AWS Directory Service

Administration Guide

Version 1.0

AWS Directory Service: Administration Guide

Table of Contents

What is AWS Directory Service?

AWS Directory Service provides multiple ways to use Microsoft Active Directory (AD) with other AWS services. Directories store information about users, groups, and devices, and administrators use them to manage access to information and resources. AWS Directory Service provides multiple directory choices for customers who want to use existing Microsoft AD or Lightweight Directory Access Protocol (LDAP)–aware applications in the cloud. It also offers those same choices to developers who need a directory to manage users, groups, devices, and access.

Which to choose

You can choose directory services with the features and scalability that best meets your needs. Use the following table to help you determine which AWS Directory Service directory option works best for your organization.

What do you need to do?	Recommended AWS Directory Service options
I need Active Directory or LDAP for my applications in the cloud	Select AWS Directory Service for Microsoft Active Directory (p. 1) (Standard Edition or Enterprise Edition) if you need an actual Microsoft Active Directory in the AWS Cloud that supports Active Directory–aware workloads, or AWS applications and services such as Amazon WorkSpaces and Amazon QuickSight, or you need LDAP support for Linux applications. Use AD Connector if you only need to allow your on-premises users to log in to AWS applications and services with their Active Directory credentials. You can also use AD Connector to join Amazon EC2 instances to your existing Active Directory domain. Use Simple AD if you need a low-scale, low-cost directory with basic Active Directory compatibility that supports Samba 4–compatible applications, or you need LDAP compatibility for LDAP-aware applications.
I develop SaaS applications	Use Amazon Cognito if you develop high-scale SaaS applications and need a scalable directory to manage and authenticate your subscribers and that works with social media identities.

AWS Directory Service options

AWS Directory Service includes several directory types to choose from. For more information, select one of the following tabs:

AWS Directory Service for Microsoft Active Directory

Also known as AWS Managed Microsoft AD, AWS Directory Service for Microsoft Active Directory is powered by an actual Microsoft Windows Server Active Directory (AD), managed by AWS in the AWS

Cloud. It enables you to migrate a broad range of Active Directory–aware applications to the AWS Cloud. AWS Managed Microsoft AD works with Microsoft SharePoint, Microsoft SQL Server Always On Availability Groups, and many .NET applications. It also supports AWS managed applications and services including Amazon WorkSpaces, Amazon WorkDocs, Amazon QuickSight, Amazon Chime, Amazon Connect, and Amazon Relational Database Service for Microsoft SQL Server (Amazon RDS for SQL Server, Amazon RDS for Oracle, and Amazon RDS for PostgreSQL).

AWS Managed Microsoft AD is approved for applications in the AWS Cloud that are subject to U.S. Health Insurance Portability and Accountability Act (HIPAA) or Payment Card Industry Data Security Standard (PCI DSS) compliance when you enable compliance for your directory.

All compatible applications work with user credentials that you store in AWS Managed Microsoft AD, or you can connect to your existing AD infrastructure with a trust and use credentials from an Active Directory running on-premises or on EC2 Windows. If you join EC2 instances to your AWS Managed Microsoft AD, your users can access Windows workloads in the AWS Cloud with the same Windows single sign-on (SSO) experience as when they access workloads in your on-premises network.

AWS Managed Microsoft AD also supports federated use cases using Active Directory credentials. Alone, AWS Managed Microsoft AD enables you to sign in to the AWS Management Console. With AWS Single Sign-On, you can also obtain short-term credentials for use with the AWS SDK and CLI, and use preconfigured SAML integrations to sign in to many cloud applications. By adding Azure AD Connect, and optionally Active Directory Federation Service (AD FS), you can sign in to Microsoft Office 365 and other cloud applications with credentials stored in AWS Managed Microsoft AD.

The service includes key features that enable you to extend your schema, manage password policies, and enable secure LDAP communications through Secure Socket Layer (SSL)/Transport Layer Security (TLS). You can also enable multi-factor authentication (MFA) for AWS Managed Microsoft AD to provide an additional layer of security when users access AWS applications from the Internet. Because Active Directory is an LDAP directory, you can also use AWS Managed Microsoft AD for Linux Secure Shell (SSH) authentication and for other LDAP-enabled applications.

AWS provides monitoring, daily snapshots, and recovery as part of the service—you add users and groups to AWS Managed Microsoft AD, and administer Group Policy using familiar Active Directory tools running on a Windows computer joined to the AWS Managed Microsoft AD domain. You can also scale the directory by deploying additional domain controllers and help improve application performance by distributing requests across a larger number of domain controllers.

AWS Managed Microsoft AD is available in two editions: Standard and Enterprise.

- **Standard Edition:** AWS Managed Microsoft AD (Standard Edition) is optimized to be a primary directory for small and midsize businesses with up to 5,000 employees. It provides you enough storage capacity to support up to 30,000* directory objects, such as users, groups, and computers.
- **Enterprise Edition:** AWS Managed Microsoft AD (Enterprise Edition) is designed to support enterprise organizations with up to 500,000* directory objects.

* Upper limits are approximations. Your directory may support more or less directory objects depending on the size of your objects and the behavior and performance needs of your applications.

When to use

AWS Managed Microsoft AD is your best choice if you need actual Active Directory features to support AWS applications or Windows workloads, including Amazon Relational Database Service for Microsoft SQL Server. It's also best if you want a standalone AD in the AWS Cloud that supports Office 365 or you need an LDAP directory to support your Linux applications. For more information, see AWS Managed Microsoft AD (p. 8).

AD Connector

AD Connector is a proxy service that provides an easy way to connect compatible AWS applications, such as Amazon WorkSpaces, Amazon QuickSight, and Amazon EC2 for Windows Server instances, to your existing on-premises Microsoft Active Directory. With AD Connector , you can simply add one service account to your Active Directory. AD Connector also eliminates the need of directory synchronization or the cost and complexity of hosting a federation infrastructure.

When you add users to AWS applications such as Amazon QuickSight, AD Connector reads your existing Active Directory to create lists of users and groups to select from. When users log in to the AWS applications, AD Connector forwards sign-in requests to your on-premises Active Directory domain controllers for authentication. AD Connector works with many AWS applications and services including Amazon WorkSpaces, Amazon WorkDocs, Amazon QuickSight, Amazon Chime, Amazon Connect, and Amazon WorkMail. You can also join your EC2 Windows instances to your on-premises Active Directory domain through AD Connector using seamless domain join. AD Connector also allows your users to access the AWS Management Console and manage AWS resources by logging in with their existing Active Directory credentials. AD Connector is not compatible with RDS SQL Server.

You can also use AD Connector to enable multi-factor authentication (MFA) for your AWS application users by connecting it to your existing RADIUS-based MFA infrastructure. This provides an additional layer of security when users access AWS applications.

With AD Connector, you continue to manage your Active Directory as you do now. For example, you add new users and groups and update passwords using standard Active Directory administration tools in your on-premises Active Directory. This helps you consistently enforce your security policies, such as password expiration, password history, and account lockouts, whether users are accessing resources on premises or in the AWS Cloud.

When to use

AD Connector is your best choice when you want to use your existing on-premises directory with compatible AWS services. For more information, see Active Directory Connector (p. 186).

Simple AD

Simple AD is a Microsoft Active Directory–*compatible* directory from AWS Directory Service that is powered by Samba 4. Simple AD supports basic Active Directory features such as user accounts, group memberships, joining a Linux domain or Windows based EC2 instances, Kerberos-based SSO, and group policies. AWS provides monitoring, daily snap-shots, and recovery as part of the service.

Simple AD is a standalone directory in the cloud, where you create and manage user identities and manage access to applications. You can use many familiar Active Directory–aware applications and tools that require basic Active Directory features. Simple AD is compatible with the following AWS applications: Amazon WorkSpaces, Amazon WorkDocs, Amazon QuickSight, and Amazon WorkMail. You can also sign in to the AWS Management Console with Simple AD user accounts and to manage AWS resources.

Simple AD does not support multi-factor authentication (MFA), trust relationships, DNS dynamic update, schema extensions, communication over LDAPS, PowerShell AD cmdlets, or FSMO role transfer. Simple AD is not compatible with RDS SQL Server. Customers who require the features of an actual Microsoft Active Directory, or who envision using their directory with RDS SQL Server should use AWS Managed Microsoft AD instead. Please verify your required applications are fully compatible with Samba 4 before using Simple AD. For more information, see https://www.samba.org.

When to use

You can use Simple AD as a standalone directory in the cloud to support Windows workloads that need basic AD features, compatible AWS applications, or to support Linux workloads that need LDAP service. For more information, see Simple Active Directory (p. 226).

Amazon Cognito

> Amazon Cognito is a user directory that adds sign-up and sign-in to your mobile app or web application using Amazon Cognito User Pools.
>
> ***When to use***
>
> You can also use Amazon Cognito when you need to create custom registration fields and store that metadata in your user directory. This fully managed service scales to support hundreds of millions of users. For more information, see Amazon Cognito user pools in the *Amazon Cognito Developer Guide*.

See Region availability for AWS Directory Service (p. 294) for a list of supported directory types per Region.

Working with Amazon EC2

A basic understanding of Amazon EC2 is essential to using AWS Directory Service. We recommend that you begin by reading the following topics:

- What is Amazon EC2? in the *Amazon EC2 User Guide for Windows Instances*.
- Launching EC2 instances in the *Amazon EC2 User Guide for Windows Instances*.
- Security groups in the *Amazon EC2 User Guide for Windows Instances*.
- What is Amazon VPC? in the *Amazon VPC User Guide*.
- Adding a Hardware Virtual Private Gateway to Your VPC in the *Amazon VPC User Guide*.

Setting up AWS Directory Service

To work with AWS Directory Service, you need to meet the prerequisites for AWS Directory Service for Microsoft Active Directory, AD Connector, or Simple AD. For more information, see AWS Managed Microsoft AD prerequisites (p. 9), AD Connector prerequisites (p. 187), or Simple AD prerequisites (p. 227).

If you haven't already done so, you'll also need to create an AWS account and use the AWS Identity and Access Management service to control access.

Topics

Sign up for an AWS account

Your AWS account gives you access to all services, but you are charged only for the resources that you use.

If you do not have an AWS account, complete the following steps to create one.

To sign up for an AWS account

1. Open https://portal.aws.amazon.com/billing/signup.
2. Follow the online instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

Your root account credentials identify you to services in AWS and grant you unlimited use of your AWS resources, such as your Amazon WorkSpaces. To allow other users to manage AWS Directory Service resources without sharing your security credentials, use AWS Identity and Access Management (IAM). We recommend that everyone work as an IAM user, even the account owner. You should create an IAM user for yourself, give that IAM user administrative privileges, and use it for all your work.

Create an IAM user

The AWS Management Console requires your username and password so that the service can determine whether you have permission to access its resources. However, we recommend that you avoid accessing AWS using the credentials for your root AWS account; instead, we recommend that you use AWS Identity and Access Management (IAM) to create an IAM user and add the IAM user to an IAM group with administrative permissions. This grants the IAM user administrative permissions. You then access the AWS Management Console using the credentials for the IAM user.

If you signed up for AWS but have not created an IAM user for yourself, you can create one using the IAM console.

To create an administrator user for yourself and add the user to an administrators group (console)

1. Sign in to the IAM console as the account owner by choosing **Root user** and entering your AWS account email address. On the next page, enter your password.

   Note

   We strongly recommend that you adhere to the best practice of using the **`Administrator`** IAM user that follows and securely lock away the root user credentials. Sign in as the root user only to perform a few account and service management tasks.
2. In the navigation pane, choose **Users** and then choose **Add users**.
3. For **User name**, enter **`Administrator`**.
4. Select the check box next to **AWS Management Console access**. Then select **Custom password**, and then enter your new password in the text box.
5. (Optional) By default, AWS requires the new user to create a new password when first signing in. You can clear the check box next to **User must create a new password at next sign-in** to allow the new user to reset their password after they sign in.
6. Choose **Next: Permissions**.
7. Under **Set permissions**, choose **Add user to group**.
8. Choose **Create group**.
9. In the **Create group** dialog box, for **Group name** enter **`Administrators`**.
10. Choose **Filter policies**, and then select **AWS managed - job function** to filter the table contents.
11. In the policy list, select the check box for **AdministratorAccess**. Then choose **Create group**.

    Note

    You must activate IAM user and role access to Billing before you can use the `AdministratorAccess` permissions to access the AWS Billing and Cost Management console. To do this, follow the instructions in step 1 of the tutorial about delegating access to the billing console.
12. Back in the list of groups, select the check box for your new group. Choose **Refresh** if necessary to see the group in the list.
13. Choose **Next: Tags**.
14. (Optional) Add metadata to the user by attaching tags as key-value pairs. For more information about using tags in IAM, see Tagging IAM entities in the *IAM User Guide*.
15. Choose **Next: Review** to see the list of group memberships to be added to the new user. When you are ready to proceed, choose **Create user**.

You can use this same process to create more groups and users and to give your users access to your AWS account resources. To learn about using policies that restrict user permissions to specific AWS resources, see Access management and Example policies.

To sign in as this new IAM user, sign out of the AWS Management Console, then use the following URL, where *your_aws_account_id* is your AWS account number without the hyphens (for example, if your AWS account number is `1234-5678-9012`, your AWS account ID is `123456789012`):

```
https://your_aws_account_id.signin.aws.amazon.com/console/
```

Enter the IAM user name and password that you just created. When you're signed in, the navigation bar displays "*your_user_name @ your_aws_account_id*".

If you don't want the URL for your sign-in page to contain your AWS account ID, you can create an account alias. From the IAM dashboard, click **Customize** and enter an alias, such as your company name. To sign in after you create an account alias, use the following URL:

```
https://your_account_alias.signin.aws.amazon.com/console/
```

For more information about using IAM policies to control access to your AWS Directory Service resources, see Using identity-based policies (IAM policies) for AWS Directory Service (p. 282).

AWS Managed Microsoft AD

AWS Directory Service lets you run Microsoft Active Directory (AD) as a managed service. AWS Directory Service for Microsoft Active Directory, also referred to as AWS Managed Microsoft AD, is powered by Windows Server 2012 R2. When you select and launch this directory type, it is created as a highly available pair of domain controllers connected to your virtual private cloud (VPC). The domain controllers run in different Availability Zones in a Region of your choice. Host monitoring and recovery, data replication, snapshots, and software updates are automatically configured and managed for you.

With AWS Managed Microsoft AD, you can run directory-aware workloads in the AWS Cloud, including Microsoft SharePoint and custom .NET and SQL Server-based applications. You can also configure a trust relationship between AWS Managed Microsoft AD in the AWS Cloud and your existing on-premises Microsoft Active Directory, providing users and groups with access to resources in either domain, using single sign-on (SSO).

AWS Directory Service makes it easy to set up and run directories in the AWS Cloud, or connect your AWS resources with an existing on-premises Microsoft Active Directory. Once your directory is created, you can use it for a variety of tasks:

- Manage users and groups
- Provide single sign-on to applications and services
- Create and apply group policy
- Simplify the deployment and management of cloud-based Linux and Microsoft Windows workloads
- You can use AWS Managed Microsoft AD to enable multi-factor authentication by integrating with your existing RADIUS-based MFA infrastructure to provide an additional layer of security when users access AWS applications.
- Securely connect to Amazon EC2 Linux and Windows instances

> **Note**
> AWS manages the licensing of your Windows Server instances for you; all you need to do is pay for the instances you use. There is also no need to buy additional Windows Server CALs, as access is included in the price. Each instance comes with two remote connections for admin purposes only. If you require more than two connections, or need those connections for purposes other than admin, you may have to bring in additional Remote Desktop Services CALs for use on AWS.

Read the topics in this section to get started creating a AWS Managed Microsoft AD directory, creating a trust relationship between AWS Managed Microsoft AD and your on-premises directories, and extending your AWS Managed Microsoft AD schema.

Topics

Related AWS Security blog articles

- How to delegate administration of your AWS Managed Microsoft AD directory to your on-premises Active Directory users
- How to configure even stronger password policies to help meet your security standards by using AWS Directory Service for AWS Managed Microsoft AD
- How to increase the redundancy and performance of your AWS Directory Service for AWS Managed Microsoft AD by adding Domain controllers
- How to enable the use of remote desktops by deploying Microsoft remote desktop licensing manager on AWS Managed Microsoft AD
- How to access the AWS Management Console using AWS Managed Microsoft AD and your on-premises credentials
- How to enable multi-factor authentication for AWS services by using AWS Managed Microsoft AD and on-premises credentials
- How to easily log on to AWS services by using your on-premises Active Directory

Getting started with AWS Managed Microsoft AD

AWS Managed Microsoft AD creates a fully managed, Microsoft Active Directory in the AWS Cloud and is powered by Windows Server 2012 R2 and operates at the 2012 R2 Forest and Domain functional levels. When you create a directory with AWS Managed Microsoft AD, AWS Directory Service creates two domain controllers and adds the DNS service on your behalf. The domain controllers are created in different subnets in a VPC; this redundancy helps ensure that your directory remains accessible even if a failure occurs. If you need more domain controllers, you can add them later. For more information, see Deploy additional domain controllers (p. 143).

Topics

AWS Managed Microsoft AD prerequisites

To create a AWS Managed Microsoft AD directory, you need a VPC with the following:

- At least two subnets. Each of the subnets must be in a different Availability Zone.
- The VPC must have default hardware tenancy.
- You cannot create a AWS Managed Microsoft AD in a VPC using addresses in the 198.18.0.0/15 address space.

If you need to integrate your AWS Managed Microsoft AD domain with an existing on-premises Active Directory domain, you must have the Forest and Domain functional levels for your on-premises domain set to Windows Server 2003 or higher.

AWS Directory Service uses a two VPC structure. The EC2 instances which make up your directory run outside of your AWS account, and are managed by AWS. They have two network adapters, `ETH0` and `ETH1`. `ETH0` is the management adapter, and exists outside of your account. `ETH1` is created within your account.

The management IP range of your directory's ETH0 network is 198.18.0.0/15.

AWS Single Sign-On prerequisites

If you plan to use AWS Single Sign-On (AWS SSO) with AWS Managed Microsoft AD, you need to ensure that the following are true:

- Your AWS Managed Microsoft AD directory is set up in your AWS organization's management account.
- Your instance of AWS SSO is in the same Region where your AWS Managed Microsoft AD directory is set up.

For more information, see AWS SSO prerequisites in the AWS Single Sign-On User Guide.

Multi-factor authentication prerequisites

To support multi-factor authentication with your AWS Managed Microsoft AD directory, you must configure either your on-premises or cloud-based Remote Authentication Dial-In User Service (RADIUS) server in the following way so that it can accept requests from your AWS Managed Microsoft AD directory in AWS.

1. On your RADIUS server, create two RADIUS clients to represent both of the AWS Managed Microsoft AD domain controllers (DCs) in AWS. You must configure both clients using the following common parameters (your RADIUS server may vary):
   - **Address (DNS or IP)**: This is the DNS address for one of the AWS Managed Microsoft AD DCs. Both DNS addresses can be found in the AWS Directory Service Console on the **Details** page of the AWS Managed Microsoft AD directory in which you plan to use MFA. The DNS addresses displayed represent the IP addresses for both of the AWS Managed Microsoft AD DCs that are used by AWS.

     > **Note**
     > If your RADIUS server supports DNS addresses, you must create only one RADIUS client configuration. Otherwise, you must create one RADIUS client configuration for each AWS Managed Microsoft AD DC.
   - **Port number**: Configure the port number for which your RADIUS server accepts RADIUS client connections. The standard RADIUS port is 1812.
   - **Shared secret**: Type or generate a shared secret that the RADIUS server will use to connect with RADIUS clients.
   - **Protocol**: You might need to configure the authentication protocol between the AWS Managed Microsoft AD DCs and the RADIUS server. Supported protocols are PAP, CHAP MS-CHAPv1, and MS-CHAPv2. MS-CHAPv2 is recommended because it provides the strongest security of the three options.
   - **Application name**: This may be optional in some RADIUS servers and usually identifies the application in messages or reports.
2. Configure your existing network to allow inbound traffic from the RADIUS clients (AWS Managed Microsoft AD DCs DNS addresses, see Step 1) to your RADIUS server port.
3. Add a rule to the Amazon EC2 security group in your AWS Managed Microsoft AD domain that allows inbound traffic from the RADIUS server DNS address and port number defined previously. For more information, see Adding rules to a security group in the *EC2 User Guide*.

For more information about using AWS Managed Microsoft AD with MFA, see Enable multi-factor authentication for AWS Managed Microsoft AD (p. 31).

Create your AWS Managed Microsoft AD directory

To create a new directory, perform the following steps. Before starting this procedure, make sure that you have completed the prerequisites identified in AWS Managed Microsoft AD prerequisites (p. 9).

To create an AWS Managed Microsoft AD directory

1. In the AWS Directory Service console navigation pane, choose **Directories** and then choose **Set up directory**.
2. On the **Select directory type** page, choose **AWS Managed Microsoft AD**, and then choose **Next**.
3. On the **Enter directory information** page, provide the following information:

   Edition

      Choose from either the **Standard Edition** or **Enterprise Edition** of AWS Managed Microsoft AD. For more information about editions, see AWS Directory Service for Microsoft Active Directory (p. 1).

   Directory DNS name

      The fully qualified name for the directory, such as `corp.example.com`.

   Directory NetBIOS name

      The short name for the directory, such as `CORP`.

   Directory description

      An optional description for the directory.

   Admin password

      The password for the directory administrator. The directory creation process creates an administrator account with the user name `Admin` and this password.

      The password cannot include the word "admin."

      The directory administrator password is case-sensitive and must be between 8 and 64 characters in length, inclusive. It must also contain at least one character from three of the following four categories:

      - Lowercase letters (a-z)
      - Uppercase letters (A-Z)
      - Numbers (0-9)
      - Non-alphanumeric characters (~!@#$%^&*_-+=`|\(){}[]:;"'<>,.?/)

   Confirm password

      Retype the administrator password.
4. On the **Choose VPC and subnets** page, provide the following information, and then choose **Next**.

   VPC

      The VPC for the directory.

   Subnets

      Choose the subnets for the domain controllers. The two subnets must be in different Availability Zones.
5. On the **Review & create** page, review the directory information and make any necessary changes. When the information is correct, choose **Create directory**. Creating the directory takes 20 to 40 minutes. Once created, the **Status** value changes to **Active**.

What gets created

When you create a directory with AWS Managed Microsoft AD, AWS Directory Service performs the following tasks on your behalf:

- Automatically creates and associates an elastic network interface (ENI) with each of your domain controllers. Each of these ENIs are essential for connectivity between your VPC and AWS Directory Service domain controllers and should never be deleted. You can identify all network interfaces reserved for use with AWS Directory Service by the description: "AWS created network interface for directory *directory-id*". For more information, see Elastic Network Interfaces in the Amazon EC2 User Guide for Windows Instances.

  Note
  Domain controllers are deployed across two Availability Zones in a region by default and connected to your Amazon Virtual Private Cloud (VPC). Backups are automatically taken once per day, and the Amazon Elastic Block Store (EBS) volumes are encrypted to ensure that data is secured at rest. Domain controllers that fail are automatically replaced in the same Availability Zone using the same IP address, and a full disaster recovery can be performed using the latest backup.
- Provisions Active Directory within your VPC using two domain controllers for fault tolerance and high availability. More domain controllers can be provisioned for higher resiliency and performance after the directory has been successfully created and is Active. For more information, see Deploy additional domain controllers (p. 143).

  Note
  AWS does not allow the installation of monitoring agents on AWS Managed Microsoft AD domain controllers.
- Creates an AWS security group that establishes network rules for traffic in and out of your domain controllers. The default outbound rule permits all traffic ENIs or instances attached to the created AWS Security Group. The default inbound rules allows only traffic through ports that are required by Active Directory from any source (0.0.0.0/0). The 0.0.0.0/0 rules do not introduce security vulnerabilities as traffic to the domain controllers is limited to traffic from your VPC, from other peered VPCs, or from networks that you have connected using AWS Direct Connect, AWS Transit Gateway, or Virtual Private Network. For additional security, the ENIs that are created do not have Elastic IPs attached to them and you do not have permission to attach an Elastic IP to those ENIs. Therefore, the only inbound traffic that can communicate with your AWS Managed Microsoft AD is local VPC and VPC routed traffic. Use extreme caution if you attempt to change these rules as you may break your ability to communicate with your domain controllers. The following AWS Security Group rules are created by default:

  Inbound Rules

Protocol	Port range	Source	Type of traffic	Active Directory usage
ICMP	N/A	0.0.0.0/0	Ping	None
TCP & UDP	53	0.0.0.0/0	DNS	User and computer authentication, name resolution, trusts
TCP & UDP	88	0.0.0.0/0	Kerberos	User and computer authentication, forest level trusts

Protocol	Port range	Source	Type of traffic	Active Directory usage
TCP & UDP	389	0.0.0.0/0	LDAP	Directory, replication, user and computer authentication group policy, trusts
TCP & UDP	445	0.0.0.0/0	SMB / CIFS	Replication, user and computer authentication, group policy, trusts
TCP & UDP	464	0.0.0.0/0	Kerberos change / set password	Replication, user and computer authentication, trusts
TCP	135	0.0.0.0/0	Replication	RPC, EPM
TCP	636	0.0.0.0/0	LDAP SSL	Directory, replication, user and computer authentication, group policy, trusts
TCP	1024 - 65535	0.0.0.0/0	RPC	Replication, user and computer authentication, group policy, trusts
TCP	3268 - 3269	0.0.0.0/0	LDAP GC & LDAP GC SSL	Directory, replication, user and computer authentication, group policy, trusts
UDP	123	0.0.0.0/0	Windows Time	Windows Time, trusts
UDP	138	0.0.0.0/0	DFSN & NetLogon	DFS, group policy
All	All	sg-##################	All Traffic	

Outbound Rules

Protocol	Port range	Destination	Type of traffic	Active Directory usage
All	All	sg-#################	All Traffic	

- Creates a directory administrator account with the user name Admin and the specified password. This account is located under the Users OU (For example, Corp > Users). You use this account to manage your directory in the AWS Cloud. For more information, see Admin account (p. 17).

  Important
  Be sure to save this password. AWS Directory Service does not store this password, and it cannot be retrieved. However, you can reset a password from the AWS Directory Service console or by using the ResetUserPassword API.

- Creates the following three organizational units (OUs) under the domain root:

OU name	Description
AWS Delegated Groups	Stores all of the groups that you can use to delegate AWS specific permissions to your users.
AWS Reserved	Stores all AWS management specific accounts.
<yourdomainname>	The name of this OU is based off of the NetBIOS name you typed when you created your directory. If you did not specify a NetBIOS name, it will default to the first part of your Directory DNS name (for example, in the case of corp.example.com, the NetBIOS name would be *corp*). This OU is owned by AWS and contains all of your AWS-related directory objects, which you are granted Full Control over. Two child OUs exist under this OU by default; Computers and Users. For example: • Corp • Computers • Users

- Creates the following groups in the AWS Delegated Groups OU:

Group name	Description
AWS Delegated Account Operators	Members of this security group have limited account management capability such as password resets
AWS Delegated Active Directory Based Activation Administrators	Members of this security group can create Active Directory volume licensing activation objects, which enables enterprises to activate computers through a connection to their domain.
AWS Delegated Add Workstations To Domain Users	Members of this security group can join 10 computers to a domain.

Group name	Description
AWS Delegated Administrators	Members of this security group can manage AWS Managed Microsoft AD, have full control of all the objects in your OU and can manage groups contained in the AWS Delegated Groups OU.
AWS Delegated Allowed to Authenticate Objects	Members of this security group are provided the ability to authenticate to computer resources in the AWS Reserved OU (Only needed for on-premises objects with Selective Authentication enabled Trusts).
AWS Delegated Allowed to Authenticate to Domain Controllers	Members of this security group are provided the ability to authenticate to computer resources in the Domain Controllers OU (Only needed for on-premises objects with Selective Authentication enabled Trusts).
AWS Delegated Deleted Object Lifetime Administrators	Members of this security group can modify the msDS-DeletedObjectLifetime object, which defines how long a deleted object will be available to recover from the AD Recycle Bin.
AWS Delegated Distributed File System Administrators	Members of this security group can add and remove FRS, DFS-R, and DFS name spaces.
AWS Delegated Domain Name System Administrators	Members of this security group can manage Active Directory integrated DNS.
AWS Delegated Dynamic Host Configuration Protocol Administrators	Members of this security group can authorize Windows DHCP servers in the enterprise.
AWS Delegated Enterprise Certificate Authority Administrators	Members of this security group can deploy and manage Microsoft Enterprise Certificate Authority infrastructure.
AWS Delegated Fine Grained Password Policy Administrators	Members of this security group can modify precreated fine-grained password policies.
AWS Delegated FSx Administrators	Members of this security group are provided the ability to manage Amazon FSx resources.
AWS Delegated Group Policy Administrators	Members of this security group can perform group policy management tasks (create, edit, delete, link).
AWS Delegated Kerberos Delegation Administrators	Members of this security group can enable delegation on computer and user account objects.
AWS Delegated Managed Service Account Administrators	Members of this security group can create and delete Managed Service Accounts.
AWS Delegated MS-NPRC Non-Compliant Devices	Members of this security group will be provided an exclusion from requiring secure channel communications with domain controllers. This group is for computer accounts.

Group name	Description
AWS Delegated Remote Access Service Administrators	Members of this security group can add and remove RAS servers from the RAS and IAS Servers group.
AWS Delegated Replicate Directory Changes Administrators	Members of this security group can synchronize profile information in Active Directory with SharePoint Server.
AWS Delegated Server Administrators	Members of this security group are included in the local administrators group on all domain joined computers.
AWS Delegated Sites and Services Administrators	Members of this security group can rename the Default-First-Site-Name object in Active Directory Sites and Services.
AWS Delegated System Management Administrators	Members of this security group can create and manage objects in the System Management container.
AWS Delegated Terminal Server Licensing Administrators	Members of this security group can add and remove Terminal Server License Servers from the Terminal Server License Servers group.
AWS Delegated User Principal Name Suffix Administrators	Members of this security group can add and remove user principal name suffixes.

- Creates and applies the following Group Policy Objects (GPOs):

  Note
  You do not have permissions to delete, modify, or unlink these GPOs. This is by design as they are reserved for AWS use. You may link them to OUs that you control if needed.

Group policy name	Applies to	Description
Default Domain Policy	Domain	Includes domain password and Kerberos policies.
ServerAdmins	All non domain controller computer accounts	Adds the 'AWS Delegated Server Administrators' as a member of the BUILTIN\Administrators Group.
AWS Reserved Policy:User	AWS Reserved user accounts	Sets recommended security settings on all user accounts in the AWS Reserved OU.
AWS Managed Active Directory Policy	All domain controllers	Sets recommended security settings on all domain controllers.
TimePolicyNT5DS	All non PDCe domain controllers	Sets all non PDCe domain controllers time policy to use Windows Time (NT5DS).

Group policy name	Applies to	Description
TimePolicyPDC	The PDCe domain controller	Sets the PDCe domain controller's time policy to use Network Time Protocol (NTP).
Default Domain Controllers Policy	Not used	Provisioned during domain creation, AWS Managed Active Directory Policy is used in its place.

If you would like to see the settings of each GPO, you can view them from a domain joined Windows instance with the Group policy management console (GPMC) enabled.

Admin account

When you create an AWS Directory Service for Microsoft Active Directory directory, AWS creates an organizational unit (OU) to store all AWS related groups and accounts. For more information about this OU, see What gets created (p. 12). This includes the Admin account. The Admin account has permissions to perform the following common administrative activities for your OU:

- Add, update, or delete users, groups, and computers. For more information, see Manage users and groups in AWS Managed Microsoft AD (p. 100).
- Add resources to your domain such as file or print servers, and then assign permissions for those resources to users and groups in your OU.
- Create additional OUs and containers.
- Delegate authority of additional OUs and containers. For more information, see Delegate directory join privileges for AWS Managed Microsoft AD (p. 98).
- Create and link group policies.
- Restore deleted objects from the Active Directory Recycle Bin.
- Run Active Directory and DNS Windows PowerShell modules on the Active Directory Web Service.
- Create and configure group Managed Service Accounts. For more information, see Group Managed Service Accounts (p. 20).
- Configure Kerberos constrained delegation. For more information, see Kerberos constrained delegation (p. 20).

The Admin account also has rights to perform the following domainwide activities:

- Manage DNS configurations (add, remove, or update records, zones, and forwarders)
- View DNS event logs
- View security event logs

Only the actions listed here are allowed for the Admin account. The Admin account also lacks permissions for any directory-related actions outside of your specific OU, such as on the parent OU.

> **Important**
> AWS Domain Administrators have full administrative access to all domains hosted on AWS. See your agreement with AWS and the AWS data protection FAQ for more information about how AWS handles content, including directory information, that you store on AWS systems.

Note
We recommend that you do not delete or rename this account. If you no longer want to use the account, we recommend you set a long password (at most 64 random characters) and then disable the account.

Enterprise and domain administrator privileged accounts

AWS automatically rotates the built-in Administrator password to a random password every 90 days. Anytime the built in Administrator password is requested for human use an AWS ticket is created and logged with the AWS Directory Service team. Account credentials are encrypted and handled over secure channels. Also the Administrator account credentials can only be requested by the AWS Directory Service management team.

To perform operational management of your directory, AWS has exclusive control of accounts with Enterprise Administrator and Domain Administrator privileges. This includes exclusive control of the AD administrator account. AWS protects this account by automating password management through the use of a password vault. During automated rotation of the administrator password, AWS creates a temporary user account and grants it Domain Administrator privileges. This temporary account is used as a back-up in the event of password rotation failure on the administrator account. After AWS successfully rotates the administrator password, AWS deletes the temporary administrator account.

Normally AWS operates the directory entirely through automation. In the event that an automation process is unable to resolve an operational problem, AWS may need to have a support engineer sign in to your domain controller (DC) to perform diagnosis. In these rare cases, AWS implements a request/notification system to grant access. In this process, AWS automation creates a time-limited user account in your directory that has Domain Administrator permissions. AWS associates the user account with the engineer who is assigned to work on your directory. AWS records this association in our log system and provides the engineer with the credentials to use. All actions taken by the engineer are logged in the Windows event logs. When the allocated time elapses, automation deletes the user account.

You can monitor administrative account actions by using the log forwarding feature of your directory. This feature enables you to forward the AD Security events to your CloudWatch system where you can implement monitoring solutions. For more information, see Enable log forwarding (p. 55).

Security Event IDs 4624, 4672 and 4648 are all logged when someone logs onto a DC interactively. You can view each DC’s Windows Security event log using the Event Viewer Microsoft Management Console (MMC) from a domain joined Windows computer. You can also Enable log forwarding (p. 55) to send all of the Security event logs to CloudWatch Logs in your account.

You might occasionally see users created and deleted within the AWS Reserved OU. AWS is responsible for the management and security of all objects in this OU and any other OU or container where we have not delegated permissions for you to access and manage. You may see creations and deletions in that OU. This is because AWS Directory Service uses automation to rotate the Domain Administrator password on a regular basis. When the password is rotated, a backup is created in the event that the rotation fails. Once the rotation is successful, the backup account is automatically deleted. Also in the rare event that interactive access is needed on the DCs for troubleshooting purposes, a temporary user account is created for an AWS Directory Service engineer to use. Once an engineer has completed their work, the temporary user account will be deleted. Note that every time interactive credentials are requested for a directory, the AWS Directory Service management team is notified.

Key concepts for AWS Managed Microsoft AD

You'll get more out of AWS Managed Microsoft AD if you become familiar with the following key concepts.

Topics

- Active Directory schema (p. 19)
- Patching and maintenance for AWS Managed Microsoft AD (p. 20)
- Group Managed Service Accounts (p. 20)
- Kerberos constrained delegation (p. 20)

Active Directory schema

A schema is the definition of attributes and classes that are part of a distributed directory and is similar to fields and tables in a database. Schemas include a set of rules which determine the type and format of data that can be added or included in the database. The User class is one example of a *class* that is stored in the database. Some example of User class attributes can include the user's first name, last name, phone number, and so on.

Schema elements

Attributes, classes and objects are the basic elements that are used to build object definitions in the schema. The following provides details about schema elements that are important to know before you begin the process to extend your AWS Managed Microsoft AD schema.

Attributes

Each schema attribute, which is similar to a field in a database, has several properties that define the characteristics of the attribute. For example, the property used by LDAP clients to read and write the attribute is `LDAPDisplayName`. The `LDAPDisplayName` property must be unique across all attributes and classes. For a complete list of attribute characteristics, see Characteristics of Attributes on the MSDN website. For additional guidance on how to create a new attribute, see Defining a New Attribute on the MSDN website.

Classes

The classes are analogous to tables in a database and also have several properties to be defined. For example, the `objectClassCategory` defines the class category. For a complete list of class characteristics, see Characteristics of Object Classes on the MSDN website. For more information about how to create a new class, see Defining a New Class on the MSDN website.

Object identifier (OID)

Each class and attribute must have an OID that is unique for all of your objects. Software vendors must obtain their own OID to ensure uniqueness. Uniqueness avoids conflicts when the same attribute is used by more than one application for different purposes. To ensure uniqueness, you can obtain a root OID from an ISO Name Registration Authority. Alternatively, you can obtain a base OID from Microsoft. For more information about OIDs and how to obtain them, see Object Identifiers on the MSDN website.

Schema linked attributes

Some attributes are linked between two classes with forward and back links. The best example is groups. When you look at a group it shows you the members of the group; if you look at a user you can see what groups it belongs to. When you add a user to a group, Active Directory creates a forward link to the group. Then Active Directory adds a back link from the group to the user. A unique link ID must be generated when creating an attribute that will be linked. For more information, see Linked Attributes on the MSDN website.

Related topics

- When to extend your AWS Managed Microsoft AD schema (p. 121)

- Tutorial: Extending your AWS Managed Microsoft AD schema (p. 121)

Patching and maintenance for AWS Managed Microsoft AD

AWS Directory Service for Microsoft Active Directory, also known as AWS DS for AWS Managed Microsoft AD, is actually Microsoft Active Directory Domain Services (AD DS), delivered as a managed service. The system uses Microsoft Windows Server 2012 R2 for the domain controllers (DCs), and AWS adds software to the DCs for service management purposes. AWS updates (patches) DCs to add new functionality and keep the Microsoft Windows Server software current. During the patching process, your directory remains available for use.

Ensuring availability

By default each directory consists of two DCs, each installed in a different Availability Zone. At your option, you may add DCs to further increase availability. AWS patches your DCs sequentially, during which time the DC that AWS is actively patching is unavailable. In the event that one or more of your DCs is temporarily out of service, AWS defers patching until your directory has at least two operational DCs. This lets you use the other operating DCs during the patch process, which typically takes 30 to 45 minutes per DC, although this time may vary. To ensure your applications can reach an operating DC in the event that one or more DCs is unavailable for any reason, including patching, your applications should use the Windows DC locator service and not use static DC addresses.

Understanding the patching schedule

To keep the Microsoft Windows Server software current on your DCs, AWS utilizes Microsoft updates. As Microsoft makes monthly rollup patches available for Windows Server, AWS makes a best effort to test and apply the rollup to all customer DCs within three calendar weeks. In addition, AWS reviews updates that Microsoft releases outside of the monthly rollup based on applicability to DCs and urgency. For security patches that Microsoft rates as *Critical* or *Important*, and that are relevant to DCs, AWS makes every effort to test and deploy the patch within five days.

Group Managed Service Accounts

With Windows Server 2012, Microsoft introduced a new method that administrators could use to manage service accounts called group Managed Service Accounts (gMSAs). Using gMSAs, service administrators no longer needed to manually manage password synchronization between service instances. Instead, an administrator could simply create a gMSA in Active Directory and then configure multiple service instances to use that single gMSA.

To grant permissions so users in AWS Managed Microsoft AD can create a gMSA, you must add their accounts as a member of the *AWS Delegated Managed Service Account Administrators* security group. By default, the Admin account is a member of this group. For more information about gMSAs, see Group Managed Service Accounts Overview on the Microsoft TechNet website.

Related AWS Security Blog post

- How AWS Managed Microsoft AD Helps to Simplify the Deployment and Improve the Security of Active Directory–Integrated .NET Applications

Kerberos constrained delegation

Kerberos constrained delegation is a feature in Windows Server. This feature gives service administrators the ability to specify and enforce application trust boundaries by limiting the scope where application services can act on a user's behalf. This can be useful when you need to configure which front-end service accounts can delegate to their backend services. Kerberos constrained delegation also prevents your gMSA from connecting to any and all services on behalf of your Active Directory users, avoiding the potential for abuse by a rogue developer.

For example, let's say user jsmith logs into an HR application. You want the SQL Server to apply jsmith's database permissions. However, by default SQL Server opens the database connection using the service account credentials that apply hr-app-service's permissions instead of jsmith's configured permissions. You must make it possible for the HR payroll application to access the SQL Server database using the jsmith's credentials. To do that, you enable Kerberos constrained delegation for the hr-app-service service account on your AWS Managed Microsoft AD directory in AWS. When jsmith logs on, Active Directory provides a Kerberos ticket that Windows automatically uses when jsmith attempts to access other services in the network. Kerberos delegation enables the hr-app-service account to reuse the jsmith Kerberos ticket when accessing the database, thus applying permissions specific to jsmith when opening the database connection.

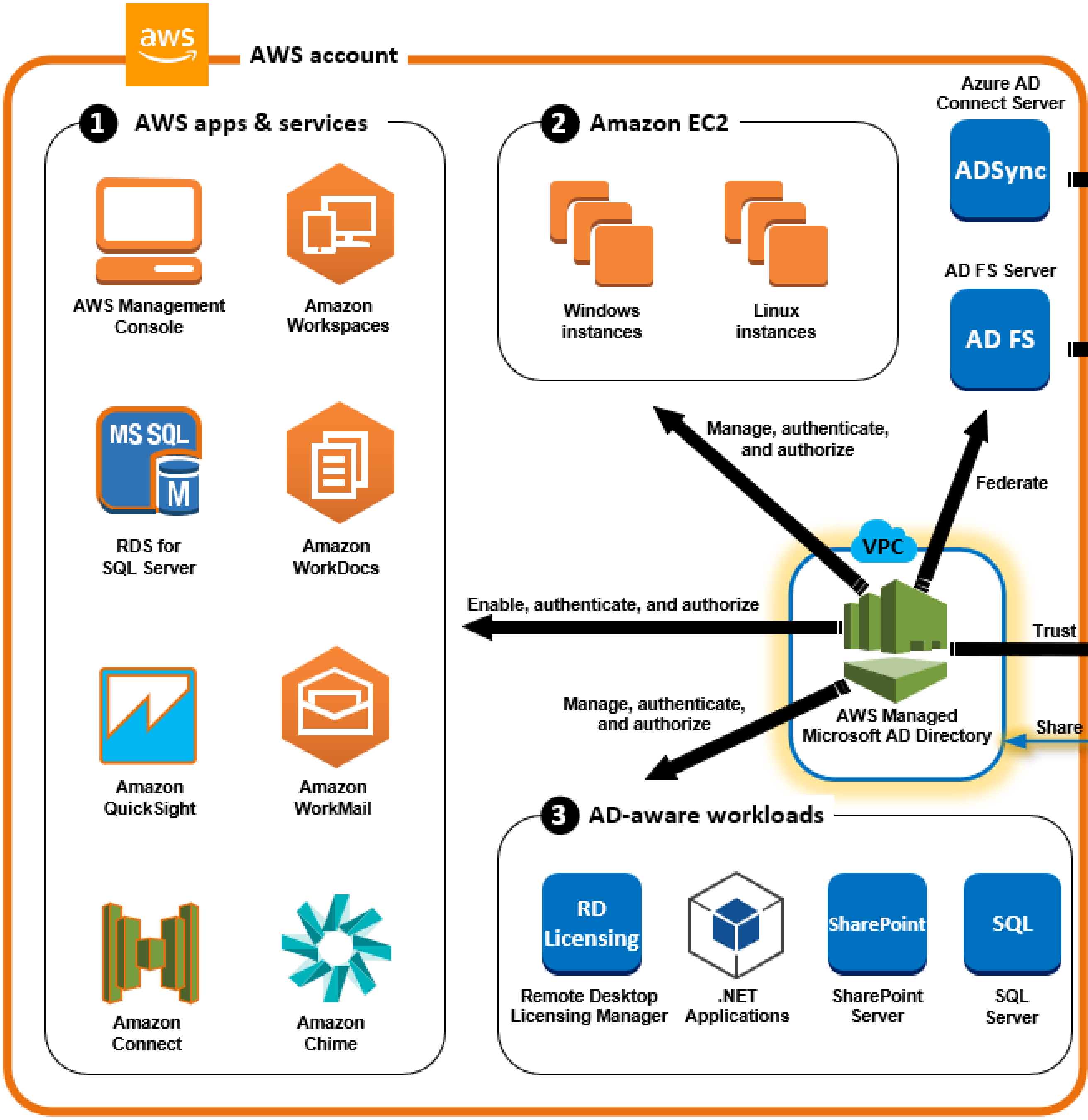
To grant permissions that allow users in AWS Managed Microsoft AD to configure Kerberos constrained delegation, you must add their accounts as a member of the *AWS Delegated Kerberos Delegation Administrators* security group. By default, the Admin account is a member of this group. For more information about Kerberos constrained delegation, see Kerberos Constrained Delegation Overview on the Microsoft TechNet website.

Resource-based constrained delegation was introduced with Windows Server 2012. It provides the back-end service administrator the ability to configure constrained delegation for the service.

Use cases for AWS Managed Microsoft AD

With AWS Managed Microsoft AD, you can share a single directory for multiple use cases. For example, you can share a directory to authenticate and authorize access for .NET applications, Amazon RDS for SQL Server with Windows authentication enabled, and Amazon Chime for messaging and video conferencing.

The following diagram shows some of the use cases for your AWS Managed Microsoft AD directory. These include the ability to grant your users access to external cloud applications and allow your on-premises AD users to manage and have access to resources in the AWS Cloud.



Use AWS Managed Microsoft AD for either of the following business use cases.

Topics

- Use Case 6: Share your directory to seamlessly join Amazon EC2 instances to a domain across AWS accounts (p. 28)

Use Case 1: Sign in to AWS applications and services with AD credentials

You can enable multiple AWS applications and services such as AWS Client VPN, AWS Management Console, AWS Single Sign-On, Amazon Chime, Amazon Connect, Amazon FSx, Amazon QuickSight, Amazon RDS for SQL Server, Amazon WorkDocs, Amazon WorkMail, and WorkSpaces to use your AWS Managed Microsoft AD directory. When you enable an AWS application or service in your directory, your users can access the application or service with their AD credentials.

For example, you can enable your users to sign in to the AWS Management Console with their AD credentials. To do this, you enable the AWS Management Console as an application in your directory, and then assign your AD users and groups to IAM roles. When your users sign in to the AWS Management Console, they assume an IAM role to manage AWS resources. This makes it easy for you to grant your users access to the AWS Management Console without needing to configure and manage a separate SAML infrastructure.

To further enhance the end user experience you can enable Single sign-on (SSO) capabilities for Amazon WorkDocs, which provides your users the ability to access Amazon WorkDocs from a computer joined to the directory without having to enter their credentials separately.

You can grant access to user accounts in your directory or in your on-premises AD, so they can sign in to the AWS Management Console or through the AWS CLI using their existing credentials and permissions to manage AWS resources by assigning IAM roles directly to the existing user accounts.

FSx for Windows File Server integration with AWS Managed Microsoft AD

Integrating FSx for Windows File Server with AWS Managed Microsoft AD provides a fully managed native Microsoft Windows based Server Message Block (SMB) protocol file system that allows you to easily move your Windows-based applications and clients (that utilize shared file storage) to AWS. Although FSx for Windows File Server can be integrated with a self-managed Microsoft Active Directory, we do not discuss that scenario here.

Common Amazon FSx use cases and resources

This section provides a reference to resources on common FSx for Windows File Server integrations with AWS Managed Microsoft AD use cases. Each of the use cases in this section start with a basic AWS Managed Microsoft AD and FSx for Windows File Server configuration. For more information about how to create these configurations, see:

- Getting started with AWS Managed Microsoft AD (p. 9)
- Getting started with Amazon FSx

FSx for Windows File Server as persistent storage on Windows containers

Amazon Elastic Container Service (ECS) supports Windows containers on container instances that are launched with the Amazon ECS-optimized Windows AMI. Windows container instances use their own version of the Amazon ECS container agent. On the Amazon ECS-optimized Windows AMI, the Amazon ECS container agent runs as a service on the host.

Amazon ECS supports Active Directory authentication for Windows containers through a special kind of service account called a group Managed Service Account (gMSA). Because Windows containers cannot be domain-joined, you must configure a Windows container to run with gMSA.

Related Items

- Using FSx for Windows File Server as persistent storage on Windows Containers
- Group Managed Service Accounts

Amazon AppStream 2.0 support

Amazon AppStream 2.0 is a fully managed application streaming service. It provides a range of solutions for users to save and access data through their applications. Amazon FSx with AppStream 2.0 provides a personal persistent storage drive using Amazon FSx and can be configured to provide a shared folder to access common files.

Related Items

- Walkthrough 4: Using Amazon FSx with Amazon AppStream 2.0
- Using Amazon FSx with Amazon AppStream 2.0
- Using Active Directory with AppStream 2.0

Microsoft SQL Server support

FSx for Windows File Server can be used as a storage option for Microsoft SQL Server 2012 (starting with 2012 version 11.x) and newer system databases (including Master, Model, MSDB, and TempDB), and for Database Engine user databases.

Related Items

- Install SQL Server with SMB fileshare storage
- Simplify your Microsoft SQL Server high availability deployments using FSx for Windows File Server
- Group Managed Service Accounts

Home folders and roaming user profile support

FSx for Windows File Server can be used to store data from Active Directory user home folders and My Documents in a central location. FSx for Windows File Server can also be used to store data from Roaming User Profiles.

Related items

- Windows home directories made easy with Amazon FSx
- Deploying roaming user profiles
- Using FSx for Windows File Server with WorkSpaces

Networked file share support

Networked file shares on an FSx for Windows File Server provide a managed and scalable file sharing solution. One use case is mapped drives for clients that can be created manually or via Group Policy.

Related items

- Walkthrough 6: Scaling out performance with Shards

- Drive mapping
- Using FSx for Windows File Server with WorkSpaces

Group policy software installation support

Because the size and performance of the SYSVOL folder is limited, you should as a best practice, avoid storing data such as software installation files in that folder. As a possible solution to this, FSx for Windows File Server can be configured to store all software files that are installed using Group Policy.

Related items

- How to use Group Policy to remotely install software in Windows Server 2008 and in Windows Server 2003

Windows Server Backup target support

FSx for Windows File Server can be configured as a target drive in Windows Server Backup using the UNC file share. In this case, you would specify the UNC path to your FSx for Windows File Server instead of to the attached EBS volume.

Related Items

- Perform a system state recovery of your server

Amazon FSx also supports AWS Managed Microsoft AD Directory Sharing. For more information, see:

- Share your directory (p. 63)
- Using Amazon FSx with AWS Managed Microsoft AD in a different VPC or account

Amazon RDS integration with AWS Managed Microsoft AD

Amazon RDS supports external authentication of database users using Kerberos with Microsoft Active Directory. Kerberos is a network authentication protocol that uses tickets and symmetric-key cryptography to eliminate the need to transmit passwords over the network. Amazon RDS support for Kerberos and Active Directory provides the benefits of single sign-on and centralized authentication of database users so you can keep your user credentials in Active Directory.

To get started with this use case you'll first need to set up a basic AWS Managed Microsoft AD and Amazon RDS configuration.

- Getting started with AWS Managed Microsoft AD (p. 9)
- Getting started with Amazon RDS

All of the use cases referenced below will start with a base AWS Managed Microsoft AD and Amazon RDS and cover how to integrate Amazon RDS with AWS Managed Microsoft AD .

- Using Windows authentication with an Amazon RDS for SQL Server DB instance
- Using Kerberos authentication for MySQL
- Using Kerberos authentication with Amazon RDS for Oracle
- Using Kerberos authentication with Amazon RDS for PostgreSQL

Amazon RDS also supports AWS Managed Microsoft AD Directory Sharing. For more information, see:

- Share your directory (p. 63)
- Joining your Amazon RDS DB instances across accounts to a single shared domain

.NET application using Amazon RDS for SQL Server with group Managed Service Accounts

You can integrate Amazon RDS for SQL Server with a basic .NET application and group Managed Service Accounts (gMSAs). For more information, see How AWS Managed Microsoft AD Helps to Simplify the Deployment and Improve the Security of Active Directory–Integrated .NET Applications

Use Case 2: Manage Amazon EC2 instances

Using familiar AD administration tools, you can apply AD group policy objects (GPOs) to centrally manage your Amazon EC2 for Windows or Linux instances by joining your instances to your AWS Managed Microsoft AD domain.

In addition, your users can sign in to your instances with their AD credentials. This eliminates the need to use individual instance credentials or distribute private key (PEM) files. This makes it easier for you to instantly grant or revoke access to users by using AD user administration tools you already use.

Use Case 3: Provide directory services to your AD-aware workloads

AWS Managed Microsoft AD is an actual Microsoft AD that enables you to run traditional AD-aware workloads such as Remote Desktop Licensing Manager and Microsoft SharePoint and Microsoft SQL Server Always On in the AWS Cloud. AWS Managed Microsoft AD also helps you to simplify and improve the security of AD-integrated .NET applications by using group Managed Service Accounts (gMSAs) and Kerberos constrained delegation (KCD).

Use Case 4: SSO to Office 365 and other cloud applications

You can use AWS Managed Microsoft AD to provide SSO for cloud applications. You can use Azure AD Connect to synchronize your users into Azure AD, and then use Active Directory Federation Services (AD FS) so that your users can access Microsoft Office 365 and other SAML 2.0 cloud applications by using their AD credentials.

Integrating AWS Managed Microsoft AD with AWS SSO adds SAML capabilities to your AWS Managed Microsoft AD and / or your on-premises trusted domains. Once integrated your users can then use AWS SSO with services that support SAML, including the AWS Management Console and third-party cloud applications such as Office 365, Concur, and Salesforce without having to configure a SAML infrastructure. For a demonstration on the process of allowing your on-premise users to use AWS SSO, see the following YouTube video.

Use Case 5: Extend your on-premises AD to the AWS Cloud

If you already have an AD infrastructure and want to use it when migrating AD-aware workloads to the AWS Cloud, AWS Managed Microsoft AD can help. You can use AD trusts to connect AWS Managed Microsoft AD to your existing AD. This means your users can access AD-aware and AWS applications with their on-premises AD credentials, without needing you to synchronize users, groups, or passwords.

For example, your users can sign in to the AWS Management Console and Amazon WorkSpaces by using their existing AD user names and passwords. Also, when you use AD-aware applications such as SharePoint with AWS Managed Microsoft AD, your logged-in Windows users can access these applications without needing to enter credentials again.

You can also migrate your on-premises Active Directory (AD) domain to AWS to be free of the operational burden of your AD infrastructure using the Active Directory Migration Toolkit (ADMT) along with the Password Export Service (PES) to perform the migration.

Use Case 6: Share your directory to seamlessly join Amazon EC2 instances to a domain across AWS accounts

Sharing your directory across multiple AWS accounts enables you to manage AWS services such as Amazon EC2 easily without the need to operate a directory for each account and each VPC. You can use your directory from any AWS account and from any Amazon VPC within an AWS Region. This capability makes it easier and more cost effective to manage directory-aware workloads with a single directory across accounts and VPCs. For example, you can now manage your Windows workloads deployed in EC2 instances across multiple accounts and VPCs easily by using a single AWS Managed Microsoft AD directory.

When you share your AWS Managed Microsoft AD directory with another AWS account, you can use the Amazon EC2 console or AWS Systems Manager to seamlessly join your instances from any Amazon VPC within the account and AWS Region. You can quickly deploy your directory-aware workloads on EC2 instances by eliminating the need to manually join your instances to a domain or to deploy directories in each account and VPC. For more information, see Share your directory (p. 63).

How to administer AWS Managed Microsoft AD

This section lists all of the procedures for operating and maintaining an AWS Managed Microsoft AD environment.

Topics

Secure your AWS Managed Microsoft AD directory

This section describes considerations for securing your AWS Managed Microsoft AD environment.

Topics

- Enable secure LDAP (LDAPS) (p. 34)
- Manage compliance for AWS Managed Microsoft AD (p. 41)
- Enhance your AWS Managed Microsoft AD network security configuration (p. 43)

Manage password policies for AWS Managed Microsoft AD

AWS Managed Microsoft AD enables you to define and assign different fine-grained password and account lockout policies (also referred to as fine-grained password policies) for groups of users you manage in your AWS Managed Microsoft AD domain. When you create an AWS Managed Microsoft AD directory, a default domain policy is created and applied to the directory. This policy includes the following settings:

Policy	Setting
Enforce password history	24 passwords remembered
Maximum password age	42 days *
Minimum password age	1 day
Minimum password length	7 characters
Password must meet complexity requirements	Enabled
Store passwords using reversible encryption	Disabled

* Note: The 42 day maximum password age includes the admin password.

For example, you can assign a less strict policy setting for employees that have access to low sensitivity information only. For senior managers who regularly access confidential information you can apply more strict settings.

AWS provides a set of fine-grained password policies in AWS Managed Microsoft AD that you can configure and assign to your groups. To configure the policies, you can use standard Microsoft policy tools such as Active Directory Administrative Center. To get started with the Microsoft policy tools, see Installing the Active Directory administration tools (p. 101).

Topics

- Supported policy settings (p. 29)
- Delegate who can manage your password policies (p. 31)
- Assign password policies to your users (p. 31)

Related AWS Security blog article

- How to configure even stronger password policies to help meet your security standards by using AWS Directory Service for AWS Managed Microsoft AD

Supported policy settings

AWS Managed Microsoft AD includes five fine-grained policies with a non-editable precedence value. The policies have a number of properties you can configure to enforce the strength of passwords, and account lock-out actions in the event of login failures. You can assign the policies to zero or more Active Directory groups. If an end-user is a member of multiple groups and receives more than one password policy, Active Directory enforces the policy with the lowest precedence value.

AWS pre-defined password policies

The following table lists the five policies included in your AWS Managed Microsoft AD directory and their assigned precedence value. For more information, see Precedence (p. 30).

Policy name	Precedence
CustomerPSO-01	10
CustomerPSO-02	20
CustomerPSO-03	30
CustomerPSO-04	40
CustomerPSO-05	50

Password policy properties

You may edit the following properties in your password policies to conform to the compliance standards that meet your business needs.

- Policy name
- Enforce password history
- Minimum password length
- Minimum password age
- Maximum password age
- Store passwords using reversible encryption
- Password must meet complexity requirements

You cannot modify the precedence values for these policies. For more details about how these settings affect password enforcement, see AD DS: Fine-grained password policies on the *Microsoft TechNet* website. For general information about these policies, see Password policy on the *Microsoft TechNet* website.

Account lockout policies

You may also modify the following properties of your password policies to specify if and how Active Directory should lockout an account after login failures:

- Number of failed logon attempts allowed
- Account lockout duration
- Reset failed logon attempts after some duration

For general information about these policies, see Account lockout policy on the *Microsoft TechNet* website.

Precedence

Policies with a lower precedence value have higher priority. You assign password policies to Active Directory security groups. While you should apply a single policy to a security group, a single user may receive more than one password policy. For example, suppose `jsmith` is a member of the HR group and also a member of the MANAGERS group. If you assign **CustomerPSO-05** (which has a precedence of 50) to the HR group, and **CustomerPSO-04** (which has a precedence of 40) to MANAGERS, **CustomerPSO-04** has the higher priority and Active Directory applies that policy to `jsmith`.

If you assign multiple policies to a user or group, Active Directory determines the resultant policy as follows:

1. A policy you assign directly to the user object applies.
2. If no policy is assigned directly to the user object, the policy with the lowest precedence value of all policies received by the user as a result of group membership applies.

For additional details, see AD DS: Fine-grained password policies on the *Microsoft TechNet* website.

Delegate who can manage your password policies

You can delegate permissions to manage password policies to specific user accounts you created in your AWS Managed Microsoft AD by adding the accounts to the **AWS Delegated Fine Grained Password Policy Administrators** security group. When an account becomes a member of this group, the account has permissions to edit and configure any of the password policies listed previously (p. 30).

To delegate who can manage password policies

1. Launch Active Directory administrative center (ADAC) from any managed EC2 instance that you joined to your AWS Managed Microsoft AD domain.
2. Switch to the **Tree View** and navigate to the **AWS Delegated Groups** OU. For more information about this OU, see What gets created (p. 12).
3. Find the **AWS Delegated Fine Grained Password Policy Administrators** user group. Add any users or groups from your domain to this group.

Assign password policies to your users

User accounts that are a member of the **AWS Delegated Fine Grained Password Policy Administrators** security group can use the following procedure to assign policies to users and security groups.

To assign password policies to your users

1. Launch Active Directory administrative center (ADAC) from any managed EC2 instance that you joined to your AWS Managed Microsoft AD domain.
2. Switch to the **Tree View** and navigate to **System\Password Settings Container**.
3. Double click on the fine-grained policy you want to edit. Click **Add** to edit the policy properties, and add users or security groups to the policy. For more information about the default fine-grained policies provided with AWS Managed Microsoft AD, see AWS pre-defined password policies (p. 30).
4. To verify the password policy has been applied, run the following PowerShell command:

```
Get-ADUserResultantPasswordPolicy -Identity 'username'
```

If you do not configure any of the five password policies in your AWS Managed Microsoft AD directory, Active Directory uses the default domain group policy. For additional details on using **Password Settings Container**, see this Microsoft blog post.

Enable multi-factor authentication for AWS Managed Microsoft AD

You can enable multi-factor authentication (MFA) for your AWS Managed Microsoft AD directory to increase security when your users specify their AD credentials to access Supported Amazon Enterprise applications (p. 33). When you enable MFA, your users enter their username and password (first factor) as usual, and they must also enter an authentication code (the second factor) they obtain from your virtual or hardware MFA solution. These factors together provide additional security by preventing access to your Amazon Enterprise applications, unless users supply valid user credentials and a valid MFA code.

To enable MFA, you must have an MFA solution that is a Remote authentication dial-in user service (RADIUS) server, or you must have an MFA plugin to a RADIUS server already implemented in your on-premises infrastructure. Your MFA solution should implement One Time Passcodes (OTP) that users obtain from a hardware device or from software running on a device such as a cell phone.

RADIUS is an industry-standard client/server protocol that provides authentication, authorization, and accounting management to enable users to connect to network services. AWS Managed Microsoft AD includes a RADIUS client that connects to the RADIUS server upon which you have implemented your MFA solution. Your RADIUS server validates the username and OTP code. If your RADIUS server successfully validates the user, AWS Managed Microsoft AD then authenticates the user against AD. Upon successful AD authentication, users can then access the AWS application. Communication between the AWS Managed Microsoft AD RADIUS client and your RADIUS server require you to configure AWS security groups that enable communication over port 1812.

You can enable multi-factor authentication for your AWS Managed Microsoft AD directory by performing the following procedure. For more information about how to configure your RADIUS server to work with AWS Directory Service and MFA, see Multi-factor authentication prerequisites (p. 10).

> **Note**
> Multi-factor authentication is not available for Simple AD. However, MFA can be enabled for your AD Connector directory. For more information, see Enable multi-factor authentication for AD Connector (p. 199).

> **Note**
> MFA is a Regional feature of AWS Managed Microsoft AD. If you are using Multi-Region replication (p. 58), the following procedures must be applied separately in each Region. For more information, see Global vs Regional features (p. 58).

To enable multi-factor authentication for AWS Managed Microsoft AD

1. Identify the IP address of your RADIUS MFA server and your AWS Managed Microsoft AD directory.
2. Edit your Virtual Private Cloud (VPC) security groups to enable communications over port 1812 between your AWS Managed Microsoft AD IP end points and your RADIUS MFA server.
3. In the AWS Directory Service console navigation pane, select **Directories**.
4. Choose the directory ID link for your AWS Managed Microsoft AD directory.
5. On the **Directory details** page, do one of the following:
   - If you have multiple Regions showing under **Multi-Region replication**, select the Region where you want to enable MFA, and then choose the **Networking & security** tab. For more information, see Primary vs additional Regions (p. 59).
   - If you do not have any Regions showing under **Multi-Region replication**, choose the **Networking & security** tab.
6. In the **Multi-factor authentication** section, choose **Actions**, and then choose **Enable**.
7. On the **Enable multi-factor authentication (MFA)** page, provide the following values:

   Display label

   Provide a label name.

   RADIUS server DNS name or IP addresses

   The IP addresses of your RADIUS server endpoints, or the IP address of your RADIUS server load balancer. You can enter multiple IP addresses by separating them with a comma (e.g., `192.0.0.0,192.0.0.12`).

> **Note**
> RADIUS MFA is applicable only to authenticate access to the AWS Management Console, or to Amazon Enterprise applications and services such as WorkSpaces, Amazon QuickSight, or Amazon Chime. It does not provide MFA to Windows workloads running on EC2 instances, or for signing into an EC2 instance. AWS Directory Service does not support RADIUS Challenge/Response authentication.
> Users must have their MFA code at the time they enter their user name and password. Alternatively, you must use a solution that performs MFA out-of-band such as SMS text verification for the user. In out-of-band MFA solutions, you must make sure you set the RADIUS time-out value appropriately for your solution. When using an out-of-band MFA solution, the sign-in page will prompt the user for an MFA code. In this case, users must enter their password in both the password field and the MFA field.

Port

The port that your RADIUS server is using for communications. Your on-premises network must allow inbound traffic over the default RADIUS server port (UDP:1812) from the AWS Directory Service servers.

Shared secret code

The shared secret code that was specified when your RADIUS endpoints were created.

Confirm shared secret code

Confirm the shared secret code for your RADIUS endpoints.

Protocol

Select the protocol that was specified when your RADIUS endpoints were created.

Server timeout (in seconds)

The amount of time, in seconds, to wait for the RADIUS server to respond. This must be a value between 1 and 50.

> **Note**
> We recommend configuring your RADIUS server timeout to 20 seconds or less. If the timeout exceeds 20 seconds, the system cannot retry with another RADIUS server and may result in a timeout failure.

Max RADIUS request retries

The number of times that communication with the RADIUS server is attempted. This must be a value between 0 and 10.

Multi-factor authentication is available when the **RADIUS Status** changes to **Enabled**.

8. Choose **Enable**.

Supported Amazon Enterprise applications

All Amazon Enterprise IT applications including WorkSpaces, Amazon WorkDocs, Amazon WorkMail, Amazon QuickSight, and access to AWS Single Sign-On and AWS Management Console are supported when using AWS Managed Microsoft AD and AD Connector with MFA.

For information about how to configure basic user access to Amazon Enterprise applications, AWS Single Sign-On and the AWS Management Console using AWS Directory Service, see Enable access to AWS applications and services (p. 134) and Enable access to the AWS Management Console with AD credentials (p. 141).

Related AWS Security blog article

- How to enable multi-factor authentication for AWS services by using AWS Managed Microsoft AD and on-premises credentials

Enable secure LDAP (LDAPS)

Lightweight Directory Access Protocol (LDAP) is a standard communications protocol used to read and write data to and from Active Directory. Some applications use LDAP to add, remove, or search users and groups in Active Directory or to transport credentials for authenticating users in Active Directory. Every LDAP communication includes a client (such as an application) and a server (such as Active Directory).

By default, communications over LDAP are not encrypted. This makes it possible for a malicious user to use network monitoring software to view data packets over the wire. This is why many corporate security policies typically require that organizations encrypt all LDAP communication.

To mitigate this form of data exposure, AWS Managed Microsoft AD provides an option: You can enable LDAP over Secure Sockets Layer (SSL)/Transport Layer Security (TLS), also known as LDAPS. With LDAPS, you can improve security across the wire. You can also meet compliance requirements by encrypting all communications between your LDAP-enabled applications and AWS Managed Microsoft AD.

AWS Managed Microsoft AD provides support for LDAPS in both of the following deployment scenarios:

- **Server-side LDAPS** encrypts LDAP communications between your commercial or homegrown LDAP-aware applications (acting as LDAP clients) and AWS Managed Microsoft AD (acting as an LDAP server). For more information, see Enable server-side LDAPS using AWS Managed Microsoft AD (p. 34).
- **Client-side LDAPS** encrypts LDAP communications between AWS applications such as WorkSpaces (acting as LDAP clients) and your self-managed Active Directory (acting as LDAP server). For more information, see Enable client-side LDAPS using AWS Managed Microsoft AD (p. 37).

Topics

Enable server-side LDAPS using AWS Managed Microsoft AD

Server-side LDAPS support encrypts LDAP communications between your commercial or homegrown LDAP-aware applications and your AWS Managed Microsoft AD directory. This helps to improve security across the wire and meet compliance requirements using the Secure Sockets Layer (SSL) cryptographic protocol.

Enable server-side LDAPS

For detailed instructions on how to set up and configure server-side LDAPS and your certificate authority (CA) server, see How to Enable Server-Side LDAPS for Your AWS Managed Microsoft AD Directory on the AWS Security Blog.

You must do most of the setup from the Amazon EC2 instance that you use to manage your AWS Managed Microsoft AD domain controllers. The following steps guide you through enabling LDAPS for your domain in the AWS Cloud.

If you would like to use automation to setup your PKI Infrastructure, you can use the Microsoft Public Key Infrastructure on AWS QuickStart Guide. Specifically you will want to follow the instructions in the guide to load the template for Deploy Microsoft PKI into an existing VPC on AWS. Once you load the template, be sure to choose `AWSManaged` when you get to the **Active Directory Domain Services Type** option. If you used the QuickStart guide, you can jump directly to Step 3: Create a certificate template (p. 35).

Topics

Step 1: Delegate who can enable LDAPS

To enable server-side LDAPS, you must be a member of the Admins or AWS Delegated Enterprise Certificate Authority Administrators group in your AWS Managed Microsoft AD directory. Alternatively, you can be the default administrative user (Admin account). If you prefer, you can have a user other than the Admin account setup LDAPS. In that case, add that user to the Admins or AWS Delegated Enterprise Certificate Authority Administrators group in your AWS Managed Microsoft AD directory.

Step 2: Set up your certificate authority

Before you can enable server-side LDAPS, you must create a certificate. This certificate must be issued by a Microsoft enterprise CA server that is joined to your AWS Managed Microsoft AD domain. Once created, the certificate must be installed on each of your domain controllers in that domain. This certificate lets the LDAP service on the domain controllers listen for and automatically accept SSL connections from LDAP clients.

> **Note**
> Server-side LDAPS with AWS Managed Microsoft AD does not support certificates that are issued by a standalone CA. It also does not support certificates issued by a third-party certification authority.

Depending on your business need, you have the following options for setting up or connecting to a CA in your domain:

- **Create a subordinate Microsoft Enterprise CA** – (Recommended) With this option, you can deploy a subordinate Microsoft enterprise CA server in the AWS Cloud. The server can use Amazon EC2 so that it works with your existing root Microsoft CA. For more information about how to set up a subordinate Microsoft enterprise CA, see **Step 4: Add a Microsoft Enterprise CA to your AWS Microsoft AD directory** in How to Enable Server-Side LDAPS for Your AWS Managed Microsoft AD Directory.
- **Create a root Microsoft enterprise CA** – With this option, you can create a root Microsoft enterprise CA in the AWS Cloud using Amazon EC2 and join it to your AWS Managed Microsoft AD domain. This root CA can issue the certificate to your domain controllers. For more information about setting up a new root CA, see **Step 3: Install and configure an offline CA** in How to Enable Server-Side LDAPS for Your AWS Managed Microsoft AD Directory.

For more information about how to join your EC2 instance to the domain, see Join an EC2 instance to your AWS Managed Microsoft AD directory (p. 71).

Step 3: Create a certificate template

After your enterprise CA has been set up, you can configure the Kerberos Authentication certificate template.

To create a certificate template

1. Launch **Microsoft Windows Server Manager**. Select **Tools > Certification Authority**.
2. In the **Certificate Authority** window, expand the **Certificate Authority** tree in the left pane. Right-click **Certificate Templates**, and choose **Manage**.
3. In the **Certificate Templates Console** window, right-click **Kerberos Authentication** and choose **Duplicate Template**.
4. The **Properties of New Template** window will pop up.

5. In the **Properties of New Template** window, go to the **Compatibility** tab, and then do the following:

    a. Change **Certification Authority** to the OS that matches your CA.

    b. If a **Resulting changes** window pops up, select **OK**.

    c. Change **Certification recipient** to **Windows 8.1 / Windows Server 2012 R2**.

        Note
        AWS Managed Microsoft AD is powered by Windows Server 2012 R2.

    d. If a **Resulting changes** windows pops up, select **OK**.

6. Click the **General** tab and change the **Template display name** to **LDAPOverSSL** or any other name you would prefer.
7. Click the **Security** tab, and choose **Domain Controllers** in the **Group or user names** section. In the **Permissions for Domain Controllers** section, verify that the **Allow** check boxes for **Read**, **Enroll**, and **Autoenroll** are checked.
8. Choose **OK** to create the **LDAPOverSSL** (or the name you specified above) certificate template. Close the **Certificate Templates Console** window.
9. In the **Certificate Authority** window, right-click **Certificate Templates**, and choose **New > Certificate Template to Issue**.
10. In the **Enable Certificate Templates** window, choose **LDAPOverSSL** (or the name you specified above), and then choose **OK**.

Step 4: Add security group rules

In the final step, you must open the Amazon EC2 console and add security group rules. These rules allow your domain controllers to connect to your enterprise CA to request a certificate. To do this, you add inbound rules so that your enterprise CA can accept incoming traffic from your domain controllers. Then you add outbound rules to allow traffic from your domain controllers to the enterprise CA.

Once both rules have been configured, your domain controllers request a certificate from your enterprise CA automatically and enable LDAPS for your directory. The LDAP service on your domain controllers is now ready to accept LDAPS connections.

To configure security group rules

1. Navigate to your Amazon EC2 console at https://console.aws.amazon.com/ec2 and sign in with administrator credentials.
2. In the left pane, choose **Security Groups** under **Network & Security**.
3. In the main pane, choose the AWS security group for your CA.
4. Choose the **Inbound** tab, and then choose **Edit**.
5. In the **Edit inbound rules** dialog box, do the following:

    - Choose **Add Rule**.
    - Choose **All traffic** for **Type** and **Custom** for **Source**.
    - Enter your directory’s AWS security group (for example, sg-123456789) in the box next to **Source**.
    - Choose **Save**.

6. Now choose the AWS security group of your AWS Managed Microsoft AD directory. Choose the **Outbound** tab and then choose **Edit**.
7. In the **Edit outbound rules** dialog box, do the following:

    - Choose **Add Rule**.
    - Choose **All traffic** for **Type** and **Custom** for **Destination**.
    - Type your CA's AWS security group in the box next to **Destination**.
    - Choose **Save**.

You can test the LDAPS connection to the AWS Managed Microsoft AD directory using the LDP tool. The LDP tool comes with the Active Directory Administrative Tools. For more information, see Installing the Active Directory administration tools (p. 101).

> **Note**
> Before you test the LDAPS connection, you must wait up to 30 minutes for the subordinate CA to issue a certificate to your domain controllers.

For additional details about server-side LDAPS and to see an example use case on how to set it up, see How to Enable Server-Side LDAPS for Your AWS Managed Microsoft AD Directory on the AWS Security Blog.

Enable client-side LDAPS using AWS Managed Microsoft AD

Client-side LDAPS support in AWS Managed Microsoft AD encrypts communications between Microsoft Active Directory (AD) and AWS applications. Examples of such applications include WorkSpaces, AWS SSO, Amazon QuickSight, and Amazon Chime. This encryption helps you to better protect your organization's identity data and meet your security requirements.

Prerequisites

Before you enable client-side LDAPS, you need to meet the following requirements.

Topics

Deploy server certificates in Active Directory

In order to enable client-side LDAPS, you need to obtain and install server certificates for each domain controller in Active Directory. These certificates will be used by the LDAP service to listen for and automatically accept SSL connections from LDAP clients. You can use SSL certificates that are either issued by an in-house Active Directory Certificate Services (ADCS) deployment or purchased from a commercial issuer. For more information on Active Directory server certificate requirements, see LDAP over SSL (LDAPS) Certificate on the Microsoft website.

CA certificate requirements

A certificate authority (CA) certificate, which represents the issuer of your server certificates, is required for client-side LDAPS operation. CA certificates are matched with the server certificates that are presented by your Active Directory domain controllers to encrypt LDAP communications. Note the following CA certificate requirements:

- To register a certificate, it must be more than 90 days away from expiration.
- Certificates must be in Privacy-Enhanced Mail (PEM) format. If exporting CA certificates from inside Active Directory, choose base64 encoded X.509 (.CER) as the export file format.
- A maximum of five (5) CA certificates can be stored per AWS Managed Microsoft AD directory.
- Certificates using the RSASSA-PSS signature algorithm are not supported.
- CA certificates that chain to every server certificate in every trusted domain must be registered.

Networking requirements

AWS application LDAP traffic will run exclusively on TCP port 636, with no fallback to LDAP port 389. However, Windows LDAP communications supporting replication, trusts, and more will continue using

LDAP port 389 with Windows-native security. Configure AWS security groups and network firewalls to allow TCP communications on port 636 in AWS Managed Microsoft AD (outbound) and self-managed Active Directory (inbound). Leave open LDAP port 389 between AWS Managed Microsoft AD and self-managed Active Directory.

Enable client-side LDAPS

To enable client-side LDAPS, you import your certificate authority (CA) certificate into AWS Managed Microsoft AD, and then enable LDAPS on your directory. Upon enabling, all LDAP traffic between AWS applications and your self-managed Active Directory will flow with Secure Sockets Layer (SSL) channel encryption.

You can use two different methods to enable client-side LDAPS for your directory. You can use either the AWS Management Console method or the AWS CLI method.

> **Note**
> Client-Side LDAPS is a Regional feature of AWS Managed Microsoft AD. If you are using Multi-Region replication (p. 58), the following procedures must be applied separately in each Region. For more information, see Global vs Regional features (p. 58).

Topics

Step 1: Register a certificate in AWS Directory Service

Use either of the following methods to register a certificate in AWS Directory Service.

Method 1: To register your certificate in AWS Directory Service (AWS Management Console)

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. Choose the directory ID link for your directory.
3. On the **Directory details** page, do one of the following:

   - If you have multiple Regions showing under **Multi-Region replication**, select the Region where you want to register your certificate, and then choose the **Networking & security** tab. For more information, see Primary vs additional Regions (p. 59).
   - If you do not have any Regions showing under **Multi-Region replication**, choose the **Networking & security** tab.
4. In the **Client-side LDAPS** section, select the **Actions** menu, and then select **Register certificate**.
5. In the **Register a CA certificate** dialog box, select **Browse**, and then select the certificate and choose **Open**.
6. Choose **Register certificate**.

Method 2: To register your certificate in AWS Directory Service (AWS CLI)

- Run the following command. For the certificate data, point to the location of your CA certificate file. A certificate ID will be provided in the response.

```
aws ds register-certificate --directory-id your_directory_id --certificate-data
 file://your_file_path
```

Step 2: Check registration status

To see the status of a certificate registration or a list of registered certificates, use either of the following methods.

Method 1: To check certificate registration status in AWS Directory Service (AWS Management Console)

1. Go to the **Client-side LDAPS** section on the **Directory details** page.
2. Review the current certificate registration state that is displayed under the **Registration status** column. When the registration status value changes to **Registered**, your certificate has been successfully registered.

Method 2: To check certificate registration status in AWS Directory Service (AWS CLI)

- Run the following command. If the status value returns `Registered`, your certificate has been successfully registered.

```
aws ds list-certificates --directory-id your_directory_id
```

Step 3: Enable client-side LDAPS

Use either of the following methods to enable client-side LDAPS in AWS Directory Service.

> **Note**
> You must have successfully registered at least one certificate before you can enable client-side LDAPS.

Method 1: To enable client-side LDAPS in AWS Directory Service (AWS Management Console)

1. Go to the **Client-side LDAPS** section on the **Directory details** page.
2. Choose **Enable**. If this option is not available, verify that a valid certificate has been successfully registered, and then try again.
3. In the **Enable client-side LDAPS** dialog box, choose **Enable**.

Method 2: To enable client-side LDAPS in AWS Directory Service (AWS CLI)

- Run the following command.

```
aws ds enable-ldaps --directory-id your_directory_id --type Client
```

Step 4: Check LDAPS status

Use either of the following methods to check the LDAPS status in AWS Directory Service.

Method 1: To check LDAPS status in AWS Directory Service (AWS Management Console)

1. Go to the **Client-side LDAPS** section on the **Directory details** page.
2. If the status value is displayed as **Enabled**, LDAPS has been successfully configured.

Method 2: To check LDAPS status in AWS Directory Service (AWS CLI)

- Run the following command. If the status value returns `Enabled`, LDAPS has been successfully configured.

```
aws ds describe-ldaps-settings --directory-id your_directory_id
```

Manage client-side LDAPS

Use these commands to manage your LDAPS configuration.

You can use two different methods to manage client-side LDAPS settings. You can use either the AWS Management Console method or the AWS CLI method.

View certificate details

Use either of the following methods to see when a certificate is set to expire.

Method 1: To view certificate details in AWS Directory Service (AWS Management Console)

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. Choose the directory ID link for your directory.
3. On the **Directory details** page, do one of the following:
   - If you have multiple Regions showing under **Multi-Region replication**, select the Region where you want to view the certificate, and then choose the **Networking & security** tab. For more information, see Primary vs additional Regions (p. 59).
   - If you do not have any Regions showing under **Multi-Region replication**, choose the **Networking & security** tab.
4. In the **Client-side LDAPS** section, under **CA certificates**, information about the certificate will be displayed.

Method 2: To view certificate details in AWS Directory Service (AWS CLI)

- Run the following command. For the certificate ID, use the identifier returned by `register-certificate` or `list-certificates`.

```
aws ds describe-certificate --directory-id your_directory_id --certificate-
id your_cert_id
```

Deregister a certificate

Use either of the following methods to deregister a certificate.

> **Note**
> If only one certificate is registered, you must first disable LDAPS before you can deregister the certificate.

Method 1: To deregister a certificate in AWS Directory Service (AWS Management Console)

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. Choose the directory ID link for your directory.
3. On the **Directory details** page, do one of the following:
   - If you have multiple Regions showing under **Multi-Region replication**, select the Region where you want to deregister a certificate, and then choose the **Networking & security** tab. For more information, see Primary vs additional Regions (p. 59).

  - If you do not have any Regions showing under **Multi-Region replication**, choose the **Networking & security** tab.

4. In the **Client-side LDAPS** section, choose **Actions**, and then choose **Deregister certificate**.
5. In the **Deregister a CA certificate** dialog box, choose **Deregister**.

Method 2: To deregister a certificate in AWS Directory Service (AWS CLI)

- Run the following command. For the certificate ID, use the identifier returned by `register-certificate` or `list-certificates`.

```
aws ds deregister-certificate --directory-id your_directory_id --certificate-
id your_cert_id
```

Disable client-side LDAPS

Use either of the following methods to disable client-side LDAPS.

Method 1: To disable client-side LDAPS in AWS Directory Service (AWS Management Console)

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. Choose the directory ID link for your directory.
3. On the **Directory details** page, do one of the following:
   - If you have multiple Regions showing under **Multi-Region replication**, select the Region where you want to disable client-side LDAPS, and then choose the **Networking & security** tab. For more information, see Primary vs additional Regions (p. 59).
   - If you do not have any Regions showing under **Multi-Region replication**, choose the **Networking & security** tab.
4. In the **Client-side LDAPS** section, choose **Disable**.
5. In the **Disable client-side LDAPS** dialog box, choose **Disable**.

Method 2: To disable client-side LDAPS in AWS Directory Service (AWS CLI)

- Run the following command.

```
aws ds disable-ldaps --directory-id your_directory_id --type Client
```

Manage compliance for AWS Managed Microsoft AD

You can use AWS Managed Microsoft AD to support your Active Directory–aware applications, in the AWS Cloud, that are subject to the following compliance requirements. However, your applications will not adhere to compliance requirements if you use Simple AD or AD Connector.

Supported compliance standards

AWS Managed Microsoft AD has undergone auditing for the following standards and is eligible for use as part of solutions for which you need to obtain compliance certification.

FR FedRAMP	AWS Managed Microsoft AD meets Federal Risk and Authorization Management Program (FedRAMP) security requirements and has received a FedRAMP Joint Authorization Board (JAB) Provisional Authority to Operate (P-ATO) at the FedRAMP Moderate and High Baseline. For more information about FedRAMP, see FedRAMP compliance.
PCI Security Standards Council ® PARTICIPATING ORGANIZATION™	AWS Managed Microsoft AD has an Attestation of Compliance for Payment Card Industry (PCI) Data Security Standard (DSS) version 3.2 at Service Provider Level 1. Customers who use AWS products and services to store, process, or transmit cardholder data can use AWS Managed Microsoft AD as they manage their own PCI DSS compliance certification. For more information about PCI DSS, including how to request a copy of the AWS PCI Compliance Package, see PCI DSS level 1. Importantly, you must configure fine-grained password policies in AWS Managed Microsoft AD to be consistent with PCI DSS version 3.2 standards. For details on which policies must be enforced, see the section below titled Enable PCI Compliance for Your AWS Managed Microsoft AD Directory.
HIPAA	AWS has expanded its Health Insurance Portability and Accountability Act (HIPAA) compliance program to include AWS Managed Microsoft AD as a HIPAA eligible service. If you have an executed Business Associate Agreement (BAA) with AWS, you can use AWS Managed Microsoft AD to help build your HIPAA-compliant applications. AWS offers a HIPAA-focused whitepaper for customers who are interested in learning more about how they can leverage AWS for the processing and storage of health information. For more information, see HIPAA compliance.

Shared responsibility

Security, including FedRAMP, HIPAA and PCI compliance, is a shared responsibility. It is important to understand that AWS Managed Microsoft AD compliance status does not automatically apply to applications that you run in the AWS Cloud. You need to ensure that your use of AWS services complies with the standards.

For a complete list of all the various AWS compliance programs that AWS Managed Microsoft AD supports, see AWS services in scope by compliance program.

Enable PCI compliance for your AWS Managed Microsoft AD directory

To enable PCI compliance for your AWS Managed Microsoft AD directory, you must configure fine-grained password policies as specified in the PCI DSS Attestation of Compliance (AOC) and Responsibility Summary document provided by AWS Artifact.

For more information about using fine-grained password policies, see Manage password policies for AWS Managed Microsoft AD (p. 29).

Enhance your AWS Managed Microsoft AD network security configuration

The AWS Security Group that is provisioned for the AWS Managed Microsoft AD directory is configured with the minimum inbound network ports required to support all known use cases for your AWS Managed Microsoft AD directory. For more information on the provisioned AWS Security Group, see What gets created (p. 12).

To further enhance the network security of your AWS Managed Microsoft AD directory you can modify the AWS Security Group based on common scenarios listed below.

Topics

AWS applications only support

All user accounts are provisioned only in your AWS Managed Microsoft AD to be used with supported AWS applications, such as the following:

- Amazon Chime
- Amazon Connect
- Amazon QuickSight
- AWS Single Sign-On
- Amazon WorkDocs
- Amazon WorkMail
- AWS Client VPN
- AWS Management Console

You can use the following AWS Security Group configuration to block all non-essential traffic to your AWS Managed Microsoft AD domain controllers.

Note

- The following are not compatible with this AWS Security Group configuration:
  - Amazon EC2 instances
  - Amazon FSx

    - Amazon RDS for MySQL
    - Amazon RDS for Oracle
    - Amazon RDS for PostgreSQL
    - Amazon RDS for SQL Server
    - WorkSpaces
    - Active Directory trusts
    - Domain joined clients or servers

Inbound Rules

None.

Outbound Rules

None.

AWS applications only with trust support

All user accounts are provisioned in your AWS Managed Microsoft AD or trusted Active Directory to be used with supported AWS applications, such as the following:

- Amazon Chime
- Amazon Connect
- Amazon QuickSight
- AWS Single Sign-On
- Amazon WorkDocs
- Amazon WorkMail
- Amazon WorkSpaces
- AWS Client VPN
- AWS Management Console

You can modify the provisioned AWS Security Group configuration to block all non-essential traffic to your AWS Managed Microsoft AD domain controllers.

> **Note**
>
> - The following are not compatible with this AWS Security Group configuration:
>   - Amazon EC2 instances
>   - Amazon FSx
>   - Amazon RDS for MySQL
>   - Amazon RDS for Oracle
>   - Amazon RDS for PostgreSQL
>   - Amazon RDS for SQL Server
>   - WorkSpaces
>   - Active Directory trusts
>   - Domain joined clients or servers
> - This configuration requires you to ensure the “On-premise CIDR” network is secure.
> - TCP 445 is used for trust creation only and can be removed after the trust has been established.
> - TCP 636 is only required when LDAP over SSL is in use.

Inbound Rules

Protocol	Port range	Source	Type of traffic	Active Directory usage
TCP & UDP	53	On-premise CIDR	DNS	User and computer authentication, name resolution, trusts
TCP & UDP	88	On-premise CIDR	Kerberos	User and computer authentication, forest level trusts
TCP & UDP	389	On-premise CIDR	LDAP	Directory, replication, user and computer authentication group policy, trusts
TCP & UDP	464	On-premise CIDR	Kerberos change / set password	Replication, user and computer authentication, trusts
TCP	445	On-premise CIDR	SMB / CIFS	Replication, user and computer authentication, group policy trusts
TCP	135	On-premise CIDR	Replication	RPC, EPM
TCP	636	On-premise CIDR	LDAP SSL	Directory, replication, user and computer authentication group policy, trusts
TCP	49152 - 65535	On-premise CIDR	RPC	Replication, user and computer authentication, group policy, trusts
TCP	3268 - 3269	On-premise CIDR	LDAP GC & LDAP GC SSL	Directory, replication, user and computer authentication group policy, trusts
UDP	123	On-premise CIDR	Windows Time	Windows Time, trusts

Outbound Rules

Protocol	Port range	Source	Type of traffic	Active Directory usage
All	All	On-premise CIDR	All traffic	

AWS applications and native Active Directory workload support

User accounts are provisioned only in your AWS Managed Microsoft AD to be used with supported AWS applications, such as the following:

- Amazon Chime
- Amazon Connect
- Amazon EC2 instances
- Amazon FSx
- Amazon QuickSight
- Amazon RDS for MySQL
- Amazon RDS for Oracle
- Amazon RDS for PostgreSQL
- Amazon RDS for SQL Server
- AWS Single Sign-On
- Amazon WorkDocs
- Amazon WorkMail
- WorkSpaces
- AWS Client VPN
- AWS Management Console

You can modify the provisioned AWS Security Group configuration to block all non-essential traffic to your AWS Managed Microsoft AD domain controllers.

> **Note**
>
> - Active Directory trusts cannot be created and maintained between your AWS Managed Microsoft AD directory and on-premise domain.
> - It requires you to ensure the “Client CIDR” network is secure.
> - TCP 636 is only required when LDAP over SSL is in use.
> - If you want to use an Enterprise CA with this configuration you will need to create an outbound rule “TCP, 443, CA CIDR”.

Inbound Rules

Protocol	Port range	Source	Type of traffic	Active Directory usage
TCP & UDP	53	Client CIDR	DNS	User and computer authentication, name resolution, trusts

Protocol	Port range	Source	Type of traffic	Active Directory usage
TCP & UDP	88	Client CIDR	Kerberos	User and computer authentication, forest level trusts
TCP & UDP	389	Client CIDR	LDAP	Directory, replication, user and computer authentication group policy, trusts
TCP & UDP	445	Client CIDR	SMB / CIFS	Replication, user and computer authentication, group policy trusts
TCP & UDP	464	Client CIDR	Kerberos change / set password	Replication, user and computer authentication, trusts
TCP	135	Client CIDR	Replication	RPC, EPM
TCP	636	Client CIDR	LDAP SSL	Directory, replication, user and computer authentication group policy, trusts
TCP	49152 - 65535	Client CIDR	RPC	Replication, user and computer authentication, group policy, trusts
TCP	3268 - 3269	Client CIDR	LDAP GC & LDAP GC SSL	Directory, replication, user and computer authentication group policy, trusts
TCP	9389	Client CIDR	SOAP	AD DS web services
UDP	123	Client CIDR	Windows Time	Windows Time, trusts
UDP	138	Client CIDR	DFSN & NetLogon	DFS, group policy

Outbound Rules

None.

AWS applications and native Active Directory workload support with trust support

All user accounts are provisioned in your AWS Managed Microsoft AD or trusted Active Directory to be used with supported AWS applications, such as the following:

- Amazon Chime
- Amazon Connect
- Amazon EC2 instances
- Amazon FSx
- Amazon QuickSight
- Amazon RDS for MySQL
- Amazon RDS for Oracle
- Amazon RDS for PostgreSQL
- Amazon RDS for SQL Server
- AWS Single Sign-On
- Amazon WorkDocs
- Amazon WorkMail
- WorkSpaces
- AWS Client VPN
- AWS Management Console

You can modify the provisioned AWS Security Group configuration to block all non-essential traffic to your AWS Managed Microsoft AD domain controllers.

> **Note**
>
> - It requires you to ensure the “On-premise CIDR” and “Client CIDR” networks are secure.
> - TCP 445 with the “On-premise CIDR” is used for trust creation only and can be removed after the trust has been established.
> - TCP 445 with the “Client CIDR” should be left open as it is required for Group Policy processing.
> - TCP 636 is only required when LDAP over SSL is in use.
> - If you want to use an Enterprise CA with this configuration you will need to create an outbound rule “TCP, 443, CA CIDR”.

Inbound Rules

Protocol	Port range	Source	Type of traffic	Active Directory usage
TCP & UDP	53	On-premise CIDR	DNS	User and computer authentication, name resolution, trusts
TCP & UDP	88	On-premise CIDR	Kerberos	User and computer authentication, forest level trusts

Protocol	Port range	Source	Type of traffic	Active Directory usage
TCP & UDP	389	On-premise CIDR	LDAP	Directory, replication, user and computer authentication group policy, trusts
TCP & UDP	464	On-premise CIDR	Kerberos change / set password	Replication, user and computer authentication, trusts
TCP	445	On-premise CIDR	SMB / CIFS	Replication, user and computer authentication, group policy trusts
TCP	135	On-premise CIDR	Replication	RPC, EPM
TCP	636	On-premise CIDR	LDAP SSL	Directory, replication, user and computer authentication group policy, trusts
TCP	49152 - 65535	On-premise CIDR	RPC	Replication, user and computer authentication, group policy, trusts
TCP	3268 - 3269	On-premise CIDR	LDAP GC & LDAP GC SSL	Directory, replication, user and computer authentication group policy, trusts
UDP	123	On-premise CIDR	Windows Time	Windows Time, trusts
TCP & UDP	53	Client CIDR	DNS	User and computer authentication, name resolution, trusts
TCP & UDP	88	Client CIDR	Kerberos	User and computer authentication, forest level trusts

Protocol	Port range	Source	Type of traffic	Active Directory usage
TCP & UDP	389	Client CIDR	LDAP	Directory, replication, user and computer authentication group policy, trusts
TCP & UDP	445	Client CIDR	SMB / CIFS	Replication, user and computer authentication, group policy trusts
TCP & UDP	464	Client CIDR	Kerberos change / set password	Replication, user and computer authentication, trusts
TCP	135	Client CIDR	Replication	RPC, EPM
TCP	636	Client CIDR	LDAP SSL	Directory, replication, user and computer authentication group policy, trusts
TCP	49152 - 65535	Client CIDR	RPC	Replication, user and computer authentication, group policy, trusts
TCP	3268 - 3269	Client CIDR	LDAP GC & LDAP GC SSL	Directory, replication, user and computer authentication group policy, trusts
TCP	9389	Client CIDR	SOAP	AD DS web services
UDP	123	Client CIDR	Windows Time	Windows Time, trusts
UDP	138	Client CIDR	DFSN & NetLogon	DFS, group policy

Outbound Rules

Protocol	Port range	Source	Type of traffic	Active Directory usage
All	All	On-premise CIDR	All traffic	

Monitor your AWS Managed Microsoft AD

You can monitor your AWS Managed Microsoft AD directory with the following methods:

Topics

Understanding your directory status

The following are the various statuses for a directory.

Active

The directory is operating normally. No issues have been detected by the AWS Directory Service for your directory.

Creating

The directory is currently being created. Directory creation typically takes between 20 to 45 minutes but may vary depending on the system load.

Deleted

The directory has been deleted. All resources for the directory have been released. Once a directory enters this state, it cannot be recovered.

Deleting

The directory is currently being deleted. The directory will remain in this state until it has been completely deleted. Once a directory enters this state, the delete operation cannot be cancelled, and the directory cannot be recovered.

Failed

The directory could not be created. Please delete this directory. If this problem persists, please contact the AWS Support Center.

Impaired

The directory is running in a degraded state. One or more issues have been detected, and not all directory operations may be working at full operational capacity. There are many potential reasons for the directory being in this state. These include normal operational maintenance activity such as patching or EC2 instance rotation, temporary hot spotting by an application on one of your domain controllers, or changes you made to your network that inadvertently disrupt directory communications. For more information, see either Troubleshooting AWS Managed Microsoft AD (p. 175), Troubleshooting AD Connector (p. 223), Troubleshooting Simple AD (p. 272). For normal maintenance related issues, AWS resolves these issues within 40 minutes. If after reviewing the troubleshooting topic, your directory is in an Impaired state longer than 40 minutes, we recommend that you contact the AWS Support Center.

> **Important**
> Do not restore a snapshot while a directory is in an Impaired state. It is rare that snapshot restore is necessary to resolve impairments. For more information, see Snapshot or restore your directory (p. 128).

Inoperable

The directory is not functional. All directory endpoints have reported issues.

Requested

A request to create your directory is currently pending.

RestoreFailed

Restoring the directory from a snapshot failed. Please retry the restore operation. If this continues, try a different snapshot, or contact the AWS Support Center.

Restoring

The directory is currently being restored from an automatic or manual snapshot. Restoring from a snapshot typically takes several minutes, depending on the size of the directory data in the snapshot.

For more information, see Simple AD directory status reasons (p. 274).

Configure directory status notifications

Using Amazon Simple Notification Service (Amazon SNS), you can receive email or text (SMS) messages when the status of your directory changes. You get notified if your directory goes from an Active status to an Impaired or Inoperable status. You also receive a notification when the directory returns to an Active status.

How It Works

Amazon SNS uses “topics” to collect and distribute messages. Each topic has one or more subscribers who receive the messages that have been published to that topic. Using the steps below you can add AWS Directory Service as publisher to an Amazon SNS topic. When AWS Directory Service detects a change in your directory’s status, it publishes a message to that topic, which is then sent to the topic's subscribers.

You can associate multiple directories as publishers to a single topic. You can also add directory status messages to topics that you’ve previously created in Amazon SNS. You have detailed control over who can publish to and subscribe to a topic. For complete information about Amazon SNS, see What is Amazon SNS?.

> **Note**
> Directory status notifications is a Regional feature of AWS Managed Microsoft AD. If you are using Multi-Region replication (p. 58), the following procedures must be applied separately in each Region. For more information, see Global vs Regional features (p. 58).

To enable SNS messaging for your directory

1. Sign in to the AWS Management Console and open the AWS Directory Service console at https://console.aws.amazon.com/directoryservicev2/.
2. On the **Directories** page, choose your directory ID.
3. On the **Directory details** page, do one of the following:
   - If you have multiple Regions showing under **Multi-Region replication**, select the Region where you want to enable SNS messaging, and then choose the **Maintenance** tab. For more information, see Primary vs additional Regions (p. 59).
   - If you do not have any Regions showing under **Multi-Region replication**, choose the **Maintenance** tab.
4. In the **Directory monitoring** section, choose **Actions**, and then select **Create notification**.

5. On the **Create notification** page, select **Choose a notification type**, and then choose **Create a new notification**. Alternatively, if you already have an existing SNS topic, you can choose **Associate existing SNS topic** to send status messages from this directory to that topic.

   Note
   If you choose **Create a new notification** but then use the same topic name for an SNS topic that already exists, Amazon SNS does not create a new topic, but just adds the new subscription information to the existing topic.
   If you choose **Associate existing SNS topic**, you will only be able to choose an SNS topic that is in the same Region as the directory.
6. Choose the **Recipient type** and enter the **Recipient** contact information. If you enter a phone number for SMS, use numbers only. Do not include dashes, spaces, or parentheses.
7. (Optional) Provide a name for your topic and an SNS display name. The display name is a short name up to 10 characters that is included in all SMS messages from this topic. When using the SMS option, the display name is required.

   Note
   If you are logged in using an IAM user or role that has only the DirectoryServiceFullAccess managed policy, your topic name must start with "DirectoryMonitoring". If you'd like to further customize your topic name you'll need additional privileges for SNS.
8. Choose **Create**.

If you want to designate additional SNS subscribers, such as an additional email address, Amazon SQS queues or AWS Lambda, you can do this from the Amazon SNS console at https://console.aws.amazon.com/sns/v3/home.

To remove directory status messages from a topic

1. Sign in to the AWS Management Console and open the AWS Directory Service console at https://console.aws.amazon.com/directoryservicev2/.
2. On the **Directories** page, choose your directory ID.
3. On the **Directory details** page, do one of the following:
   - If you have multiple Regions showing under **Multi-Region replication**, select the Region where you want to remove status messages, and then choose the **Maintenance** tab. For more information, see Primary vs additional Regions (p. 59).
   - If you do not have any Regions showing under **Multi-Region replication**, choose the **Maintenance** tab.
4. In the **Directory monitoring** section, select an SNS topic name in the list, choose **Actions**, and then select **Remove**.
5. Choose **Remove**.

This removes your directory as a publisher to the selected SNS topic. If you want to delete the entire topic, you can do this from the Amazon SNS console at https://console.aws.amazon.com/sns/v3/home.

> **Note**
> Before deleting an Amazon SNS topic using the SNS console, you should ensure that a directory is not sending status messages to that topic.
> If you delete an Amazon SNS topic using the SNS console, this change will not immediately be reflected within the Directory Services console. You would only be notified the next time a directory publishes a notification to the deleted topic, in which case you would see an updated status on the directory's **Monitoring** tab indicating the topic could not be found.
> Therefore, to avoid missing important directory status messages, before deleting any topic that receives messages from AWS Directory Service, associate your directory with a different Amazon SNS topic.

Review your AWS Managed Microsoft AD directory logs

Security logs from AWS Managed Microsoft AD domain controller instances are archived for a year. You can also configure your AWS Managed Microsoft AD directory to forward domain controller logs to Amazon CloudWatch Logs in near real time. For more information, see Enable log forwarding (p. 55).

AWS logs the following events for compliance.

Monitoring category	Policy setting	Audit state
Account Logon	Audit Credential Validation	Success, Failure
	Audit Other Account Logon Events	Success, Failure
Account Management	Audit Computer Account Management	Success, Failure
	Audit Other Account Management Events	Success, Failure
	Audit Security Group Management	Success, Failure
	Audit User Account Management	Success, Failure
Detailed Tracking	Audit DPAPI Activity	Success, Failure
	Audit PNP Activity	Success
	Audit Process Creation	Success, Failure
DS Access	Audit Directory Service Access	Success, Failure
	Audit Directory Service Changes	Success, Failure
Logon/Logoff	Audit Account Lockout	Success, Failure
	Audit Logoff	Success
	Audit Logon	Success, Failure
	Audit Other Logon/Logoff Events	Success, Failure
	Audit Special Logon	Success, Failure
Object Access	Audit Other Object Access Events	Success, Failure
	Audit Removable Storage	Success, Failure
	Audit Central Access Policy Staging	Success, Failure
Policy Change	Audit Policy Change	Success, Failure
	Audit Authentication Policy Change	Success, Failure

Monitoring category	Policy setting	Audit state
	Audit Authorization Policy Change	Success, Failure
	Audit MPSSVC Rule-Level Policy Change	Success
	Audit Other Policy Change Events	Failure
Privilege Use	Audit Sensitive Privilege Use	Success, Failure
System	Audit IPsec Driver	Success, Failure
	Audit Other System Events	Success, Failure
	Audit Security State Change	Success, Failure
	Audit Security System Extension	Success, Failure
	Audit System Integrity	Success, Failure

Enable log forwarding

You can use either the AWS Directory Service console or APIs to forward domain controller security event logs to Amazon CloudWatch Logs. This helps you to meet your security monitoring, audit, and log retention policy requirements by providing transparency of the security events in your directory.

CloudWatch Logs can also forward these events to other AWS accounts, AWS services, or third party applications. This makes it easier for you to centrally monitor and configure alerts to detect and respond proactively to unusual activities in near real time.

Once enabled, you can then use the CloudWatch Logs console to retrieve the data from the log group you specified when you enabled the service. This log group contains the security logs from your domain controllers.

For more information about log groups and how to read their data, see Working with log groups and log streams in the *Amazon CloudWatch Logs User Guide*.

> **Note**
> Log forwarding is a Regional feature of AWS Managed Microsoft AD. If you are using Multi-Region replication (p. 58), the following procedures must be applied separately in each Region. For more information, see Global vs Regional features (p. 58).

To enable log forwarding

1. In the AWS Directory Service console navigation pane, choose **Directories**.
2. Choose the directory ID of the AWS Managed Microsoft AD directory that you want to share.
3. On the **Directory details** page, do one of the following:
   - If you have multiple Regions showing under **Multi-Region replication**, select the Region where you want to enable log forwarding, and then choose the **Networking & security** tab. For more information, see Primary vs additional Regions (p. 59).
   - If you do not have any Regions showing under **Multi-Region replication**, choose the **Networking & security** tab.
4. In the **Log forwarding** section, choose **Enable**.
5. On the **Enable log forwarding to CloudWatch** dialog, choose either of the following options:

   a. Select **Create a new CloudWatch log group**, under **CloudWatch Log group name**, specify a name that you can refer to in CloudWatch Logs.
   b. Select **Choose an existing CloudWatch log group**, and under **Existing CloudWatch log groups**, select a log group from the menu.

6. Review the pricing information and link, and then choose **Enable**.

To disable log forwarding

1. In the AWS Directory Service console navigation pane, choose **Directories**.
2. Choose the directory ID of the AWS Managed Microsoft AD directory that you want to share.
3. On the **Directory details** page, do one of the following:
   - If you have multiple Regions showing under **Multi-Region replication**, select the Region where you want to disable log forwarding, and then choose the **Networking & security** tab. For more information, see Primary vs additional Regions (p. 59).
   - If you do not have any Regions showing under **Multi-Region replication**, choose the **Networking & security** tab.
4. In the **Log forwarding** section, choose **Disable**.
5. Once you've read the information in the **Disable log forwarding** dialog, choose **Disable**.

Using the CLI to enable log forwarding

Before you can use the `ds create-log-subscription` command, you must first create an Amazon CloudWatch log group and then create an IAM resource policy that will grant the necessary permission to that group. To enable log forwarding using the CLI, complete all of the steps below.

Step 1: Create a log group in CloudWatch Logs

Create a log group that will be used to receive the security logs from your domain controllers. We recommend pre-pending the name with `/aws/directoryservice/`, but that is not required. For example:

EXAMPLE CLI COMMAND

```
aws logs create-log-group --log-group-name '/aws/directoryservice/d-9876543210'
```

EXAMPLE POWERSHELL COMMAND

```
New-CWLLogGroup -LogGroupName '/aws/directoryservice/d-9876543210'
```

For instructions on how to create a CloudWatch Logs group, see Create a log group in CloudWatch Logs in the *Amazon CloudWatch Logs User Guide*.

Step 2: Create a CloudWatch Logs resource policy in IAM

Create a CloudWatch Logs resource policy granting AWS Directory Service rights to add logs into the new log group you created in Step 1. You can either specify the exact ARN to the log group to limit AWS Directory Service's access to other log groups or use a wild card to include all log groups. The following sample policy uses the wild card method to identify that all log groups that start with `/aws/directoryservice/` for the AWS account where your directory resides will be included.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "ds.amazonaws.com"
            },
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:YOUR_REGION:YOUR_ACCOUNT_NUMBER:log-group:/aws/
directoryservice/*"
        }
    ]
}
```

You will need to save this policy to a text file (for example DSPolicy.json) on your local workstation as you will need to run it from the CLI. For example:

EXAMPLE CLI COMMAND

```
aws logs put-resource-policy --policy-name DSLogSubscription --policy-document
file://DSPolicy.json
```

EXAMPLE POWERSHELL COMMAND

```
$PolicyDocument = Get-Content .\DSPolicy.json -Raw
```

```
Write-CWLResourcePolicy -PolicyName DSLogSubscription -PolicyDocument
$PolicyDocument
```

Step 3: Create an AWS Directory Service log subscription

In this final step, you can now proceed to enable log forwarding by creating the log subscription. For example:

EXAMPLE CLI COMMAND

```
aws ds create-log-subscription --directory-id 'd-9876543210' --log-group-name
'/aws/directoryservice/d-9876543210'
```

EXAMPLE POWERSHELL COMMAND

```
New-DSLogSubscription -DirectoryId 'd-9876543210' -LogGroupName '/aws/
directoryservice/d-9876543210'
```

Monitor your domain controllers with performance metrics

AWS Directory Service integrates with Amazon CloudWatch to help provide you with important performance metrics for each domain controller in your directory. This means that you can monitor domain controller performance counters, such as CPU and memory utilization. You can also configure alarms and initiate automated actions to respond to periods of high utilization. For example, you can configure an alarm for domain controller CPU utilization above 70 percent and create an SNS topic to notify you when this occurs. You can use this SNS topic to initiate automation, such as AWS Lambda functions, to increase the number of domain controllers to your directory.

For more information about monitoring your domain controllers, see Use Amazon CloudWatch metrics to determine when to add domain controllers (p. 145).

Find domain controller performance metrics in CloudWatch

In the CloudWatch console, metrics for a given service are grouped first by the service's namespace. You can add metric filters that are subordinate to that namespace. Use the following procedure to locate the correct namespace and subordinate metric that is required to set up AWS Managed Microsoft AD domain controller metrics in CloudWatch.

To find domain controller metrics in the CloudWatch console

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose **Metrics**.
3. From the list of metrics, select the **Directory Service** namespace, and then from the list, select the **AWS Managed Microsoft AD** metric.

For instructions on how to set up domain controller metrics using the CloudWatch console, see How to automate AWS Managed Microsoft AD scaling based on utilization metrics in the AWS Security Blog.

Multi-Region replication

Multi-Region replication can be used to automatically replicate your AWS Managed Microsoft AD directory data across multiple Regions. This replication can improve performance for users and applications in disperse geographic locations. AWS Managed Microsoft AD uses native Active Directory replication to replicate your directory's data securely to the new Region.

Multi-Region replication is only supported for the **Enterprise Edition** of AWS Managed Microsoft AD. You can use automated multi-Region replication in all Regions where AWS Managed Microsoft AD is available.

Benefits

With multi-Region replication in AWS Managed Microsoft AD, AD-aware applications use the directory locally for high performance and the multi-Region feature for resiliency. You can use multi-Region replication with AD-aware applications like SharePoint and SQL Server Always On as well as AWS services like Amazon RDS for SQL Server and FSx for Windows File Server. The following are additional benefits of multi-Region replication.

- It lets you deploy a single AWS Managed Microsoft AD instance globally, quickly, and eliminates the heavy lifting of self-managing a global AD infrastructure.
- It makes it easier and more cost-effective for you to deploy and manage Windows and Linux workloads in multiple AWS Regions. Automated multi-Region replication enables optimal performance in your global AD-aware applications. All applications deployed in Windows or Linux instances use AWS Managed Microsoft AD locally in the Region, which enables responses to user requests from the closest Region possible.
- It provides multi-Region resiliency. Deployed in the highly available AWS managed infrastructure, AWS Managed Microsoft AD handles automated software updates, monitoring, recovery, and the security of the underlying AD infrastructure across all Regions. This allows you to focus on building your applications.

Topics

Global vs Regional features

When you add an AWS Region to your directory using multi-Region replication, AWS Directory Service enhances the scope of all features so that they become Region-aware. These features are listed on various tabs of the details page that appears when you choose the ID of a directory in the AWS Directory Service console. This means that all features are enabled, configured, or managed based on the Region that you select in the **Multi-Region replication** section of the console. Changes you make to features in each Region are either applied globally or per Region.

Global features

Any changes that you make to global features while the Primary Region (p. 59) is selected will be applied across all Regions.

You can identify the features that are used globally on the **Directory details** page because they display **Applied to all replicated Regions** next to them. Alternatively, if you selected another Region in the list that is not the primary Region, you can identify the globally used features because they display **Inherited from primary Region**.

Regional features

Any changes that you make to a feature in an Additional Region (p. 59) will be applied only to that Region.

You can identify the features that are Regional on the **Directory details** page because they do ***not*** display **Applied to all replicated Regions** or **Inherited from primary Region** next to them.

Primary vs additional Regions

With multi-Region replication, AWS Managed Microsoft AD uses the following two types of Regions to differentiate how global or Regional features should be applied across your directory.

Primary Region

The initial Region where you first created your directory is referred to as the *primary* Region. You can perform only global directory level operations such as creating Active Directory trusts and updating the AD schema from the primary Region.

The primary Region can always be identified as the first Region showing at the top of the list in the **Multi-Region replication** section, and ends with **- Primary**. For example, **US East (N. Virginia) - Primary**.

Any changes that you make to Global features (p. 59) while the primary Region is selected will be applied across all Regions.

You can only add Regions while the primary Region is selected. For more information, see Add a replicated Region (p. 61).

Additional Region

Any Regions that you have added to your directory are referred to as *additional* Regions.

Although some features can be managed globally for all Regions, others are managed individually per Region. To manage a feature for an additional Region (non-primary Region), you must first select the additional Region from the list in the **Multi-Region replication** section on the **Directory details** page. Then you can proceed to manage the feature.

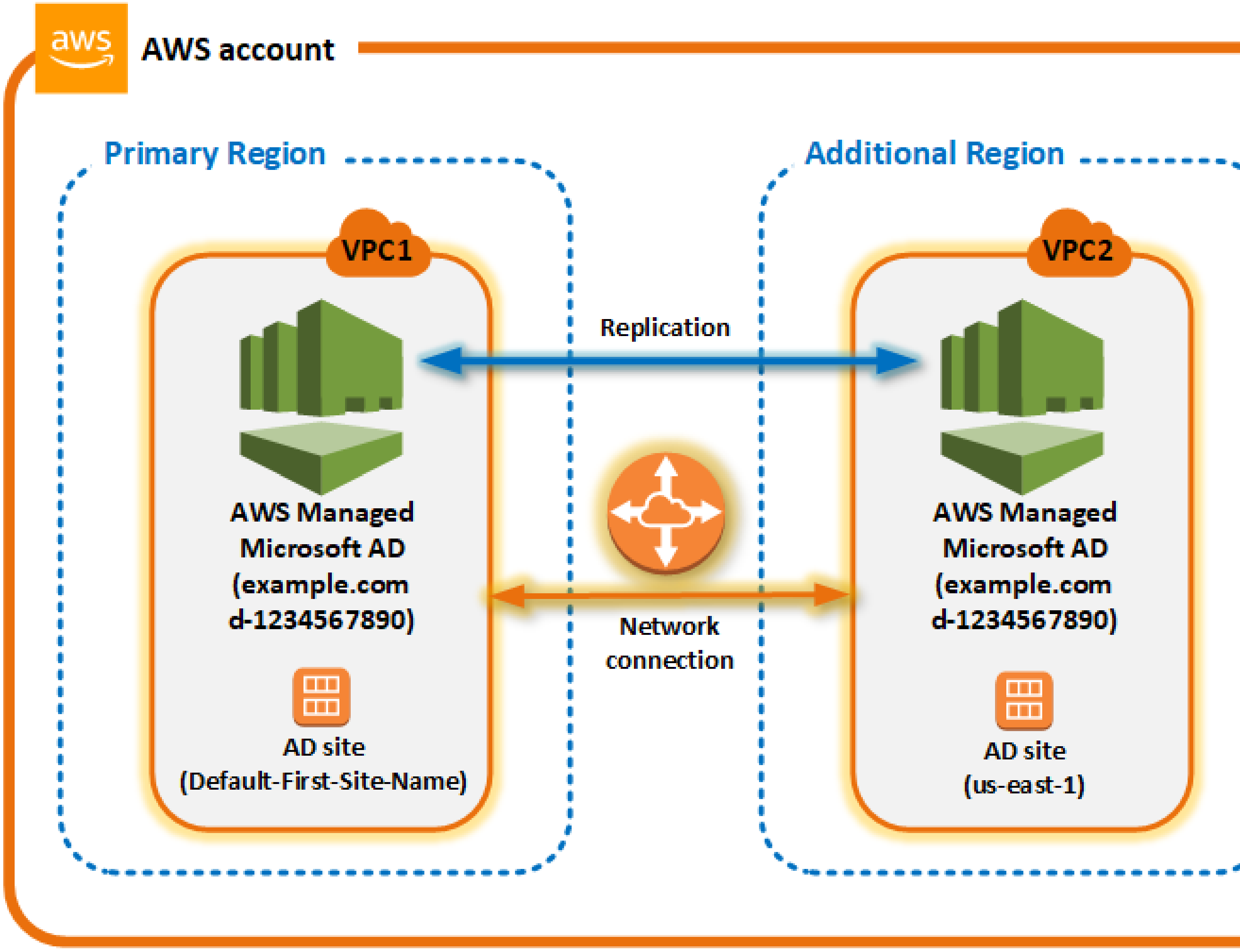
Any changes that you make to Regional features (p. 59) while an additional Region is selected will be applied only to that Region.

How multi-Region replication works

With the multi-Region replication feature, AWS Managed Microsoft AD eliminates the undifferentiated heavy lifting of managing a global AD infrastructure. When configured, AWS replicates all customer directory data including users, groups, group policies, and schema across multiple AWS Regions.

Once a new Region has been added, the following operations automatically occur as shown in the illustration:

- AWS Managed Microsoft AD creates two domain controllers in the selected VPC and deploys them to the new Region in the same AWS account. Your directory identifier (`directory_id`) remains the same across all Regions. You can add additional domain controllers later if you want.
- AWS Managed Microsoft AD configures the networking connection between the primary Region and the new Region.
- AWS Managed Microsoft AD creates a new Active Directory site and gives it the same name as the Region, such as us-east-1. You can also rename this later using the Active Directory Sites and Services tool.
- AWS Managed Microsoft AD replicates all AD objects and configurations to the new Region, including users, groups, group policies, AD trusts, organizational units, and AD schema. Active Directory site links are configured to use Change Notification. With change notification between sites enabled, changes propagate to the remote site with the same frequency that they are propagated within the source site, including changes that warrant urgent replication.
- If this is the first Region you've added, AWS Managed Microsoft AD makes all features multi-Region aware. For more information, see Global vs Regional features (p. 58).


aws
AWS account
Primary Region
Additional Region
VPC1
VPC2
Replication
AWS Managed Microsoft AD (example.com d-1234567890)
AWS Managed Microsoft AD (example.com d-1234567890)
Network connection
AD site (Default-First-Site-Name)
AD site (us-east-1)

Active Directory sites

Multi-Region replication supports multiple Active Directory sites (one AD site per Region). When a new Region is added, it is given the same name as the Region—for example, us-east-1. You can also rename this later using Active Directory Sites and Services.

AWS services

AWS services such as Amazon RDS for SQL Server and Amazon FSx connect to the local instances of the global directory. This allows your users to sign in once to AD-aware applications that run in AWS as well as AWS services like Amazon RDS for SQL Server in any AWS Region. To do so, users need credentials from AWS Managed Microsoft AD or on-premises Active Directory when you have a trust with your AWS Managed Microsoft AD.

You can use the following AWS services with the multi-Region replication feature.

- Amazon EC2
- FSx for Windows File Server
- Amazon RDS for SQL Server
- Amazon RDS for Oracle
- Amazon RDS for MySQL
- Amazon RDS for PostgreSQL
- Amazon RDS for MariaDB
- Amazon Aurora for MySQL
- Amazon Aurora for PostgreSQL

Failover

In the event that all domain controllers in one Region are down, AWS Managed Microsoft AD recovers the domain controllers and replicates the directory data automatically. Meanwhile domain controllers in other Regions stay up and running.

Add a replicated Region

When you add a Region using the Multi-Region replication (p. 58) feature, AWS Managed Microsoft AD creates two domain controllers in the selected AWS Region, Amazon Virtual Private Cloud (VPC), and subnet. AWS Managed Microsoft AD also creates the related security groups that enable Windows workloads to connect to your directory in the new Region. It also creates these resources using the same AWS account where your directory is already deployed. You do this by choosing the Region, specifying the VPC, and providing the configurations for the new Region.

Multi-Region replication is only supported for the **Enterprise Edition** of AWS Managed Microsoft AD.

Prerequisites

Before you proceed with the steps to add a new replication Region, we recommend that you first review the following prerequisite tasks.

- Verify that you have the necessary AWS Identity and Access Management (IAM) permissions, Amazon VPC setup, and the subnet setup in the new Region to which you want to replicate the directory.
- If you want to use your existing on-premises Active Directory credentials to access and manage AD-aware workloads in AWS, you must create an AD trust between AWS Managed Microsoft AD and your

on-premises AD infrastructure. For more information about trusts, see Connect to your existing AD infrastructure (p. 104).

Add a Region

Use the following procedure to add a replicated Region for your AWS Managed Microsoft AD directory.

To add a replicated Region

1. In the AWS Directory Service console navigation pane, choose **Directories**.
2. On the **Directories** page, choose your directory ID.
3. On the **Directory details** page, under **Multi-Region replication**, choose the **Primary** Region from the list, and then choose **Add Region**.

   Note
   You can only add Regions while the **Primary** Region is selected. For more information, see Primary Region (p. 59).
4. On the **Add Region** page, under **Region**, choose the Region you want to add from the list.
5. Under **VPC**, choose the VPC to use for this Region.

   Note
   This VPC must not have a Classless Inter-Domain Routing (CIDR) that overlaps with a VPC used by this directory in another Region.
6. Under **Subnets**, choose the subnet to use for this Region.
7. Review the information under **Pricing**, and then choose **Add**.
8. When AWS Managed Microsoft AD completes the domain controller deployment process, the Region will display **Active** status. You can now make updates to this Region as needed.

Next steps

After you add your new Region, you should consider doing the following next steps:

- Deploy additional domain controllers (up to 20) to your new Region as needed. The number of domain controllers when you add a new Region is 2 by default, which is the minimum required for fault-tolerance and high availability purposes. For more information, see Add or remove additional domain controllers (p. 144).
- Share your directory with more AWS accounts per Region. Directory sharing configurations are not replicated from the primary Region automatically. For more information, see Share your directory (p. 63).
- Enable log forwarding to retrieve your directory’s security logs using Amazon CloudWatch Logs from the new Region. When you enable log forwarding, you must provide a log group name in each Region where you replicated your directory. For more information, see Enable log forwarding (p. 55).
- Enable Amazon Simple Notification Service (Amazon SNS) monitoring for the new Region to track your directory health status per Region. For more information, see Configure directory status notifications (p. 52).

Delete a replicated Region

Use the following procedure to delete a Region for your AWS Managed Microsoft AD directory. Before you delete a Region, make sure it does not have either of the following:

- Authorized applications attached to it.

- Shared directories associated with it.

To delete a replicated Region

1. In the AWS Directory Service console navigation pane, choose **Directories**.
2. On the **Directories** page, choose your directory ID.
3. On the **Directory details** page, under **Multi-Region replication** choose **Delete Region**.
4. In the **Delete Region** dialog box, review the information, and then enter in the Region name to confirm. Then choose **Delete**.

   Note
   You cannot make updates to the Region while it's being deleted.

Share your directory

AWS Managed Microsoft AD integrates tightly with AWS Organizations to allow seamless directory sharing across multiple AWS accounts. You can share a single directory with other trusted AWS accounts within the same organization or share the directory with other AWS accounts that are outside your organization. You can also share your directory when your AWS account is not currently a member of an organization.

> **Note**
> AWS charges an additional fee for directory sharing. To learn more, see the Pricing page on the AWS Directory Service web site.

Directory sharing makes AWS Managed Microsoft AD a more cost-effective way of integrating with Amazon EC2 in multiple accounts and VPCs. Directory sharing is available in all AWS regions where AWS managed Microsoft AD is offered.

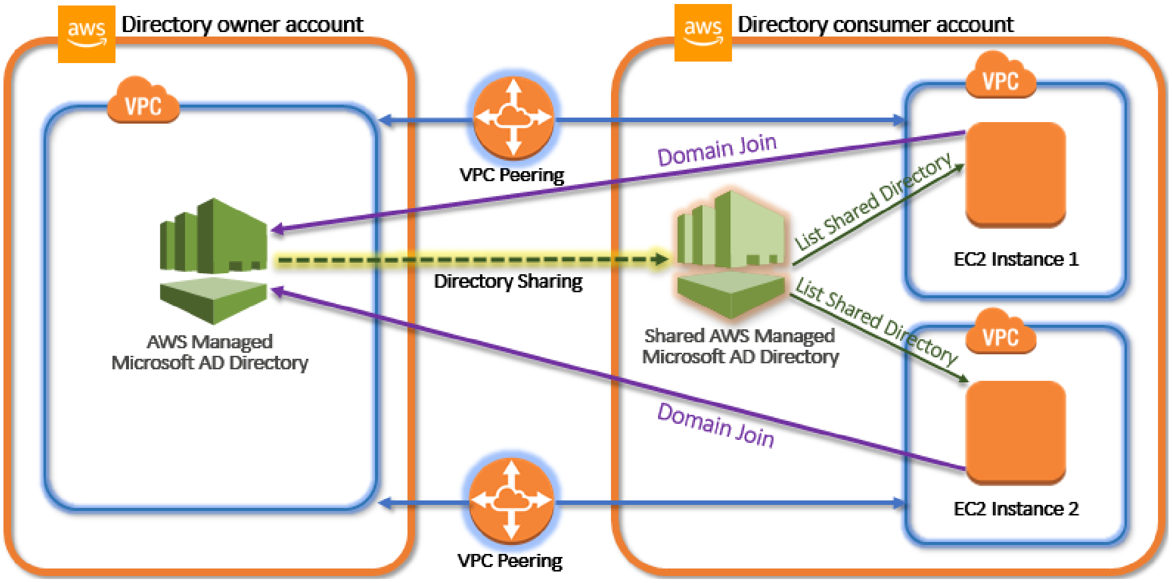
> **Note**
> In the AWS China (Ningxia) Region, this feature is available only when using AWS Systems Manager (SSM) to seamlessly join your Amazon EC2 instances.

For more information about directory sharing and how to extend the reach of your AWS Managed Microsoft AD directory across AWS account boundaries, see the following topics.

Topics

Key directory sharing concepts

You'll get more out of the directory sharing feature if you become familiar with the following key concepts.



Directory owner account

A directory owner is the AWS account holder that owns the originating directory in the shared directory relationship. An administrator in this account initiates the directory sharing workflow by specifying which AWS accounts to share their directory with. Directory owners can see who they've shared a directory with using the **Scale & Share** tab for a given directory in the AWS Directory Service console.

Directory consumer account

In a shared directory relationship, a directory consumer represents the AWS account to which the directory owner shared the directory with. Depending on the sharing method used, an administrator in this account may need to accept an invite sent from the directory owner before they can start using the shared directory.

The directory sharing process creates a shared directory in the directory consumer account. This shared directory contains the metadata that enables the EC2 instance to seamlessly join the domain, which locates the originating directory in the directory owner account. Each shared directory in the directory consumer account has a unique identifier (**Shared directory ID**).

Sharing methods

AWS Managed Microsoft AD provides the following two directory sharing methods:

- **AWS Organizations** – This method makes it easier to share the directory within your organization because you can browse and validate the directory consumer accounts. To use this option, your organization must have **All features** enabled, and your directory must be in the organization management account. This method of sharing simplifies your setup because it doesn’t require the directory consumer accounts to accept your directory sharing request. In the console, this method is referred to as **Share this directory with AWS accounts inside your organization**.
- **Handshake** – This method enables directory sharing when you aren’t using AWS Organizations. The handshake method requires the directory consumer account to accept the directory sharing request. In the console, this method is referred to as **Share this directory with other AWS accounts**.

Network connectivity

Network connectivity is a prerequisite to use a directory sharing relationship across AWS accounts. AWS supports many solutions to connect your VPCs, some of these include VPC peering, Transit Gateway, and

VPN. To get started, see Tutorial: Sharing your AWS Managed Microsoft AD directory for seamless EC2 domain-join (p. 65).

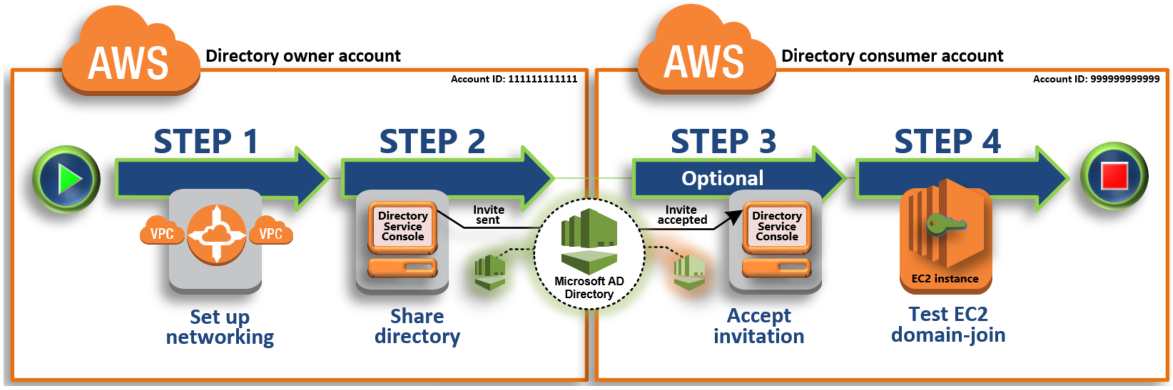
Tutorial: Sharing your AWS Managed Microsoft AD directory for seamless EC2 domain-join

This tutorial shows you how to share your AWS Managed Microsoft AD directory (the directory owner account) with another AWS account (the directory consumer account). Once the networking prerequisites have been completed, you will share a directory between two AWS accounts. Then you'll learn how to seamlessly join an EC2 instance to a domain in the directory consumer account.

We recommend that you first review directory sharing key concepts and use case content before you start work on this tutorial. For more information, see Key directory sharing concepts (p. 63).

The process for sharing your directory differs depending on whether you share the directory with another AWS account in the same AWS organization or with an account that is outside of the AWS organization. For more information about how sharing works, see Sharing methods (p. 64).

This workflow has four basic steps.



Step 1: Set up your networking environment (p. 66)

In the directory owner account, you set up all of the networking prerequisites necessary for the directory sharing process.

Step 2: Share your directory (p. 67)

While signed in with directory owner administrator credentials, you open the AWS Directory Service console and start the share directory workflow, which sends an invitation to the directory consumer account.

Step 3: Accept shared directory invite (optional) (p. 68)

While signed in with directory consumer administrator credentials, you open the AWS Directory Service console and accept the directory sharing invite.

Step 4: Test seamlessly joining an EC2 instance for Windows Server to a domain (p. 69)

Finally, as the directory consumer administrator, you attempt to join an EC2 instance to your domain and verify that it works.

Additional resources

- Use case: Share your directory to seamlessly join Amazon EC2 instances to a domain across AWS accounts
- AWS Security Blog Article: How to Join Amazon EC2 Instances From Multiple Accounts and VPCs to a Single AWS Managed Microsoft AD Directory

Step 1: Set up your networking environment

Before you begin the steps in this tutorial, you must first do the following:

- Create two new AWS accounts for testing purposes in the same Region. When you create an AWS account, it automatically creates a dedicated virtual private cloud (VPC) in each account. Take note of the VPC ID in each account. You will need this later.
- Create a VPC peering connection between the two VPCs in each account using the procedures in this step.

  > **Note**
  > While there are many ways to connect Directory owner and Directory consumer account VPCs, this tutorial will use the VPC peering method. For additional VPC connectivity options, see Network connectivity (p. 64).

Configure a VPC peering connection between the directory owner and the directory consumer account

The VPC peering connection you will create is between the directory consumer and directory owner VPCs. Follow these steps to configure a VPC peering connection for connectivity with the directory consumer account. With this connection you can route traffic between both VPCs using private IP addresses.

To create a VPC peering connection between the directory owner and directory consumer account

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/. Makes sure to sign in as a user with administrator credentials in the directory owner account.
2. In the navigation pane, choose **Peering Connections**. Then choose **Create Peering Connection**.
3. Configure the following information:
   - **Peering connection name tag**: Provide a name that clearly identifies this connection with the VPC in the directory consumer account.
   - **VPC (Requester)**: Select the VPC ID for the directory owner account.
   - Under **Select another VPC to peer with**, ensure that **My account** and **This region** are selected.
   - **VPC (Accepter)**: Select the VPC ID for the directory consumer account.
4. Choose **Create Peering Connection**. In the confirmation dialog box, choose **OK**.

Since both VPCs are in the same Region, the administrator of the directory owner account who sent the VPC peering request can also accept the peering request on behalf of the directory consumer account.

To accept the peering request on behalf of the directory consumer account

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Peering Connections**.
3. Select the pending VPC peering connection. (Its status is Pending Acceptance.) Choose **Actions**, **Accept Request**.
4. In the confirmation dialog, choose **Yes, Accept**. In the next confirmation dialog box, choose **Modify my route tables now** to go directly to the route tables page.

Now that your VPC peering connection is active, you must add an entry to your VPC route table in the directory owner account. Doing so enables traffic to be directed to the VPC in the directory consumer account.

To add an entry to the VPC route table in the directory owner account

1. While in the **Route Tables** section of the Amazon VPC console, select the route table for the directory owner VPC.
2. Choose the **Routes** tab, choose **Edit routes**, and then choose **Add route**.
3. In the **Destination** column, enter the CIDR block for the directory consumer VPC.
4. In the **Target** column, enter the VPC peering connection ID (such as `pcx-123456789abcde000`) for the peering connection that you created earlier in the directory owner account.
5. Choose **Save changes**.

To add an entry to the VPC route table in the directory consumer account

1. While in the **Route Tables** section of the Amazon VPC console, select the route table for the directory consumer VPC.
2. Choose the **Routes** tab, choose **Edit routes**, and then choose **Add route**.
3. In the **Destination** column, enter the CIDR block for the directory owner VPC.
4. In the **Target** column, type in the VPC peering connection ID (such as `pcx-123456789abcde001`) for the peering connection that you created earlier in the directory consumer account.
5. Choose **Save changes**.

Make sure to configure your directory consumer VPCs' security group to enable outbound traffic by adding the Active Directory protocols and ports to the outbound rules table. For more information, see Security groups for your VPC and AWS Managed Microsoft AD prerequisites.

Next Step

Step 2: Share your directory (p. 67)

Step 2: Share your directory

Use the following procedures to begin the directory sharing workflow from within the directory owner account.

> **Note**
> Directory sharing is a Regional feature of AWS Managed Microsoft AD. If you are using Multi-Region replication (p. 58), the following procedures must be applied separately in each Region. For more information, see Global vs Regional features (p. 58).

To share your directory from the directory owner account

1. Sign into the AWS Management Console with administrator credentials in the directory owner account and open the AWS Directory Service console at https://console.aws.amazon.com/directoryservicev2/.
2. In the navigation pane, choose **Directories**.
3. Choose the directory ID of the AWS Managed Microsoft AD directory that you want to share.
4. On the **Directory details** page, do one of the following:
   - If you have multiple Regions showing under **Multi-Region replication**, select the Region where you want to share your directory, and then choose the **Scale & share** tab. For more information, see Primary vs additional Regions (p. 59).
   - If you do not have any Regions showing under **Multi-Region replication**, choose the **Scale & share** tab.
5. In the **Shared directories** section, choose **Actions**, and then choose **Create new shared directory**.

6. On the **Choose which AWS accounts to share with** page, choose one of the following sharing methods depending on your business needs:

   a. **Share this directory with AWS accounts inside your organization** – With this option you can select the AWS accounts you want to share your directory with from a list showing all the AWS accounts inside your AWS organization. You must enable trusted access with AWS Directory Service before you share a directory. For more information, see How to enable or disable trusted access.

      Note
      To use this option, your organization must have **All features** enabled, and your directory must be in the organization management account.

      i. Under **AWS accounts in your organization**, select the AWS accounts that you want to share the directory with and click **Add**.
      ii. Review the pricing details, and then choose **Share**.
      iii. Proceed to Step 4 (p. 69) in this guide. Because all AWS accounts are in the same organization, you do not need to follow Step 3.

   b. **Share this directory with other AWS accounts** - With this option, you can share a directory with accounts inside or outside your AWS organization. You can also use this option when your directory is not a member of an AWS organization and you want to share with another AWS account.

      i. In **AWS account ID(s)**, enter all the AWS account IDs that you want to share the directory with, and then click **Add**.
      ii. In **Send a note**, type a message to the administrator in the other AWS account.
      iii. Review the pricing details, and then choose **Share**.
      iv. Proceed to Step 3.

Next Step

Step 3: Accept shared directory invite (optional) (p. 68)

Step 3: Accept shared directory invite (optional)

If you chose the **Share this directory with other AWS accounts** (handshake method) option in the previous procedure, you should use this procedure to finish the shared directory workflow. If you chose the **Share this directory with AWS accounts inside your organization** option, skip this step and proceed to Step 4.

To accept the shared directory invite

1. Sign into the AWS Management Console with administrator credentials in the directory consumer account and open the AWS Directory Service console at https://console.aws.amazon.com/directoryservicev2/.
2. In the navigation pane, choose **Directories shared with me**.
3. In the **Shared directory ID** column, choose the directory ID that is in the **Pending acceptance** state.
4. On the **Shared directory details** page, choose **Review**.
5. In the **Pending shared directory invitation** dialog, review the note, directory owner details, and information about pricing. If you agree, choose **Accept** to start using the directory.

Next Step

Step 4: Test seamlessly joining an EC2 instance for Windows Server to a domain (p. 69)

Step 4: Test seamlessly joining an EC2 instance for Windows Server to a domain

You can use either of the following two methods to test seamless domain join.

Method 1: Test domain join using the Amazon EC2 console

Use this step in the directory consumer account.

1. Sign in to the AWS Management Console and open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the Region selector in the navigation bar, choose the same Region as the existing directory.
3. Choose **Launch Instance**.
4. On the **Step 1** page, choose **Select** for the appropriate AMI.
5. On the **Step 2** page, select the appropriate instance type, and then choose **Next: Configure Instance Details**.
6. On the **Step 3** page, do the following, and then choose **Next: Add Storage**:
   1. For **Network**, choose the VPC that your directory was created in.
   2. For **Subnet**, choose one of the public subnets in your VPC. The subnet that you choose must have all external traffic routed to an internet gateway. If this is not the case, you won't be able to connect to the instance remotely.
   3. For **Auto-assign Public IP**, choose **Enable**.

      For more information about public and private IP addressing, see Amazon EC2 instance IP addressing in the *Amazon EC2 User Guide for Windows Instances*.
   4. For **Domain join directory**, choose your domain from the list.
   5. For **IAM role**, do one of the following:

      Select an IAM role that has the AWS managed policies **AmazonSSMManagedInstanceCore** and **AmazonSSMDirectoryServiceAccess** attached to it.

      -or-

      If you haven't created an IAM role that has the **AmazonSSMManagedInstanceCore** and **AmazonSSMDirectoryServiceAccess** managed policies attached to it, choose the **Create new IAM role** link, and then do the following:

      a. Choose **Create role**.

      b. Under **Select type of trusted entity**, choose **AWS service**.

      c. Under **Choose the service that this role will use**, in the full list of services, choose **EC2**.

      d. Under **Select your use case**, choose **EC2**, and the choose **Next: Permissions**.

      e. In the list of policies, select the **AmazonSSMManagedInstanceCore** and **AmazonSSMDirectoryServiceAccess** policies. (To filter the list, type `SSM` in the search box.)

         > **Note**
         > **AmazonSSMDirectoryServiceAccess** provides the permissions to join instances to an Active Directory managed by AWS Directory Service. **AmazonSSMManagedInstanceCore** provides the minimum permissions necessary to use the Systems Manager service. For more information about creating a role with these permissions, and for information about other permissions and policies you can assign to your IAM role, see Create an IAM instance profile for Systems Manager in the *AWS Systems Manager User Guide*.

      f. Choose **Next: Tags**.

      g. (Optional) Add one or more tag key-value pairs to organize, track, or control access for this role, and then choose **Next: Review**.

h. For **Role name**, enter a name for your new role, such as `EC2DomainJoin` or another name that you prefer.

i. (Optional) For **Role description**, enter a description.

j. Choose **Create role**.

k. Go back to the **Step 3** page. For **IAM role**, choose the refresh icon next to **IAM role**. Your new role should be visible in the menu. Choose it and leave the rest of the settings on this page with their default values, and then choose **Next: Add Storage**.

7. On both the **Step 4** and **Step 5** pages, leave the default settings or make changes as needed, and then choose the **Next** buttons.
8. On the **Step 6** page, select a security group for the instance that has been configured to allow remote access to the instance from your network, and then choose **Review and Launch**.
9. On the **Step 7** page, choose **Launch**, select a key pair, and then choose **Launch Instance**.

Method 2: Test domain join using the AWS Systems Manager console

Use this step in the directory consumer account. To complete this procedure, you'll need some information about the directory owner account.

> **Note**
> Make sure to attach the **AmazonSSMManagedInstanceCore** and **AmazonSSMDirectoryServiceAccess** managed policies to the IAM role permissions for your instance before starting the steps in this procedure. For information about these managed policies and other policies you can attach to an IAM instance profile for Systems Manager, see Create an IAM instance profile for Systems Manager in the *AWS Systems Manager User Guide*.For information about managed policies, see AWS Managed policies in the *IAM User Guide*.

1. Sign into the AWS Management Console and open the AWS Systems Manager console at https://console.aws.amazon.com/systems-manager/.
2. In the navigation pane, choose **Run Command**.
3. Choose **Run command**.
4. On the **Run a command** page, search for `AWS-JoinDirectoryServiceDomain`. When it is displayed in the search results, select the `AWS-JoinDirectoryServiceDomain` option.
5. Scroll down to the **Command parameters** section. You must provide the following parameters:

   - For **Directory Id**, enter the name of the AWS Directory Service directory.

     > **Note**
     > You can locate the **Directory Id** value by going back to the AWS Directory Service console, choosing **Directories shared with me**, selecting your directory, and then finding the value in the **Shared directory details** section.

   - For **Directory Name**, enter the name of the directory (for the directory owner account).
   - For **Dns Ip Addresses**, enter the IP addresses of the DNS servers in the directory (for the directory owner account).

   > **Note**
   > You can locate the values for **Directory Name** and **Dns Ip Addresses** by going back to the AWS Directory Service console, choosing **Directories shared with me**, selecting your directory, and then reviewing the attributes found in the **Owner directory details** section.

6. For **Targets**, select the instances that you want to domain join.
7. Leave the remainder of the form set to their default values, scroll down the page, and then choose **Run**.
8. In the navigation pane, choose **Managed Instances**.

9. Verify the instance successfully joined the domain by reviewing the instance in the list. If the **Association Status** displays **Success**, your instance has successfully joined the domain.

After completing either of these steps, you should now be able to join your EC2 instance to the domain. Once you do that, you can then log into your instance using a Remote Desktop Protocol (RDP) client with the credentials from your AWS Managed Microsoft AD user account.

Unshare your directory

Use the following procedure to unshare an AWS Managed Microsoft AD directory.

To unshare your directory

1. In the AWS Directory Service console navigation pane, under **Active Directory**, select **Directories**.
2. Choose the directory ID of the AWS Managed Microsoft AD directory that you want to unshare.
3. On the **Directory details** page, do one of the following:
   - If you have multiple Regions showing under **Multi-Region replication**, select the Region where you want to unshare your directory, and then choose the **Scale & share** tab. For more information, see Primary vs additional Regions (p. 59).
   - If you do not have any Regions showing under **Multi-Region replication**, choose the **Scale & share** tab.
4. In the **Shared directories** section, select the shared directory you want to unshare, choose **Actions**, and then choose **Unshare**.
5. In the **Unshare directory** dialog box, choose **Unshare**.

Additional resources

- Use case: Share your directory to seamlessly join amazon EC2 instances to a domain across AWS accounts
- AWS security blog article: How to join Amazon EC2 instances from multiple accounts and VPCs to a single AWS Managed Microsoft AD directory
- Joining your Amazon RDS DB instances across accounts to a single shared domain

Join an EC2 instance to your AWS Managed Microsoft AD directory

You can seamlessly join an EC2 instance to your directory domain when the instance is launched using AWS Systems Manager. For more information, see Seamlessly joining a Windows instance to an AWS Directory Service domain in the *Amazon EC2 User Guide for Windows Instances*.

If you need to manually join an EC2 instance to your domain, you must launch the instance in the proper Region and security group or subnet, then join the instance to the domain.

To be able to connect remotely to these instances, you must have IP connectivity to the instances from the network you are connecting from. In most cases, this requires that an internet gateway be attached to your VPC and that the instance has a public IP address.

Topics

- Seamlessly join a Linux EC2 instance to your AWS Managed Microsoft AD directory (p. 75)
- Manually join a Linux instance (p. 79)
- Manually join a Linux instance using Winbind (p. 90)
- Delegate directory join privileges for AWS Managed Microsoft AD (p. 98)
- Create a DHCP options set (p. 99)

Seamlessly join a Windows EC2 instance

This procedure seamlessly joins a Windows EC2 instance to your AWS Managed Microsoft AD directory. If you need to perform seamless domain join across multiple AWS accounts, you can optionally choose to enable Directory sharing. For more information, see Tutorial: Sharing your AWS Managed Microsoft AD directory for seamless EC2 domain-join (p. 65).

To seamlessly join a Windows EC2 instance

1. Sign in to the AWS Management Console and open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the Region selector in the navigation bar, choose the same Region as the existing directory.
3. Choose **Launch Instance**.
4. On the **Step 1** page, choose **Select** for the appropriate AMI.
5. On the **Step 2** page, select the appropriate instance type, and then choose **Next: Configure Instance Details**.
6. On the **Step 3** page, do the following, and then choose **Next: Add Storage**:

    1. For **Network**, choose the VPC that your directory was created in.
    2. For **Subnet**, choose one of the public subnets in your VPC. The subnet that you choose must have all external traffic routed to an internet gateway. If this is not the case, you won't be able to connect to the instance remotely.
    3. For **Auto-assign Public IP**, choose **Enable**.

       For more information about public and private IP addressing, see Amazon EC2 instance IP addressing in the *Amazon EC2 User Guide for Windows Instances*.
    4. For **Domain join directory**, choose your domain from the list.
    5. For **IAM role**, do one of the following:

       Select an IAM role that has the AWS managed policies **AmazonSSMManagedInstanceCore** and **AmazonSSMDirectoryServiceAccess** attached to it.

       -or-

       If you haven't created an IAM role that has the **AmazonSSMManagedInstanceCore** and **AmazonSSMDirectoryServiceAccess** managed policies attached to it, choose the **Create new IAM role** link, and then do the following:

       a. Choose **Create role**.
       b. Under **Select type of trusted entity**, choose **AWS service**.
       c. Under **Choose the service that this role will use**, in the full list of services, choose **EC2**.
       d. Under **Select your use case**, choose **EC2**, and the choose **Next: Permissions**.
       e. In the list of policies, select the **AmazonSSMManagedInstanceCore** and **AmazonSSMDirectoryServiceAccess** policies. (To filter the list, type `SSM` in the search box.)

          Note
          AmazonSSMDirectoryServiceAccess provides the permissions to join instances to an Active Directory managed by AWS Directory Service.

  AmazonSSMManagedInstanceCore provides the minimum permissions necessary to use the Systems Manager service. For more information about creating a role with these permissions, and for information about other permissions and policies you can assign to your IAM role, see Create an IAM instance profile for Systems Manager in the *AWS Systems Manager User Guide*.

   f. Choose **Next: Tags**.

   g. (Optional) Add one or more tag key-value pairs to organize, track, or control access for this role, and then choose **Next: Review**.

   h. For **Role name**, enter a name for your new role, such as **`EC2DomainJoin`** or another name that you prefer.

   i. (Optional) For **Role description**, enter a description.

   j. Choose **Create role**.

   k. Go back to the **Step 3** page. For **IAM role**, choose the refresh icon next to **IAM role**. Your new role should be visible in the menu. Choose it and leave the rest of the settings on this page with their default values, and then choose **Next: Add Storage**.

7. On both the **Step 4** and **Step 5** pages, leave the default settings or make changes as needed, and then choose the **Next** buttons.
8. On the **Step 6** page, select a security group for the instance that has been configured to allow remote access to the instance from your network, and then choose **Review and Launch**.
9. On the **Step 7** page, choose **Launch**, select a key pair, and then choose **Launch Instance**.

Manually join a Windows instance

To manually join an existing Amazon EC2 Windows instance to a Simple AD or AWS Directory Service for Microsoft Active Directory directory, the instance must be launched as specified in Seamlessly join a Windows EC2 instance (p. 72).

To join a Windows instance to a Simple AD or AWS Managed Microsoft AD directory

1. Connect to the instance using any Remote Desktop Protocol client.
2. Open the TCP/IPv4 properties dialog box on the instance.

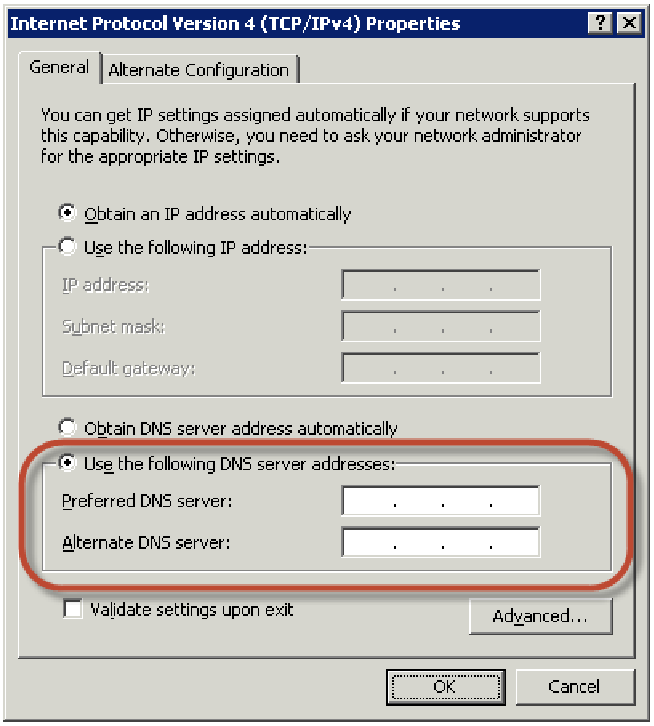
   a. Open **Network Connections**.

      Tip
      You can open **Network Connections** directly by running the following from a command prompt on the instance.

      ```
      %SystemRoot%\system32\control.exe ncpa.cpl
      ```

   b. Open the context menu (right-click) for any enabled network connection and then choose **Properties**.

   c. In the connection properties dialog box, open (double-click) **Internet Protocol Version 4**.
3. Select **Use the following DNS server addresses**, change the **Preferred DNS server** and **Alternate DNS server** addresses to the IP addresses of the AWS Directory Service-provided DNS servers, and choose **OK**.



You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

● Obtain an IP address automatically

○ Use the following IP address:

IP address: . . .

Subnet mask: . . .

Default gateway: . . .

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

□ Validate settings upon exit

4. Open the **System Properties** dialog box for the instance, select the **Computer Name** tab, and choose **Change**.

   Tip
   You can open the **System Properties** dialog box directly by running the following from a command prompt on the instance.

   ```
   %SystemRoot%\system32\control.exe sysdm.cpl
   ```

5. In the **Member of** field, select **Domain**, enter the fully qualified name of your AWS Directory Service directory, and choose **OK**.

6. When prompted for the name and password for the domain administrator, enter the user name and password of an account that has domain join privileges. For more information about delegating these privileges, see Delegate directory join privileges for AWS Managed Microsoft AD (p. 98).

   Note
   You can enter either the fully qualified name of your domain or the NetBios name, followed by a backslash (\), and then the user name.
   If using AWS Managed Microsoft AD, the user name would be **Admin**. For example, **corp.example.com\admin** or **corp\admin**.
   If using Simple AD, the user name would be **Administrator**. For example, **corp.example.com\administrator** or **corp\administrator**.

7. After you receive the message welcoming you to the domain, restart the instance to have the changes take effect.

Now that your instance has been joined to the domain, you can log into that instance remotely and install utilities to manage the directory, such as adding users and groups.

Seamlessly join a Linux EC2 instance to your AWS Managed Microsoft AD directory

This procedure seamlessly joins a Linux EC2 instance to your AWS Managed Microsoft AD directory. If you need to perform seamless domain join across multiple AWS accounts, you can optionally choose to enable Directory sharing.

The following Linux instance distributions and versions are supported:

- Amazon Linux AMI 2018.03.0
- Amazon Linux 2 (64-bit x86)
- Red Hat Enterprise Linux 8 (HVM) (64-bit x86)
- Ubuntu Server 18.04 LTS & Ubuntu Server 16.04 LTS
- CentOS 7 x86-64
- SUSE Linux Enterprise Server 15 SP1

> **Note**
> Distributions prior to Ubuntu 14 and Red Hat Enterprise Linux 7 do not support the seamless domain join feature.

Prerequisites

Before you can set up seamless domain join to a Linux EC2 instance, you need to complete the procedures in this section.

Select your seamless domain join service account

You can seamlessly join Linux computers to your AWS Managed Active Directory domain. To do that, you must use a user account with create computer account permissions to join the machines to the domain. Although members of the *AWS delegated administrators* or other groups might have sufficient privileges to join computers to the domain, we do not recommend using these. As a best practice, we recommend that you use a service account that has the minimum privileges necessary to join the computers to the domain.

To delegate an account with the minimum privileges necessary to join the computers to the domain, you can run the following PowerShell commands. You must run these commands from a domain-joined Windows computer with the Installing the Active Directory administration tools (p. 101) installed. In addition, you must use an account that has permission to modify the permissions on your Computers OU or container. The PowerShell command sets permissions allowing the service account to create computer objects in your domain's default computers container.

```
$AccountName = 'awsSeamlessDomain'
# DO NOT modify anything below this comment.
# Getting Active Directory information.
Import-Module 'ActiveDirectory'
$Domain = Get-ADDomain -ErrorAction Stop
$BaseDn = $Domain.DistinguishedName
$ComputersContainer = $Domain.ComputersContainer
$SchemaNamingContext = Get-ADRootDSE | Select-Object -ExpandProperty 'schemaNamingContext'
[System.GUID]$ServicePrincipalNameGuid = (Get-ADObject -SearchBase $SchemaNamingContext -
Filter { lDAPDisplayName -eq 'Computer' } -Properties 'schemaIDGUID').schemaIDGUID
# Getting Service account Information.
$AccountProperties = Get-ADUser -Identity $AccountName
$AccountSid = New-Object -TypeName 'System.Security.Principal.SecurityIdentifier'
 $AccountProperties.SID.Value
# Getting ACL settings for the Computers container.
$ObjectAcl = Get-ACL -Path "AD:\$ComputersContainer"
```

```
# Setting ACL allowing the service account the ability to create child computer objects in
 the Computers container.
$AddAccessRule = New-Object -TypeName 'System.DirectoryServices.ActiveDirectoryAccessRule'
 $AccountSid, 'CreateChild', 'Allow', $ServicePrincipalNameGUID, 'All'
$ObjectAcl.AddAccessRule($AddAccessRule)
Set-ACL -AclObject $ObjectAcl -Path "AD:\$ComputersContainer"
```

If you prefer using a graphical user interface (GUI) you can use the manual process that is described in Delegate privileges to your service account (p. 189).

Create the secrets to store the domain service account

You can use AWS Secrets Manager to store the domain service account.

To create secrets and store the domain service account information

1. Sign in to the AWS Management Console and open the AWS Secrets Manager console at https://console.aws.amazon.com/secretsmanager/.
2. Choose **Store a new secret**.
3. On the **Store a new secret** page, do the following:

   a. Under **Select secret type**, choose **Other type of secrets**.

   b. Under **Specify the key/value pairs to be stored in the secret**, do the following:

      i. In the first box, enter **awsSeamlessDomainUsername**. On the same row, in the next box, enter the user name for your service account. For example, if you used the PowerShell command previously, the service account name would be awsSeamlessDomain.

         Note
         You must enter **awsSeamlessDomainUsername** exactly as it is. Make sure there are not any leading or ending spaces. Otherwise the domain join will fail.

      ii. Choose **Add row**.

      iii. On the new row, in the first box, enter **awsSeamlessDomainPassword**. On the same row, in the next box, enter the password for your service account.

         Note
         You must enter **awsSeamlessDomainPassword** exactly as it is. Make sure there are not any leading or ending spaces. Otherwise the domain join will fail.

      iv. Under **Select the encryption key**, choose **DefaultEncryptionKey** from the menu. Secrets Manager always encrypts the secret when you choose this option, and provides it at no charge to you. You also may choose a key you created.

      v. Choose **Next**.
4. Under **Secret name**, enter a secret name that includes your directory ID using the following format:

   ```
   aws/directory-services/d-xxxxxxxxx/seamless-domain-join
   ```

   This will be used to retrieve secrets in the application.

   Note
   You must enter **aws/directory-services/*d-xxxxxxxxx*/seamless-domain-join** exactly as it is but replace *d-xxxxxxxxxx* with your directory ID. Make sure that there are no leading or ending spaces. Otherwise the domain join will fail.
5. Leave everything else set to defaults, and then choose **Next**.
6. Under **Configure automatic rotation**, choose **Disable automatic rotation**, and then choose **Next**.
7. Review the settings, and then choose **Store** to save your changes. The Secrets Manager console returns you to the list of secrets in your account with your new secret now included in the list.

8. Choose your newly created secret name from the list, and take note of the **Secret ARN** value. You will need it in the next section.

Create the required IAM policy and role

Use the following prerequisite steps to create a custom policy that allows read-only access to your Secrets Manager seamless domain join secret (which you created earlier), and to create a new LinuxEC2DomainJoin IAM role.

Create the Secrets Manager IAM read policy

You use the IAM console to create a policy that grants read-only access to your Secrets Manager secret.

To create the Secrets Manager IAM read policy

1. Sign in to the AWS Management Console as a user that has permission to create IAM policies. Then open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Policies**.
3. Choose **Create policy**.
4. Choose the **JSON** tab and copy the text from the following JSON policy document. Then paste it into the **JSON** text box.

   Note
   Make sure you replace the `Resource` ARN with the actual ARN of the secret that you created earlier.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue",
                "secretsmanager:DescribeSecret"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:############:secret:aws/directory-
service/d-xxxxxxxxxx/seamless-domain-join-example"
            ]
        }
    ]
}
```

5. When you are finished, choose **Review policy**. The policy validator reports any syntax errors. For more information, see Validating IAM policies.
6. On the **Review policy** page, enter a policy name, such as **`SM-Secret-Linux-DJ-`*`d-xxxxxxxxxx`*`-Read`**. Review the **Summary** section to see the permissions that your policy grants. Then choose **Create policy** to save your changes. The new policy appears in the list of managed policies and is now ready to attach to an identity.

   Note
   We recommend you create one policy per secret. Doing so ensures that instances only have access to the appropriate secret and minimizes the impact if an instance is compromised.

Create the LinuxEC2DomainJoin role

You use the IAM console to create the role that you will use to domain join your Linux EC2 instance.

To create the LinuxEC2DomainJoin role

1. Sign in to the AWS Management Console as a user that has permission to create IAM policies. Then open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Roles**.
3. In the content pane, choose **Create role**.
4. Under **Select type of trusted entity**, choose **AWS service**.
5. Under **Choose a use case** choose **EC2**, and then choose **Next: Permissions**.
6. For **Filter policies**, do the following:

   a. Enter `AmazonSSMManagedInstanceCore`. Then select the check box for that item in the list.

   b. Enter `AmazonSSMDirectoryServiceAccess`. Then select the check box for that item in the list.

   c. Enter `SM-Secret-Linux-DJ-d-xxxxxxxxxx-Read` (or the name of the policy that you created in the previous procedure). Then select the check box for that item in the list.

   Note
   AmazonSSMDirectoryServiceAccess provides the permissions to join instances to an Active Directory managed by AWS Directory Service. AmazonSSMManagedInstanceCore provides the minimum permissions necessary to use the AWS Systems Manager service. For more information about creating a role with these permissions, and for information about other permissions and policies you can assign to your IAM role, see Create an IAM instance profile for Systems Manager in the *AWS Systems Manager User Guide*.
7. Choose **Next: Tags**.
8. (Optional) Add one or more tag key-value pairs to organize, track, or control access for this role. Then choose **Next: Review**.
9. Next to **Role name**, enter a name for your new role, such as `LinuxEC2DomainJoin` or another name that you prefer.
10. (Optional) For **Role description**, enter a description.
11. Choose **Create role**.

Seamlessly join your Linux EC2 instance

Now that you have configured all of the prerequisite tasks, you can use the following procedure to seamlessly join your Linux EC2 instance.

To seamlessly join your Linux EC2 instance

1. Sign in to the AWS Management Console and open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the Region selector in the navigation bar, choose the same Region as the existing directory.
3. Choose **Launch Instance**.
4. On the **Step 1** page, choose **Select** for the appropriate Amazon Machine Image (AMI).

   Note
   The AMI used must have AWS Systems Manager (SSM Agent) version 2.3.1644.0 or higher. To check the installed SSM Agent version in your AMI by launching an instance from that AMI, see Getting the currently installed SSM Agent version. If you need to upgrade the SSM Agent, see Installing and configuring SSM Agent on EC2 instances for Linux.
5. On the **Step 2** page, select the appropriate instance type, and then choose **Next: Configure Instance Details**.
6. On the **Step 3** page, do the following, and then choose **Next: Add Storage**:

a. For **Network**, choose the VPC that your directory was created in.

b. For **Subnet**, choose one of the public subnets in your VPC. The subnet that you choose must have all external traffic routed to an internet gateway. If this is not the case, you won't be able to connect to the instance remotely.

c. For **Auto-assign Public IP**, choose **Enable**. For more information about public and private IP addressing, see Amazon EC2 instance IP addressing in the *Amazon EC2 User Guide for Windows Instances*.

d. For **Domain join directory**, choose your domain from the list.

e. For **IAM role**, choose the IAM role that you previously created in the prerequisites section **Step 2: Create the LinuxEC2DomainJoin Role**.

7. On both the **Step 4** and **Step 5** pages, leave the default settings or make changes as needed. Then choose **Next** on each.
8. On the **Step 6** page, select a security group that has been configured to allow remote access to the instance from your network. Then choose **Review and Launch**.
9. On the **Step 7** page, choose **Launch**, choose a key pair, and then choose **Launch Instance**.

Note
If you are performing a seamless domain join with SUSE Linux, a reboot is required before authentications will work. To reboot SUSE from the Linux terminal, type **sudo reboot**.

Manually join a Linux instance

In addition to Amazon EC2 Windows instances, you can also join certain Amazon EC2 Linux instances to your AWS Directory Service for Microsoft Active Directory directory. The following Linux instance distributions and versions are supported:

- Amazon Linux AMI 2018.03.0
- Amazon Linux 2 (64-bit x86)
- Red Hat Enterprise Linux 8 (HVM) (64-bit x86)
- Ubuntu Server 18.04 LTS & Ubuntu Server 16.04 LTS
- CentOS 7 x86-64
- SUSE Linux Enterprise Server 15 SP1

Note
Other Linux distributions and versions may work but have not been tested.

Join an instance to your directory

Before you can join either an Amazon Linux, CentOS, Red Hat, or Ubuntu instance to your directory, the instance must first be launched as specified in Seamlessly join a Windows EC2 instance (p. 72).

Important
Some of the following procedures, if not performed correctly, can render your instance unreachable or unusable. Therefore, we strongly suggest you make a backup or take a snapshot of your instance before performing these procedures.

To join a linux instance to your directory

Follow the steps for your specific Linux instance using one of the following tabs:

Amazon Linux

1. Connect to the instance using any SSH client.

2. Configure the Linux instance to use the DNS server IP addresses of the AWS Directory Service-provided DNS servers. You can do this either by setting it up in the DHCP Options set attached to the VPC or by setting it manually on the instance. If you want to set it manually, see How do I assign a static DNS server to a private Amazon EC2 instance in the AWS Knowledge Center for guidance on setting the persistent DNS server for your particular Linux distribution and version.
3. Make sure your Amazon Linux - 64bit instance is up to date.

```
sudo yum -y update
```

4. Install the required Amazon Linux packages on your Linux instance.

   Note
   Some of these packages may already be installed.
   As you install the packages, you might be presented with several pop-up configuration screens. You can generally leave the fields in these screens blank.

   Amazon Linux 1

   ```
   sudo yum -y install sssd realmd krb5-workstation
   ```

   Amazon Linux 2

   ```
   sudo yum -y install sssd realmd krb5-workstation samba-common-tools
   ```

   Note
   For help with determining the Amazon Linux version you are using, see Identifying Amazon Linux images in the *Amazon EC2 User Guide for Linux Instances*.

5. Join the instance to the directory with the following command.

```
sudo realm join -U join_account@example.com example.com --verbose
```

join_account@example.com

   An account in the *example.com* domain that has domain join privileges. Enter the password for the account when prompted. For more information about delegating these privileges, see Delegate directory join privileges for AWS Managed Microsoft AD (p. 98).

example.com

   The fully qualified DNS name of your directory.

```
...
 * Successfully enrolled machine in realm
```

6. Set the SSH service to allow password authentication.

   a. Open the `/etc/ssh/sshd_config` file in a text editor.

   ```
   sudo vi /etc/ssh/sshd_config
   ```

   b. Set the `PasswordAuthentication` setting to `yes`.

   ```
   PasswordAuthentication yes
   ```

   c. Restart the SSH service.

   ```
   sudo systemctl restart sshd.service
   ```

Alternatively:

```
sudo service sshd restart
```

7. After the instance has restarted, connect to it with any SSH client and add the AWS Delegated Administrators group to the sudoers list by performing the following steps:

   a. Open the `sudoers` file with the following command:

   ```
   sudo visudo
   ```

   b. Add the following to the bottom of the `sudoers` file and save it.

   ```
   ## Add the "AWS Delegated Administrators" group from the example.com domain.
   %AWS\ Delegated\ Administrators@example.com ALL=(ALL:ALL) ALL
   ```

   (The above example uses "\<space>" to create the Linux space character.)

CentOS

1. Connect to the instance using any SSH client.
2. Configure the Linux instance to use the DNS server IP addresses of the AWS Directory Service-provided DNS servers. You can do this either by setting it up in the DHCP Options set attached to the VPC or by setting it manually on the instance. If you want to set it manually, see How do I assign a static DNS server to a private Amazon EC2 instance in the AWS Knowledge Center for guidance on setting the persistent DNS server for your particular Linux distribution and version.
3. Make sure your CentOS 7 instance is up to date.

```
sudo yum -y update
```

4. Install the required CentOS 7 packages on your Linux instance.

   Note
   Some of these packages may already be installed.
   As you install the packages, you might be presented with several pop-up configuration screens. You can generally leave the fields in these screens blank.

```
sudo yum -y install sssd realmd krb5-workstation samba-common-tools
```

5. Join the instance to the directory with the following command.

```
sudo realm join -U join_account@example.com example.com --verbose
```

`join_account@example.com`

   An account in the *`example.com`* domain that has domain join privileges. Enter the password for the account when prompted. For more information about delegating these privileges, see Delegate directory join privileges for AWS Managed Microsoft AD (p. 98).

`example.com`

   The fully qualified DNS name of your directory.

```
...
 * Successfully enrolled machine in realm
```

6. Set the SSH service to allow password authentication.

   a. Open the `/etc/ssh/sshd_config` file in a text editor.

   ```
   sudo vi /etc/ssh/sshd_config
   ```

   b. Set the `PasswordAuthentication` setting to `yes`.

   ```
   PasswordAuthentication yes
   ```

   c. Restart the SSH service.

   ```
   sudo systemctl restart sshd.service
   ```

   Alternatively:

   ```
   sudo service sshd restart
   ```

7. After the instance has restarted, connect to it with any SSH client and add the AWS Delegated Administrators group to the sudoers list by performing the following steps:

   a. Open the `sudoers` file with the following command:

   ```
   sudo visudo
   ```

   b. Add the following to the bottom of the `sudoers` file and save it.

   ```
   ## Add the "AWS Delegated Administrators" group from the example.com domain.
   %AWS\ Delegated\ Administrators@example.com ALL=(ALL:ALL) ALL
   ```

   (The above example uses "\<space>" to create the Linux space character.)

Red Hat

1. Connect to the instance using any SSH client.
2. Configure the Linux instance to use the DNS server IP addresses of the AWS Directory Service-provided DNS servers. You can do this either by setting it up in the DHCP Options set attached to the VPC or by setting it manually on the instance. If you want to set it manually, see How do I assign a static DNS server to a private Amazon EC2 instance in the AWS Knowledge Center for guidance on setting the persistent DNS server for your particular Linux distribution and version.
3. Make sure the Red Hat - 64bit instance is up to date.

```
sudo yum -y update
```

4. Install the required Red Hat packages on your Linux instance.

   Note
   Some of these packages may already be installed.
   As you install the packages, you might be presented with several pop-up configuration screens. You can generally leave the fields in these screens blank.

```
sudo yum -y install sssd realmd krb5-workstation samba-common-tools
```

5. Join the instance to the directory with the following command.

```
sudo realm join -v -U join_account example.com --install=/
```

join_account

The **sAMAccountName** for an account in the *example.com* domain that has domain join privileges. Enter the password for the account when prompted. For more information about delegating these privileges, see Delegate directory join privileges for AWS Managed Microsoft AD (p. 98).

example.com

The fully qualified DNS name of your directory.

```
...
 * Successfully enrolled machine in realm
```

6. Set the SSH service to allow password authentication.

   a. Open the `/etc/ssh/sshd_config` file in a text editor.

   ```
   sudo vi /etc/ssh/sshd_config
   ```

   b. Set the `PasswordAuthentication` setting to `yes`.

   ```
   PasswordAuthentication yes
   ```

   c. Restart the SSH service.

   ```
   sudo systemctl restart sshd.service
   ```

   Alternatively:

   ```
   sudo service sshd restart
   ```

7. After the instance has restarted, connect to it with any SSH client and add the AWS Delegated Administrators group to the sudoers list by performing the following steps:

   a. Open the `sudoers` file with the following command:

   ```
   sudo visudo
   ```

   b. Add the following to the bottom of the `sudoers` file and save it.

   ```
   ## Add the "AWS Delegated Administrators" group from the example.com domain.
   %AWS\ Delegated\ Administrators@example.com ALL=(ALL:ALL) ALL
   ```

   (The above example uses "\<space>" to create the Linux space character.)

SUSE

1. Connect to the instance using any SSH client.
2. Configure the Linux instance to use the DNS server IP addresses of the AWS Directory Service-provided DNS servers. You can do this either by setting it up in the DHCP Options set attached to the VPC or by setting it manually on the instance. If you want to set it manually, see How do I assign a static DNS server to a private Amazon EC2 instance in the AWS Knowledge Center for guidance on setting the persistent DNS server for your particular Linux distribution and version.
3. Make sure your SUSE Linux 15 instance is up to date.

   a. Connect the package repository.

   ```
   sudo SUSEConnect -p PackageHub/15.1/x86_64
   ```

   b. Update SUSE.

   ```
   sudo zypper update -y
   ```

4. Install the required SUSE Linux 15 packages on your Linux instance.

   Note
   Some of these packages may already be installed.
   As you install the packages, you might be presented with several pop-up configuration screens. You can generally leave the fields in these screens blank.

   ```
   sudo zypper -n install realmd adcli sssd sssd-tools sssd-ad samba-client krb5-client
   ```

5. Join the instance to the directory with the following command.

   ```
   sudo realm join -U join_account example.com --verbose
   ```

   join_account

   The sAMAccountName in the *example.com* domain that has domain join privileges. Enter the password for the account when prompted. For more information about delegating these privileges, see Delegate directory join privileges for AWS Managed Microsoft AD (p. 98).

   example.com

   The fully-qualified DNS name of your directory.

   ```
   …
   realm: Couldn't join realm: Enabling SSSD in nsswitch.conf and PAM failed.
   ```

   Note that both of the following returns are expected.

   ```
   ! Couldn't authenticate with keytab while discovering which salt to use:
   ! Enabling SSSD in nsswitch.conf and PAM failed.
   ```

6. Manually enable **SSSD** in **PAM**.

   ```
   sudo pam-config --add --sss
   ```

7. Edit nsswitch.conf to enable SSSD in nsswitch.conf

   ```
   sudo vi /etc/nsswitch.conf
   ```

   ```
   passwd: compat sss
   group:  compat sss
   ```

```
shadow: compat sss
```

8. Add the following line to /etc/pam.d/common-session to auto create a home directory at initial login

   ```
   sudo vi /etc/pam.d/common-session
   ```

   ```
   session optional        pam_mkhomedir.so skel=/etc/skel umask=077
   ```

9. Reboot the instance to complete the domain joined process.

   ```
   sudo reboot
   ```

10. Reconnect to the instance using any SSH client to verify the domain join has completed successfully and finalize additional steps.

    a. To confirm the instance has been enrolled on the domain

    ```
    sudo realm list
    ```

    ```
    example.com
      type: kerberos
      realm-name: EXAMPLE.COM
      domain-name: example.com
      configured: kerberos-member
      server-software: active-directory
      client-software: sssd
      required-package: sssd-tools
      required-package: sssd
      required-package: adcli
      required-package: samba-client
      login-formats: %U@example.com
      login-policy: allow-realm-logins
    ```

    b. To verify the status of SSSD daemon

    ```
    systemctl status sssd
    ```

    ```
    sssd.service - System Security Services Daemon
       Loaded: loaded (/usr/lib/systemd/system/sssd.service; enabled; vendor preset:
     disabled)
       Active: active (running) since Wed 2020-04-15 16:22:32 UTC; 3min 49s ago
     Main PID: 479 (sssd)
        Tasks: 4
       CGroup: /system.slice/sssd.service
               ##479 /usr/sbin/sssd -i --logger=files
               ##505 /usr/lib/sssd/sssd_be --domain example.com --uid 0 --gid 0 --
    logger=files
               ##548 /usr/lib/sssd/sssd_nss --uid 0 --gid 0 --logger=files
               ##549 /usr/lib/sssd/sssd_pam --uid 0 --gid 0 --logger=files
    ```

11. To permit a user access via SSH and console

    ```
    sudo realm permit join_account@example.com
    ```

    To permit a domain group access via SSH and console

```
sudo realm permit -g 'AWS Delegated Administrators'
```

Or to permit all users access

```
sudo realm permit --all
```

12Set the SSH service to allow password authentication.

a. Open the `/etc/ssh/sshd_config` file in a text editor.

```
sudo vi /etc/ssh/sshd_config
```

b. Set the `PasswordAuthentication` setting to `yes`.

```
PasswordAuthentication yes
```

c. Restart the SSH service.

```
sudo systemctl restart sshd.service
```

Alternatively:

```
sudo service sshd restart
```

13.13. After the instance has restarted, connect to it with any SSH client and add the AWS Delegated Administrators group to the sudoers list by performing the following steps:

a. Open the sudoers file with the following command:

```
sudo visudo
```

b. Add the following to the bottom of the sudoers file and save it.

```
## Add the "Domain Admins" group from the awsad.com domain.
%AWS\ Delegated\ Administrators@example.com ALL=(ALL) NOPASSWD: ALL
```

Ubuntu

1. Connect to the instance using any SSH client.
2. Configure the Linux instance to use the DNS server IP addresses of the AWS Directory Service-provided DNS servers. You can do this either by setting it up in the DHCP Options set attached to the VPC or by setting it manually on the instance. If you want to set it manually, see How do I assign a static DNS server to a private Amazon EC2 instance in the AWS Knowledge Center for guidance on setting the persistent DNS server for your particular Linux distribution and version.
3. Make sure your Ubuntu - 64bit instance is up to date.

```
sudo apt-get update
sudo apt-get -y upgrade
```

4. Install the required Ubuntu packages on your Linux instance.

Note
Some of these packages may already be installed.

As you install the packages, you might be presented with several pop-up configuration screens. You can generally leave the fields in these screens blank.

```
sudo apt-get -y install sssd realmd krb5-user samba-common packagekit adcli
```

5. Disable Reverse DNS resolution and set the default realm to your domain's FQDN. Ubuntu Instances **must** be reverse-resolvable in DNS before the realm will work. Otherwise, you have to disable reverse DNS in /etc/krb5.conf as follows:

```
sudo vi /etc/krb5.conf
```

```
[libdefaults]
default_realm = EXAMPLE.COM
rdns = false
```

6. Join the instance to the directory with the following command.

```
sudo realm join -U join_account example.com --verbose
```

join_account@example.com

The **sAMAccountName** for an account in the *example.com* domain that has domain join privileges. Enter the password for the account when prompted. For more information about delegating these privileges, see Delegate directory join privileges for AWS Managed Microsoft AD (p. 98).

example.com

The fully qualified DNS name of your directory.

```
...
 * Successfully enrolled machine in realm
```

7. Set the SSH service to allow password authentication.

   a. Open the `/etc/ssh/sshd_config` file in a text editor.

   ```
   sudo vi /etc/ssh/sshd_config
   ```

   b. Set the `PasswordAuthentication` setting to `yes`.

   ```
   PasswordAuthentication yes
   ```

   c. Restart the SSH service.

   ```
   sudo systemctl restart sshd.service
   ```

   Alternatively:

   ```
   sudo service sshd restart
   ```

8. After the instance has restarted, connect to it with any SSH client and add the AWS Delegated Administrators group to the sudoers list by performing the following steps:

   a. Open the `sudoers` file with the following command:

   ```
   sudo visudo
   ```

b. Add the following to the bottom of the `sudoers` file and save it.

```
## Add the "AWS Delegated Administrators" group from the example.com domain.
%AWS\ Delegated\ Administrators@example.com ALL=(ALL:ALL) ALL
```

(The above example uses "\<space>" to create the Linux space character.)

Restricting account login access

Since all accounts are defined in Active Directory, by default, all the users in the directory can log in to the instance. You can allow only specific users to log in to the instance with **ad_access_filter** in **sssd.conf**. For example:

```
ad_access_filter = (memberOf=cn=admins,ou=Testou,dc=example,dc=com)
```

memberOf

Indicates that users should only be allowed access to the instance if they are a member of a specific group.

cn

The common name of the group that should have access. In this example, the group name is *admins*.

ou

This is the organizational unit in which the above group is located. In this example, the OU is *Testou*.

dc

This is the domain component of your domain. In this example, *example*.

dc

This is an additional domain component. In this example, *com*.

You must manually add **ad_access_filter** to your **/etc/sssd/sssd.conf**.

Open the **/etc/sssd/sssd.conf** file in a text editor.

```
sudo vi /etc/sssd/sssd.conf
```

After you do this, your **sssd.conf** might look like this:

```
[sssd]
domains = example.com
config_file_version = 2
services = nss, pam

[domain/example.com]
ad_domain = example.com
krb5_realm = EXAMPLE.COM
realmd_tags = manages-system joined-with-samba
cache_credentials = True
id_provider = ad
```

```
krb5_store_password_if_offline = True
default_shell = /bin/bash
ldap_id_mapping = True
use_fully_qualified_names = True
fallback_homedir = /home/%u@%d
access_provider = ad
ad_access_filter = (memberOf=cn=admins,ou=Testou,dc=example,dc=com)
```

In order for the configuration to take effect, you need to restart the sssd service:

```
sudo systemctl restart sssd.service
```

Alternatively, you could use:

```
sudo service sssd restart
```

Connect to the instance

When a user connects to the instance using an SSH client, they are prompted for their user name. The user can enter the user name in either the `username@example.com` or `EXAMPLE\username` format. The response will appear similar to the following, depending on which linux distribution you are using:

Amazon Linux, Red Hat Enterprise Linux, and CentOS Linux

```
login as: johndoe@example.com
johndoe@example.com's password:
Last login: Thu Jun 25 16:26:28 2015 from XX.XX.XX.XX
```

SUSE Linux

```
SUSE Linux Enterprise Server 15 SP1 x86_64 (64-bit)

As "root" (sudo or sudo -i) use the:
  - zypper command for package management
  - yast command for configuration management

Management and Config: https://www.suse.com/suse-in-the-cloud-basics
Documentation: https://www.suse.com/documentation/sles-15/
Forum: https://forums.suse.com/forumdisplay.php?93-SUSE-Public-Cloud

Have a lot of fun...
```

Ubuntu Linux

```
login as: admin@example.com
admin@example.com@10.24.34.0's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-1057-aws x86_64)

* Documentation:  https://help.ubuntu.com
* Management:     https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

  System information as of Sat Apr 18 22:03:35 UTC 2020

  System load:  0.01               Processes:           102
  Usage of /:   18.6% of 7.69GB    Users logged in:     2
  Memory usage: 16%                IP address for eth0: 10.24.34.1
  Swap usage:   0%
```

Manually join a Linux instance using Winbind

You can use the Winbind service to manually join your Linux instances to an AWS Managed Microsoft AD domain. This enables your existing on-premises Active Directory users to use their AD credentials when accessing the Linux instances joined to your AWS Managed Microsoft AD. The following Linux instance distributions and versions are supported:

- Amazon Linux AMI 2018.03.0
- Amazon Linux 2 (64-bit x86)
- Red Hat Enterprise Linux 8 (HVM) (64-bit x86)
- Ubuntu Server 18.04 LTS & Ubuntu Server 16.04 LTS
- CentOS 7 x86-64
- SUSE Linux Enterprise Server 15 SP1

Note
Other Linux distributions and versions may work but have not been tested.

Join an instance to your directory

Before you can join either an Amazon Linux, CentOS, Red Hat, or Ubuntu instance to your directory, the instance must first be launched as specified in Seamlessly join a Windows EC2 instance (p. 72).

Important
Some of the following procedures, if not performed correctly, can render your instance unreachable or unusable. Therefore, we strongly suggest you make a backup or take a snapshot of your instance before performing these procedures.

To join a linux instance to your directory

Follow the steps for your specific Linux instance using one of the following tabs:

Amazon Linux/CENTOS/REDHAT

1. Connect to the instance using any SSH client.
2. Configure the Linux instance to use the DNS server IP addresses of the AWS Directory Service-provided DNS servers. You can do this either by setting it up in the DHCP Options set attached to the VPC or by setting it manually on the instance. If you want to set it manually, see How do I assign a static DNS server to a private Amazon EC2 instance in the AWS Knowledge Center for guidance on setting the persistent DNS server for your particular Linux distribution and version.
3. Make sure your Linux instance is up to date.

```
sudo yum -y update
```

4. Install the required Samba / Winbind packages on your Linux instance.

```
sudo yum -y install authconfig samba samba-client samba-winbind samba-winbind-clients
```

5. Make a backup of the main `smb.conf` file so you can revert back to it in case of any failure:

```
sudo cp /etc/samba/smb.conf /etc/samba/smb.bk
```

6. Open the original configuration file [`/etc/samba/smb.conf`] in a text editor.

```
sudo vim /etc/samba/smb.conf
```

Fill in your Active directory domain environment information as shown in the below example:

```
[global]
 workgroup = example
 security = ads
 realm = example.com
 idmap config * : rangesize = 1000000
 idmap config * : range = 1000000-19999999
 idmap config * : backend = autorid
 winbind enum users = no
 winbind enum groups = no
 template homedir = /home/%U@%D
 template shell = /bin/bash
 winbind use default domain = false
```

7. Open the hosts file [`/etc/hosts`] in a text editor.

```
sudo vim /etc/hosts
```

Add your Linux instance private IP address as follows:

```
10.x.x.x  Linux_hostname.example.com Linux_hostname
```

> **Note**
>
> If you did not specify your IP Address in the `/etc/hosts` file, you might receive the following DNS error while joining the instance to the domain.:
> `No DNS domain configured for linux-instance. Unable to perform DNS Update. DNS update failed: NT_STATUS_INVALID_PARAMETER`
> This error means that the join was successful but the [net ads] command was unable to register the DNS record in DNS.

8. Join the Linux instance to Active Directory using the net utility.

```
sudo net ads join -U join_account@example.com
```

`join_account@example.com`

An account in the *`example.com`* domain that has domain join privileges. Enter the password for the account when prompted. For more information about delegating these privileges, see Delegate directory join privileges for AWS Managed Microsoft AD (p. 98).

`example.com`

The fully qualified DNS name of your directory.

```
Enter join_account@example.com's password:
Using short domain name -- example
Joined 'IP-10-x-x-x' to dns domain 'example.com'
```

9. Modify PAM Configuration file, Use the command below to add the necessary entries for winbind authentication:

```
sudo authconfig --enablewinbind --enablewinbindauth  --enablemkhomedir   --update
```

10Set the SSH service to allow password authentication by editing the `/etc/ssh/sshd_config` file..

a. Open the `/etc/ssh/sshd_config` file in a text editor.

```
sudo vi /etc/ssh/sshd_config
```

b. Set the `PasswordAuthentication` setting to `yes`.

```
PasswordAuthentication yes
```

c. Restart the SSH service.

```
sudo systemctl restart sshd.service
```

Alternatively:

```
sudo service sshd restart
```

11. After the instance has restarted, connect to it with any SSH client and add the root privileges for a domain user or group to the sudoers list by performing the following steps:

a. Open the `sudoers` file with the following command:

```
sudo visudo
```

b. Add the required groups or users from your Trusting or Trusted domain as follows, and then save it.

```
## Adding Domain Users/Groups.
%domainname\\AWS\ Delegated\ Administrators ALL=(ALL:ALL) ALL
%domainname\\groupname ALL=(ALL:ALL) ALL
domainname\\username ALL=(ALL:ALL) ALL
%Trusted_DomainName\\groupname ALL=(ALL:ALL) ALL
Trusted_DomainName\\username ALL=(ALL:ALL) ALL
```

(The above example uses "\<space>" to create the Linux space character.)

SUSE

1. Connect to the instance using any SSH client.
2. Configure the Linux instance to use the DNS server IP addresses of the AWS Directory Service-provided DNS servers. You can do this either by setting it up in the DHCP Options set attached to the VPC or by setting it manually on the instance. If you want to set it manually, see How do I assign a static DNS server to a private Amazon EC2 instance in the AWS Knowledge Center for guidance on setting the persistent DNS server for your particular Linux distribution and version.
3. Make sure your SUSE Linux 15 instance is up to date.

a. Connect the package repository.

```
sudo SUSEConnect -p PackageHub/15.1/x86_64
```

b. Update SUSE.

```
sudo zypper update -y
```

4. Install the required Samba / Winbind packages on your Linux instance.

```
sudo zypper in -y samba samba-winbind
```

5. Make a backup of the main smb.conf file so you can revert back to it in case of any failure:

   ```
   sudo cp /etc/samba/smb.conf /etc/samba/smb.bk
   ```

6. Open the original configuration file [/etc/samba/smb.conf] in a text editor.

   ```
   sudo vim /etc/samba/smb.conf
   ```

   Fill in your Active directory domain environment information as shown in the below example:

   ```
   [global]
    workgroup = example
    security = ads
    realm = example.com
    idmap config * : rangesize = 1000000
    idmap config * : range = 1000000-19999999
    idmap config * : backend = autorid
    winbind enum users = no
    winbind enum groups = no
    template homedir = /home/%U@%D
    template shell = /bin/bash
    winbind use default domain = false
   ```

7. Open the hosts file [/etc/hosts] in a text editor.

   ```
   sudo vim /etc/hosts
   ```

   Add your Linux instance private IP address as follows:

   ```
   10.x.x.x  Linux_hostname.example.com Linux_hostname
   ```

   > **Note**
   > If you did not specify your IP Address in the /etc/hosts file, you might receive the following DNS error while joining the instance to the domain.:
   > No DNS domain configured for linux-instance. Unable to perform DNS Update. DNS update failed: NT_STATUS_INVALID_PARAMETER
   > This error means that the join was successful but the [net ads] command was unable to register the DNS record in DNS.

8. Join the Linux instance to the directory with the following command.

   ```
   sudo net ads join -U join_account@example.com
   ```

   join_account

   The sAMAccountName in the *example.com* domain that has domain join privileges. Enter the password for the account when prompted. For more information about delegating these privileges, see Delegate directory join privileges for AWS Managed Microsoft AD (p. 98).

   example.com

   The fully-qualified DNS name of your directory.

   ```
   Enter join_account@example.com's password:
   Using short domain name -- example
   Joined 'IP-10-x-x-x' to dns domain 'example.com'
   ```

9. Modify PAM Configuration file, Use the command below to add the necessary entries for Winbind authentication:

```
sudo pam-config --add --winbind --mkhomedir
```

10. Open the Name Service Switch configuration file [`/etc/nsswitch.conf`] in a text editor.

```
vim /etc/nsswitch.conf
```

Add the Winbind directive as shown below.

```
passwd: files winbind
shadow: files winbind
group:  files winbind
```

11. Set the SSH service to allow password authentication by editing the `/etc/ssh/sshd_config` file..

a. Open the `/etc/ssh/sshd_config` file in a text editor.

```
sudo vim /etc/ssh/sshd_config
```

b. Set the `PasswordAuthentication` setting to `yes`.

```
PasswordAuthentication yes
```

c. Restart the SSH service.

```
sudo systemctl restart sshd.service
```

Alternatively:

```
sudo service sshd restart
```

12. After the instance has restarted, connect to it with any SSH client and add root privileges for a domain user or group, to the sudoers list by performing the following steps:

a. Open the `sudoers` file with the following command:

```
sudo visudo
```

b. Add the required groups or users from your Trusting or Trusted domain as follows, and then save it.

```
## Adding Domain Users/Groups.
%domainname\\AWS\ Delegated\ Administrators ALL=(ALL:ALL) ALL
%domainname\\groupname ALL=(ALL:ALL) ALL
domainname\\username ALL=(ALL:ALL) ALL
%Trusted_DomainName\\groupname ALL=(ALL:ALL) ALL
Trusted_DomainName\\username ALL=(ALL:ALL) ALL
```

(The above example uses "\<space>" to create the Linux space character.)

Ubuntu

1. Connect to the instance using any SSH client.

2. Configure the Linux instance to use the DNS server IP addresses of the AWS Directory Service-provided DNS servers. You can do this either by setting it up in the DHCP Options set attached to the VPC or by setting it manually on the instance. If you want to set it manually, see How do I assign a static DNS server to a private Amazon EC2 instance in the AWS Knowledge Center for guidance on setting the persistent DNS server for your particular Linux distribution and version.
3. Make sure your Linux instance is up to date.

   ```
   sudo yum -y update
   ```

   ```
   sudo apt-get -y upgrade
   ```
4. Install the required Samba / Winbind packages on your Linux instance.

   ```
   sudo apt -y install samba winbind libnss-winbind libpam-winbind
   ```
5. Make a backup of the main `smb.conf` file so you can revert back to it in case of any failure.

   ```
   sudo cp /etc/samba/smb.conf /etc/samba/smb.bk
   ```
6. Open the original configuration file [`/etc/samba/smb.conf`] in a text editor.

   ```
   sudo vim /etc/samba/smb.conf
   ```

   Fill in your Active directory domain environment information as shown in the below example:

   ```
   [global]
    workgroup = example
    security = ads
    realm = example.com
    idmap config * : rangesize = 1000000
    idmap config * : range = 1000000-19999999
    idmap config * : backend = autorid
    winbind enum users = no
    winbind enum groups = no
    template homedir = /home/%U@%D
    template shell = /bin/bash
    winbind use default domain = false
   ```
7. Open the hosts file [`/etc/hosts`] in a text editor.

   ```
   sudo vim /etc/hosts
   ```

   Add your Linux instance private IP address as follows:

   ```
   10.x.x.x  Linux_hostname.example.com Linux_hostname
   ```

   > **Note**
   > If you did not specify your IP Address in the `/etc/hosts` file, you might receive the following DNS error while joining the instance to the domain.:
   > `No DNS domain configured for linux-instance. Unable to perform DNS Update. DNS update failed: NT_STATUS_INVALID_PARAMETER`
   > This error means that the join was successful but the [net ads] command was unable to register the DNS record in DNS.
8. Join the Linux instance to Active Directory using the net utility.

```
sudo net ads join -U join_account@example.com
```

join_account@example.com

An account in the *example.com* domain that has domain join privileges. Enter the password for the account when prompted. For more information about delegating these privileges, see Delegate directory join privileges for AWS Managed Microsoft AD (p. 98).

example.com

The fully qualified DNS name of your directory.

```
Enter join_account@example.com's password:
Using short domain name -- example
Joined 'IP-10-x-x-x' to dns domain 'example.com'
```

9. Modify PAM Configuration file, Use the command below to add the necessary entries for Winbind authentication:

```
sudo pam-auth-update --add --winbind --enable mkhomedir
```

10. Open the Name Service Switch configuration file [`/etc/nsswitch.conf`] in a text editor.

```
vim /etc/nsswitch.conf
```

Add the Winbind directive as shown below.

```
passwd: compat winbind
group:  compat winbind
shadow: compat winbind
```

11. Set the SSH service to allow password authentication by editing the `/etc/ssh/sshd_config` file..

    a. Open the `/etc/ssh/sshd_config` file in a text editor.

    ```
    sudo vim /etc/ssh/sshd_config
    ```

    b. Set the `PasswordAuthentication` setting to `yes`.

    ```
    PasswordAuthentication yes
    ```

    c. Restart the SSH service.

    ```
    sudo systemctl restart sshd.service
    ```

    Alternatively:

    ```
    sudo service sshd restart
    ```

12. After the instance has restarted, connect to it with any SSH client and add root privileges for a domain user or group, to the sudoers list by performing the following steps:

    a. Open the `sudoers` file with the following command:

    ```
    sudo visudo
    ```

b. Add the required groups or users from your Trusting or Trusted domain as follows, and then save it.

```
## Adding Domain Users/Groups.
%domainname\\AWS\ Delegated\ Administrators ALL=(ALL:ALL) ALL
%domainname\\groupname ALL=(ALL:ALL) ALL
domainname\\username ALL=(ALL:ALL) ALL
%Trusted_DomainName\\groupname ALL=(ALL:ALL) ALL
Trusted_DomainName\\username ALL=(ALL:ALL) ALL
```

(The above example uses "\<space>" to create the Linux space character.)

Connect to the instance

When a user connects to the instance using an SSH client, they are prompted for their user name. The user can enter the user name in either the `username@example.com` or `EXAMPLE\username` format. The response will appear similar to the following, depending on which linux distribution you are using:

Amazon Linux, Red Hat Enterprise Linux, and CentOS Linux

```
login as: johndoe@example.com
johndoe@example.com's password:
Last login: Thu Jun 25 16:26:28 2015 from XX.XX.XX.XX
```

SUSE Linux

```
SUSE Linux Enterprise Server 15 SP1 x86_64 (64-bit)

As "root" (sudo or sudo -i) use the:
  - zypper command for package management
  - yast command for configuration management

Management and Config: https://www.suse.com/suse-in-the-cloud-basics
Documentation: https://www.suse.com/documentation/sles-15/
Forum: https://forums.suse.com/forumdisplay.php?93-SUSE-Public-Cloud

Have a lot of fun...
```

Ubuntu Linux

```
login as: admin@example.com
admin@example.com@10.24.34.0's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-1057-aws x86_64)

* Documentation:  https://help.ubuntu.com
* Management:     https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

  System information as of Sat Apr 18 22:03:35 UTC 2020

  System load:  0.01              Processes:           102
  Usage of /:   18.6% of 7.69GB   Users logged in:     2
  Memory usage: 16%               IP address for eth0: 10.24.34.1
  Swap usage:   0%
```

Delegate directory join privileges for AWS Managed Microsoft AD

To join a computer to your directory, you need an account that has privileges to join computers to the directory.

With AWS Directory Service for Microsoft Active Directory, members of the **Admins** and **AWS Delegated Server Administrators** groups have these privileges.

However, as a best practice, you should use an account that has only the minimum privileges necessary. The following procedure demonstrates how to create a new group called `Joiners` and delegate the privileges to this group that are needed to join computers to the directory.

You must perform this procedure on a computer that is joined to your directory and has the **Active Directory User and Computers** MMC snap-in installed. You must also be logged in as a domain administrator.

To delegate join privileges for AWS Managed Microsoft AD

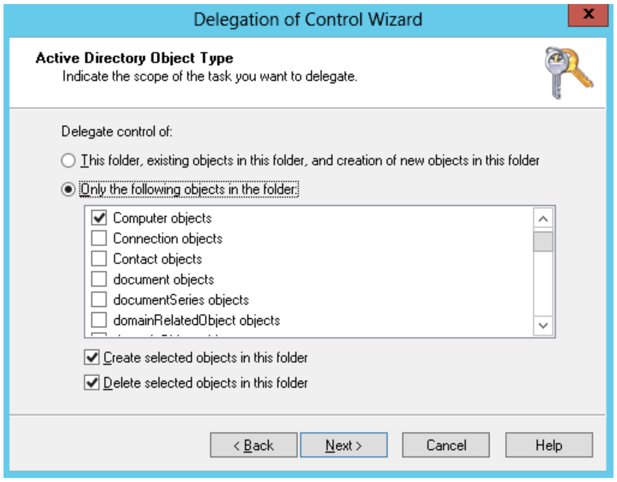
1. Open **Active Directory User and Computers** and select the organizational unit (OU) that has your NetBIOS name in the navigation tree, then select the **Users** OU.

   > **Important**
   > When you launch a AWS Directory Service for Microsoft Active Directory, AWS creates an organizational unit (OU) that contains all your directory's objects. This OU, which has the NetBIOS name that you typed when you created your directory, is located in the domain root. The domain root is owned and managed by AWS. You cannot make changes to the domain root itself, therefore, you must create the **`Joiners`** group within the OU that has your NetBIOS name.

2. Open the context menu (right-click) for **Users**, choose **New**, and then choose **Group**.
3. In the **New Object - Group** box, type the following and choose **OK**.

   - For **Group name**, type **`Joiners`**.
   - For **Group scope**, choose **Global**.
   - For **Group type**, choose **Security**.

4. In the navigation tree, select the **Computers** container under your NetBIOS name. From the **Action** menu, choose **Delegate Control**.
5. On the **Delegation of Control Wizard** page, choose **Next**, and then choose **Add**.
6. In the **Select Users, Computers, or Groups** box, type `Joiners` and choose **OK**. If more than one object is found, select the `Joiners` group created above. Choose **Next**.
7. On the **Tasks to Delegate** page, select **Create a custom task to delegate**, and then choose **Next**.
8. Select **Only the following objects in the folder**, and then select **Computer objects**.
9. Select **Create selected objects in this folder** and **Delete selected objects in this folder**. Then choose **Next**.



Delegation of Control Wizard

Active Directory Object Type
Indicate the scope of the task you want to delegate.

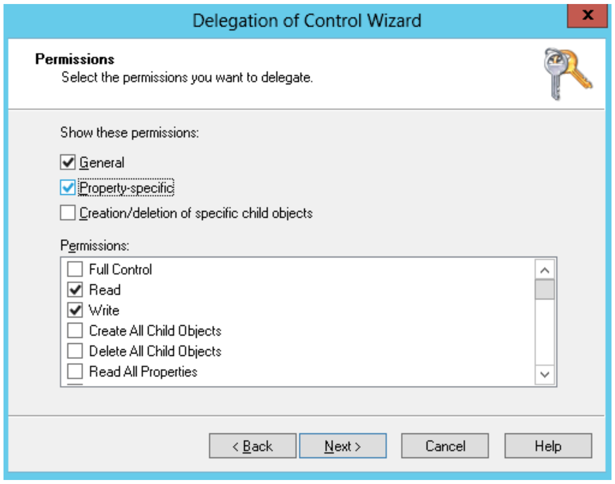
Delegate control of:

○ This folder, existing objects in this folder, and creation of new objects in this folder
● Only the following objects in the folder:

- ☑ Computer objects
- □ Connection objects
- □ Contact objects
- □ document objects
- □ documentSeries objects
- □ domainRelatedObject objects

☑ Create selected objects in this folder
☑ Delete selected objects in this folder

10. Select **Read** and **Write**, and then choose **Next**.



Delegation of Control Wizard

Permissions
Select the permissions you want to delegate.

Show these permissions:

☑ General
☑ Property-specific
□ Creation/deletion of specific child objects

Permissions:

- □ Full Control
- ☑ Read
- ☑ Write
- □ Create All Child Objects
- □ Delete All Child Objects
- □ Read All Properties

11. Verify the information on the **Completing the Delegation of Control Wizard** page and choose **Finish**.

12. Create a user with a strong password and add that user to the `Joiners` group. This user must be in the **Users** container that is under your NetBIOS name. The user will then have sufficient privileges to connect instances to the directory.

Create a DHCP options set

AWS recommends that you create a DHCP options set for your AWS Directory Service directory and assign the DHCP options set to the VPC that your directory is in. This allows any instances in that VPC to point to the specified domain and DNS servers to resolve their domain names.

For more information about DHCP options sets, see DHCP options sets in the *Amazon VPC User Guide*.

To create a DHCP options set for your directory

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **DHCP Options Sets**, and then choose **Create DHCP options set**.
3. On the **Create DHCP options set** page, enter the following values for your directory:

Name

An optional tag for the options set.

Domain name

The fully qualified name of your directory, such as `corp.example.com`.

Domain name servers

The IP addresses of your AWS-provided directory's DNS servers.

> **Note**
> You can find these addresses by going to the AWS Directory Service console navigation pane, selecting **Directories** and then choosing the correct directory ID.

NTP servers

Leave this field blank.

NetBIOS name servers

Leave this field blank.

NetBIOS node type

Leave this field blank.

4. Choose **Create DHCP options set**. The new set of DHCP options appears in your list of DHCP options.
5. Make a note of the ID of the new set of DHCP options (dopt-*xxxxxxxx*). You use it to associate the new options set with your VPC.

To change the DHCP options set associated with a VPC

After you create a set of DHCP options, you can't modify them. If you want your VPC to use a different set of DHCP options, you must create a new set and associate them with your VPC. You can also set up your VPC to use no DHCP options at all.

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Your VPCs**
3. Select the VPC, and then choose **Actions**, **Edit DHCP options set**.
4. For **DHCP options set**, select an options set or choose **No DHCP options set**, and then choose **Save**.

Manage users and groups in AWS Managed Microsoft AD

Users represent individual people or entities that have access to your directory. Groups are very useful for giving or denying privileges to groups of users, rather than having to apply those privileges to each individual user. If a user moves to a different organization, you move that user to a different group and they automatically receive the privileges needed for the new organization.

To create users and groups in an AWS Directory Service directory, you must use any instance (from either on-premises or EC2) that has been joined to your AWS Directory Service directory, and be logged in as a user that has privileges to create users and groups. You will also need to install the Active Directory Tools on your EC2 instance so you can add your users and groups with the Active Directory Users and Computers snap-in. For more information about how to set up an EC2 instance and install the necessary tools, see Step 3: Deploy an EC2 instance to manage your AWS Managed Microsoft AD (p. 163).

Note

Your user accounts must have Kerberos preauthentication enabled. This is the default setting for new user accounts, but it should not be modified. For more information about this setting, go to Preauthentication on Microsoft TechNet.

The following topics include instructions on how to create and manage users and groups.

Topics

Installing the Active Directory administration tools

To manage your directory from an EC2 Windows instance, you need to install the Active Directory Domain Services and Active Directory Lightweight Directory Services Tools on the instance. Use the following procedure to install these tools on either Windows Server 2012, Windows Server 2016, or Windows Server 2019.

You can optionally choose to install the Active Directory administration tools using Windows PowerShell. For example, you can install the Active Directory remote administration tools from a PowerShell prompt using `Install-WindowsFeature RSAT-ADDS`. For more information, see Install-WindowsFeature on the Microsoft Website.

Install the Active Directory Administration Tools on Windows Server 2012 through Windows Server 2019

To install the Active Directory administration tools on Windows Server 2012 through Windows Server 2019

1. Open Server Manager from the Start screen by choosing **Server Manager**.
2. In the **Server Manager Dashboard**, choose **Add roles and features**,
3. In the **Add Roles and Features Wizard** choose **Installation Type**, select **Role-based or feature-based installation**, and choose **Next**.
4. Under **Server Selection**, make sure the local server is selected, and choose **Features** in the left navigation pane.
5. In the **Features** tree, open **Remote Server Administration Tools**, **Role Administration Tools**, select **AD DS and AD LDS Tools**, scroll down and select **DNS Server Tools**, and then choose **Next**.
6. Review the information and choose **Install**. When the feature installation is finished, the Active Directory Domain Services and Active Directory Lightweight Directory Services Tools are available on the Start screen in the **Administrative Tools** folder.

Create a user

Use the following procedure to create a user with an EC2 instance that is joined to your AWS Managed Microsoft AD directory.

To create a user

1. Open the Active Directory Users and Computers tool. There is a shortcut to this tool in the **Administrative Tools** folder.

Tip
You can run the following from a command prompt on the instance to open the Active Directory Users and Computers tool box directly.

```
%SystemRoot%\system32\dsa.msc
```

2. In the directory tree, select an OU under your directory's NetBIOS name OU where you want to store your user (for example, Corp\Users). For more information about the OU structure used by directories in AWS, see What gets created (p. 12).
3. On the **Action** menu, click **New**, and then click **User** to open the new user wizard.
4. On the first page of the wizard, enter the values for the following fields, and then click **Next**.
   - **First name**
   - **Last name**
   - **User logon name**
5. On the second page of the wizard, type a temporary password in **Password** and **Confirm Password**. Make sure the **User must change password at next logon** option is selected. None of the other options should be selected. Click **Next**.
6. On the third page of the wizard, verify that the new user information is correct and click **Finish**. The new user will appear in the **Users** folder.

Reset a user password

Users must adhere to password policies as defined in the directory. Sometimes this can get the best of users, including the directory admin, and they forget their password. When this happens, you can quickly reset the user's password using AWS Directory Service if the user resides in either a Simple AD or AWS Managed Microsoft AD directory.

You can reset the password for any user in your directory with the following exceptions:

- For Simple AD, you cannot reset the password for any user that is a member of either the **Domain Admins** or **Enterprise Admins** group except for the Administrator user.
- For AWS Managed Microsoft AD, you cannot reset the password for any user that is in an OU other than the OU that is based off of the NetBIOS name you typed when you created your directory. For example, you cannot reset the password for a user in the **AWS Reserved** OU. For more information about the OU structure for an AWS Managed Microsoft AD directory, see What gets created (p. 12).

You can use any of the following methods to reset a user's password.

Method 1: To reset a user password (AWS Management Console)

1. In the AWS Directory Service console navigation pane, under **Active Directory**, choose **Directories**, and then select the directory in the list where you want to reset a user's password.
2. On the **Directory details** page, choose **Reset user password**.
3. In the **Reset user password** dialog, in **Username** type the user name of the user whose password needs to change.
4. Type a password in **New password** and **Confirm password**, and then choose **Reset password**.

Method 2: To reset a user password (Windows PowerShell)

1. Open Windows PowerShell.

2. Type the following command and replace the user name "joebob" and password "P@ssw0rd" with your desired credentials. See Reset-DSUserPassword Cmdlet for more information.

   ```
   Reset-DSUserPassword -UserName joebob -DirectoryId d-1234567890 -NewPassword
   P@ssw0rd
   ```

Method 3: To reset a user password (AWS CLI)

1. Open the AWS CLI.
2. Type the following command and replace the user name "joebob" and password "P@ssw0rd" with your desired credentials. See reset-user-password in the *AWS CLI Command Reference* for more information.

   ```
   aws ds reset-user-password --directory-id d-1234567890 --user-name joebob --
   new-password P@ssw0rd
   ```

Create a group

Use the following procedure to create a security group with an EC2 instance that is joined to your AWS Managed Microsoft AD directory.

To create a group

1. Open the Active Directory Users and Computers tool. There is a shortcut to this tool in the **Administrative Tools** folder.

   Tip
   You can run the following from a command prompt on the instance to open the Active Directory Users and Computers tool box directly.

   ```
   %SystemRoot%\system32\dsa.msc
   ```

2. In the directory tree, select an OU under your directory's NetBIOS name OU where you want to store your group (for example, Corp\Users). For more information about the OU structure used by directories in AWS, see What gets created (p. 12).
3. On the **Action** menu, click **New**, and then click **Group** to open the new group wizard.
4. Type a name for the group in **Group name**, select a **Group scope**, and select **Security** for the **Group type**.
5. Click **OK**. The new security group will appear in the **Users** folder.

Add a user to a group

Use the following procedure to add a user to a security group with an EC2 instance that is joined to your AWS Managed Microsoft AD directory.

To add a user to a group

1. Open the Active Directory Users and Computers tool. There is a shortcut to this tool in the **Administrative Tools** folder.

   Tip
   You can run the following from a command prompt on the instance to open the Active Directory Users and Computers tool box directly.

   ```
   %SystemRoot%\system32\dsa.msc
   ```

2. In the directory tree, select the OU under your directory's NetBIOS name OU where you stored your group, and select the group that you want to add a user as a member.
3. On the **Action** menu, click **Properties** to open the properties dialog box for the group.
4. Select the **Members** tab and click **Add**.
5. For **Enter the object names to select**, type the user name you want to add and click **OK**. The name will be displayed in the **Members** list. Click **OK** again to update the group membership.
6. Verify that the user is now a member of the group by selecting the user in the **Users** folder and clicking **Properties** in the **Action** menu to open the properties dialog box. Select the **Member Of** tab. You should see the name of the group in the list of groups that the user belongs to.

Connect to your existing AD infrastructure

This section describes how to configure trust relationships between AWS Managed Microsoft AD and your existing AD infrastructure.

Topics

Creating a trust relationship

You can configure one and two-way external and forest trust relationships between your AWS Directory Service for Microsoft Active Directory and self-managed (on-premises) directories, as well as between multiple AWS Managed Microsoft AD directories in the AWS cloud. AWS Managed Microsoft AD supports all three trust relationship directions: Incoming, Outgoing and Two-way (Bi-directional).

> **Note**
> When setting up trust relationships, you must ensure that your self-managed directory is and remains compatible with AWS Directory Services. For more information on your responsibilities, please see our shared responsibility model.

AWS Managed Microsoft AD supports both external and forest trusts. To walk through an example scenario showing how to create a forest trust, see Tutorial: Create a trust relationship between your AWS Managed Microsoft AD and your self-managed Active Directory domain (p. 108).

A two-way trust is required for AWS Enterprise Apps such as Amazon Chime, Amazon Connect, Amazon QuickSight, AWS Single Sign-On, Amazon WorkDocs, Amazon WorkMail, Amazon WorkSpaces, and the AWS Management Console. AWS Managed Microsoft AD must be able to query the users and groups in your self-managed AD.

Amazon EC2, Amazon RDS, and Amazon FSx will work with either a one-way or two-way trust.

Prerequisites

Creating the trust requires only a few steps, but you must first complete several prerequisite steps prior to setting up the trust.

> **Note**
> AWS Managed Microsoft AD does not support trust with Single Label Domains.

Connect to VPC

If you are creating a trust relationship with your self-managed directory, you must first connect your self-managed network to the VPC containing your AWS Managed Microsoft AD. The firewall for your self-managed network must have the following ports open to the CIDRs for both subnets in the VPC.

- TCP/UDP 53 - DNS
- TCP/UDP 88 - Kerberos authentication
- TCP/UDP 389 - LDAP
- TCP 445 - SMB

  Note
  SMBv1 is no longer supported.

These are the minimum ports that are needed to be able to connect to your directory. Your specific configuration may require additional ports be open.

Configure your VPC

The VPC that contains your AWS Managed Microsoft AD must have the appropriate outbound and inbound rules.

To configure your VPC outbound rules

1. In the AWS Directory Service console, on the **Directory Details** page, note your AWS Managed Microsoft AD directory ID.
2. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
3. Choose **Security Groups**.
4. Search for your AWS Managed Microsoft AD directory ID. In the search results, select the item with the description "AWS created security group for *directory ID* directory controllers".

   Note
   The selected security group is a security group that is automatically created when you initially create your directory.
5. Go to the **Outbound Rules** tab of that security group. Select **Edit**, then **Add another rule**. For the new rule, enter the following values:

   - **Type**: All Traffic
   - **Protocol**: All
   - **Destination** determines the traffic that can leave your domain controllers and where it can go in your self-managed network. Specify a single IP address or an IP address range in CIDR notation (for example, 203.0.113.5/32). You can also specify the name or ID of another security group in the same Region. For more information, see Understand your directory's AWS security group configuration and use (p. 148).
6. Select **Save**.

Enable Kerberos pre-authentication

Your user accounts must have Kerberos pre-authentication enabled. For more information about this setting, review Preauthentication on Microsoft TechNet.

Configure DNS conditional forwarders on your self-managed domain

You must set up DNS conditional forwarders on your self-managed domain. Refer to Assign a Conditional Forwarder for a Domain Name on Microsoft TechNet for details on conditional forwarders.

To perform the following steps, you must have access to following Windows Server tools for your self-managed domain:

- AD DS and AD LDS Tools
- DNS

To configure conditional forwarders on your self-managed domain

1. First you must get some information about your AWS Managed Microsoft AD. Sign into the AWS Management Console and open the AWS Directory Service console at https://console.aws.amazon.com/directoryservicev2/.
2. In the navigation pane, select **Directories**.
3. Choose the directory ID of your AWS Managed Microsoft AD.
4. Take note of the fully qualified domain name (FQDN) and the DNS addresses of your directory.
5. Now, return to your self-managed domain controller. Open Server Manager.
6. On the **Tools** menu, choose **DNS**.
7. In the console tree, expand the DNS server of the domain for which you are setting up the trust.
8. In the console tree, choose **Conditional Forwarders**.
9. On the **Action** menu, choose **New conditional forwarder**.
10. In **DNS domain**, type the fully qualified domain name (FQDN) of your AWS Managed Microsoft AD, which you noted earlier.
11. Choose **IP addresses of the master servers** and type the DNS addresses of your AWS Managed Microsoft AD directory, which you noted earlier.

    After entering the DNS addresses, you might get a "timeout" or "unable to resolve" error. You can generally ignore these errors.
12. Select **Store this conditional forwarder in Active Directory and replicate as follows: All DNS servers in this domain**. Choose **OK**.

Trust relationship password

If you are creating a trust relationship with an existing domain, set up the trust relationship on that domain using Windows Server Administration tools. As you do so, note the trust password that you use. You will need to use this same password when setting up the trust relationship on the AWS Managed Microsoft AD. For more information, see Managing Trusts on Microsoft TechNet.

You are now ready to create the trust relationship on your AWS Managed Microsoft AD.

Create, verify, or delete a trust relationship

> **Note**
> Trust relationships is a global feature of AWS Managed Microsoft AD. If you are using Multi-Region replication (p. 58), the following procedures must be performed in the Primary Region (p. 59). The changes will be applied across all replicated Regions automatically. For more information, see Global vs Regional features (p. 58).

To create a trust relationship with your AWS Managed Microsoft AD

1. Open the AWS Directory Service console.
2. On the **Directories** page, choose your AWS Managed Microsoft AD ID.
3. On the **Directory details** page, do one of the following:

- If you have multiple Regions showing under **Multi-Region replication**, select the primary Region, and then choose the **Networking & security** tab. For more information, see Primary vs additional Regions (p. 59).
- If you do not have any Regions showing under **Multi-Region replication**, choose the **Networking & security** tab.

4. In the **Trust relationships** section, choose **Actions**, and then select **Add trust relationship**.
5. On the **Add a trust relationship** page, provide the required information, including the trust type, fully qualified domain name (FQDN) of your trusted domain, the trust password and the trust direction.
6. (Optional) If you want to allow only authorized users to access resources in your AWS Managed Microsoft AD directory, you can optionally choose the **Selective authentication** check box. For general information about selective authentication, see Security Considerations for Trusts on Microsoft TechNet.
7. For **Conditional forwarder**, type the IP address of your self-managed DNS server. If you have previously created conditional forwarders, you can type the FQDN of your self-managed domain instead of a DNS IP address.
8. (Optional) Choose **Add another IP address** and type the IP address of an additional self-managed DNS server. You can repeat this step for each applicable DNS server address for a total of four addresses.
9. Choose **Add**.
10. If the DNS server or the network for your self-managed domain uses a public (non-RFC 1918) IP address space, go to the **IP routing** section, choose **Actions**, and then choose **Add route**. Type the IP address block of your DNS server or self-managed network using CIDR format, for example 203.0.113.0/24. This step is not necessary if both your DNS server and your self-managed network are using RFC 1918 IP address spaces.

    Note
    When using a public IP address space, make sure that you do not use any of the AWS IP address ranges as these cannot be used.
11. (Optional) We recommend that while you are on the **Add routes** page that you also select **Add routes to the security group for this directory's VPC**. This will configure the security groups as detailed above in the "Configure your VPC." These security rules impact an internal network interface that is not exposed publicly. If this option is not available, you will instead see a message indicating that you have already customized your security groups.

You must set up the trust relationship on both domains. The relationships must be complementary. For example, if you create an outgoing trust on one domain, you must create an incoming trust on the other.

If you are creating a trust relationship with an existing domain, set up the trust relationship on that domain using Windows Server Administration tools.

You can create multiple trusts between your AWS Managed Microsoft AD and various Active Directory domains. However, only one trust relationship per pair can exist at a time. For example, if you have an existing, one-way trust in the “Incoming direction” and you then want to set up another trust relationship in the “Outgoing direction,” you will need to delete the existing trust relationship, and create a new “Two-way” trust.

To verify an outgoing trust relationship

1. Open the AWS Directory Service console.
2. On the **Directories** page, choose your AWS Managed Microsoft AD ID.
3. On the **Directory details** page, do one of the following:

- If you have multiple Regions showing under **Multi-Region replication**, select the primary Region, and then choose the **Networking & security** tab. For more information, see Primary vs additional Regions (p. 59).
- If you do not have any Regions showing under **Multi-Region replication**, choose the **Networking & security** tab.

4. In the **Trust relationships** section, select the trust you want to verify, choose **Actions**, and then select **Verify trust relationship**.

This process verifies only the outgoing direction of a two-way trust. AWS does not support verification of an incoming trusts. For more information on how to verify a trust to or from your self-managed Active Directory, refer to Verify a Trust on Microsoft TechNet.

To delete an existing trust relationship

1. Open the AWS Directory Service console.
2. On the **Directories** page, choose your AWS Managed Microsoft AD ID.
3. On the **Directory details** page, do one of the following:
   - If you have multiple Regions showing under **Multi-Region replication**, select the primary Region, and then choose the **Networking & security** tab. For more information, see Primary vs additional Regions (p. 59).
   - If you do not have any Regions showing under **Multi-Region replication**, choose the **Networking & security** tab.
4. In the **Trust relationships** section, select the trust you want to delete, choose **Actions**, and then select **Delete trust relationship**.
5. Choose **Delete**.

Adding IP routes when using public IP addresses

You can use AWS Directory Service for Microsoft Active Directory to take advantage of many powerful Active Directory features, including establishing trusts with other directories. However, if the DNS servers for the networks of the other directories use public (non-RFC 1918) IP addresses, you must specify those IP addresses as part of configuring the trust. Instructions for doing this can be found in Creating a trust relationship (p. 104).

Similarly, you must also enter the IP address information when routing traffic from your AWS Managed Microsoft AD on AWS to a peer AWS VPC, if the VPC uses public IP ranges.

When you add the IP addresses as described in Creating a trust relationship (p. 104), you have the option of selecting **Add routes to the security group for this directory's VPC**. This option should be selected unless you have previously customized your security group to allow the necessary traffic as shown below. For more information, see Understand your directory's AWS security group configuration and use (p. 148).

Tutorial: Create a trust relationship between your AWS Managed Microsoft AD and your self-managed Active Directory domain

This tutorial walks you through all the steps necessary to set up a trust relationship between AWS Directory Service for Microsoft Active Directory and your self-managed (on-premises) Microsoft Active Directory. Although creating the trust requires only a few steps, you must first complete the following prerequisite steps.

Topics

See Also

Creating a trust relationship (p. 104)

Prerequisites

This tutorial assumes you already have the following:

> **Note**
> AWS Managed Microsoft AD does not support trust with Single label domains.

- An AWS Managed Microsoft AD directory created on AWS. If you need help doing this, see Getting started with AWS Managed Microsoft AD (p. 9).
- An EC2 instance running Windows added to that AWS Managed Microsoft AD. If you need help doing this, see Manually join a Windows instance (p. 73).

  > **Important**
  > The admin account for your AWS Managed Microsoft AD must have administrative access to this instance.
- The following Windows Server tools installed on that instance:
  - AD DS and AD LDS Tools
  - DNS

  If you need help doing this, see Installing the Active Directory administration tools (p. 101).
- A self-managed (on-premises) Microsoft Active Directory

  You must have administrative access to this directory. The same Windows Server tools as listed above must also be available for this directory.
- An active connection between your self-managed network and the VPC containing your AWS Managed Microsoft AD. If you need help doing this, see Amazon Virtual Private Cloud Connectivity Options.
- A correctly set local security policy. Check `Local Security Policy > Local Policies > Security Options > Network access: Named Pipes that can be accessed anonymously` and ensure that it contains at least the following three named pipes:
  - netlogon
  - samr
  - lsarpc

Tutorial configuration

For this tutorial, we've already created a AWS Managed Microsoft AD and a self-managed domain. The self-managed network is connected to the AWS Managed Microsoft AD's VPC. Following are the properties of the two directories:

AWS Managed Microsoft AD running on AWS

- Domain name (FQDN): MyManagedAD.example.com
- NetBIOS name: MyManagedAD
- DNS Addresses: 10.0.10.246, 10.0.20.121

- VPC CIDR: 10.0.0.0/16

The AWS Managed Microsoft AD resides in VPC ID: vpc-12345678.

Self-managed or AWS Managed Microsoft AD domain

- Domain name (FQDN): corp.example.com
- NetBIOS name: CORP
- DNS Addresses: 172.16.10.153
- Self-managed CIDR: 172.16.0.0/16

Next Step

Step 1: Prepare your self-managed AD Domain (p. 110)

Step 1: Prepare your self-managed AD Domain

First you need to complete several prerequisite steps on your self-managed (on-premises) domain.

Configure your self-managed firewall

You must configure your self-managed firewall so that the following ports are open to the CIDRs for all subnets used by the VPC that contains your AWS Managed Microsoft AD. In this tutorial, we allow both incoming and outgoing traffic from 10.0.0.0/16 (the CIDR block of our AWS Managed Microsoft AD's VPC) on the following ports:

- TCP/UDP 53 - DNS
- TCP/UDP 88 - Kerberos authentication
- TCP/UDP 389 - LDAP
- TCP 445 - SMB

  Note
  SMBv1 is no longer supported.

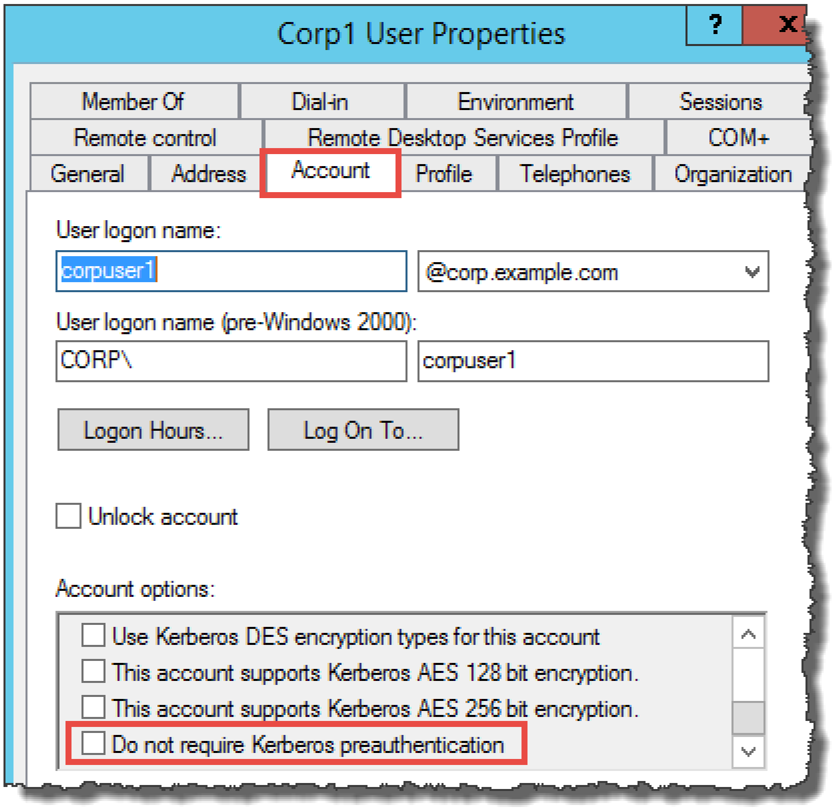
Note
These are the minimum ports that are needed to connect the VPC to the self-managed directory. Your specific configuration may require additional ports be open.

Ensure that Kerberos pre-authentication is enabled

User accounts in both directories must have Kerberos preauthentication enabled. This is the default, but let's check the properties of any random user to make sure nothing has changed.

To view user kerberos settings

1. On your self-managed domain controller, open Server Manager.
2. On the **Tools** menu, choose **Active Directory Users and Computers**.
3. Choose the **Users** folder and open the context (right-click) menu. Select any random user account listed in the right pane. Choose **Properties**.
4. Choose the **Account** tab. In the **Account options** list, scroll down and ensure that **Do not require Kerberos preauthentication** is *not* checked.



User logon name:
corpuser1 | @corp.example.com

User logon name (pre-Windows 2000):
CORP\ | corpuser1

□ Unlock account

Account options:

□ Use Kerberos DES encryption types for this account
□ This account supports Kerberos AES 128 bit encryption.
□ This account supports Kerberos AES 256 bit encryption.
□ Do not require Kerberos preauthentication

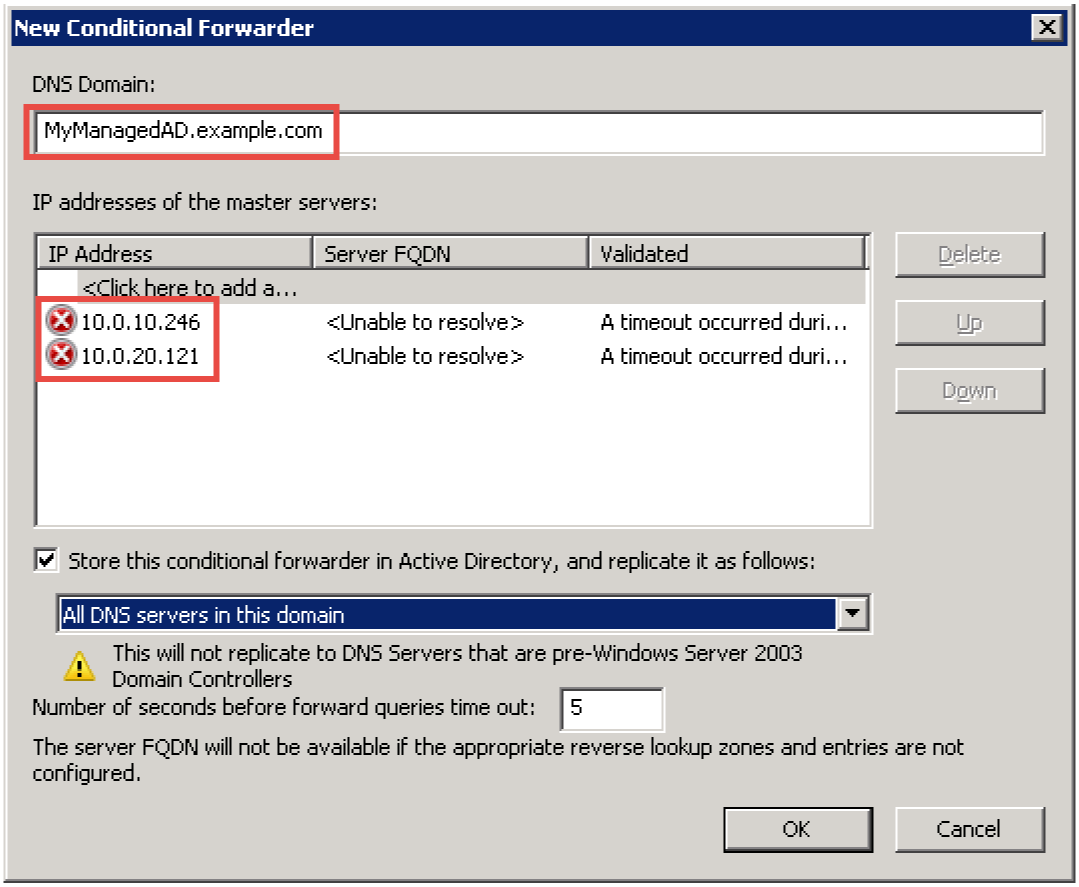
Configure DNS conditional forwarders for your self-managed domain

You must set up DNS conditional forwarders on each domain. Before doing this on your self-managed domain, you will first get some information about your AWS Managed Microsoft AD.

To configure conditional forwarders on your self-managed domain

1. Sign into the AWS Management Console and open the AWS Directory Service console at https://console.aws.amazon.com/directoryservicev2/.
2. In the navigation pane, select **Directories**.
3. Choose the directory ID of your AWS Managed Microsoft AD.
4. On the **Details** page, take note of the values in **Directory name** and the **DNS address** of your directory.
5. Now, return to your self-managed domain controller. Open Server Manager.
6. On the **Tools** menu, choose **DNS**.
7. In the console tree, expand the DNS server of the domain for which you are setting up the trust. Our server is WIN-5V70CN7VJ0.corp.example.com.
8. In the console tree, choose **Conditional Forwarders**.
9. On the **Action** menu, choose **New conditional forwarder**.
10. In **DNS domain**, type the fully qualified domain name (FQDN) of your AWS Managed Microsoft AD, which you noted earlier. In this example, the FQDN is MyManagedAD.example.com.
11. Choose **IP addresses of the master servers** and type the DNS addresses of your AWS Managed Microsoft AD directory, which you noted earlier. In this example those are: 10.0.10.246, 10.0.20.121

After entering the DNS addresses, you might get a "timeout" or "unable to resolve" error. You can generally ignore these errors.


New Conditional Forwarder

DNS Domain: MyManagedAD.example.com

IP addresses of the master servers:

IP Address	Server FQDN	Validated
<Click here to add a...		
☒ 10.0.10.246	<Unable to resolve>	A timeout occurred duri...
☒ 10.0.20.121	<Unable to resolve>	A timeout occurred duri...

☑ Store this conditional forwarder in Active Directory, and replicate it as follows:

All DNS servers in this domain

⚠ This will not replicate to DNS Servers that are pre-Windows Server 2003 Domain Controllers

Number of seconds before forward queries time out: 5

The server FQDN will not be available if the appropriate reverse lookup zones and entries are not configured.

12. Select **Store this conditional forwarder in Active Directory, and replicate it as follows**.
13. Select **All DNS servers in this domain**, and then choose **OK**.

Next Step

Step 2: Prepare your AWS Managed Microsoft AD (p. 112)

Step 2: Prepare your AWS Managed Microsoft AD

Now let's get your AWS Managed Microsoft AD ready for the trust relationship. Many of the following steps are almost identical to what you just completed for your self-managed domain. This time, however, you are working with your AWS Managed Microsoft AD.

Configure your VPC subnets and security groups

You must allow traffic from your self-managed network to the VPC containing your AWS Managed Microsoft AD. To do this, you will need to make sure that the ACLs associated with the subnets used to deploy your AWS Managed Microsoft AD and the security group rules configured on your domain controllers, both allow the requisite traffic to support trusts.

Port requirements vary based on the version of Windows Server used by your domain controllers and the services or applications that will be leveraging the trust. For the purposes of this tutorial, you will need to open the following ports:

Inbound

- TCP/UDP 53 - DNS
- TCP/UDP 88 - Kerberos authentication
- UDP 123 - NTP
- TCP 135 - RPC
- TCP/UDP 389 - LDAP
- TCP/UDP 445 - SMB

  Note
  SMBv1 is no longer supported.
- TCP/UDP 464 - Kerberos authentication
- TCP 636 - LDAPS (LDAP over TLS/SSL)
- TCP 3268-3269 - Global Catalog
- TCP/UDP 49152-65535 - Ephemeral ports for RPC

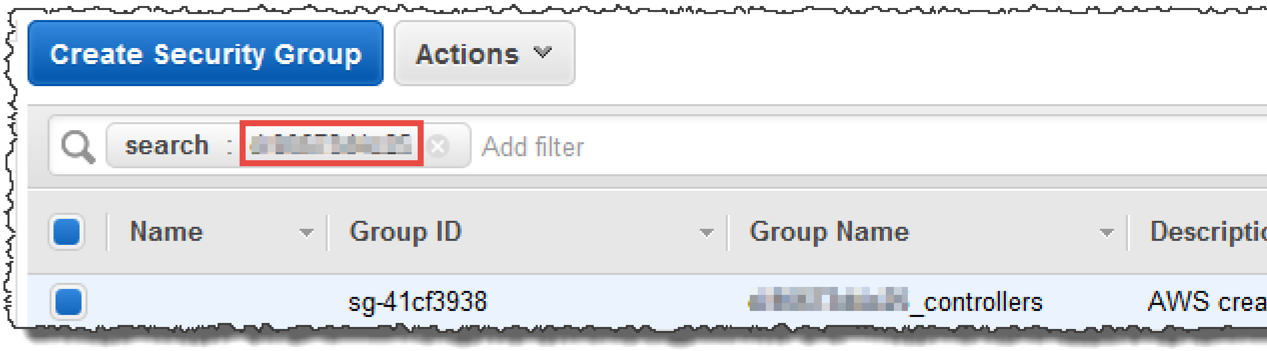
Outbound

- ALL

> **Note**
> These are the minimum ports that are needed to be able to connect the VPC and self-managed directory. Your specific configuration may require additional ports be open.

To configure your AWS Managed Microsoft AD domain controller outbound and inbound rules

1. Return to the AWS Directory Service console. In the list of directories, take note the directory ID for your AWS Managed Microsoft AD directory.
2. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
3. In the navigation pane, choose **Security Groups**.
4. Use the search box to search for your AWS Managed Microsoft AD directory ID. In the search results, select the item with the description **AWS created security group for <yourdirectoryID> directory controllers**.



search : [redacted]

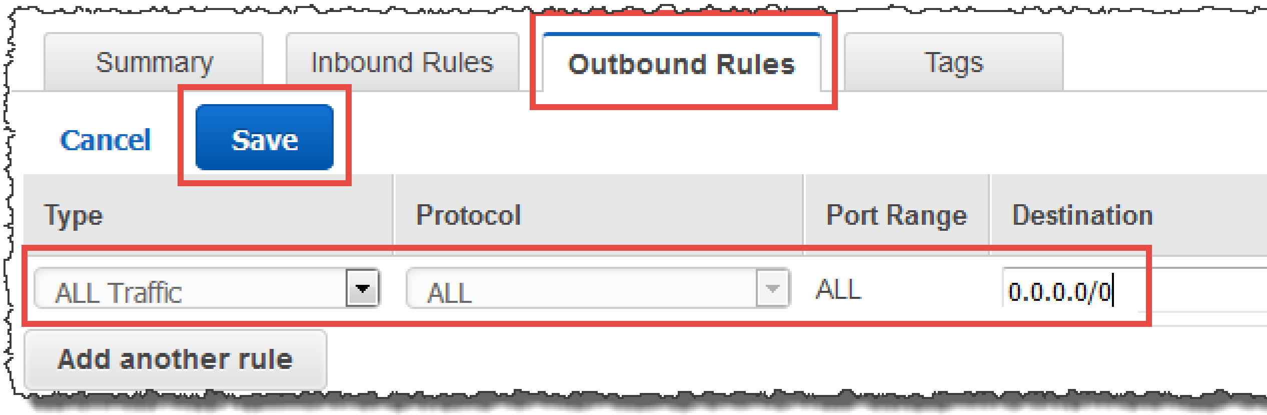
	Name	Group ID	Group Name	Descriptio
■		sg-41cf3938	[redacted]_controllers	AWS crea

5. Go to the **Outbound Rules** tab for that security group. Choose **Edit**, and then **Add another rule**. For the new rule, enter the following values:

   - **Type**: ALL Traffic
   - **Protocol**: ALL
   - **Destination** determines the traffic that can leave your domain controllers and where it can go. Specify a single IP address or an IP address range in CIDR notation (for example, 203.0.113.5/32).

You can also specify the name or ID of another security group in the same Region. For more information, see Understand your directory’s AWS security group configuration and use (p. 148).

6. Select **Save**.



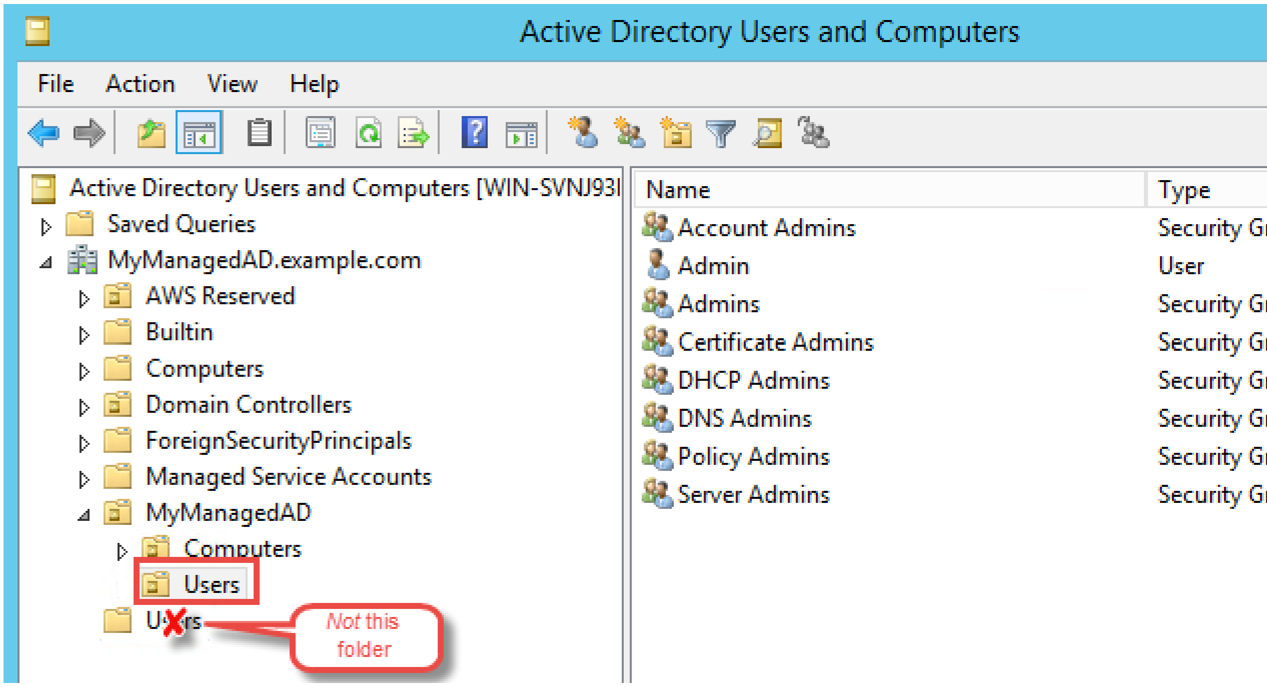
Type	Protocol	Port Range	Destination
ALL Traffic	ALL	ALL	0.0.0.0/0

Ensure that Kerberos pre-authentication is enabled

Now you want to confirm that users in your AWS Managed Microsoft AD also have Kerberos pre-authentication enabled. This is the same process you completed for your self-managed directory. This is the default, but let's check to make sure nothing has changed.

To view user kerberos settings

1. Log in to an instance that is a member of your AWS Managed Microsoft AD directory using either the Admin account (p. 17) for the domain or an account that has been delegated permissions to manage users in the domain.
2. If they are not already installed, install the Active Directory Users and Computers tool and the DNS tool. Learn how to install these tools in Installing the Active Directory administration tools (p. 101).
3. Open Server Manager. On the **Tools** menu, choose **Active Directory Users and Computers**.
4. Choose the **Users** folder in your domain. Note that this is the **Users** folder under your NetBIOS name, not the **Users** folder under the fully qualified domain name (FQDN).



5. In the list of users, right-click on a user, and then choose **Properties**.
6. Choose the **Account** tab. In the **Account options** list, ensure that **Do not require Kerberos preauthentication** is *not* checked.

Next Step

Step 3: Create the trust relationship (p. 115)

Step 3: Create the trust relationship

Now that the preparation work is complete, the final steps are to create the trusts. First you create the trust on your self-managed domain, and then finally on your AWS Managed Microsoft AD. If you have any issues during the trust creation process, see Trust creation status reasons (p. 182) for assistance.

Configure the trust in your self-managed Active Directory

In this tutorial, you configure a two-way forest trust. However, if you create a one-way forest trust, be aware that the trust directions on each of your domains must be complementary. For example, if you create a one-way, outgoing trust on your self-managed domain, you need to create a one-way, incoming trust on your AWS Managed Microsoft AD.

> **Note**
> AWS Managed Microsoft AD also supports external trusts. However, for the purposes of this tutorial, you will create a two-way forest trust.

To configure the trust in your self-managed AD

1. Open Server Manager and on the **Tools** menu, choose **Active Directory Domains and Trusts**.
2. Open the context (right-click) menu of your domain and choose **Properties**.
3. Choose the **Trusts** tab and choose **New trust**. Type the name of your AWS Managed Microsoft AD and choose **Next**.
4. Choose **Forest trust**. Choose **Next**.

5. Choose **Two-way**. Choose **Next**.
6. Choose **This domain only**. Choose **Next**.
7. Choose **Forest-wide authentication**. Choose **Next**.
8. Type a **Trust password**. Make sure to remember this password as you will need it when setting up the trust for your AWS Managed Microsoft AD.
9. In the next dialog box, confirm your settings and choose **Next**. Confirm that the trust was created successfully and again choose **Next**.
10. Choose **No, do not confirm the outgoing trust**. Choose **Next**.
11. Choose **No, do not confirm the incoming trust**. Choose **Next**.

Configure the trust in your AWS Managed Microsoft AD directory

Finally, you configure the forest trust relationship with your AWS Managed Microsoft AD directory. Because you created a two-way forest trust on the self-managed domain, you also create a two-way trust using your AWS Managed Microsoft AD directory.

> **Note**
> Trust relationships is a global feature of AWS Managed Microsoft AD. If you are using Multi-Region replication (p. 58), the following procedures must be performed in the Primary Region (p. 59). The changes will be applied across all replicated Regions automatically. For more information, see Global vs Regional features (p. 58).

To configure the trust in your AWS Managed Microsoft AD directory

1. Return to the AWS Directory Service console.
2. On the **Directories** page, choose your AWS Managed Microsoft AD ID.
3. On the **Directory details** page, do one of the following:
   - If you have multiple Regions showing under **Multi-Region replication**, select the primary Region, and then choose the **Networking & security** tab. For more information, see Primary vs additional Regions (p. 59).
   - If you do not have any Regions showing under **Multi-Region replication**, choose the **Networking & security** tab.
4. In the **Trust relationships** section, choose **Actions**, and then select **Add trust relationship**.
5. On the **Add a trust relationship** page, specify the Trust type. In this case, we choose **Forest trust**. Type the FQDN of your self-managed domain (in this tutorial **`corp.example.com`**). Type the same trust password that you used when creating the trust on your self-managed domain. Specify the direction. In this case, we choose **Two-way**.
6. In the **Conditional forwarder** field, enter the IP address of your self-managed DNS server. In this example, enter 172.16.10.153.
7. (Optional) Choose **Add another IP address** and enter a second IP address for your self-managed DNS server. You can specify up to a total of four DNS servers.
8. Choose **Add**.

Congratulations. You now have a trust relationship between your self-managed domain (corp.example.com) and your AWS Managed Microsoft AD (MyManagedAD.example.com). Only one relationship can be set up between these two domains. If for example, you want to change the trust direction to one-way, you would first need to delete this existing trust relationship and create a new one.

For more information, including instructions about verifying or deleting trusts, see Creating a trust relationship (p. 104).

Tutorial: Create a trust relationship between two AWS Managed Microsoft AD domains

This tutorial walks you through all the steps necessary to set up a trust relationship between two AWS Directory Service for Microsoft Active Directory domains.

Topics

See Also

Creating a trust relationship (p. 104)

Step 1: Prepare your AWS Managed Microsoft AD

In this section, you will get your AWS Managed Microsoft AD ready for the trust relationship with another AWS Managed Microsoft AD. Many of the following steps are almost identical to what you completed in Tutorial: Create a trust relationship between your AWS Managed Microsoft AD and your self-managed Active Directory domain (p. 108). This time, however, you are configuring your AWS Managed Microsoft AD environments to work with each other.

Configure your VPC subnets and security groups

You must allow traffic from one AWS Managed Microsoft AD network to the VPC containing your other AWS Managed Microsoft AD. To do this, you will need to make sure that the ACLs associated with the subnets used to deploy your AWS Managed Microsoft AD and the security group rules configured on your domain controllers, both allow the requisite traffic to support trusts.

Port requirements vary based on the version of Windows Server used by your domain controllers and the services or applications that will be leveraging the trust. For the purposes of this tutorial, you will need to open the following ports:

Inbound

- TCP/UDP 53 - DNS
- TCP/UDP 88 - Kerberos authentication
- UDP 123 - NTP
- TCP 135 - RPC
- TCP/UDP 389 - LDAP
- TCP/UDP 445 - SMB

  Note
  SMBv1 is no longer supported.
- TCP/UDP 464 - Kerberos authentication
- TCP 636 - LDAPS (LDAP over TLS/SSL)
- TCP 3268-3269 - Global Catalog
- TCP/UDP 1024-65535 - Ephemeral ports for RPC

Outbound

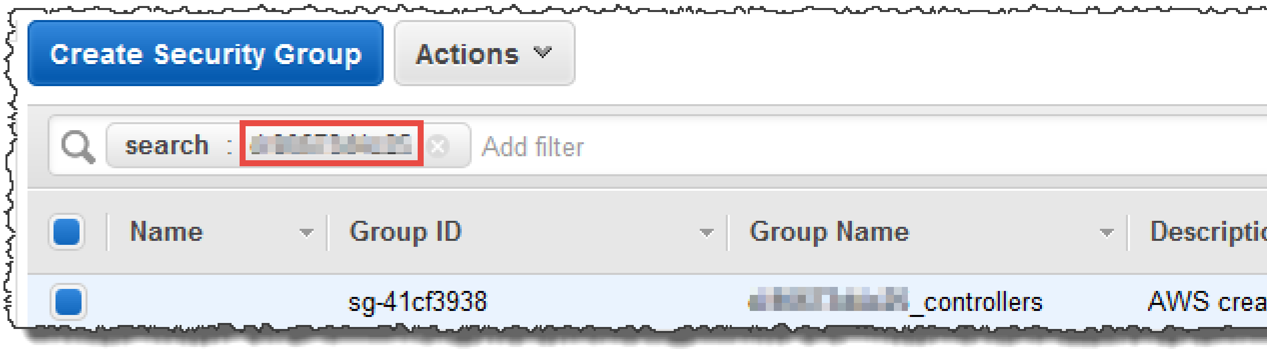
- ALL

Note
These are the minimum ports that are needed to be able to connect the VPCs from both AWS Managed Microsoft AD's. Your specific configuration may require additional ports be open. For more information, see How to configure a firewall for Active Directory domains and trusts on Microsoft's website.

To configure your AWS Managed Microsoft AD domain controller outbound rules

Note
Repeat steps 1-6 below for each directory.

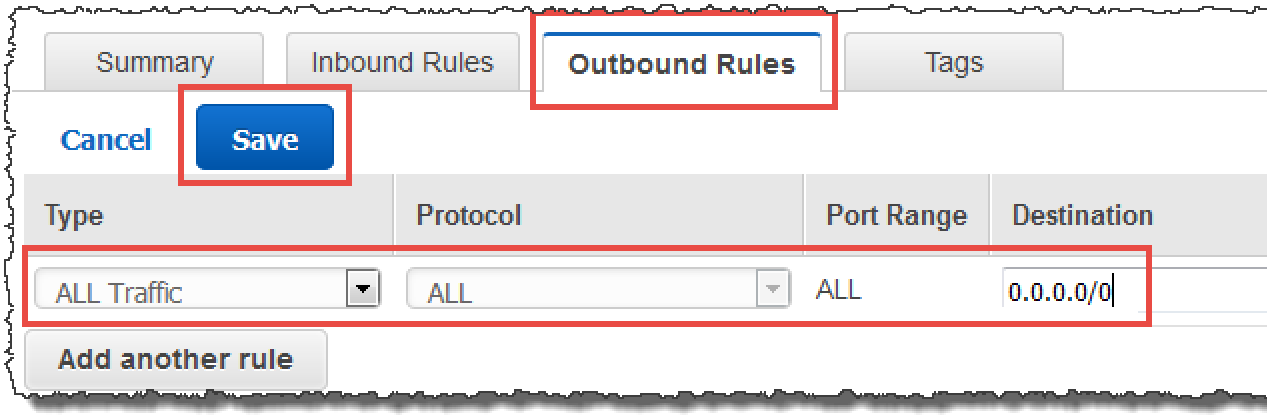
1. Go to the AWS Directory Service console. In the list of directories, take note the directory ID for your AWS Managed Microsoft AD directory.
2. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
3. In the navigation pane, choose **Security Groups**.
4. Use the search box to search for your AWS Managed Microsoft AD directory ID. In the search results, select the item with the description **AWS created security group for <yourdirectoryID> directory controllers**.



■	Name	Group ID	Group Name	Descriptic
■		sg-41cf3938	[redacted]_controllers	AWS crea

5. Go to the **Outbound Rules** tab for that security group. Choose **Edit**, and then **Add another rule**. For the new rule, enter the following values:

   - **Type**: ALL Traffic
   - **Protocol**: ALL
   - **Destination** determines the traffic that can leave your domain controllers and where it can go. Specify a single IP address or an IP address range in CIDR notation (for example, 203.0.113.5/32). You can also specify the name or ID of another security group in the same Region. For more information, see Understand your directory's AWS security group configuration and use (p. 148).
6. Select **Save**.



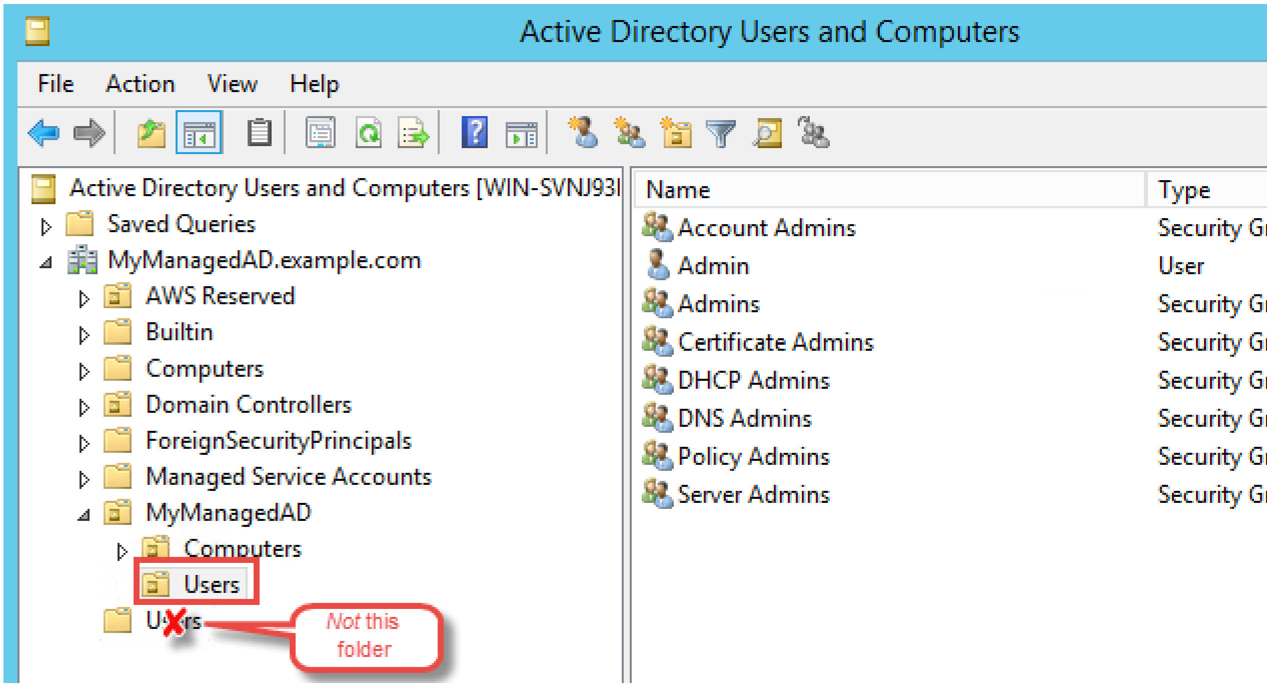
Type	Protocol	Port Range	Destination
ALL Traffic	ALL	ALL	0.0.0.0/0

Ensure that Kerberos pre-authentication is enabled

Now you want to confirm that users in your AWS Managed Microsoft AD also have Kerberos pre-authentication enabled. This is the same process you completed for your on-premises directory. This is the default, but let's check to make sure nothing has changed.

To view user kerberos settings

1. Log in to an instance that is a member of your AWS Managed Microsoft AD directory using either the Admin account (p. 17) for the domain or an account that has been delegated permissions to manage users in the domain.
2. If they are not already installed, install the Active Directory Users and Computers tool and the DNS tool. Learn how to install these tools in Installing the Active Directory administration tools (p. 101).
3. Open Server Manager. On the **Tools** menu, choose **Active Directory Users and Computers**.
4. Choose the **Users** folder in your domain. Note that this is the **Users** folder under your NetBIOS name, not the **Users** folder under the fully qualified domain name (FQDN).



Name	Type
Account Admins	Security G
Admin	User
Admins	Security G
Certificate Admins	Security G
DHCP Admins	Security G
DNS Admins	Security G
Policy Admins	Security G
Server Admins	Security G

5. In the list of users, right-click on a user, and then choose **Properties**.
6. Choose the **Account** tab. In the **Account options** list, ensure that **Do not require Kerberos preauthentication** is *not* checked.

Next Step

Step 2: Create the trust relationship with another AWS Managed Microsoft AD domain (p. 119)

Step 2: Create the trust relationship with another AWS Managed Microsoft AD domain

Now that the preparation work is complete, the final steps are to create the trusts between your two AWS Managed Microsoft AD domains. If you have any issues during the trust creation process, see Trust creation status reasons (p. 182) for assistance.

Configure the trust in your first AWS Managed Microsoft AD domain

In this tutorial, you configure a two-way forest trust. However, if you create a one-way forest trust, be aware that the trust directions on each of your domains must be complementary. For example, if you create a one-way, outgoing trust on this first domain, you need to create a one-way, incoming trust on your second AWS Managed Microsoft AD domain.

> **Note**
> AWS Managed Microsoft AD also supports external trusts. However, for the purposes of this tutorial, you will create a two-way forest trust.

To configure the trust in your first AWS Managed Microsoft AD domain

1. Open the AWS Directory Service console.
2. On the **Directories** page, choose your first AWS Managed Microsoft AD ID.
3. On the **Directory details** page, do one of the following:
   - If you have multiple Regions showing under **Multi-Region replication**, select the primary Region, and then choose the **Networking & security** tab. For more information, see Primary vs additional Regions (p. 59).
   - If you do not have any Regions showing under **Multi-Region replication**, choose the **Networking & security** tab.
4. In the **Trust relationships** section, choose **Actions**, and then select **Add trust relationship**.
5. On the **Add a trust relationship** page, Type the FQDN of your second AWS Managed Microsoft AD domain. Make sure to remember this password as you will need it when setting up the trust for your second AWS Managed Microsoft AD. Specify the direction. In this case, choose **Two-way**.
6. In the **Conditional forwarder** field, enter the IP address of your second AWS Managed Microsoft AD DNS server.
7. (Optional) Choose **Add another IP address** and enter a second IP address for your second AWS Managed Microsoft AD DNS server. You can specify up to a total of four DNS servers.
8. Choose **Add**. The trust will fail at this point which is expected until we create the other side of the trust.

Configure the trust in your second AWS Managed Microsoft AD domain

Now, you configure the forest trust relationship with your second AWS Managed Microsoft AD directory. Because you created a two-way forest trust on the first AWS Managed Microsoft AD domain, you also create a two-way trust using this AWS Managed Microsoft AD domain.

To configure the trust in your second AWS Managed Microsoft AD domain

1. Return to the AWS Directory Service console.
2. On the **Directories** page, choose your second AWS Managed Microsoft AD ID.
3. On the **Directory details** page, do one of the following:
   - If you have multiple Regions showing under **Multi-Region replication**, select the primary Region, and then choose the **Networking & security** tab. For more information, see Primary vs additional Regions (p. 59).
   - If you do not have any Regions showing under **Multi-Region replication**, choose the **Networking & security** tab.
4. In the **Trust relationships** section, choose **Actions**, and then select **Add trust relationship**.
5. On the **Add a trust relationship** page, Type the FQDN of your first AWS Managed Microsoft AD domain. Type the same trust password that you used when creating the trust on your on-premises domain. Specify the direction. In this case, choose **Two-way**.

6. In the **Conditional forwarder** field, enter the IP address of your first AWS Managed Microsoft AD DNS server.
7. (Optional) Choose **Add another IP address** and enter a second IP address for your first AWS Managed Microsoft AD DNS server. You can specify up to a total of four DNS servers.
8. Choose **Add**. The trust should be verified shortly afterwards.
9. Now, go back to the trust you created in the first domain and verify the trust relationship again.

Congratulations. You now have a trust relationship between your two AWS Managed Microsoft AD domains. Only one relationship can be set up between these two domains. If for example, you want to change the trust direction to one-way, you would first need to delete this existing trust relationship and create a new one.

Extend your schema

AWS Managed Microsoft AD uses schemas to organize and enforce how directory data is stored. The process of adding definitions to the schema is referred to as “extending the schema.” Schema extensions make it possible for you to modify the schema of your AWS Managed Microsoft AD directory using a valid LDAP Data Interchange Format (LDIF) file. For more information about AD schemas and how to extend your schema, see the topics listed below.

Topics

When to extend your AWS Managed Microsoft AD schema

You can extend your AWS Managed Microsoft AD schema by adding new object classes and attributes. For example, you might do this if you have an application that requires changes to your schema in order to support single sign-on capabilities.

You can also use schema extensions to enable support for applications that rely on specific Active Directory object classes and attributes. This can be especially useful in the case where you need to migrate corporate applications that are dependent on AWS Managed Microsoft AD, to the AWS cloud.

Each attribute or class that is added to an existing Active Directory schema must be defined with a unique ID. That way when companies add extensions to the schema, they can be guaranteed to be unique and not to conflict with each other. These IDs are referred to as AD Object Identifiers (OIDs) and are stored in AWS Managed Microsoft AD.

To get started, see Tutorial: Extending your AWS Managed Microsoft AD schema (p. 121).

Related topics

- Extend your schema (p. 121)
- Schema elements (p. 19)

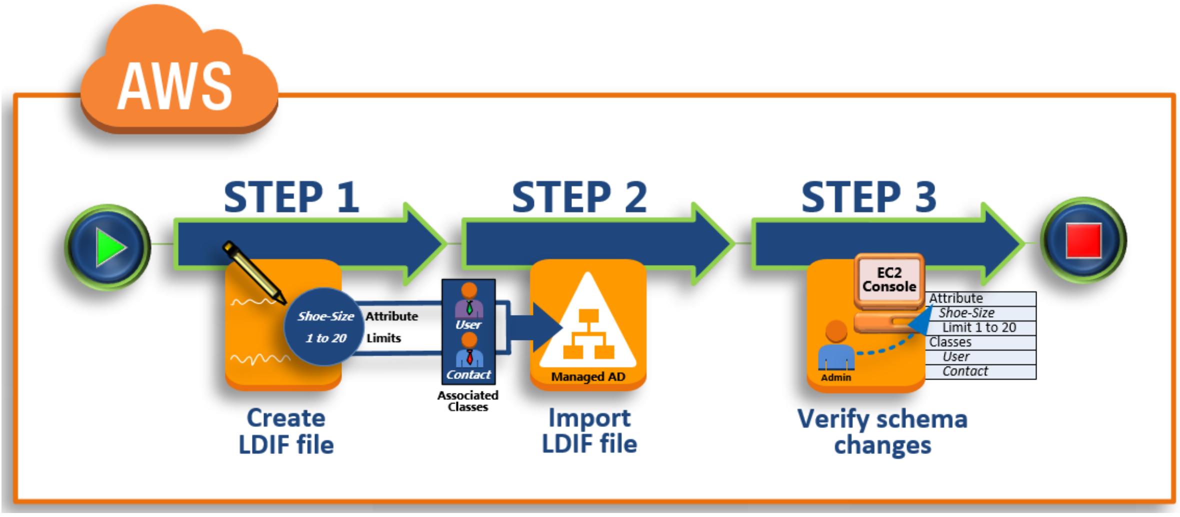
Tutorial: Extending your AWS Managed Microsoft AD schema

In this tutorial, you will learn how to extend the schema for your AWS Directory Service for Microsoft Active Directory directory, also known as AWS Managed Microsoft AD, by adding unique *attributes* and *classes* that meet your specific requirements. AWS Managed Microsoft AD schema extensions can only be uploaded and applied using a valid LDIF (Lightweight Directory Interchange Format) script file.

Attributes (attributeSchema) define the fields in the database while classes (classSchema) define the tables in the database. For example, all of the user objects in Active Directory are defined by the schema class *User* while the individual properties of a user, such as email address or phone number, are each defined by an attribute.

If you wanted to add a new property, such as Shoe-Size, you would define a new attribute, which would be of type *integer*. You could also define lower and upper limits like 1 to 20. Once the Shoe-Size attributeSchema object has been created, you would then alter the *User* classSchema object to contain that attribute. Attributes can be linked to multiple classes. Shoe-Size could also be added to the *Contact* class for example. For more information about Active Directory schemas, see When to extend your AWS Managed Microsoft AD schema (p. 121).

This workflow has three basic steps.



Step 1: Create your LDIF file (p. 122)

First, you create an LDIF file and define the new attributes and any classes that the attributes should be added to. You use this file for the next phase of the workflow.

Step 2: Import your LDIF file (p. 123)

In this step, you use the AWS Directory Service console to import the LDIF file to your Microsoft AD environment.

Step 3: Verify if the schema extension was successful (p. 125)

Finally, as an administrator, you use an EC2 instance to verify that the new extensions appear in the Active Directory Schema Snap-in.

Step 1: Create your LDIF file

An LDIF file is a standard plain text data interchange format for representing LDAP (Lightweight Directory Access Protocol) directory content and update requests. LDIF conveys directory content as a set of records, one record for each object (or entry). It also represents update requests, such as Add, Modify, Delete, and Rename, as a set of records, one record for each update request.

The AWS Directory Service imports your LDIF file with the schema changes by running the `ldifde.exe` application on your AWS Managed Microsoft AD directory. Therefore, you'll find it helpful to understand the LDIF script syntax. For more information, see LDIF Scripts.

Several third-party LDIF tools can extract, clean-up, and update your schema updates. Regardless of which tool you use, it is important to understand that all identifiers used in your LDIF file must be unique.

We highly recommend that you review the following concepts and tips prior to creating your LDIF file.

- **Schema elements** – Learn about schema elements such as attributes, classes, object IDs, and linked attributes. For more information, see Schema elements (p. 19).

- **Sequence of items** – Make sure that the order in which the items in your LDIF file are laid out follow the Directory Information Tree (DIT) from the top down. The general rules for sequencing in an LDIF file include the following:
  - Separate items with a blank line.
  - List child items after their parent items.
  - Ensure that items such as attributes or object classes exist in the schema. If they are not present, you must add them to the schema before they can be used. For example, before you can assign an attribute to a class, the attribute must be created.
- **Format of the DN** – For each new instruction in the LDIF file, define the distinguished name (DN) as the first line of the instruction. The DN identifies an Active Directory object within the Active Directory object's tree and must contain the domain components for your directory. For example, the domain components for the directory in this tutorial are `DC=example,DC=com`.

  The DN also must contain the common name (CN) of the Active Directory object. The first CN entry is the attribute or class name. Next, you must use `CN=Schema,CN=Configuration`. This CN ensures that you are able to extend the Active Directory schema. As mentioned before, you cannot add or modify Active Directory objects' content. The general format for a DN follows.

```
dn: CN=[attribute or class name],CN=Schema,CN=Configuration,DC=[domain_name]
```

  For this tutorial, the DN for the new Shoe-Size attribute would look like:

```
dn: CN=Shoe-Size,CN=Schema,CN=Configuration,DC=example,DC=com
```

- **Warnings** – Review the warnings below before you extend your schema.
  - Before you extend your Active Directory schema, it is important to review Microsoft's warnings on the impact of this operation. For more information, see What You Must Know Before Extending the Schema.
  - You cannot delete a schema attribute or class. Therefore, if you make a mistake and don't want to restore from backup, you can only disable the object. For more information, see Disabling Existing Classes and Attributes.

To learn more about how LDIF files are constructed and see a sample LDIF file that can be used for testing AWS Managed Microsoft AD schema extensions, see the article How to Extend your AWS Managed Microsoft AD Directory Schema on the AWS Security Blog.

Next Step

Step 2: Import your LDIF file (p. 123)

Step 2: Import your LDIF file

You can extend your schema by importing an LDIF file from either the AWS Directory Service console or by using the API. For more information about how to do this with the schema extension APIs, see the

AWS Directory Service API Reference. At this time, AWS does not support external applications, such as Microsoft Exchange, to perform schema updates directly.

> **Important**
> When you make an update to your AWS Managed Microsoft AD directory schema, the operation is not reversible. In other words, once you create a new class or attribute, Active Directory doesn't allow you to remove it. However, you can disable it.
> If you must delete the schema changes, one option is to restore the directory from a previous snapshot. Restoring a snapshot rolls both the schema and the directory data back to a previous point, not just the schema. Note, the maximum supported age of a snapshot is 180 days. For more information, see Useful shelf life of a system-state backup of Active Directory on the Microsoft website.

Before the update process begins, AWS Managed Microsoft AD takes a snapshot to preserve the current state of your directory.

> **Note**
> Schema extensions is a global feature of AWS Managed Microsoft AD. If you are using Multi-Region replication (p. 58), the following procedures must be performed in the Primary Region (p. 59). The changes will be applied across all replicated Regions automatically. For more information, see Global vs Regional features (p. 58).

To import your LDIF file

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. On the **Directories** page, choose your directory ID.
3. On the **Directory details** page, do one of the following:
   - If you have multiple Regions showing under **Multi-Region replication**, select the primary Region, and then choose the **Maintenance** tab. For more information, see Primary vs additional Regions (p. 59).
   - If you do not have any Regions showing under **Multi-Region replication**, choose the **Maintenance** tab.
4. In the **Schema extensions** section, choose **Actions**, and then select **Upload and update schema**.
5. In the dialog box, click **Browse**, select a valid LDIF file, type a description, and then choose **Update Schema**.

   > **Important**
   > Extending the schema is a critical operation. Don't apply any schema update in production environment without first testing it with your application in a development or test environment.

How is the LDIF file applied

After your LDIF file has been uploaded, AWS Managed Microsoft AD takes steps to protect your directory against errors as it applies the changes in the following order.

1. **Validates the LDIF file.** Since LDIF scripts can manipulate any object in the domain, AWS Managed Microsoft AD runs checks right after you upload to help ensure that the import operation will not fail. These include checks to ensure the following:
   - The objects to be updated are only held in the schema container
   - The DC (domain controllers) part matches the name of the domain where the LDIF script is running
2. **Takes a snapshot of your directory.** You can use the snapshot to restore your directory in case you encounter any problems with your application after updating the schema.
3. **Applies the changes to a single DC.** AWS Managed Microsoft AD isolates one of your DCs and applies the updates in the LDIF file to the isolated DC. It then selects one of your DCs to be the schema master, removes that DC from directory replication, and applies your LDIF file using `Ldifde.exe`.

4. **Replication occurs to all DCs.** AWS Managed Microsoft AD adds the isolated DC back in to replication to complete the update. While this is all happening, your directory continues to provide the Active Directory service to your applications without disruption.

Next step

Step 3: Verify if the schema extension was successful (p. 125)

Step 3: Verify if the schema extension was successful

After you have finished the import process, it is important to verify that schema updates were applied to your directory. This is especially critical before you migrate or update any application that relies on the schema update. You can do this using a variety of different LDAP tools or by writing a test tool that issues the appropriate LDAP commands.

This procedure uses the Active Directory Schema Snap-in and/or PowerShell to verify that the schema updates were applied. You must run these tools from a computer that is domain joined to your AWS Managed Microsoft AD. This can be a Windows server running in your on-premises network with access to your virtual private cloud (VPC) or through a virtual private network (VPN) connection. You can also run these tools on an Amazon EC2 Windows instance (see How to launch a new EC2 instance with seamless domain join).

To verify using the Active Directory Schema Snap-in

1. Install the Active Directory Schema Snap-In using the instructions on the TechNet website.
2. Open the Microsoft Management Console (MMC) and expand the **AD Schema** tree for your directory.
3. Navigate through the **Classes** and **Attributes** folders until you find the schema changes that you made earlier.

To verify using PowerShell

1. Open a PowerShell window.
2. Use the `Get-ADObject` cmdlet as shown below to verify the schema change. For example:

```
get-adobject -Identity 'CN=Shoe-
Size,CN=Schema,CN=Configuration,DC=example,DC=com' -Properties *
```

Optional step

(Optional) Add a value to the new attribute (p. 125)

(Optional) Add a value to the new attribute

Use this optional step when you have created a new attribute and want to add a new value to the attribute in your AWS Managed Microsoft AD directory.

To add a value to an attribute

1. Open the Windows PowerShell command line utility and set the new attribute with the following command. In this example, we will add a new EC2InstanceID value to the attribute for a specific computer.

```
PS C:\> set-adcomputer -Identity computer name -add @{example-EC2InstanceID
= 'EC2 instance ID'}
```

2. You can validate if the EC2InstanceID value was added to the computer object by running the following command:

   ```
   PS C:\> get-adcomputer -Identity computer name -Property example-
   EC2InstanceID
   ```

Related resources

The following resource links are located on the Microsoft website and provide related information.

- Extending the Schema (Windows)
- Active Directory Schema (Windows)
- Active Directory Schema
- Windows Administration: Extending the Active Directory Schema
- Restrictions on Schema Extension (Windows)
- Ldifde

Maintain your AWS Managed Microsoft AD directory

This section describes how to maintain common administrative tasks for your AWS Managed Microsoft AD environment.

Topics

Add alternate UPN suffixes

You can simplify the management of Active Directory (AD) login names and improve the user login experience by adding alternate user principal name (UPN) suffixes to your AWS Managed Microsoft AD directory. To do that, you must be logged on with the **Admin** account or with an account that is a member of the **AWS Delegated User Principal Name Suffix Administrators** group. For more information about this group, see What gets created (p. 12).

To add alternate UPN suffixes

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Locate an Amazon EC2 instance that is joined to your AWS Managed Microsoft AD directory. Select the instance and then choose **Connect**.
3. In the **Server Manager** window, choose **Tools**. Then choose **Active Directory Domains and Trusts**.
4. In the left pane, right-click **Active Directory Domains and Trusts** and then choose **Properties** .
5. In the **UPN Suffixes** tab, type an alternative UPN suffix (such as **`sales.example.com`**). Choose **Add** and then choose **Apply**.
6. If you need to add additional alternative UPN suffixes, repeat step 5 until you have the UPN suffixes you require.

Delete your directory

When a Simple AD or AWS Directory Service for Microsoft Active Directory directory is deleted, all of the directory data and snapshots are deleted and cannot be recovered. After the directory is deleted, all instances that are joined to the directory remain intact. You cannot, however, use your directory credentials to log in to these instances. You need to log in to these instances with a user account that is local to the instance.

When an AD Connector directory is deleted, your on-premises directory remains intact. All instances that are joined to the directory also remain intact and remain joined to your on-premises directory. You can still use your directory credentials to log in to these instances.

To delete a directory

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. Ensure that you are not using Multi-Region replication (p. 58) for the directory you intend to delete. You must delete all replicated Regions before you can proceed to delete this directory. For more information, see Delete a replicated Region (p. 62).
3. Ensure that no AWS applications are enabled for the directory.

   a. On the **Directories** page, choose your directory ID.

   b. On the **Directory details** page, select the **Application management** tab. In the **AWS apps & services** section, you see which AWS applications are enabled for your directory.

      - To disable WorkSpaces, you must deregister the service from the directory in the WorkSpaces console. For more information, see Deregistering from a directory in the *Amazon WorkSpaces Administration Guide*.
      - To disable Amazon WorkSpaces Application Manager, you must remove all application assignments in the Amazon WAM console. For more information, see Removing all application assignments in the *Amazon WAM Administration Guide*.
      - To disable Amazon WorkDocs, you must delete the Amazon WorkDocs site in the Amazon WorkDocs console. For more information, see Delete a site in the *Amazon WorkDocs Administration Guide*.
      - To disable Amazon WorkMail, you must remove the Amazon WorkMail organization in the Amazon WorkMail console. For more information, see Remove an organization in the *Amazon WorkMail Administrator Guide*.
      - Disable AWS Management Console access.
      - To disable Amazon Relational Database Service, you must remove the Amazon RDS instance from the domain. For more information, see Managing a DB instance in a domain in the *Amazon RDS User Guide*.
      - To disable Amazon QuickSight, you must unsubscribe from Amazon QuickSight. For more information, see Closing your Amazon QuickSight account in the *Amazon QuickSight User Guide*.
      - To disable Amazon Connect, you must delete the Amazon Connect Instance. For more information, see Deleting an Amazon Connect instance in the *Amazon Connect Administration Guide*.
      - To disable FSx for Windows File Server, you must remove the Amazon FSx file system from the domain. For more information, see Working with Active Directory in FSx for Windows File Server in the *Amazon FSx for Windows File Server User Guide*.

      Note
      If you are using AWS Single Sign-On and have previously connected it to the AWS Managed Microsoft AD directory you plan to delete, you must first change your identity

source before you can delete it. For more information, see Change your identity source in the *AWS SSO User Guide*.

4. In the navigation pane, choose **Directories**.
5. Select only the directory to be deleted and click **Delete**. It takes several minutes for the directory to be deleted. When the directory has been deleted, it is removed from your directory list.

Rename your directory's site name

You can rename your AWS Managed Microsoft AD directory’s default site name so that it matches with your existing Microsoft Active Directory (AD) site names. This makes it faster for AWS Managed Microsoft AD to find and authenticate your existing AD users in your on-premises directory. The result is a better experience when users login to AWS resources such as Amazon EC2 and Amazon RDS for SQL Server instances that you have joined to your AWS Managed Microsoft AD directory.

To do that, you must be logged in with the **Admin** account or with an account that is a member of the **AWS Delegated Sites and Services Administrators** group. For more information about this group, see What gets created (p. 12).

For additional benefits on renaming your site in relation to trusts, see Domain Locator Across a Forest Trust on Microsoft's website.

To rename the AWS Managed Microsoft AD site name

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Locate an Amazon EC2 instance that is joined to your AWS Managed Microsoft AD directory. Select the instance and then choose **Connect**.
3. In the **Server Manager** window, choose **Tools**. Then choose **Active Directory Sites and Services**.
4. In the left pane, expand the **Sites** folder, right-click the site name (default is **Default-Site-Name**), and then choose **Rename**.
5. Type the new site name, and then choose **Enter**.

Snapshot or restore your directory

AWS Directory Service provides the ability to take manual snapshots of data for your AWS Managed Microsoft AD directory. These snapshots can be used to perform a point-in-time restore for your directory. You cannot take snapshots of AD Connector directories.

> **Note**
> Snapshot is a global feature of AWS Managed Microsoft AD. If you are using Multi-Region replication (p. 58), the following procedures must be performed in the Primary Region (p. 59). The changes will be applied across all replicated Regions automatically. For more information, see Global vs Regional features (p. 58).

Topics

Creating a snapshot of your directory

A snapshot can be used to restore your directory to what it was at the point in time that the snapshot was taken. To create a manual snapshot of your directory, perform the following steps.

> **Note**
> You are limited to 5 manual snapshots for each directory. If you have already reached this limit, you must delete one of your existing manual snapshots before you can create another.

To create a manual snapshot

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. On the **Directories** page, choose your directory ID.
3. On the **Directory details** page, choose the **Maintenance** tab.
4. In the **Snapshots** section, choose **Actions**, and then select **Create snapshot**.
5. In the **Create directory snapshot** dialog box, provide a name for the snapshot, if desired. When ready, choose **Create**.

Depending on the size of your directory, it may take several minutes to create the snapshot. When the snapshot is ready, the **Status** value changes to `Completed`.

Restoring your directory from a snapshot

Restoring a directory from a snapshot is equivalent to moving the directory back in time. Directory snapshots are unique to the directory they were created from. A snapshot can only be restored to the directory from which it was created. In addition, the maximum supported age of a manual snapshot is 180 days. For more information, see Useful shelf life of a system-state backup of Active Directory on the Microsoft website.

> **Warning**
> We recommend that you contact the AWS Support Center before any snapshot restore; we may be able to help you avoid the need to do a snapshot restore. Any restore from snapshot can result in data loss as they are a point in time. It is important you understand that all of the DCs and DNS servers associated with the directory will be offline until the restore operation has been completed.

To restore your directory from a snapshot, perform the following steps.

To restore a directory from a snapshot

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. On the **Directories** page, choose your directory ID.
3. On the **Directory details** page, choose the **Maintenance** tab.
4. In the **Snapshots** section, select a snapshot in the list, choose **Actions**, and then select **Restore snapshot**.
5. Review the information in the **Restore directory snapshot** dialog box, and choose **Restore**.

For an AWS Managed Microsoft AD directory, it can take from two to three hours for the directory to be restored. When it has been successfully restored, the **Status** value of the directory changes to `Active`. Any changes made to the directory after the snapshot date are overwritten.

Deleting a snapshot

To delete a snapshot

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. On the **Directories** page, choose your directory ID.
3. On the **Directory details** page, choose the **Maintenance** tab.
4. In the **Snapshots** section, choose **Actions**, and then select **Delete snapshot**.
5. Verify that you want to delete the snapshot, and then choose **Delete**.

View directory information

You can view detailed information about a directory.

To view detailed directory information

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. Click the directory ID link for your directory. Information about the directory is displayed in the **Directory details** page.

For more information about the **Status** field, see Understanding your directory status (p. 51).

Grant users and groups access to AWS resources

AWS Directory Service provides the ability to give your directory users and groups access to AWS services and resources, such as access to the Amazon EC2 console. Similar to granting IAM users access to manage directories as described in Identity-based policies (IAM policies) (p. 281), in order for users in your directory to have access to other AWS resources, such as Amazon EC2 you must assign IAM roles and policies to those users and groups. For more information, see IAM roles in the *IAM User Guide*.

For information about how to grant users access to the AWS Management Console, see Enable access to the AWS Management Console with AD credentials (p. 141).

Topics

Creating a new role

If you need to create a new IAM role for use with AWS Directory Service, you must create it using the IAM console. Once the role has been created, you must then set up a trust relationship with that role before you can see that role in the AWS Directory Service console. For more information, see Editing the trust relationship for an existing role (p. 131).

> **Note**
> The user performing this task must have permission to perform the following IAM actions. For more information, see Identity-based policies (IAM policies) (p. 281).
>
> - iam:PassRole
> - iam:GetRole
> - iam:CreateRole
> - iam:PutRolePolicy

To create a new role in the IAM console

1. In the navigation pane of the IAM console, choose **Roles**. For more information, see Creating a role (AWS Management Console) in the *IAM User Guide*.

2. Choose **Create role**.
3. Under **Choose the service that will use this role**, choose **Directory Service**, and then choose **Next**.
4. Select the check box next to the policy (for example, **AmazonEC2FullAccess**) that you want to apply to your directory users, and then choose **Next**.
5. If necessary, add a tag to the role, and then choose **Next**.
6. Provide a **Role name** and optional **Description**, and then choose **Create role**.

Example: Create a role to enable AWS Management Console access

The following checklist provides an example of the tasks you must complete to create a new role that will give specific directory users access to the Amazon EC2 console.

1. Create a role with the IAM console using the procedure above. When prompted for a policy, choose **AmazonEC2FullAccess**.
2. Use the steps in Editing the trust relationship for an existing role (p. 131) to edit the role you just created, and then add the required trust relationship information to the policy document. This step is necessary for the role to be visible immediately after you enable access to the AWS Management Console in the next step.
3. Follow the steps in Enable access to the AWS Management Console with AD credentials (p. 141) to configure general access to the AWS Management Console.
4. Follow the steps in Assigning users or groups to an existing role (p. 132) to add the users who need full access to EC2 resources to the new role.

Editing the trust relationship for an existing role

You can assign your existing IAM roles to your AWS Directory Service users and groups. To do this, however, the role must have a trust relationship with AWS Directory Service. When you use AWS Directory Service to create a role using the procedure in Creating a new role (p. 130), this trust relationship is automatically set. You only need to establish this trust relationship for IAM roles that are not created by AWS Directory Service.

To establish a trust relationship for an existing role to AWS Directory Service

1. In the navigation pane of the IAM console, choose **Roles**.

   The console displays the roles for your account.
2. Choose the name of the role that you want to modify, and select the **Trust relationships** tab on the details page.
3. Choose **Edit trust relationship**.
4. Under **Policy Document**, paste the following, and then choose **Update Trust Policy**.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ds.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

You can also update this policy document using the IAM CLI. For more information, see update-trust in the *IAM Command Line Reference*.

Assigning users or groups to an existing role

You can assign an existing IAM role to an AWS Directory Service user or group. The role must have a trust relationship with AWS Directory Service. For more information, see Editing the trust relationship for an existing role (p. 131).

To assign users or groups to an existing IAM role

1. In the AWS Directory Service console navigation pane, choose **Directories**.
2. On the **Directories** page, choose your directory ID.
3. On the **Directory details** page, do one of the following:
    - If you have multiple Regions showing under **Multi-Region replication**, select the Region where you want to make your assignments, and then choose the **Application management** tab. For more information, see Primary vs additional Regions (p. 59).
    - If you do not have any Regions showing under **Multi-Region replication**, choose the **Application management** tab.
4. In the **AWS Management Console** section, under **Delegate console access**, choose the IAM role name for the existing IAM role that you want to assign users to. If the role has not yet been created, see Creating a new role (p. 130).
5. On the **Selected role** page, under **Manage users and groups for this role**, choose **Add**.
6. On the **Add users and groups to the role** page, under **Select Active Directory Forest**, choose either the AWS Managed Microsoft AD forest (this forest) or the on-premises forest (trusted forest), whichever contains where the accounts that need access to the AWS Management Console. For more information about how to set up a trusted forest, see Tutorial: Create a trust relationship between your AWS Managed Microsoft AD and your self-managed Active Directory domain (p. 108).
7. Under **Specify which users or groups to add**, select either **Find by user** or **Find by group**, and then type the name of the user or group. In the list of possible matches, choose the user or group that you want to add.
8. Choose **Add** to finish assigning the users and groups to the role.

    Note
    Access for users in nested groups within your directory are not supported. Members of the parent group have console access, but members of child groups do not.

Viewing users and groups assigned to a role

To view the users and groups assigned to a role, perform the following steps.

To view users and group assigned to a role

1. In the AWS Directory Service console navigation pane, choose **Directories**.
2. On the **Directories** page, choose your directory ID.
3. On the **Directory details** page, do one of the following:
    - If you have multiple Regions showing under **Multi-Region replication**, select the Region where you want to view your assignments, and then choose the **Application management** tab. For more information, see Primary vs additional Regions (p. 59).

- If you do not have any Regions showing under **Multi-Region replication**, choose the **Application management** tab.

4. Under the **AWS Management Console** section, choose the role you want to view.
5. On the **Selected role** page, under **Manage users and groups for this role**, you can view the users and groups assigned to the role.

Removing a user or group from a role

To remove a user or group from a role, perform the following steps.

To remove a user or group from a role

1. In the AWS Directory Service console navigation pane, choose **Directories**.
2. On the **Directories** page, choose your directory ID.
3. On the **Directory details** page, do one of the following:
   - If you have multiple Regions showing under **Multi-Region replication**, select the Region where you want to remove your assignments, and then choose the **Application management** tab. For more information, see Primary vs additional Regions (p. 59).
   - If you do not have any Regions showing under **Multi-Region replication**, choose the **Application management** tab.
4. Under the **AWS Management Console** section, choose the role you want to view.
5. On the **Selected role** page, under **Manage users and groups for this role**, select the users or groups to remove the role from and choose **Remove**. The role is removed from the specified users and groups, but the role is not removed from your account.

Using AWS managed policies with AWS Directory Service

AWS Directory Service provides the following AWS managed policies to give your users and groups access to AWS services and resources, such as access to the Amazon EC2 console. You must log in to the AWS Management Console before you can view these policies.

- Read only access
- Power user access
- AWS Directory Service full access
- AWS Directory Service read only access
- Amazon Cloud Directory full access
- Amazon Cloud Directory read only access
- Amazon EC2 full access
- Amazon EC2 read only access
- Amazon VPC full access
- Amazon VPC read only access
- Amazon RDS full access
- Amazon RDS read only access
- Amazon DynamoDB full access
- Amazon DynamoDB read only access
- Amazon S3 full access

- Amazon S3 read only access
- AWS CloudTrail full access
- AWS CloudTrail read only access
- Amazon CloudWatch full access
- Amazon CloudWatch read only access
- Amazon CloudWatch Logs full access
- Amazon CloudWatch Logs read only access

For more information on how to create your own policies, see Example policies for administering AWS resources in the *IAM User Guide*.

Enable access to AWS applications and services

AWS Directory Service can give other AWS applications and services, such as Amazon WorkSpaces, access to your directory users. The following AWS applications and services can be enabled or disabled to work with AWS Directory Service.

AWS application / service	More information...
Amazon Chime	For more information, see the Amazon Chime Administration Guide.
Amazon Connect	For more information, see the Amazon Connect Administration Guide.
FSx for Windows File Server	For more information, see Using Amazon FSx with AWS Directory Service for Microsoft Active Directory in the *FSx for Windows File Server User Guide*.
Amazon QuickSight	For more information, see the Amazon QuickSight User Guide.
Amazon Relational Database Service	For more information, see the Amazon RDS User Guide.
AWS Transfer Family	For more information, see the AWS Transfer Family User Guide.
Amazon WorkDocs	For more information, see the Amazon WorkDocs Administration Guide.
Amazon WorkMail	For more information, see the Amazon WorkMail Administrator Guide.
WorkSpaces	You can create a Simple AD, AWS Managed Microsoft AD, or AD Connector directly from WorkSpaces. Simply launch **Advanced Setup** when creating your Workspace. For more information, see the Amazon WorkSpaces Administration Guide.
Amazon WorkSpaces Application Manager	For more information, see the Amazon WAM Administration Guide.

AWS application / service	More information...
AWS Management Console	For more information, see Enable access to the AWS Management Console with AD credentials (p. 141).

Once enabled, you manage access to your directories in the console of the application or service that you want to give access to your directory. To find the AWS applications and services links described above in the AWS Directory Service console, perform the following steps.

To display the applications and services for a directory

1. In the AWS Directory Service console navigation pane, choose **Directories**.
2. On the **Directories** page, choose your directory ID.
3. On the **Directory details** page, select the **Application management** tab.
4. Review the list under the **AWS apps & services** section.

Topics

Creating an access URL

An access URL is used with AWS applications and services, such as Amazon WorkDocs, to reach a login page that is associated with your directory. The URL must be unique globally. You can create an access URL for your directory by performing the following steps.

> **Warning**
> Once you create an application access URL for this directory, it cannot be changed. After an access URL is created, it cannot be used by others. If you delete your directory, the access URL is also deleted and can then be used by any other account.

> **Note**
> Access URL is a Regional feature of AWS Managed Microsoft AD. If you are using Multi-Region replication (p. 58), the following procedures must be applied separately in each Region. For more information, see Global vs Regional features (p. 58).

To create an access URL

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. On the **Directories** page, choose your directory ID.
3. On the **Directory details** page, do one of the following:
   - If you have multiple Regions showing under **Multi-Region replication**, select the Region where you want to create your access URL, and then choose the **Application management** tab. For more information, see Primary vs additional Regions (p. 59).
   - If you do not have any regions showing under **Multi-Region replication**, choose the **Application management** tab.
4. In the **Application access URL** section, if an access URL has not been assigned to the directory, the **Create** button is displayed. Enter a directory alias and choose **Create**. If an **Entity Already Exists** error is returned, the specified directory alias has already been allocated. Choose another alias and repeat this procedure.

Your access URL is displayed in the format *<alias>*.awsapps.com. By default, this URL will take you to the sign-in page for Amazon WorkDocs.

Single sign-on

AWS Directory Service provides the ability to allow your users to access Amazon WorkDocs from a computer joined to the directory without having to enter their credentials separately.

Before you enable single sign-on, you need to take additional steps to enable your users web browsers to support single sign-on. Users may need to modify their web browser settings to enable single sign-on.

> **Note**
> Single sign-on only works when used on a computer that is joined to the AWS Directory Service directory. It cannot be used on computers that are not joined to the directory.

If your directory is an AD Connector directory and the AD Connector service account does not have the permission to add or remove its service principal name attribute, then for Steps 5 and 6 below, you have two options:

1. You can proceed and will be prompted for the user name and password for a directory user that has this permission to add or remove the service principal name attribute on the AD Connector service account. These credentials are only used to enable single sign-on and are not stored by the service. The AD Connector service account permissions are not changed.
2. You can delegate permissions to allow the AD Connector service account to add or remove the service principal name attribute on itself, you can run the below PowerShell commands from a domain joined computer using an account that has permissions to modify the permissions on the AD Connector service account. The below command will give the AD Connector service account the ability to add and remove a service principal name attribute only for itself.

```
$AccountName = 'ConnectorAccountName'
# DO NOT modify anything below this comment.
# Getting Active Directory information.
Import-Module 'ActiveDirectory'
$RootDse = Get-ADRootDSE
[System.GUID]$ServicePrincipalNameGuid = (Get-ADObject -SearchBase
 $RootDse.SchemaNamingContext -Filter { lDAPDisplayName -eq 'servicePrincipalName' } -
Properties 'schemaIDGUID').schemaIDGUID
# Getting AD Connector service account Information.
$AccountProperties = Get-ADUser -Identity $AccountName
$AclPath = $AccountProperties.DistinguishedName
$AccountSid = New-Object -TypeName 'System.Security.Principal.SecurityIdentifier'
 $AccountProperties.SID.Value
# Getting ACL settings for AD Connector service account.
$ObjectAcl = Get-ACL -Path "AD:\$AclPath"
# Setting ACL allowing the AD Connector service account the ability to add and remove a
 Service Principal Name (SPN) to itself
$AddAccessRule = New-Object -TypeName 'System.DirectoryServices.ActiveDirectoryAccessRule'
 $AccountSid, 'WriteProperty', 'Allow', $ServicePrincipalNameGUID, 'None'
$ObjectAcl.AddAccessRule($AddAccessRule)
Set-ACL -AclObject $ObjectAcl -Path "AD:\$AclPath"
```

To enable or disable single sign-on with Amazon WorkDocs

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. On the **Directories** page, choose your directory ID.
3. On the **Directory details** page, select the **Application management** tab.

4. In the **Application access URL** section, choose **Enable** to enable single sign-on for Amazon WorkDocs.

   If you do not see the **Enable** button, you may need to first create an Access URL before this option will be displayed. For more information about how to create an access URL, see Creating an access URL (p. 135).
5. In the **Enable Single Sign-On for this directory** dialog box, choose **Enable**. Single sign-on is enabled for the directory.
6. If you later want to disable single sign-on with Amazon WorkDocs, choose **Disable**, and then in the **Disable Single Sign-On for this directory** dialog box, choose **Disable** again.

Topics

Single sign-on for IE and Chrome

To allow Microsoft Internet Explorer (IE) and Google Chrome browsers to support single sign-on, the following tasks must be performed on the client computer:

- Add your access URL (e.g., https://*<alias>*.awsapps.com) to the list of approved sites for single sign-on.
- Enable active scripting (JavaScript).
- Allow automatic logon.
- Enable integrated authentication.

You or your users can perform these tasks manually, or you can change these settings using Group Policy settings.

Topics

Manual update for single sign-on on Windows

To manually enable single sign-on on a Windows computer, perform the following steps on the client computer. Some of these settings may already be set correctly.

To manually enable single sign-on for Internet Explorer and Chrome on Windows

1. To open the **Internet Properties** dialog box, choose the **Start** menu, type `Internet Options` in the search box, and choose **Internet Options**.
2. Add your access URL to the list of approved sites for single sign-on by performing the following steps:

   a. In the **Internet Properties** dialog box, select the **Security** tab.
   b. Select **Local intranet** and choose **Sites**.
   c. In the **Local intranet** dialog box, choose **Advanced**.
   d. Add your access URL to the list of websites and choose **Close**.
   e. In the **Local intranet** dialog box, choose **OK**.

3. To enable active scripting, perform the following steps:

   a. In the **Security** tab of the **Internet Properties** dialog box, choose **Custom level**.
   b. In the **Security Settings - Local Intranet Zone** dialog box, scroll down to **Scripting** and select **Enable** under **Active scripting**.
   c. In the **Security Settings - Local Intranet Zone** dialog box, choose **OK**.
4. To enable automatic logon, perform the following steps:

   a. In the **Security** tab of the **Internet Properties** dialog box, choose **Custom level**.
   b. In the **Security Settings - Local Intranet Zone** dialog box, scroll down to **User Authentication** and select **Automatic logon only in Intranet zone** under **Logon**.
   c. In the **Security Settings - Local Intranet Zone** dialog box, choose **OK**.
   d. In the **Security Settings - Local Intranet Zone** dialog box, choose **OK**.
5. To enable integrated authentication, perform the following steps:

   a. In the **Internet Properties** dialog box, select the **Advanced** tab.
   b. Scroll down to **Security** and select **Enable Integrated Windows Authentication**.
   c. In the **Internet Properties** dialog box, choose **OK**.
6. Close and re-open your browser to have these changes take effect.

Manual update for single sign-on on OS X

To manually enable single sign-on for Chrome on OS X, perform the following steps on the client computer. You will need administrator rights on your computer to complete these steps.

To manually enable single sign-on for Chrome on OS X

1. Add your access URL to the AuthServerWhitelist policy by running the following command:

   ```
   defaults write com.google.Chrome AuthServerWhitelist "https://<alias>.awsapps.com"
   ```
2. Open **System Preferences**, go to the **Profiles** panel, and delete the `Chrome Kerberos Configuration` profile.
3. Restart Chrome and open chrome://policy in Chrome to confirm that the new settings are in place.

Group policy settings for single sign-on

The domain administrator can implement Group Policy settings to make the single sign-on changes on client computers that are joined to the domain.

> **Note**
> If you manage the Chrome web browsers on the computers in your domain with Chrome policies, you must add your access URL to the AuthServerWhitelist policy. For more information about setting Chrome policies, go to Policy Settings in Chrome.

To enable single sign-on for Internet Explorer and Chrome using Group Policy settings

1. Create a new Group Policy object by performing the following steps:

   a. Open the Group Policy Management tool, navigate to your domain and select **Group Policy Objects**.
   b. From the main menu, choose **Action** and select **New**.

   c. In the **New GPO** dialog box, enter a descriptive name for the Group Policy object, such as `SSO Policy`, and leave **Source Starter GPO** set to **(none)**. Click **OK**.

2. Add the access URL to the list of approved sites for single sign-on by performing the following steps:

   a. In the Group Policy Management tool, navigate to your domain, select **Group Policy Objects**, open the context (right-click) menu for your SSO policy, and choose **Edit**.

   b. In the policy tree, navigate to **User Configuration** > **Preferences** > **Windows Settings**.

   c. In the **Windows Settings** list, open the context (right-click) menu for **Registry** and choose **New registry item**.

   d. In the **New Registry Properties** dialog box, enter the following settings and choose **OK**:

      Action

      `Update`

      Hive

      `HKEY_CURRENT_USER`

      Path

      `Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\Domains\awsapps.com\<alias>`

      The value for *`<alias>`* is derived from your access URL. If your access URL is `https://examplecorp.awsapps.com`, the alias is `examplecorp`, and the registry key will be `Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\Domains\awsapps.com\examplecorp`.

      Value name

      `https`

      Value type

      `REG_DWORD`

      Value data

      `1`

3. To enable active scripting, perform the following steps:

   a. In the Group Policy Management tool, navigate to your domain, select **Group Policy Objects**, open the context (right-click) menu for your SSO policy, and choose **Edit**.

   b. In the policy tree, navigate to **Computer Configuration** > **Policies** > **Administrative Templates** > **Windows Components** > **Internet Explorer** > **Internet Control Panel** > **Security Page** > **Intranet Zone**.

   c. In the **Intranet Zone** list, open the context (right-click) menu for **Allow active scripting** and choose **Edit**.

   d. In the **Allow active scripting** dialog box, enter the following settings and choose **OK**:

      - Select the **Enabled** radio button.
      - Under **Options** set **Allow active scripting** to **Enable**.

4. To enable automatic logon, perform the following steps:

   a. In the Group Policy Management tool, navigate to your domain, select Group Policy Objects, open the context (right-click) menu for your SSO policy, and choose **Edit**.

   b. In the policy tree, navigate to **Computer Configuration** > **Policies** > **Administrative Templates** > **Windows Components** > **Internet Explorer** > **Internet Control Panel** > **Security Page** > **Intranet Zone**.

   c. In the **Intranet Zone** list, open the context (right-click) menu for **Logon options** and choose **Edit**.

   d. In the **Logon options** dialog box, enter the following settings and choose **OK**:

      - Select the **Enabled** radio button.
      - Under **Options** set **Logon options** to **Automatic logon only in Intranet zone**.

5. To enable integrated authentication, perform the following steps:

   a. In the Group Policy Management tool, navigate to your domain, select **Group Policy Objects**, open the context (right-click) menu for your SSO policy, and choose **Edit**.

   b. In the policy tree, navigate to **User Configuration** > **Preferences** > **Windows Settings**.

   c. In the **Windows Settings** list, open the context (right-click) menu for **Registry** and choose **New registry item**.

   d. In the **New Registry Properties** dialog box, enter the following settings and choose **OK**:

      Action

      ```
      Update
      ```

      Hive

      ```
      HKEY_CURRENT_USER
      ```

      Path

      ```
      Software\Microsoft\Windows\CurrentVersion\Internet Settings
      ```

      Value name

      ```
      EnableNegotiate
      ```

      Value type

      ```
      REG_DWORD
      ```

      Value data

      ```
      1
      ```

6. Close the **Group Policy Management Editor** window if it is still open.

7. Assign the new policy to your domain by following these steps:

   a. In the Group Policy Management tree, open the context (right-click) menu for your domain and choose **Link an Existing GPO**.

   b. In the **Group Policy Objects** list, select your SSO policy and choose **OK**.

These changes will take effect after the next Group Policy update on the client, or the next time the user logs in.

Single sign-on for Firefox

To allow Mozilla Firefox browser to support single sign-on, add your access URL (e.g., https://*<alias>*.awsapps.com) to the list of approved sites for single sign-on. This can be done manually, or automated with a script.

Topics

Manual update for single sign-on

To manually add your access URL to the list of approved sites in Firefox, perform the following steps on the client computer.

To manually add your access URL to the list of approved sites in Firefox

1. Open Firefox and open the `about:config` page.
2. Open the `network.negotiate-auth.trusted-uris` preference and add your access URL to the list of sites. Use a comma (,) to separate multiple entries.

Automatic update for single sign-on

As a domain administrator, you can use a script to add your access URL to the Firefox `network.negotiate-auth.trusted-uris` user preference on all computers on your network. For more information, go to https://support.mozilla.org/en-US/questions/939037.

Enable access to the AWS Management Console with AD credentials

AWS Directory Service allows you to grant members of your directory access to the AWS Management Console. By default, your directory members do not have access to any AWS resources. You assign IAM roles to your directory members to give them access to the various AWS services and resources. The IAM role defines the services, resources, and level of access that your directory members have.

Before you can grant console access to your directory members, your directory must have an access URL. For more information about how to view directory details and get your access URL, see View directory information (p. 130). For more information about how to create an access URL, see Creating an access URL (p. 135).

For more information about how to create and assign IAM roles to your directory members, see Grant users and groups access to AWS resources (p. 130).

Topics

Related AWS Security Blog Article

- How to Access the AWS Management Console Using AWS Managed Microsoft AD and Your On-Premises Credentials

Note

Access to the AWS Management Console is a Regional feature of AWS Managed Microsoft AD. If you are using Multi-Region replication (p. 58), the following procedures must be applied separately in each Region. For more information, see Global vs Regional features (p. 58).

Enable AWS Management Console access

By default, console access is not enabled for any directory. To enable console access for your directory users and groups, perform the following steps:

> **Note**
> You cannot currently use AWS SSO to access the AWS Management Console in opt-in regions. For a list of these regions, see Regions and Zones .

To enable console access

1. In the AWS Directory Service console navigation pane, choose **Directories**.
2. On the **Directories** page, choose your directory ID.
3. On the **Directory details** page, do one of the following:
   - If you have multiple Regions showing under **Multi-Region replication**, select the Region where you want to enable access to the AWS Management Console, and then choose the **Application management** tab. For more information, see Primary vs additional Regions (p. 59).
   - If you do not have any Regions showing under **Multi-Region replication**, choose the **Application management** tab.
4. Under the **AWS Management Console** section, choose **Enable**. Console access is now enabled for your directory.

   Before users can sign-in to the console with your access URL, you must first add your users to the role. For general information about assigning users to IAM roles, see Assigning users or groups to an existing role (p. 132). After the IAM roles have been assigned, users can then access the console using your access URL. For example, if your directory access URL is example-corp.awsapps.com, the URL to access the console is https://example-corp.awsapps.com/console/.

Disable AWS Management Console access

To disable console access for your directory users and groups, perform the following steps:

To disable console access

1. In the AWS Directory Service console navigation pane, choose **Directories**.
2. On the **Directories** page, choose your directory ID.
3. On the **Directory details** page, do one of the following:
   - If you have multiple Regions showing under **Multi-Region replication**, select the Region where you want to disable access to the AWS Management Console, and then choose the **Application management** tab. For more information, see Primary vs additional Regions (p. 59).
   - If you do not have any Regions showing under **Multi-Region replication**, choose the **Application management** tab.
4. Under the **AWS Management Console** section, choose **Disable**. Console access is now disabled for your directory.
5. If any IAM roles have been assigned to users or groups in the directory, the **Disable** button may be unavailable. In this case, you must remove all IAM role assignments for the directory before proceeding, including assignments for users or groups in your directory that have been deleted, which will show as **Deleted User** or **Deleted Group**.

   After all IAM role assignments have been removed, repeat the steps above.

Set login session length

By default, users have 1 hour to use their session after successfully signing in to the console before they are logged out. After that, users must sign in again to start the next 1 hour session before being logged off again. You can use the following procedure to change the length of time to up to 12 hours per session.

To set login session length

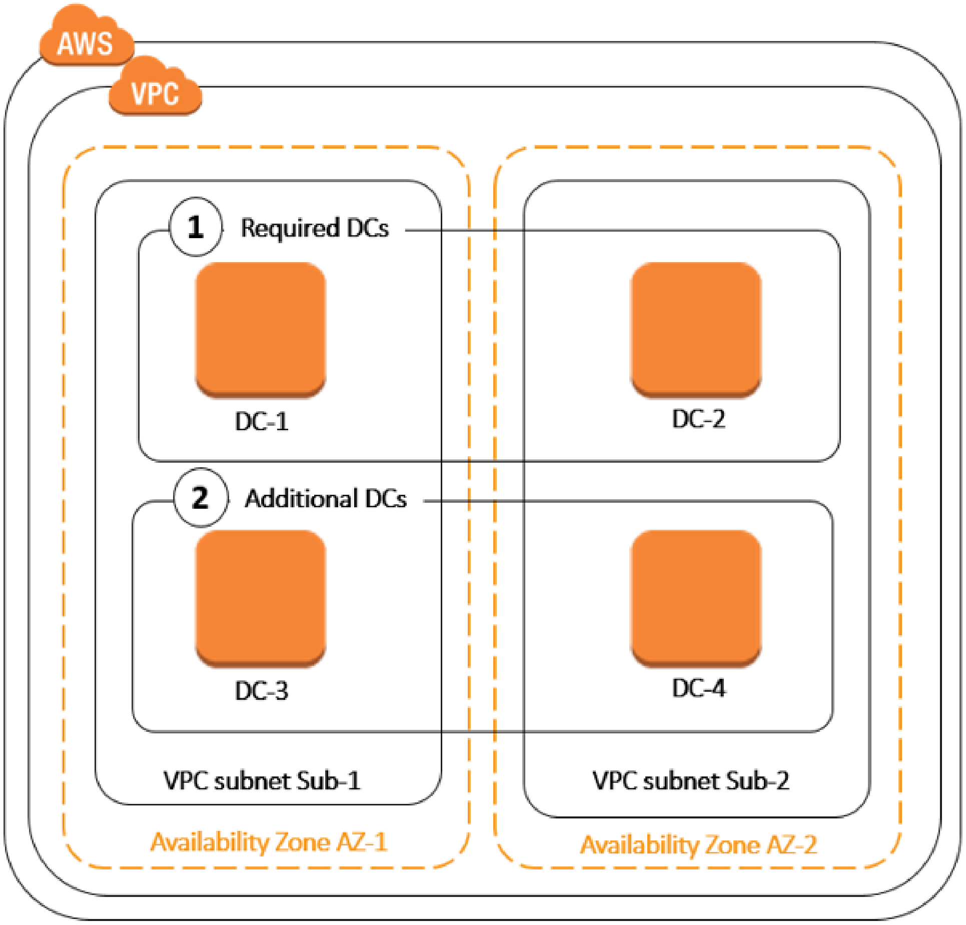
1. In the AWS Directory Service console navigation pane, choose **Directories**.
2. On the **Directories** page, choose your directory ID.
3. On the **Directory details** page, do one of the following:
   - If you have multiple Regions showing under **Multi-Region replication**, select the Region where you want to set the login session length, and then choose the **Application management** tab. For more information, see Primary vs additional Regions (p. 59).
   - If you do not have any Regions showing under **Multi-Region replication**, choose the **Application management** tab.
4. Under the **AWS apps & services** section, choose **AWS Management Console**.
5. In the **Manage Access to AWS Resource** dialog box, choose **Continue**.
6. In the **Assign users and groups to IAM roles** page, under **Set login session length**, edit the numbered value, and then choose **Save**.

Deploy additional domain controllers

Deploying additional domain controllers increases the redundancy, which results in even greater resilience and higher availability. This also improves the performance of your directory by supporting a greater number of Active Directory requests. For example, you can now use AWS Managed Microsoft AD to support multiple .NET applications that are deployed on large fleets of Amazon EC2 and Amazon RDS for SQL Server instances.

When you first create your directory, AWS Managed Microsoft AD deploys two domain controllers across multiple Availability Zones, which is required for highly availability purposes. Later, you can easily deploy additional domain controllers via the AWS Directory Service console by just specifying the total number of domain controllers that you want. AWS Managed Microsoft AD distributes the additional domain controllers to the Availability Zones and VPC subnets on which your directory is running.

For example, in the below illustration, DC-1 and DC-2 represent the two domain controllers that were originally created with your directory. The AWS Directory Service console refers to these default domain controllers as **Required**. AWS Managed Microsoft AD intentionally locates each of these domain controllers in separate Availability Zones during the directory creation process. Later, you might decide to add two more domain controllers to help distribute the authentication load over peak login times. Both DC-3 and DC-4 represent the new domain controllers, which the console now refers to as **Additional**. As before, AWS Managed Microsoft AD again automatically places the new domain controllers in different Availability Zones to ensure your domain's high availability.



This process eliminates the need for you to manually configure directory data replication, automated daily snapshots, or monitoring for the additional domain controllers. It's also easier for you to migrate and run mission critical Active Directory–integrated workloads in the AWS Cloud without having to deploy and maintain your own Active Directory infrastructure. You can also deploy or remove additional domain controllers for AWS Managed Microsoft AD using the UpdateNumberOfDomainControllers API.

> **Note**
> Additional domain controllers is a Regional feature of AWS Managed Microsoft AD. If you are using Multi-Region replication (p. 58), the following procedures must be applied separately in each Region. For more information, see Global vs Regional features (p. 58).

Add or remove additional domain controllers

Use the following procedure to deploy or remove additional domain controllers in your AWS Managed Microsoft AD directory.

> **Note**
> If you have configured your AWS Managed Microsoft AD to enable LDAPS, any additional domain controllers you add will also have LDAPS enabled automatically. For more information, see Enable secure LDAP (LDAPS) (p. 34).

To add or remove additional domain controllers

1. In the AWS Directory Service console navigation pane, choose **Directories**.
2. On the **Directories** page, choose your directory ID.
3. On the **Directory details** page, do one of the following:

   - If you have multiple Regions showing under **Multi-Region replication**, select the Region where you want to add or remove domain controllers, and then choose the **Scale & share** tab. For more information, see Primary vs additional Regions (p. 59).
   - If you do not have any Regions showing under **Multi-Region replication**, choose the **Scale & share** tab.
4. In the **Domain controllers** section, choose **Edit**.
5. Specify the number of domain controllers to add or remove from your directory, and then choose **Modify**.
6. When AWS Managed Microsoft AD completes the deployment process, all domain controllers show **Active** status, and both the assigned Availability Zone and VPC subnets appear. New domain controllers are equally distributed across the Availability Zones and subnets where your directory is already deployed.

   Note
   After deploying additional domain controllers, you can reduce the number of domain controllers to two, which is the minimum required for fault-tolerance and high availability purposes.

Related AWS Security Blog Article

- How to increase the redundancy and performance of your AWS Directory Service for AWS Managed Microsoft AD by adding domain controllers

Use Amazon CloudWatch metrics to determine when to add domain controllers

Load balancing across all of your domain controllers is important for the resilience and performance of your directory. To help you optimize the performance of your domain controllers in AWS Managed Microsoft AD, we recommend that you first monitor important metrics in CloudWatch to form a baseline. During this process, you analyze your directory over time to identify your average and peak directory utilization. After determining your baseline, you can monitor these metrics on a regular basis to help determine when to add a domain controller to your directory. For more information, see Monitor your domain controllers with performance metrics (p. 57).

The following metrics are important to monitor on a regular basis. For a full list of available domain controller metrics in CloudWatch, see AWS Managed Microsoft AD performance counters (p. 146).

- Domain controller-specific metrics, such as:
  - Processor
  - Memory
  - Logical Disk
  - Network Interface
- AWS Managed Microsoft AD directory-specific metrics, such as:
  - LDAP searches
  - Binds
  - DNS queries
  - Directory reads
  - Directory writes

For instructions on how to set up domain controller metrics using the CloudWatch console, see How to automate AWS Managed Microsoft AD scaling based on utilization metrics in the AWS Security Blog. For general information about metrics in CloudWatch, see Using Amazon CloudWatch metrics in the *Amazon CloudWatch User Guide*.

For general information about domain controller planning, see Capacity planning for Active Directory Domain Services on the Microsoft website.

AWS Managed Microsoft AD performance counters

The following table lists all performance counters available in Amazon CloudWatch for tracking domain controller and directory performance in AWS Managed Microsoft AD.

Metric category	Metric name
Database ==> Instances (NTDSA)	Database Cache % Hit
	I/O Database Reads Average Latency
	I/O Database Reads/sec
	I/O Log Writes Average Latency
DirectoryServices (NTDS)	LDAP Bind Time
	DRA Pending Replication Operations
	DRA Pending Replication Synchronizations
DNS	Recursive Queries/sec
	Recursive Query Failure/sec
	TCP Query Received/sec
	Total Query Received/sec
	Total Response Sent/sec
	UDP Query Received/sec
LogicalDisk	Avg. Disk Queue Length
	% Free Space
Memory	% Committed Bytes in Use
	Long-Term Average Standby Cache Lifetime (s)
Network Interface	Bytes Sent/sec
	Bytes Received/sec
	Current Bandwidth
NTDS	ATQ Estimated Queue Delay
	ATQ Request Latency
	DS Directory Reads/Sec
	DS Directory Searches/Sec
	DS Directory Writes/Sec

Metric category	Metric name
	LDAP Client Sessions
	LDAP Searches/sec
	LDAP Successful Binds/sec
Processor	% Processor Time
Security System-Wide Statistics	Kerberos Authentications
	NTLM Authentications

Migrate users from Active Directory to AWS Managed Microsoft AD

You can use the Active Directory Migration Toolkit (ADMT) along with the Password Export Service (PES) to migrate users from your self-managed AD to your AWS Managed Microsoft AD directory. This enables you to migrate AD objects and encrypted passwords for your users more easily.

For detailed instructions, see How to migrate your on-premises domain to AWS Managed Microsoft AD using ADMT on the *AWS Security Blog*.

Best practices for AWS Managed Microsoft AD

Here are some suggestions and guidelines you should consider to avoid problems and get the most out of AWS Managed Microsoft AD.

Setting up: Prerequisites

Consider these guidelines before creating your directory.

Verify you have the right directory type

AWS Directory Service provides multiple ways to use Microsoft Active Directory with other AWS services. You can choose the directory service with the features you need at a cost that fits your budget:

- **AWS Directory Service for Microsoft Active Directory** is a feature-rich managed Microsoft Active Directory hosted on the AWS cloud. AWS Managed Microsoft AD is your best choice if you have more than 5,000 users and need a trust relationship set up between an AWS hosted directory and your on-premises directories.
- **AD Connector** simply connects your existing on-premises Active Directory to AWS. AD Connector is your best choice when you want to use your existing on-premises directory with AWS services.
- **Simple AD** is an inexpensive Active Directory–compatible service with the common directory features. In most cases, Simple AD is the least expensive option and your best choice if you have 5,000 or fewer users and don’t need the more advanced Microsoft Active Directory features.

For a more detailed comparison of AWS Directory Service options, see Which to choose (p. 1).

Ensure your VPCs and instances are configured correctly

In order to connect to, manage, and use your directories, you must properly configure the VPCs that the directories are associated with. See either AWS Managed Microsoft AD prerequisites (p. 9), AD Connector prerequisites (p. 187), or Simple AD prerequisites (p. 227) for information about the VPC security and networking requirements.

If you are adding an instance to your domain, ensure that you have connectivity and remote access to your instance as described in Join an EC2 instance to your AWS Managed Microsoft AD directory (p. 71).

Be aware of your limits

Learn about the various limits for your specific directory type. The available storage and the aggregate size of your objects are the only limitations on the number of objects you may store in your directory. See either AWS Managed Microsoft AD quotas (p. 152), AD Connector quotas (p. 222), or Simple AD quotas (p. 271) for details about your chosen directory.

Understand your directory’s AWS security group configuration and use

AWS creates a security group and attaches it to your directory’s domain controller elastic network interfaces. This security group blocks unnecessary traffic to the domain controller and allows traffic that is necessary for Active Directory communications. AWS configures the security group to open only the ports that are required for Active Directory communications. In the default configuration, the security group accepts traffic to these ports from any IP address. AWS attaches the security group to your domain controllers’ interfaces that are accessible from within your peered or resized VPCs. These interfaces are inaccessible from the internet even if you modify routing tables, change the network connections to your VPC, and configure the NAT Gateway service. As such, only instances and computers that have a network path into the VPC can access the directory. This simplifies setup by eliminating the requirement for you to configure specific address ranges. Instead, you configure routes and security groups into the VPC that permit traffic only from trusted instances and computers.

Modifying the directory security group

If you want to increase the security of your directories’ security groups, you can modify them to accept traffic from a more restrictive list of IP addresses. For example, you could change the accepted addresses from 0.0.0.0/0 to a CIDR range that is specific to a single subnet or computer. Similarly, you might choose to restrict the destination addresses to which your domain controllers can communicate. Make such changes only if you fully understand how security group filtering works. For more information, see Amazon EC2 security groups for Linux instances in the *Amazon EC2 User Guide*. Improper changes can result in loss of communications to intended computers and instances. AWS recommends that you do not attempt to open additional ports to the domain controller as this decreases the security of your directory. Please carefully review the AWS Shared Responsibility Model.

> **Warning**
> It is technically possible for you to associate the security groups, which your directory uses, with other EC2 instances that you create. However, AWS recommends against this practice. AWS may have reasons to modify the security group without notice to address functional or security needs of the managed directory. Such changes affect any instances with which you associate the directory security group. Furthermore, associating the directory security group with your EC2 instances creates a potential security risk for your EC2 instances. The directory security group accepts traffic on required Active Directory ports from any IP address. If you associate this Security Group with an EC2 instance that has a public IP address attached to the internet, then any computer on the internet can communicate with your EC2 instance on the opened ports.

Setting up: Creating your directory

Here are some suggestions to consider as you create your directory.

Remember your administrator ID and password

When you set up your directory, you provide a password for the administrator account. That account ID is *Admin* for AWS Managed Microsoft AD. Remember the password that you create for this account; otherwise you will not be able to add objects to your directory.

Create a DHCP options set

We recommend that you create a DHCP options set for your AWS Directory Service directory and assign the DHCP options set to the VPC that your directory is in. That way any instances in that VPC can point to the specified domain, and DNS servers can resolve their domain names.

For more information about DHCP options sets, see Create a DHCP options set (p. 99).

Deploy additional domain controllers

By default, AWS creates two domain controllers that exist in separate Availability Zones. This provides fault resiliency during software patching and other events that may make one domain controller unreachable or unavailable. We recommend that you deploy additional domain controllers to further increase resiliency and ensure scale-out performance in the event of a longer term event that affects access to a domain controller or an Availability Zone.

For more information, see Use the Windows DC locator service (p. 152).

Understand username restrictions for AWS applications

AWS Directory Service provides support for most character formats that can be used in the construction of usernames. However, there are character restrictions that are enforced on usernames that will be used for signing in to AWS applications, such as WorkSpaces, Amazon WorkDocs, Amazon WorkMail, or Amazon QuickSight. These restrictions require that the following characters not be used:

- Spaces
- Multibyte characters
- !"#$%&'()*+,/:;<=>?@[\]^`{|}~

> **Note**
> The @ symbol is allowed as long as it precedes a UPN suffix.

Using your directory

Here are some suggestions to keep in mind when using your directory.

Do not alter predefined users, groups and organizational units

When you use AWS Directory Service to launch a directory, AWS creates an organizational unit (OU) that contains all your directory's objects. This OU, which has the NetBIOS name that you typed when you created your directory, is located in the domain root. The domain root is owned and managed by AWS. Several groups and an administrative user are also created.

Do not move, delete or in any other way alter these predefined objects. Doing so can make your directory inaccessible by both yourself and AWS. For more information, see What gets created (p. 12).

Automatically join domains

When launching a Windows instance that is to be part of an AWS Directory Service domain, it is often easiest to join the domain as part of the instance creation process rather than manually adding the instance later. To automatically join a domain, simply select the correct directory for **Domain join directory** when launching a new instance. You can find details in Seamlessly join a Windows EC2 instance (p. 72).

Set up trusts correctly

When setting up trust relationship between your AWS Managed Microsoft AD directory and another directory, keep in mind these guidelines:

- The trust type must match on both sides (Forest or External)
- Ensure the trust direction is setup correctly if using a one-way trust (Outgoing on trusting domain, Incoming on trusted domain)
- Both fully qualified domain names (FQDNs) and NetBIOS names must be unique between forests / domains

For more details and specific instructions on setting up a trust relationship, see Creating a trust relationship (p. 104).

Managing your directory

Consider these suggestions for managing your directory.

Track your domain controller performance

To help optimize scaling decisions and improve directory resilience and performance, we recommend that you use CloudWatch metrics. For more information, see Monitor your domain controllers with performance metrics (p. 57).

For instructions on how to set up domain controller metrics using the CloudWatch console, see How to automate AWS Managed Microsoft AD scaling based on utilization metrics in the AWS Security Blog.

Carefully plan for schema extensions

Thoughtfully apply schema extensions to index your directory for important and frequent queries. Use care to not over-index the directory as indexes consume directory space and rapidly changing indexed values can result in performance problems. To add indexes, you must create a Lightweight Directory Access Protocol (LDAP) Directory Interchange Format (LDIF) file and extend your schema change. For more information, see Extend your schema (p. 121).

About load balancers

Do not use a load balancer in front of the AWS Managed Microsoft AD end-points. Microsoft designed Active Directory (AD) for use with a domain controller (DC) discovery algorithm that finds the most responsive operational DC without external load balancing. External network load balancers inaccurately detect active DCs and can result in your application being sent to a DC that is coming up but not ready for use. For more information, see Load balancers and Active Directory on Microsoft TechNet which recommends fixing applications to use AD correctly rather than implementing external load balancers.

Make a backup of your instance

If you decide to manually add an instance to an existing AWS Directory Service domain, make a backup or take a snapshot of that instance first. This is particularly important when joining a Linux instance. Some of the procedures used to add an instance, if not performed correctly, can render your instance unreachable or unusable. For more information, see Snapshot or restore your directory (p. 128).

Set up SNS messaging

With Amazon Simple Notification Service (Amazon SNS), you can receive email or text (SMS) messages when the status of your directory changes. You will be notified if your directory goes from an **Active** status to an **Impaired** or **Inoperable** status. You also receive a notification when the directory returns to an Active status.

Also remember that if you have an SNS topic that receives messages from AWS Directory Service, before deleting that topic from the Amazon SNS console, you should associate your directory with a different SNS topic. Otherwise you risk missing important directory status messages. For information about how to set up Amazon SNS, see Configure directory status notifications (p. 52).

Remove Amazon Enterprise applications before deleting a directory

Before deleting a directory that is associated with one or more Amazon Enterprise Applications such as, WorkSpaces, Amazon WorkSpaces Application Manager, Amazon WorkDocs, Amazon WorkMail, AWS Management Console, or Amazon Relational Database Service (Amazon RDS), you must first remove each application. For more information how to remove these applications, see Delete your directory (p. 127).

Use SMB 2.x clients when accessing the SYSVOL and NETLOGON shares

Client computers use Server Message Block (SMB) to access the SYSVOL and NETLOGON shares on AWS Managed Microsoft AD domain controllers for Group Policy, login scripts and other files. AWS Managed Microsoft AD only supports SMB version 2.0 (SMBv2) and newer.

The SMBv2 and newer version protocols add a number of features that improve client performance and increase the security of your domain controllers and clients. This change follows recommendations by the United States Computer Emergency Readiness Team and Microsoft to disable SMBv1.

> **Important**
> If you currently use SMBv1 clients to access the SYSVOL and NETLOGON shares of your domain controller, you must update those clients to use SMBv2 or newer. Your directory will work correctly but your SMBv1 clients will fail to connect to the SYSVOL and NETLOGON shares of your AWS Managed Microsoft AD domain controllers, and will also be unable to process Group Policy.

SMBv1 clients will work with any other SMBv1 compatible file servers that you have. However, AWS recommends that you update all of your SMB servers and clients to SMBv2 or newer. To learn more about disabling SMBv1 and updating it to newer SMB versions on your systems, see these postings on Microsoft TechNet and Support.

Tracking SMBv1 Remote Connections

You can review the **Microsoft-Windows-SMBServer/Audit** Windows Event log remotely connecting to the AWS Managed Microsoft AD domain controller, any events in this log indicate SMBv1 connections. Below is an example of the information you might see in one of these logs:

SMB1 access

Client Address: ###.###.###.###

Guidance:

This event indicates that a client attempted to access the server using SMB1. To stop auditing SMB1 access, use the Windows PowerShell cmdlet Set-SmbServerConfiguration.

Programming your applications

Before you program your applications, consider the following:

Use the Windows DC locator service

When developing applications, use the Windows DC locator service or use the Dynamic DNS (DDNS) service of your AWS Managed Microsoft AD to locate domain controllers (DCs). Do not hard code applications with the address of a DC. The DC locator service helps ensure directory load is distributed and enables you to take advantage of horizontal scaling by adding domain controllers to your deployment. If you bind your application to a fixed DC and the DC undergoes patching or recovery, your application will lose access to the DC instead of using one of the remaining DCs. Furthermore, hard coding of the DC can result in hot spotting on a single DC. In severe cases, hot spotting may cause your DC to become unresponsive. Such cases may also cause AWS directory automation to flag the directory as impaired and may trigger recovery processes that replace the unresponsive DC.

Load test before rolling out to production

Be sure to do lab testing with objects and requests that are representative of your production workload to confirm that the directory scales to the load of your application. Should you require additional capacity, test with additional DCs while distributing requests between the DCs. For more information, see Deploy additional domain controllers (p. 143).

Use efficient LDAP queries

Broad LDAP queries to a domain controller across tens of thousands of objects can consume significant CPU cycles in a single DC, resulting in hot spotting. This may affect applications that share the same DC during the query.

AWS Managed Microsoft AD quotas

The following are the default quotas for AWS Managed Microsoft AD. Each quota is per Region unless otherwise noted.

AWS Managed Microsoft AD quotas

Resource	Default quota
AWS Managed Microsoft AD directories	10

Resource	Default quota
Manual snapshots *	5 per AWS Managed Microsoft AD
Manual snapshots age **	180 days
Maximum number of domain controllers per directory	20
Shared domains per Standard Microsoft AD ***	5
Shared domains per Enterprise Microsoft AD ***	125
Maximum number of registered certificate authority (CA) certificates per directory	5
Maximum number of total AWS Regions in a single AWS Managed Microsoft AD (Enterprise Edition) directory ****	5

* The manual snapshot quota cannot be changed.

** The maximum supported age of a manual snapshot is 180 days and cannot be changed. This is due to the Tombstone-Lifetime attribute of deleted objects which defines the useful shelf life of a system-state backup of Active Directory. It is not possible to restore from a snapshot older than 180 days. For more information, see Useful shelf life of a system-state backup of Active Directory on the Microsoft website.

*** The shared domain default quota refers to the number of accounts that an individual directory can be shared to.

**** This includes 1 primary Region and up to 4 additional Regions. For more information, see Primary vs additional Regions (p. 59).

> **Note**
> You cannot attach a public IP address to your AWS elastic network interface (ENI).

For information regarding application design and load distribution, see Programming your applications (p. 152).

For storage and object quotas, see the **Comparison Table** on the AWS Directory Service Pricing page.

Application compatibility policy for AWS Managed Microsoft AD

AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD) is compatible with multiple AWS services and third-party applications.

The following is a list of compatible AWS applications and services:

- Amazon Chime - For detailed instructions, see Connect to your Active Directory.
- Amazon Connect - For more information, see How Amazon Connect works.
- Amazon EC2 – For more information, see Join an EC2 instance to your AWS Managed Microsoft AD directory (p. 71).
- FSx for Windows File Server – For more information, see What is FSx for Windows File Server?.

- Amazon QuickSight - For more information, see Managing user accounts in Amazon QuickSight Enterprise Edition.
- Amazon RDS for MySQL - For more information, see Using Kerberos authentication for MySQL.
- Amazon RDS for SQL Server - For more information, see Using Windows authentication with an Amazon RDS Microsoft SQL Server DB instance.
- Amazon RDS for Oracle - For more information, see Using Kerberos authentication with Amazon RDS for Oracle.
- Amazon RDS for PostgreSQL - For more information, see Using Kerberos authentication with Amazon RDS for PostgreSQL.
- AWS Single Sign-On - For detailed instructions, see Connect AWS SSO to an on-premises Active Directory.
- Amazon WorkDocs - For detailed instructions, see Connecting to your on-premises directory with AWS Managed Microsoft AD.
- Amazon WorkMail - For detailed instructions, see Integrate Amazon WorkMail with an existing directory (standard setup).
- WorkSpaces - For detailed instructions, see Launch a WorkSpace using AWS Managed Microsoft AD.
- AWS Client VPN - For detailed instructions, see Client authentication and authorization.
- AWS Management Console – For more information, see Enable access to the AWS Management Console with AD credentials (p. 141).

Due to the magnitude of custom and commercial off-the-shelf applications that use Active Directory, AWS does not and cannot perform formal or broad verification of third-party application compatibility with AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD). Although AWS works with customers in an attempt to overcome any potential application installation challenges they might encounter, we are unable to guarantee that any application is or will continue to be compatible with AWS Managed Microsoft AD.

The following third-party applications are compatible with AWS Managed Microsoft AD:

- Active Directory-Based Activation (ADBA)
- Active Directory Certificate Services (AD CS): Enterprise Certificate Authority
- Active Directory Federation Services (AD FS)
- Active Directory Users and Computers (ADUC)
- Application Server (.NET)
- Azure Active Directory (Azure AD)
- Azure Active Directory (AD) Connect
- Distributed File System Replication (DFSR)
- Distributed File System Namespaces (DFSN)
- Microsoft Remote Desktop Services Licensing Server
- Microsoft SharePoint Server
- Microsoft SQL Server (including SQL Server Always On Availability Groups)
- Microsoft System Center Configuration Manager (SCCM) - The user deploying SCCM must be a member of the AWS Delegated System Management Administrators group.
- Microsoft Windows and Windows Server OS
- Office 365

Note that not all configurations of these applications may be supported.

Compatibility guidelines

Although applications may have configurations that are incompatible, application deployment configurations can often overcome incompatibility. The following describes the most common reasons for application incompatibility. Customers can use this information to investigate compatibility characteristics of a desired application and identify potential deployment changes.

- **Domain administrator or other privileged permissions –** Some applications state that you must install them as the domain administrator. Because AWS must retain exclusive control of this permission level in order to deliver Active Directory as a managed service, you cannot act as the domain administrator to install such applications. However, you can often install such applications by delegating specific, less privileged, and AWS supported permissions to the person who performs the installation. For more details on the precise permissions that your application requires, ask your application provider. For more information about permissions that AWS allows you to delegate, see What gets created (p. 12).
- **Access to privileged Active Directory containers –** Within your directory, AWS Managed Microsoft AD provides an Organizational Unit (OU) over which you have full administrative control. You do not have create or write permissions and may have limited read permissions to containers that are higher in the Active Directory tree than your OU. Applications that create or access containers for which you have no permissions might not work. However, such applications often have an ability to use a container that you create in your OU as an alternative. Check with your application provider to find ways to create and use a container in your OU as an alternative. For more information on managing your OU, see How to administer AWS Managed Microsoft AD (p. 28).
- **Schema changes during the install workflow –** Some Active Directory applications require changes to the default Active Directory schema, and they may attempt to install those changes as part of the application installation workflow. Due to the privileged nature of schema extensions, AWS makes this possible by importing Lightweight Directory Interchange Format (LDIF) files through the AWS Directory Service console, CLI, or SDK only. Such applications often come with an LDIF file that you can apply to the directory through the AWS Directory Service schema update process. For more information about how the LDIF import process works, see Tutorial: Extending your AWS Managed Microsoft AD schema (p. 121). You can install the application in a way to bypass the schema installation during the installation process.

Known incompatible applications

The following lists commonly requested commercial off-the-shelf applications for which we have not found a configuration that works with AWS Managed Microsoft AD. AWS updates this list from time to time at its sole discretion as a courtesy to help you avoid unproductive efforts. AWS provide this information without warranty or claims regarding current or future compatibility.

- Active Directory Certificate Services (AD CS): Certificate Enrollment Web Service
- Active Directory Certificate Services (AD CS): Certificate Enrollment Policy Web Service
- Microsoft Exchange Server
- Microsoft Skype for Business Server

AWS Managed Microsoft AD test lab tutorials

This section provides a series of guided tutorials to help you establish a test lab environment in AWS where you can experiment with AWS Managed Microsoft AD.

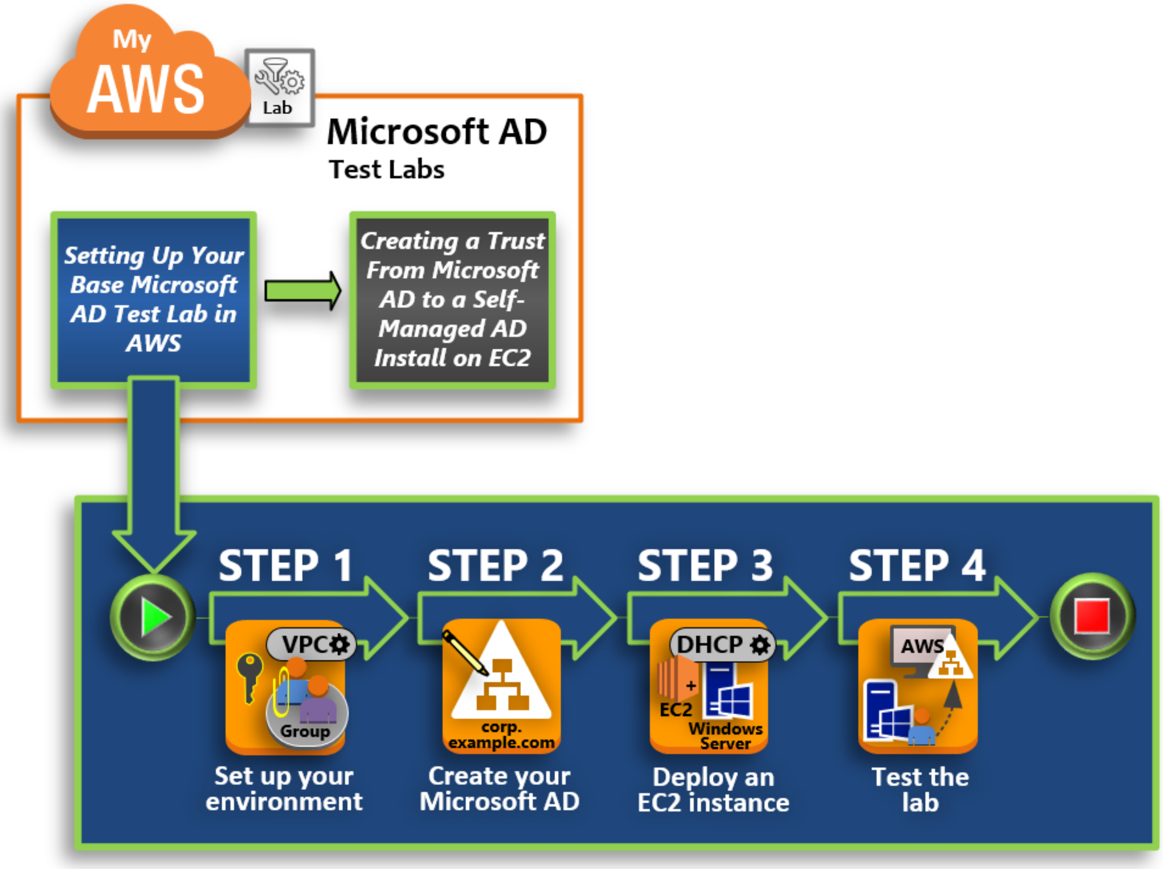
Topics

- Tutorial: Setting up your base AWS Managed Microsoft AD test lab in AWS (p. 156)
- Tutorial: Creating a trust from AWS Managed Microsoft AD to a self-managed Active Directory installation on Amazon EC2 (p. 167)

Tutorial: Setting up your base AWS Managed Microsoft AD test lab in AWS

This tutorial teaches you how to set up your AWS environment to prepare for a new AWS Managed Microsoft AD installation that uses a new EC2 instance running Windows Server 2019. It then teaches you to use typical Active Directory administration tools to manage your AWS Managed Microsoft AD environment from your Windows system. By the time you complete the tutorial, you will have set up the network prerequisites and have configured a new AWS Managed Microsoft AD forest.

As shown in the following illustration, the lab you create from this tutorial is the foundational component for hands-on learning about AWS Managed Microsoft AD. You can later add optional tutorials for more hands-on experience. This tutorial series is ideal for anyone who is new to AWS Managed Microsoft AD and wants a test lab for evaluation purposes. This tutorial takes approximately 1 hour to complete.



Step 1: Set up your AWS environment for AWS Managed Microsoft AD (p. 157)

After you've completed your prerequisite tasks, you create and configure a VPC in your EC2 instance.

Step 2: Create your AWS Managed Microsoft AD directory in AWS (p. 162)

In this step, you set up AWS Managed Microsoft AD in AWS for the first time.

Step 3: Deploy an EC2 instance to manage your AWS Managed Microsoft AD (p. 163)

Here, you walk through the various post-deployment tasks necessary for client computers to connect to your new domain and set up a new Windows Server system in EC2.

Step 4: Verify that the base test lab is operational (p. 166)

Finally, as an administrator, you verify that you can log in and connect to AWS Managed Microsoft AD from your Windows Server system in EC2. Once you've successfully tested that the lab is operational, you can continue to add other test lab guide modules.

Prerequisites

If you plan to use only the UI steps in this tutorial to create your test lab, you can skip this prerequisites section and move on to Step 1. However, if you plan to use either AWS CLI commands or AWS Tools for Windows PowerShell modules to create your test lab environment, you must first configure the following:

- **IAM user with the access and secret access key** – An IAM user with an access key is required if you want to use the AWS CLI or AWS Tools for Windows PowerShell modules. If you do not have an access key, see Creating, modifying, and viewing access keys (AWS Management Console).
- **AWS Command Line Interface (optional)** – Download and Install the AWS CLI on Windows. Once installed, open the command prompt or Windows PowerShell window, and then type `aws configure`. Note that you need the access key and secret key to complete the setup. See the first prerequisite for steps on how to do this. You will be prompted for the following:
  - AWS access key ID [None]: `AKIAIOSFODNN7EXAMPLE`
  - AWS secret access key [None]: `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`
  - Default Region name [None]: `us-west-2`
  - Default output format [None]: `json`
- **AWS Tools for Windows PowerShell (optional)** – Download and install the latest version of the AWS Tools for Windows PowerShell from https://aws.amazon.com/powershell/, and then run the following command. Note that you need your access key and secret key to complete the setup. See the first prerequisite for the steps on how to do this.

  ```
  Set-AWSCredentials -AccessKey {AKIAIOSFODNN7EXAMPLE} -SecretKey
  {wJalrXUtnFEMI/K7MDENG/ bPxRfiCYEXAMPLEKEY} -StoreAs {default}
  ```

Step 1: Set up your AWS environment for AWS Managed Microsoft AD

Before you can create AWS Managed Microsoft AD in your AWS test lab, you first need to set up your Amazon EC2 key pair so that all login data is encrypted.

Create a key pair

If you already have a key pair, you can skip this step. For more information about Amazon EC2 key pairs, see http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html.

To create a key pair

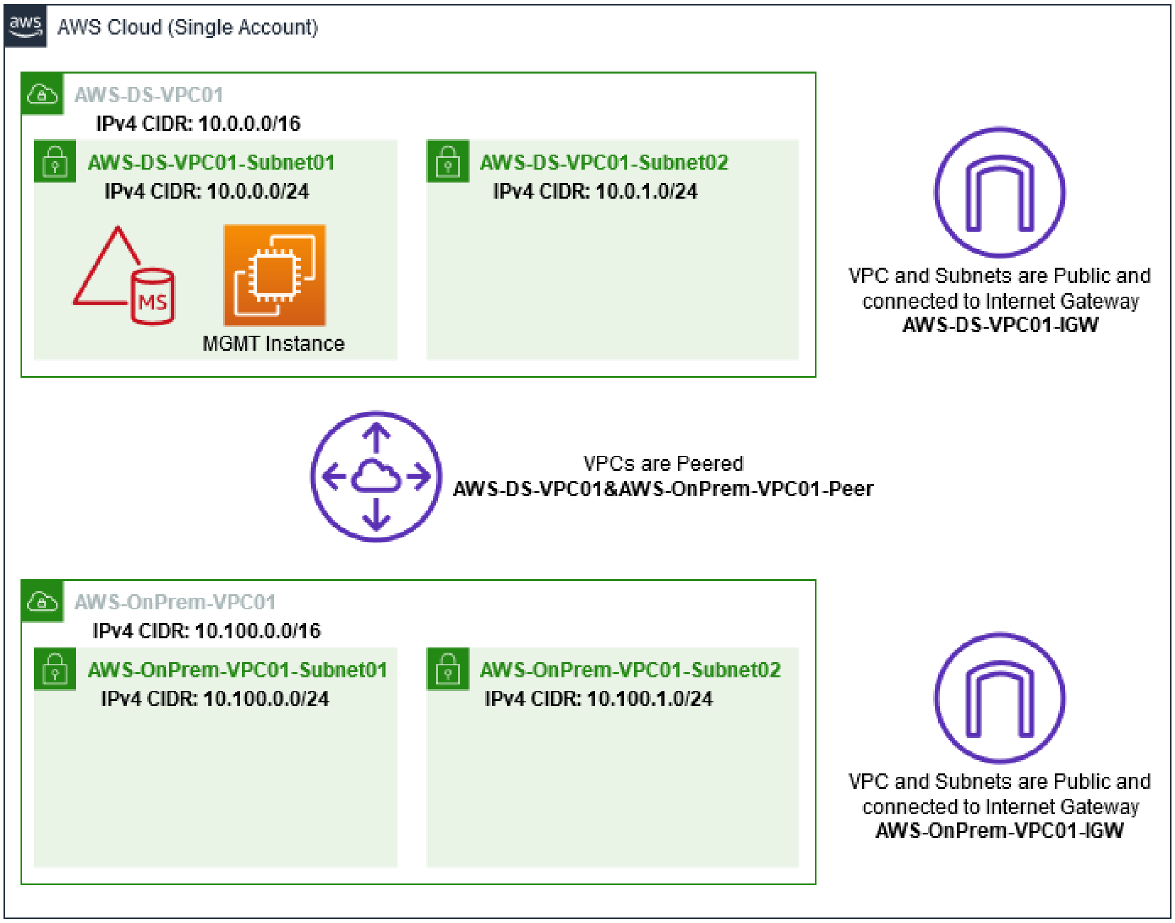
1. Sign in to the AWS Management Console and open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under **Network & Security**, choose **Key Pairs**, and then choose **Create Key Pair**.

3. For **Key pair name**, type **`AWS-DS-KP`**. For **Key pair file format**, select **pem**, and then choose **Create**.
4. The private key file is automatically downloaded by your browser. The file name is the name you specified when you created your key pair with an extension of `.pem`. Save the private key file in a safe place.

    Important
    This is the only chance for you to save the private key file. You need to provide the name of your key pair when you launch an instance and the corresponding private key each time you decrypt the password for the instance.

Create, configure, and peer two VPCs

As shown in the following illustration, by the time you finish this multi-step process you will have created and configured two public VPCs, two public subnets per VPC, one Internet Gateway per VPC, and one VPC Peering connection between the VPCs. We chose to use public VPCs and subnets for the purpose of simplicity and cost. For production workloads, we recommend that you use private VPCs. For more information about improving VPC Security, see Security in Amazon Virtual Private Cloud.



All of the AWS CLI and PowerShell examples use the VPC information from below and are built in us-west-2. You may choose any supported Region to build you environment in. For general information, see What is Amazon VPC?.

Step 1: Create two VPCs

In this step, you need to create two VPCs in the same account using the specified parameters in the following table. AWS Managed Microsoft AD supports the use of separate accounts with the Share your directory (p. 63) feature. The first VPC will be used for AWS Managed Microsoft AD. The second VPC will be used for resources that can be used later in Tutorial: Creating a trust from AWS Managed Microsoft AD to a self-managed Active Directory installation on Amazon EC2 (p. 167).

Managed AD VPC information	On-premise VPC information
Name tag: AWS-DS-VPC01 IPv4 CIDR block: 10.0.0.0/16 IPv6 CIDR block: No IPv6 CIDR Block Tenancy: Default	Name tag: AWS-OnPrem-VPC01 IPv4 CIDR block: 10.100.0.0/16 IPv6 CIDR block: No IPv6 CIDR Block Tenancy: Default

For detailed instructions, see Creating a VPC.

Step 2: Create two subnets per VPC

After you have created the VPCs you will need to create two subnets per VPC using the specified parameters in the following table. For this test lab each subnet will be a /24. This will allows up to 256 addresses to be issued per subnet. Each subnet must be a in a separate AZ. Putting each subnet in a separate in AZ is one of the AWS Managed Microsoft AD prerequisites (p. 9).

AWS-DS-VPC01 subnet Information:	AWS-OnPrem-VPC01 subnet information
Name tag: AWS-DS-VPC01-Subnet01 VPC: vpc-xxxxxxxxxxxxxxxxx AWS-DS-VPC01 Availability Zone: us-west-2a IPv4 CIDR block: 10.0.0.0/24	Name tag: AWS-OnPrem-VPC01-Subnet01 VPC: vpc-xxxxxxxxxxxxxxxxx AWS-OnPrem-VPC01 Availability Zone: us-west-2a IPv4 CIDR block: 10.100.0.0/24
Name tag: AWS-DS-VPC01-Subnet02 VPC: vpc-xxxxxxxxxxxxxxxxx AWS-DS-VPC01 Availability Zone: us-west-2b IPv4 CIDR block: 10.0.1.0/24	Name tag: AWS-OnPrem-VPC01-Subnet02 VPC: vpc-xxxxxxxxxxxxxxxxx AWS-OnPrem-VPC01 Availability Zone: us-west-2b IPv4 CIDR block: 10.100.1.0/24

For detailed instructions, see Creating a subnet in your VPC.

Step 3: Create and attach an Internet Gateway to your VPCs

Since we are using public VPCs you will need to create and attach an Internet gateway to your VPCs using the specified parameters in the following table. This will allow you to be able to connect to and manage your EC2 instances.

AWS-DS-VPC01 Internet Gateway information	AWS-OnPrem-VPC01 Internet Gateway information
Name tag: AWS-DS-VPC01-IGW VPC: vpc-xxxxxxxxxxxxxxxxx AWS-DS-VPC01	Name tag: AWS-OnPrem-VPC01-IGW VPC: vpc-xxxxxxxxxxxxxxxxx AWS-OnPrem-VPC01

For detailed instructions, see Internet gateways.

Step 4: Configure a VPC peering connection between AWS-DS-VPC01 and AWS-OnPrem-VPC01

Since you already created two VPCs earlier, you will need to network them together using VPC peering using the specified parameters in the following table. While there are many ways to connect your VPCs, this tutorial will use VPC Peering. AWS Managed Microsoft AD supports many solutions to connect your VPCs, some of these include VPC peering, Transit Gateway, and VPN.

```
Peering connection name tag: AWS-DS-VPC01&AWS-OnPrem-VPC01-Peer

VPC (Requester): vpc-xxxxxxxxxxxxxxxxx AWS-DS-VPC01

Account: My Account

Region: This Region

VPC (Accepter): vpc-xxxxxxxxxxxxxxxxx AWS-OnPrem-VPC01
```

For instructions on how to create a VPC Peering Connection with another VPC from with in your account, see Creating a VPC peering connection with another VPC in your account.

Step 5: Add two routes to each VPC's main route table

In order for the Internet Gateways and VPC Peering Connection created in the previous steps to be functional you will need to update the main route table of both VPCs using the specified parameters in the following table. You will be adding two routes; 0.0.0.0/0 which will route to all destinations not explicitly known to the route table and 10.0.0.0/16 or 10.100.0.0/16 which will route to each VPC over the VPC Peering Connection established above.

You can easily find the correct route table for each VPC by filtering on the VPC name tag (AWS-DS-VPC01 or AWS-OnPrem-VPC01).

AWS-DS-VPC01 route 1 information	AWS-DS-VPC01 route 2 information	AWS-OnPrem-VPC01 route 1 Information	AWS-OnPrem-VPC01 route 2 Information
Destination: 0.0.0.0/0 Target: igw-xxxxxxxxxxxxxxxxx AWS-DS-VPC01-IGW	Destination: 10.100.0.0/16 Target: pcx-xxxxxxxxxxxxxxxxx AWS-DS-VPC01&AWS-OnPrem-VPC01-Peer	Destination: 0.0.0.0/0 Target: igw-xxxxxxxxxxxxxxxxx AWS-Onprem-VPC01	Destination: 10.0.0.0/16 Target: pcx-xxxxxxxxxxxxxxxxx AWS-DS-VPC01&AWS-OnPrem-VPC01-Peer

For instructions on how to add routes to a VPC route table, see Adding and removing routes from a route table.

Create security groups for EC2 instances

By default, AWS Managed Microsoft AD creates a security group to manage traffic between its domain controllers. In this section, you will need to create 2 security groups (one for each VPC) which will be used to manage traffic within your VPC for your EC2 instances using the specified parameters in the following tables. You also add a rule that allows RDP (3389) inbound from anywhere and for all traffic types inbound from the local VPC. For more information, see Amazon EC2 security groups for Windows instances.

AWS-DS-VPC01 security group information:
Security group name: AWS DS Test Lab Security Group Description: AWS DS Test Lab Security Group VPC: vpc-xxxxxxxxxxxxxxxxx AWS-DS-VPC01

Security Group Inbound Rules for AWS-DS-VPC01

Type	Protocol	Port range	Source	Type of traffic
Custom TCP Rule	TCP	3389	My IP	Remote Desktop
All Traffic	All	All	10.0.0.0/16	All local VPC traffic

Security Group Outbound Rules for AWS-DS-VPC01

Type	Protocol	Port range	Destination	Type of traffic
All Traffic	All	All	0.0.0.0/0	All traffic

AWS-OnPrem-VPC01 security group information:
Security group name: AWS OnPrem Test Lab Security Group. Description: AWS OnPrem Test Lab Security Group. VPC: vpc-xxxxxxxxxxxxxxxxx AWS-OnPrem-VPC01

Security Group Inbound Rules for AWS-OnPrem-VPC01

Type	Protocol	Port range	Source	Type of traffic
Custom TCP Rule	TCP	3389	My IP	Remote Desktop
Custom TCP Rule	TCP	53	10.0.0.0/16	DNS
Custom TCP Rule	TCP	88	10.0.0.0/16	Kerberos
Custom TCP Rule	TCP	389	10.0.0.0/16	LDAP
Custom TCP Rule	TCP	464	10.0.0.0/16	Kerberos change / set password
Custom TCP Rule	TCP	445	10.0.0.0/16	SMB / CIFS
Custom TCP Rule	TCP	135	10.0.0.0/16	Replication
Custom TCP Rule	TCP	636	10.0.0.0/16	LDAP SSL
Custom TCP Rule	TCP	49152 - 65535	10.0.0.0/16	RPC

Type	Protocol	Port range	Source	Type of traffic
Custom TCP Rule	TCP	3268 - 3269	10.0.0.0/16	LDAP GC & LDAP GC SSL
Custom UDP Rule	UDP	53	10.0.0.0/16	DNS
Custom UDP Rule	UDP	88	10.0.0.0/16	Kerberos
Custom UDP Rule	UDP	123	10.0.0.0/16	Windows Time
Custom UDP Rule	UDP	389	10.0.0.0/16	LDAP
Custom UDP Rule	UDP	464	10.0.0.0/16	Kerberos change / set password
All Traffic	All	All	10.100.0.0/16	All local VPC traffic

Security Group Outbound Rules for AWS-OnPrem-VPC01

Type	Protocol	Port range	Destination	Type of traffic
All Traffic	All	All	0.0.0.0/0	All traffic

For detailed instructions on how to create and add rules to your security groups, see Working with security groups.

Step 2: Create your AWS Managed Microsoft AD directory in AWS

You can use three different methods to create your directory. You can use the AWS Management Console procedure (recommended for this tutorial) or you can use either the AWS CLI or AWS Tools for Windows PowerShell procedures to create your directory.

Method 1: To create your AWS Managed Microsoft AD directory (AWS Management Console)

1. In the AWS Directory Service console navigation pane, choose **Directories** and then choose **Set up directory**.
2. On the **Select directory type** page, choose **AWS Managed Microsoft AD**, and then choose **Next**.
3. On the **Enter directory information** page, provide the following information, and then choose **Next**.

   - For **Edition**, select either **Standard Edition** or **Enterprise Edition**. For more information about editions, see AWS Directory Service for Microsoft Active Directory (p. 1).
   - For **Directory DNS name**, type `corp.example.com`.
   - For **Directory NetBIOS name**, type `corp`.
   - For **Directory description**, type `AWS DS Managed`.
   - For **Admin password**, type the password you want to use for this account and type the password again in **Confirm password**. This **Admin** account is automatically created during the directory creation process. The password cannot include the word *admin*. The directory administrator password is case sensitive and must be between 8 and 64 characters in length, inclusive. It must also contain at least one character from three of the following four categories:
     - Lowercase letters (a-z)
     - Uppercase letters (A-Z)

   - Numbers (0-9)
   - Non-alphanumeric characters (~!@#$%^&*_-+=`|\(){}[]:;"'<>,.?/)

4. On the **Choose VPC and subnets** page, provide the following information, and then choose **Next**.

   - For **VPC**, choose the option that begins with **AWS-DS-VPC01** and ends with **(10.0.0.0/16)**.
   - For **Subnets**, choose the **10.0.0.0/24** and **10.0.1.0/24** public subnets.

5. On the **Review & create** page, review the directory information and make any necessary changes. When the information is correct, choose **Create directory**. Creating the directory takes 20 to 40 minutes. Once created, the **Status** value changes to **Active**.

Method 2: To create your AWS Managed Microsoft AD (Windows PowerShell) (Optional)

1. Open Windows PowerShell.
2. Type the following command. Make sure to use the values provided in Step 4 of the preceding AWS Management Console procedure.

   ```
   New-DSMicrosoftAD -Name corp.example.com -ShortName corp -Password
   P@ssw0rd -Description “AWS DS Managed” - VpcSettings_VpcId vpc-xxxxxxxx -
   VpcSettings_SubnetId subnet-xxxxxxxx, subnet-xxxxxxxx
   ```

Method 3: To create your AWS Managed Microsoft AD (AWS CLI) (Optional)

1. Open the AWS CLI.
2. Type the following command. Make sure to use the values provided in Step 4 of the preceding AWS Management Console procedure.

   ```
   aws ds create-microsoft-ad --name corp.example.com --short-name corp --
   password P@ssw0rd --description "AWS DS Managed" --vpc-settings VpcId= vpc-
   xxxxxxxx,SubnetIds= subnet-xxxxxxxx, subnet-xxxxxxxx
   ```

Step 3: Deploy an EC2 instance to manage your AWS Managed Microsoft AD

For this lab, we are using EC2 instances that have public IP addresses to make it easy to access the management instance from anywhere. In a production setting, you can use instances that are in a private VPC that are only accessible through a VPN or Amazon Direct Connect link. There is no requirement the instance have a public IP address.

In this section, you walk through the various post-deployment tasks necessary for client computers to connect to your domain using the Windows Server on your new EC2 instance. You use the Windows Server in the next step to verify that the lab is operational.

Optional: Create a DHCP options set in AWS-DS-VPC01 for your directory

In this optional procedure, you set up a DHCP option scope so that EC2 instances in your VPC automatically use your AWS Managed Microsoft AD for DNS resolution. For more information, see DHCP options sets.

To create a DHCP options set for your directory

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **DHCP Options Sets**, and then choose **Create DHCP options set**.
3. On the **Create DHCP options set** page, provide the following values for your directory:

- For **Name**, type `AWS DS DHCP`.
- For **Domain name**, type `corp.example.com`.
- For **Domain name servers**, type the IP addresses of your AWS provided directory's DNS servers.

  Note

  To find these addresses, go to the AWS Directory Service **Directories** page, and then choose the applicable directory ID. On the **Details** page, identify and use the IPs that are displayed in **DNS address**.

  Alternatively, to find these addresses, go to the AWS Directory Service **Directories** page, and choose the applicable directory ID. Then, choose **Scale & share**. Under **Domain controllers**, identify and use the IPs that are displayed in **IP address**.
- Leave the settings blank for **NTP servers**, **NetBIOS name servers**, and **NetBIOS node type**.

4. Choose **Create DHCP options set**, and then choose **Close**. The new set of DHCP options appear in your list of DHCP options.
5. Make a note of the ID of the new set of DHCP options (**dopt-*xxxxxxxx***). You use it at the end of this procedure when you associate the new options set with your VPC.

   Note

   Seamless domain join works without having to configure a DHCP Options Set.
6. In the navigation pane, choose **Your VPCs**.
7. In the list of VPCs, select **AWS DS VPC**, choose **Actions**, and then choose **Edit DHCP options set**.
8. On the **Edit DHCP options set** page, select the options set that you recorded in Step 5, and then choose **Save**.

Create a role to join Windows instances to your AWS Managed Microsoft AD domain

Use this procedure to configure a role that joins an EC2 Windows instance to a domain. For more information, see Seamlessly join a Windows EC2 instance in the *Amazon EC2 User Guide for Windows Instances*.

To configure EC2 to join Windows instances to your domain

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane of the IAM console, choose **Roles**, and then choose **Create role**.
3. Under **Select type of trusted entity**, choose **AWS service**.
4. Immediately under **Choose the service that will use this role**, choose **EC2**, and then choose **Next: Permissions**.
5. On the **Attached permissions policy** page, do the following:

   - Select the box next to the **AmazonSSMManagedInstanceCore** managed policy. This policy provides the minimum permissions necessary to use the Systems Manager service.
   - Select the box next to **AmazonSSMDirectoryServiceAccess** managed policy. The policy provides the permissions to join instances to an Active Directory managed by AWS Directory Service.

   For information about these managed policies and other policies you can attach to an IAM instance profile for Systems Manager, see Create an IAM instance profile for Systems Manager in the *AWS Systems Manager User Guide*. For information about managed policies, see AWS Managed policies in the *IAM User Guide*.
6. Choose **Next: Tags**.
7. (Optional) Add one or more tag key-value pairs to organize, track, or control access for this role, and then choose **Next: Review**.

8. For **Role name**, enter a name for the role that describes that it is used to join instances to a domain, such as **EC2DomainJoin**.
9. (Optional) For **Role description**, enter a description.
10. Choose **Create role**. The system returns you to the **Roles** page.

Create an EC2 instance and automatically join the directory

In this procedure you set up a Windows Server system in Amazon EC2 that can be used later to administer users, groups, and policies in Active Directory.

To create an EC2 instance and automatically join the directory

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Choose **Launch Instance**.
3. On the **Step 1** page, next to **Microsoft Windows Server 2019 Base - ami-*xxxxxxxxxxxxxxxxx*** choose **Select**.
4. On the **Step 2** page, select **t3.micro** (note, you can choose a larger instance type), and then choose **Next: Configure Instance Details**.
5. On the **Step 3** page, do the following:
   - For **Network**, choose the VPC that ends with **AWS-DS-VPC01** (for example, **vpc-*xxxxxxxxxxxxxxxxx* | AWS-DS-VPC01**).
   - For **Subnet** choose **Public subnet 1**, which should be preconfigured for your preferred Availability Zone (for example, **subnet-*xxxxxxxxxxxxxxxxx* | AWS-DS-VPC01-Subnet01 | *us-west-2a***).
   - For **Auto-assign Public IP**, choose **Enable** (if the subnet setting is not set to enable by default).
   - For **Domain join directory**, choose **corp.example.com (d-*xxxxxxxxxx*)**.
   - For **IAM role** choose the name you gave your instance role in Create a role to join Windows instances to your AWS Managed Microsoft AD domain (p. 164), such as **EC2DomainJoin**.
   - Leave the rest of the settings at their defaults.
   - Choose **Next: Add Storage**.
6. On the **Step 4** page, leave the default settings, and then choose **Next: Add Tags**.
7. On the **Step 5** page, choose **Add Tag**. Under **Key** type `corp.example.com-mgmt` and then choose **Next: Configure Security Group**.
8. On the **Step 6** page, choose **Select an existing security group**, select **AWS DS Test Lab Security Group** (which you previously set up in the Base tutorial (p. 160)), and then choose **Review and Launch** to review your instance.
9. On the **Step 7** page, review the page, and then choose **Launch**.
10. On the **Select an existing key pair or create a new key pair** dialog box, do the following:
    - Choose **Choose an existing key pair**.
    - Under **Select a key pair**, choose **AWS-DS-KP**.
    - Select the **I acknowledge...** check box.
    - Choose **Launch Instances**.
11. Choose **View Instances** to return to the Amazon EC2 console and view the status of the deployment.

Install the Active Directory tools on your EC2 instance

You can choose from two methods to install the Active Directory Domain Management Tools on your EC2 instance. You can use the Server Manager UI (recommended for this tutorial) or Windows PowerShell.

To install the Active Directory tools on your EC2 instance (Server Manager)

1. In the Amazon EC2 console, choose **Instances**, select the instance you just created, and then choose **Connect**.
2. In the **Connect To Your Instance** dialog box, choose **Get Password** to retrieve your password if you haven't already, and then choose **Download Remote Desktop File**.
3. In the **Windows Security** dialog box, type your local administrator credentials for the Windows Server computer to log in (for example, `administrator`).
4. From the **Start** menu, choose **Server Manager**.
5. In the **Dashboard**, choose **Add Roles and Features**.
6. In the **Add Roles and Features Wizard**, choose **Next**.
7. On the **Select installation type** page, choose **Role-based or feature-based installation**, and then choose **Next**.
8. On the **Select destination server** page, make sure that the local server is selected, and then choose **Next**.
9. On the **Select server roles** page, choose **Next**.
10. On the **Select features** page, do the following:
    - Select the **Group Policy Management** check box.
    - Expand **Remote Server Administration Tools**, and then expand **Role Administration Tools**.
    - Select the **AD DS and AD LDS Tools** check box.
    - Select the **DNS Server Tools** check box.
    - Choose **Next**.
11. On the **Confirm installation selections** page, review the information, and then choose **Install**. When the feature installation is finished, the following new tools or snap-ins will be available in the Windows Administrative Tools folder in the Start menu.
    - Active Directory Administrative Center
    - Active Directory Domains and Trusts
    - Active Directory Module for Windows PowerShell
    - Active Directory Sites and Services
    - Active Directory Users and Computers
    - ADSI Edit
    - DNS
    - Group Policy Management

To install the Active Directory tools on your EC2 instance (Windows PowerShell) (Optional)

1. Start Windows PowerShell.
2. Type the following command.

```
Install-WindowsFeature -Name GPMC,RSAT-AD-PowerShell,RSAT-AD-
AdminCenter,RSAT-ADDS-Tools,RSAT-DNS-Server
```

Step 4: Verify that the base test lab is operational

Use the following procedure to verify that the test lab has been set up successfully before adding on additional test lab guide modules. This procedure verifies that your Windows Server is configured appropriately, can connect to the corp.example.com domain, and be used to administer your AWS Managed Microsoft AD forest.

To verify that the test lab is operational

1. Sign out of the EC2 instance where you were logged in as the local administrator.
2. Back in the Amazon EC2 console, choose **Instances** in the navigation pane. Then select the instance that you created. Choose **Connect**.
3. In the **Connect To Your Instance** dialog box, choose **Download Remote Desktop File**.
4. In the **Windows Security** dialog box, type your administrator credentials for the CORP domain to log in (for example, **`corp\admin`**).
5. Once you are logged in, in the **Start** menu, under **Windows Administrative Tools**, choose **Active Directory Users and Computers**.
6. You should see **corp.example.com** displayed with all the default OUs and accounts associated with a new domain. Under **Domain Controllers**, notice the names of the domain controllers that were automatically created when you created your AWS Managed Microsoft AD back in Step 2 of this tutorial.

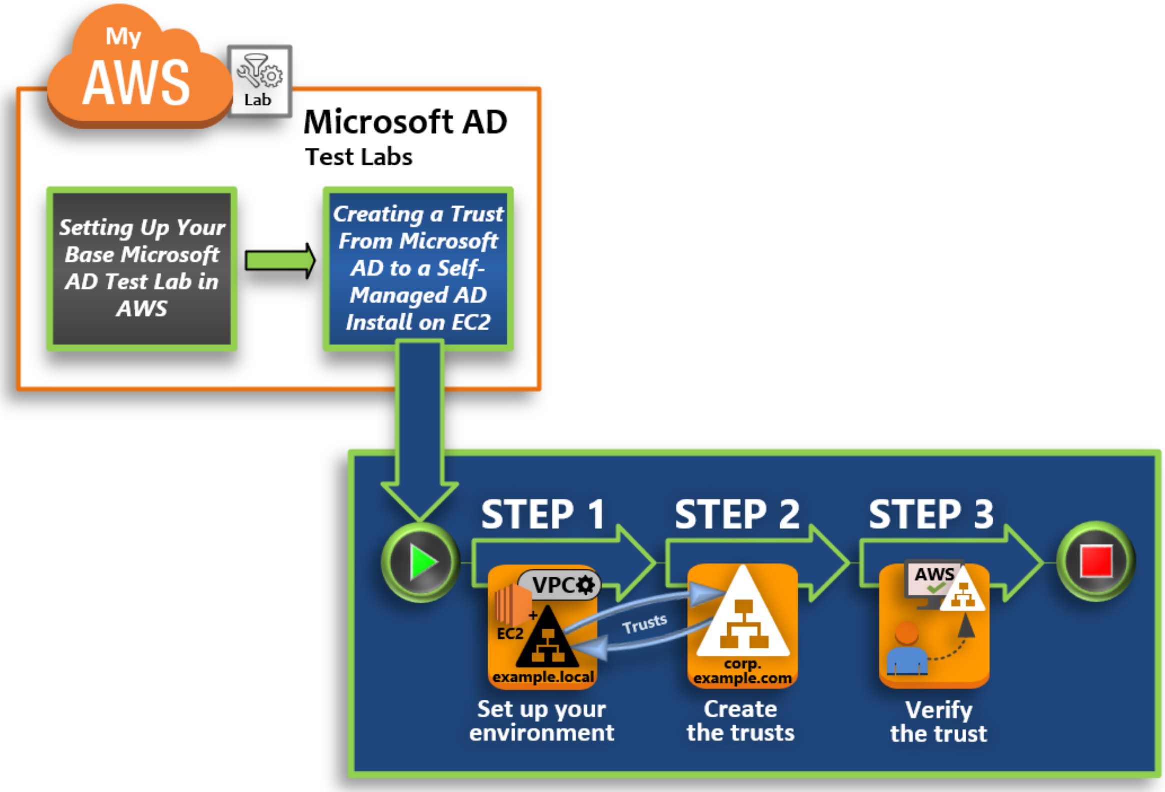
Congratulations! Your AWS Managed Microsoft AD base test lab environment has now been configured. You are ready to begin adding the next test lab in the series.

Next tutorial: Tutorial: Creating a trust from AWS Managed Microsoft AD to a self-managed Active Directory installation on Amazon EC2 (p. 167)

Tutorial: Creating a trust from AWS Managed Microsoft AD to a self-managed Active Directory installation on Amazon EC2

In this tutorial, you learn how to create a trust between the AWS Directory Service for Microsoft Active Directory forest that you created in the Base tutorial (p. 156). You also learn to create a new native Active Directory forest on a Windows Server in Amazon EC2. As shown in the following illustration, the lab that you create from this tutorial is the second building block necessary when setting up a complete AWS Managed Microsoft AD test lab. You can use the test lab to test your pure cloud or hybrid cloud–based AWS solutions.

You should only need to create this tutorial once. After that you can add optional tutorials when necessary for more experience.



Step 1: Set up your environment for trusts (p. 168)

Before you can establish trusts between a new Active Directory forest and the AWS Managed Microsoft AD forest that you created in the Base tutorial (p. 156), you need to prepare your Amazon EC2 environment. To do that, you first create a Windows Server 2019 server, promote that server to a domain controller, and then configure your VPC accordingly.

Step 2: Create the trusts (p. 172)

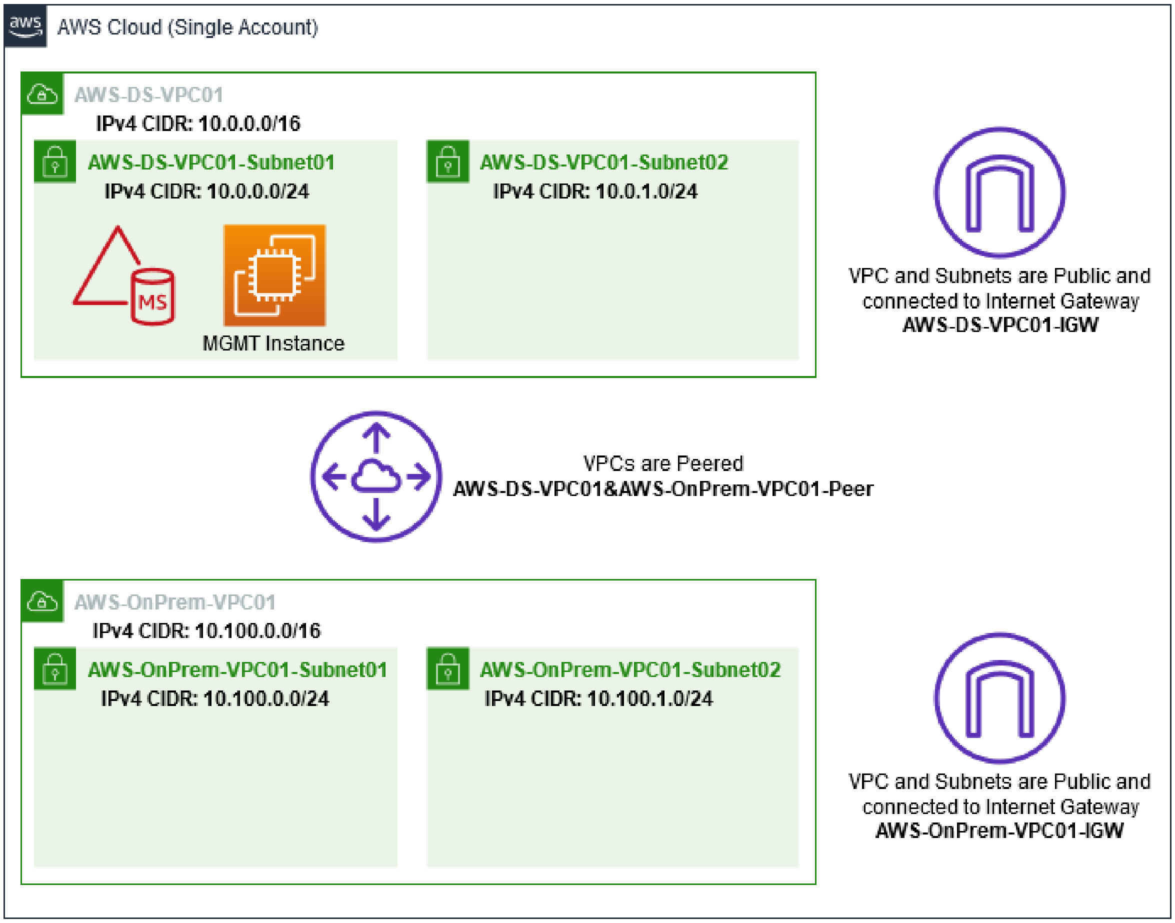
In this step, you create a two-way forest trust relationship between your newly created Active Directory forest hosted in Amazon EC2 and your AWS Managed Microsoft AD forest in AWS.

Step 3: Verify the trust (p. 174)

Finally, as an administrator, you use the AWS Directory Service console to verify that the new trusts are operational.

Step 1: Set up your environment for trusts

In this section, you set up your Amazon EC2 environment, deploy your new forest, and prepare your VPC for trusts with AWS.



Create a Windows Server 2019 EC2 instance

Use the following procedure to create a Windows Server 2019 member server in Amazon EC2.

To create a Windows Server 2019 EC2 instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the Amazon EC2 console, choose **Launch Instance**.
3. On the **Step 1** page, locate **Microsoft Windows Server 2019 Base - ami-*xxxxxxxxxxxxxxxxx*** in the list. Then choose **Select**.
4. On the **Step 2** page, select **t2.large**, and then choose **Next: Configure Instance Details**.
5. On the **Step 3** page, do the following:

   - For **Network**, select **vpc-*xxxxxxxxxxxxxxxxx* AWS-OnPrem-VPC01** (which you previously set up in the Base tutorial (p. 158)).
   - For **Subnet**, select **subnet-*xxxxxxxxxxxxxxxxx* | AWS-OnPrem-VPC01-Subnet01 | AWS-OnPrem-VPC01**.
   - For **Auto-assign Public IP** list, choose **Enable** (if the subnet setting is not set to **Enable** by default).
   - Leave the rest of the settings at their defaults.
   - Choose **Next: Add Storage**.
6. On the **Step 4** page, leave the default settings, and then choose **Next: Add Tags**.

7. On the **Step 5** page, choose **Add Tag**. Under **Key** type `example.local-DC01`, and then choose **Next: Configure Security Group**.
8. On the **Step 6** page, choose **Select an existing security group**, select **AWS On-Prem Test Lab Security Group** (which you previously set up in the Base tutorial (p. 160)), and then choose **Review and Launch** to review your instance.
9. On the **Step 7** page, review the page, and then choose **Launch**.
10. On the **Select an existing key pair or create a new key pair** dialog box, do the following:

    - Choose **Choose an existing key pair**.
    - Under **Select a key pair**, choose **AWS-DS-KP** (which you previously set up in the Base tutorial (p. 157)).
    - Select the **I acknowledge...** check box.
    - Choose **Launch Instances**.
11. Choose **View Instances** to return to the Amazon EC2 console and view the status of the deployment.

Promote your server to a domain controller

Before you can create trusts, you must build and deploy the first domain controller for a new forest. During this process you configure a new Active Directory forest, install DNS, and set this server to use the local DNS server for name resolution. You must reboot the server at the end of this procedure.

> **Note**
> If you want to create a domain controller in AWS that replicates with your on-premises network, you would first manually join the EC2 instance to your on-premises domain. After that you can promote the server to a domain controller.

To promote your server to a domain controller

1. In the Amazon EC2 console, choose **Instances**, select the instance you just created, and then choose **Connect**.
2. In the **Connect To Your Instance** dialog box, choose **Download Remote Desktop File**.
3. In the **Windows Security** dialog box, type your local administrator credentials for the Windows Server computer to login (for example, `administrator`). If you do not yet have the local administrator password, go back to the Amazon EC2 console, right-click on the instance, and choose **Get Windows Password**. Navigate to your `AWS DS KP.pem` file or your personal `.pem` key, and then choose **Decrypt Password**.
4. From the **Start** menu, choose **Server Manager**.
5. In the **Dashboard**, choose **Add Roles and Features**.
6. In the **Add Roles and Features Wizard**, choose **Next**.
7. On the **Select installation type** page, choose **Role-based or feature-based installation**, and then choose **Next**.
8. On the **Select destination server** page, make sure that the local server is selected, and then choose **Next**.
9. On the **Select server roles** page, select **Active Directory Domain Services**. In the **Add Roles and Features Wizard** dialog box, verify that the **Include management tools (if applicable)** check box is selected. Choose **Add Features**, and then choose **Next**.
10. On the **Select features** page, choose **Next**.
11. On the **Active Directory Domain Services** page, choose **Next**.
12. On the **Confirm installation selections** page, choose **Install**.
13. Once the Active Directory binaries are installed, choose **Close**.
14. When Server Manager opens, look for a flag at the top next to the word **Manage**. When this flag turns yellow, the server is ready to be promoted.

15. Choose the yellow flag, and then choose **Promote this server to a domain controller**.
16. On the **Deployment Configuration** page, choose **Add a new forest**. In **Root domain name**, type **`example.local`**, and then choose **Next**.
17. On the **Domain Controller Options** page, do the following:

    - In both **Forest functional level** and **Domain functional level**, choose **Windows Server 2016**.
    - Under **Specify domain controller capabilities**, verify that both **Domain Name System (DNS) server** and **Global Catalog (GC)** are selected.
    - Type and then confirm a Directory Services Restore Mode (DSRM) password. Then choose **Next**.
18. On the **DNS Options** page, ignore the warning about delegation and choose **Next**.
19. On the **Additional options** page, make sure that **EXAMPLE** is listed as the NetBios domain name.
20. On the **Paths** page, leave the defaults, and then choose **Next**.
21. On **Review Options** page, choose **Next**. The server now checks to make sure all the prerequisites for the domain controller are satisfied. You may see some warnings displayed, but you can safely ignore them.
22. Choose **Install**. Once the installation is complete, the server reboots and then becomes a functional domain controller.

Configure your VPC

The following three procedures guide you through the steps to configure your VPC for connectivity with AWS.

To configure your VPC outbound rules

1. In the AWS Directory Service console, make a note of the AWS Managed Microsoft AD directory ID for corp.example.com that you previously created in the Base tutorial (p. 162).
2. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
3. In the navigation pane, choose **Security Groups**.
4. Search for your AWS Managed Microsoft AD directory ID. In the search results, select the item with the description **AWS created security group for d-*xxxxxx* directory controllers**.

    Note
    This security group was automatically created when you initially created your directory.
5. Choose the **Outbound Rules** tab under that security group. Choose **Edit**, choose **Add another rule**, and then add the following values:

    - For **Type**, choose **All Traffic**.
    - For **Destination**, type **`0.0.0.0/0`**.
    - Leave the rest of the settings at their defaults.
    - Select **Save**.

To verify kerberos preauthentication is enabled

1. On the **example.local** domain controller, open **Server Manager**.
2. On the **Tools** menu, choose **Active Directory Users and Computers**.
3. Navigate to the **Users** directory, right-click on any user and select **Properties**, and then choose the **Account** tab. In the **Account options** list, scroll down and ensure that **Do not require Kerberos preauthentication** is **not** selected.
4. Perform the same steps for the **corp.example.com** domain from the **corp.example.com-mgmt** instance.

To configure DNS conditional forwarders

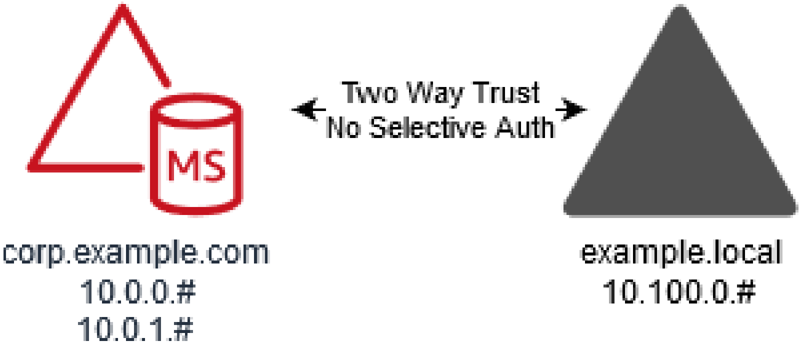
> **Note**
> A conditional forwarder is a DNS server on a network that is used to forward DNS queries according to the DNS domain name in the query. For example, a DNS server can be configured to forward all the queries it receives for names ending with widgets.example.com to the IP address of a specific DNS server or to the IP addresses of multiple DNS servers.

1. First you must get some information about your AWS Managed Microsoft AD.

   Sign in to the AWS Management Console and open the AWS Directory Service console at https://console.aws.amazon.com/directoryservicev2/.
2. In the navigation pane, choose **Directories**.
3. Select the **directory ID** of your AWS Managed Microsoft AD.
4. Take note of the fully qualified domain name (FQDN), **corp.example.com**, and the DNS addresses of your directory.
5. Now, return to your **example.local** domain controller, and then open **Server Manager**.
6. On the **Tools** menu, choose **DNS**.
7. In the console tree, expand the DNS server of the domain for which you are setting up the trust, and navigate to **Conditional Forwarders**.
8. Right-click **Conditional Forwarders**, and then choose **New Conditional Forwarder**.
9. In DNS domain, type `corp.example.com`.
10. Under **IP addresses of the master servers**, choose **<Click here to add ...>**, type the first DNS address of your AWS Managed Microsoft AD directory (which you made note of in the previous procedure), and then press **Enter**. Do the same for the second DNS address. After typing the DNS addresses, you might get a "timeout" or "unable to resolve" error. You can generally ignore these errors.
11. Select the **Store this conditional forwarder in Active Directory, and replicate as follows** check box. In the drop-down menu, choose **All DNS servers in this Forest**, and then choose **OK**.

Step 2: Create the trusts

In this section, you create two separate forest trusts. One trust is created from the Active Directory domain on your EC2 instance and the other from your AWS Managed Microsoft AD in AWS.



To create the trust from your EC2 domain to your AWS Managed Microsoft AD

1. Log into **example.local**.
2. Open **Server Manager** and in the console tree choose **DNS**. Take note of the IPv4 address listed for the server. You will need this in the next procedure when you create a conditional forwarder from **corp.example.com** to the **example.local** directory.

3. In the **Tools** menu, choose **Active Directory Domains and Trusts**.
4. In the console tree, right-click **example.local** and then choose **Properties**.
5. On the **Trusts** tab, choose **New Trust**, and then choose **Next**.
6. On the **Trust Name** page, type `corp.example.com`, and then choose **Next**.
7. On the **Trust Type** page, choose **Forest trust**, and then choose **Next**.

   Note
   AWS Managed Microsoft AD also supports external trusts. However, for the purposes of this tutorial, you will create a two-way forest trust.
8. On the **Direction of Trust** page, choose **Two-way**, and then choose **Next**.

   Note
   If you decide later to try this with a one-way trust instead, ensure that the trust directions are setup correctly (Outgoing on trusting domain, Incoming on trusted domain). For general information, see Understanding trust direction on Microsoft's website.
9. On the **Sides of Trust** page, choose **This domain only**, and then choose **Next**.
10. On the **Outgoing Trust Authentication Level** page, choose **Forest-wide authentication**, and then choose **Next**.

    Note
    Although **Selective authentication** in an option, for the simplicity of this tutorial we recommend that you do not enable it here. When configured it restricts access over an external or forest trust to only those users in a trusted domain or forest who have been explicitly given authentication permissions to computer objects (resource computers) residing in the trusting domain or forest. For more information, see Configuring selective authentication settings.
11. On the **Trust Password** page, type the trust password twice, and then choose **Next**. You will use this same password in the next procedure.
12. On the **Trust Selections Complete** page, review the results, and then choose **Next**.
13. On the **Trust Creation Complete** page, review the results, and then choose **Next**.
14. On the **Confirm Outgoing Trust** page, choose **No, do not confirm the outgoing trust**. Then choose **Next**
15. On the **Confirm Incoming Trust** page, choose **No, do not confirm the incoming trust**. Then choose **Next**
16. On the **Completing the New Trust Wizard** page, choose **Finish**.

Note
Trust relationships is a global feature of AWS Managed Microsoft AD. If you are using Multi-Region replication (p. 58), the following procedures must be performed in the Primary Region (p. 59). The changes will be applied across all replicated Regions automatically. For more information, see Global vs Regional features (p. 58).

To create the trust from your AWS Managed Microsoft AD to your EC2 domain

1. Open the AWS Directory Service console.
2. Choose the **corp.example.com** directory.
3. On the **Directory details** page, do one of the following:
   - If you have multiple Regions showing under **Multi-Region replication**, select the primary Region, and then choose the **Networking & security** tab. For more information, see Primary vs additional Regions (p. 59).
   - If you do not have any Regions showing under **Multi-Region replication**, choose the **Networking & security** tab.
4. In the **Trust relationships** section, choose **Actions**, and then select **Add trust relationship**.

5. In the **Add a trust relationship** dialog box, do the following:

   - Under **Trust type** select **Forest trust**.

     Note
     Make sure that the **Trust type** you choose here matches the same trust type configured in the previous procedure (To create the trust from your EC2 domain to your AWS Managed Microsoft AD).
   - For **Existing or new remote domain name**, type **example.local**.
   - For **Trust password**, type the same password that you provided in the previous procedure.
   - Under **Trust direction**, select **Two-Way**.

     Note

     - If you decide later to try this with a one-way trust instead, ensure that the trust directions are setup correctly (Outgoing on trusting domain, Incoming on trusted domain). For general information, see Understanding trust direction on Microsoft's website.
     - Although **Selective authentication** in an option, for the simplicity of this tutorial we recommend that you do not enable it here. When configured it restricts access over an external or forest trust to only those users in a trusted domain or forest who have been explicitly given authentication permissions to computer objects (resource computers) residing in the trusting domain or forest. For more information, see Configuring selective authentication settings.
   - For **Conditional forwarder**, type the IP address of your DNS server in the **example.local** forest (which you noted in the previous procedure).

     Note
     A conditional forwarder is a DNS server on a network that is used to forward DNS queries according to the DNS domain name in the query. For example, a DNS server can be configured to forward all the queries it receives for names ending with widgets.example.com to the IP address of a specific DNS server or to the IP addresses of multiple DNS servers.
6. Choose **Add**.

Step 3: Verify the trust

In this section, you test whether the trusts were set up successfully between AWS and Active Directory on Amazon EC2.

To verify the trust

1. Open the AWS Directory Service console.
2. Choose the **corp.example.com** directory.
3. On the **Directory details** page, do one of the following:

   - If you have multiple Regions showing under **Multi-Region replication**, select the primary Region, and then choose the **Networking & security** tab. For more information, see Primary vs additional Regions (p. 59).
   - If you do not have any Regions showing under **Multi-Region replication**, choose the **Networking & security** tab.
4. In the **Trust relationships** section, select the trust relationship you just created.
5. Choose **Actions**, and then choose **Verify trust relationship**.

Once the verification has completed, you should see **Verified** displayed under the **Status** column.

Congratulations on completing this tutorial! You now have a fully functional multiforest Active Directory environment from which you can begin testing various scenarios. Additional test lab tutorials are planned in 2018, so check back on occasion to see what's new.

Troubleshooting AWS Managed Microsoft AD

The following can help you troubleshoot some common issues you might encounter when creating or using your directory.

Issues with Netlogon and secure channel communications

As a mitigation against CVE-2020-1472, Microsoft has released patching which modifies the way that Netlogon secure channel communications are processed by domain controllers. Since the introduction of these secure Netlogon changes, some Netlogon connections (servers, workstations, and trust validations) may not be accepted by your AWS Managed Microsoft AD.

To verify if your issue is related to Netlogon or secure channel communications, search your Amazon CloudWatch Logs for event IDs 5827 (for device authentication related issues) or 5828 (for AD trust validation related issues). For information about CloudWatch in AWS Managed Microsoft AD, see Enable log forwarding (p. 55).

For more information about the mitigation against CVE-2020-1472, see How to manage the changes in Netlogon secure channel connections associated with CVE-2020-1472 on Microsoft's website.

Password recovery

If a user forgets a password or is having trouble signing in to either your Simple AD or AWS Managed Microsoft AD directory, you can reset their password using either the AWS Management Console, Windows PowerShell or the AWS CLI.

For more information, see Reset a user password (p. 102).

Topics

DNS troubleshooting

You can audit your AWS Managed Microsoft AD DNS events, making it easier to identify and troubleshoot DNS issues. For example, if a DNS record is missing, you can use the DNS audit event log to help identify the root cause and fix the issue. You can also use DNS audit event logs to improve security by detecting and blocking requests from suspicious IP addresses.

To do that, you must be logged on with the **Admin** account or with an account that is a member of the **AWS Domain Name System Administrators** group. For more information about this group, see What gets created (p. 12).

To troubleshoot AWS Managed Microsoft AD DNS

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left navigation pane, choose **Instances**.
3. Locate an Amazon EC2 instance that is joined to your AWS Managed Microsoft AD directory. Select the instance and then choose **Connect**.
4. Open the **Event Viewer** located in the **Administrative Tools** folder.
5. In the **Event Viewer** window, choose **Action** and then choose **Connect to Another Computer**.
6. Select **Another computer**, type one of your AWS Managed Microsoft AD DNS servers name or IP address, and choose **OK**.
7. In the left pane, navigate to **Applications and Services Logs**>**Microsoft**>**Windows**>**DNS-Server**, and then select **Audit**.

Linux domain join errors

The following can help you troubleshoot some error messages you might encounter when joining an EC2 Linux instance to your AWS Managed Microsoft AD directory.

Linux instances unable to join domain or authenticate

Ubuntu 14.04, 16.04, and 18.04 instances *must* be reverse-resolvable in the DNS before a realm can work with Microsoft AD. Otherwise, you might encounter one of the following two scenarios:

Scenario 1: Ubuntu instances that are not yet joined to a realm

For Ubuntu instances that are attempting to join a realm, the `sudo realm join` command might not provide the required permissions to join the domain and might display the following error:

! Couldn't authenticate to active directory: SASL(-1): generic failure: GSSAPI Error: An invalid name was supplied (Success) adcli: couldn't connect to EXAMPLE.COM domain: Couldn't authenticate to active directory: SASL(-1): generic failure: GSSAPI Error: An invalid name was supplied (Success) ! Insufficient permissions to join the domain realm: Couldn't join realm: Insufficient permissions to join the domain

Scenario 2: Ubuntu instances that are joined to a realm

For Ubuntu instances that are already joined to a Microsoft AD domain, attempts to SSH into the instance using the domain credentials might fail with following errors:

$ ssh admin@EXAMPLE.COM@198.51.100

no such identity: /Users/username/.ssh/id_ed25519: No such file or directory

admin@EXAMPLE.COM@198.51.100's password:

Permission denied, please try again.

admin@EXAMPLE.COM@198.51.100's password:

If you log in to the instance with a public key and check `/var/log/auth.log`, you might see the following errors about being unable to find the user:

May 12 01:02:12 ip-192-0-2-0 sshd[2251]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=203.0.113.0

May 12 01:02:12 ip-192-0-2-0 sshd[2251]: pam_sss(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=203.0.113.0 user=admin@EXAMPLE.COM

```
May 12 01:02:12 ip-192-0-2-0 sshd[2251]: pam_sss(sshd:auth): received for user admin@EXAMPLE.COM: 10 (User not known to the underlying authentication module)

May 12 01:02:14 ip-192-0-2-0 sshd[2251]: Failed password for invalid user admin@EXAMPLE.COM from 203.0.113.0 port 13344 ssh2

May 12 01:02:15 ip-192-0-2-0 sshd[2251]: Connection closed by 203.0.113.0 [preauth]
```

However, `kinit` for the user still works. See this example:

```
ubuntu@ip-192-0-2-0:~$ kinit admin@EXAMPLE.COM Password for admin@EXAMPLE.COM: ubuntu@ip-192-0-2-0:~$ klist Ticket cache: FILE:/tmp/krb5cc_1000 Default principal: admin@EXAMPLE.COM
```

Workaround

The current recommended workaround for both of these scenarios is to disable reverse DNS in `/etc/krb5.conf` in the [libdefaults] section as shown below:

```
[libdefaults]
default_realm = EXAMPLE.COM
rdns = false
```

One-way trust authentication issue with seamless domain join

If you have a one-way outgoing trust established between your AWS Managed Microsoft AD and your on-premises AD, you might encounter an authentication issue when attempting to authenticate against the domain joined Linux instance using your trusted AD credentials with Winbind.

Errors

```
Jul 31 00:00:00 EC2AMAZ-LSMWqT sshd[23832]: Failed password for user@corp.example.com from xxx.xxx.xxx.xxx port 18309 ssh2

Jul 31 00:05:00 EC2AMAZ-LSMWqT sshd[23832]: pam_winbind(sshd:auth): getting password (0x00000390)

Jul 31 00:05:00 EC2AMAZ-LSMWqT sshd[23832]: pam_winbind(sshd:auth): pam_get_item returned a password

Jul 31 00:05:00 EC2AMAZ-LSMWqT sshd[23832]: pam_winbind(sshd:auth): request wbcLogonUser failed: WBC_ERR_AUTH_ERROR, PAM error: PAM_SYSTEM_ERR (4), NTSTATUS: **NT_STATUS_OBJECT_NAME_NOT_FOUND**, Error message was: The object name is not found.

Jul 31 00:05:00 EC2AMAZ-LSMWqT sshd[23832]: pam_winbind(sshd:auth): internal module error (retval = PAM_SYSTEM_ERR(4), user = 'CORP\user')
```

Workaround

To resolve this issue, you will need to comment out or remove a directive from the PAM module configuration file (`/etc/security/pam_winbind.conf`) using the following steps.

1. Open the `/etc/security/pam_winbind.conf` file in a text editor.

   ```
   sudo vim /etc/security/pam_winbind.conf
   ```

2. Comment out or remove the following directive **krb5_auth = yes**.

   ```
   [global]
   ```

```
cached_login = yes
krb5_ccache_type = FILE
#krb5_auth = yes
```

3. Stop the Winbind service, and then start it again.

```
service winbind stop or systemctl stop winbind
net cache flush
service winbind start or systemctl start winbind
```

Active Directory low available storage space

When your AWS Managed Microsoft AD is impaired due to Active Directory having low available storage space, immediate action is required to return the directory to an active state. The two most common causes of this impairment are covered in the sections below:

1. SYSVOL folder is storing more than essential group policy objects (p. 178)
2. Active Directory database has filled the volume (p. 178)

For pricing information about AWS Managed Microsoft AD storage, see AWS Directory Service Pricing.

SYSVOL folder is storing more than essential group policy objects

A common cause of this impairment is due to storing non-essential files for Group Policy processing in the SYSVOL folder. These non-essential files could be EXEs, MSIs, or any other file that is not essential for Group Policy to process. The essential objects for Group Policy to process are Group Policy Objects, Logon/off Scripts, and the Central Store for Group Policy objects. Any non-essential files should be stored on a file server(s) other than your AWS Managed Microsoft AD domain controllers.

If files for Group Policy Software Installation are needed you should use a file server to store those installation files. If you would prefer to not self manage a file server, AWS provides a managed file server option, Amazon FSx.

To remove any unnecessary files you can access the SYSVOL share via it's universal naming convention (UNC) path. For example, if your domain's fully qualified domain name (FQDN) is example.com, the UNC path for the SYSVOL would be "\\example.local\SYSVOL\example.local\". Once you locate and remove objects that are not essential for Group Policy to process the directory, it should return to an Active state within 30 minutes. If after 30 minutes the directory is not active, please contact AWS Support.

Storing only essential Group Policy files in your SYSVOL share will ensure that you will not impair your directory due to SYSVOL bloat.

Active Directory database has filled the volume

A common cause of this impairment is due to the Active Directory database filling the volume. To verify if this is the case, you can review the **total** count of objects in your directory. We bold the word **total** to ensure that you understand **deleted** objects still count towards the total number of objects in a directory.

By default AWS Managed Microsoft AD keeps items in the AD Recycling Bin for 180 days before they become a Recycled-Object. Once an object becomes a Recycled-Object (tombstoned), it is retained for another 180 days before it is finally purged from the directory. So when an object is deleted it exists in the directory database for 360 day before it is purged. This is why the total number of objects need to be evaluated.

For more details on AWS Managed Microsoft AD supported object counts, see AWS Directory Service Pricing.

To get the total number of objects in a directory that includes the deleted objects, you can run the following PowerShell command from a domain joined Windows instance. For steps how to setup a management instance, see Manage users and groups in AWS Managed Microsoft AD (p. 100).

```
Get-ADObject -Filter * -IncludeDeletedObjects | Measure-Object -Property 'Count' | Select-
Object -Property 'Count'
```

Below is an example output from the above command:

```
Count
10000
```

If the total count is above the supported object count for your directory size listed in the note above, you have exceeded the capacity of your directory.

Below are the options to resolve this impairment:

1. Cleanup AD
   a. Delete any unwanted AD objects.
   b. Remove any objects that are not wanted from the AD Recycling Bin. Note this is destructive and the only way to recover those deleted objects will be to perform a restore of the directory.
   c. The following command will remove all deleted objects from the AD Recycling Bin.

      Important
      Use this command with extreme caution as this is a destructive command and the only way to recover those deleted objects will be to perform a restore of the directory.

      ```
      $DomainInfo = Get-ADDomain
      $BaseDn = $DomainInfo.DistinguishedName
      $NetBios = $DomainInfo.NetBIOSName
      $ObjectsToRemove = Get-ADObject -Filter { isDeleted -eq $true } -
      IncludeDeletedObjects -SearchBase "CN=Deleted Objects,$BaseDn" -Properties
       'LastKnownParent','DistinguishedName','msDS-LastKnownRDN' | Where-Object
       { ($_.LastKnownParent -Like "*OU=$NetBios,$BaseDn") -or ($_.LastKnownParent -Like '*
      \0ADEL:*') }
      ForEach ($ObjectToRemove in $ObjectsToRemove) { Remove-ADObject -Identity
       $ObjectToRemove.DistinguishedName -IncludeDeletedObjects }
      ```
   d. Open a case with AWS Support to request that AWS Directory Service reclaims the free space.
2. If your directory type is Standard Edition Open a case with AWS Support requesting your directory be upgraded to Enterprise Edition. This will also increase the cost of your directory. For pricing information, see AWS Directory Service Pricing.

In AWS Managed Microsoft AD, members of the **AWS Delegated Deleted Object Lifetime Administrators** group have the ability to modify the `msDS-DeletedObjectLifetime` attribute which sets the amount of time in days that deleted objects are kept in the AD Recycling Bin before they become Recycled-Objects.

> **Note**
> This is an advanced topic. If configured inappropriately, it can result in data loss. We highly recommend that you first review The AD Recycle Bin: Understanding, Implementing, Best Practices, and Troubleshooting to get a better understanding of these processes.

The ability to change the `msDS-DeletedObjectLifetime` attribute value to a lower number can help ensure your object count does not exceed supported levels. The lowest valid value this attribute can be set to is 2 days. Once that value has exceeded you will no longer be able to recover the deleted object using the AD Recycling Bin. It will require restoring your directory from a snapshot to recover the object(s). For more information, see Snapshot or restore your directory (p. 128). **Any restore from snapshot can result in data loss as they are a point in time.**

To change Deleted Object Lifetime of your directory run the following command:

> **Note**
> If you run the command as is, it will set the Deleted Object Lifetime attribute value to 30 days. If you would like to make it longer or shorter replace “30” with whatever number you prefer. However, we recommend that you go no higher than the default number of 180.

```
$DeletedObjectLifetime = 30
$DomainInfo = Get-ADDomain
$BaseDn = $DomainInfo.DistinguishedName
Set-ADObject -Identity "CN=Directory Service,CN=Windows NT,CN=Services,CN=Configuration,
$BaseDn" -Partition "CN=Configuration,$BaseDn" -Replace:@{"msDS-DeletedObjectLifetime" =
 $DeletedObjectLifetime}
```

Schema extension errors

The following can help you troubleshoot some error messages you might encounter when extending the schema for your AWS Managed Microsoft AD directory.

Referral

Error

Add error on entry starting on line 1: Referral The server side error is: 0x202b A referral was returned from the server. The extended server error is: 0000202B: RefErr: DSID-0310082F, data 0, 1 access points \tref 1: ‘example.com’ Number of Objects Modified: 0

Troubleshooting

Ensure that all of the distinguished name fields have the correct domain name. In the example above, `DC=example,dc=com` should be replaced with the `DistinguishedName` shown by the cmdlet `Get-ADDomain`.

Unable to read import file

Error

Unable to read the import file. Number of Objects Modified: 0

Troubleshooting

The imported LDIF file is empty (0 bytes). Ensure the correct file was uploaded.

Syntax error

Error

There is a syntax error in the input file Failed on line 21. The last token starts with 'q'. Number of Objects Modified: 0

Troubleshooting

The text on line 21 is not formatted correctly. The first letter of the invalid text is `A`. Update line 21 with valid LDIF syntax. For more information about how to format the LDIF file, see Step 1: Create your LDIF file (p. 122).

Attribute or value exists

Error

Add error on entry starting on line 1: Attribute Or Value Exists The server side error is: 0x2083 The specified value already exists. The extended server error is: 00002083: AtrErr: DSID-03151830, #1: \t0: 00002083: DSID-03151830, problem 1006 (ATT_OR_VALUE_EXISTS), data 0, Att 20019 (mayContain):len 4 Number of Objects Modified: 0

Troubleshooting

The schema change has already been applied.

No such attribute

Error

Add error on entry starting on line 1: No Such Attribute The server side error is: 0x2085 The attribute value cannot be removed because it is not present on the object. The extended server error is: 00002085: AtrErr: DSID-03152367, #1: \t0: 00002085: DSID-03152367, problem 1001 (NO_ATTRIBUTE_OR_VAL), data 0, Att 20019 (mayContain):len 4 Number of Objects Modified: 0

Troubleshooting

The LDIF file is trying to remove an attribute from a class, but that attribute is currently not attached to the class. Schema change was probably already applied.

Error

Add error on entry starting on line 41: No Such Attribute 0x57 The parameter is incorrect. The extended server error is: 0x208d Directory object not found. The extended server error is: "00000057: LdapErr: DSID-0C090D8A, comment: Error in attribute conversion operation, data 0, v2580" Number of Objects Modified: 0

Troubleshooting

The attribute listed on line 41 is incorrect. Double-check the spelling.

No such object

Error

Add error on entry starting on line 1: No Such Object The server side error is: 0x208d Directory object not found. The extended server error is: 0000208D: NameErr: DSID-03100238, problem 2001 (NO_OBJECT), data 0, best match of: 'CN=Schema,CN=Configuration,DC=example,DC=com' Number of Objects Modified: 0

Troubleshooting

The object referenced by the distinguished name (DN) does not exist.

Trust creation status reasons

When trust creation fails, the status message contains additional information. Here's some help understanding what those messages mean.

Access is denied

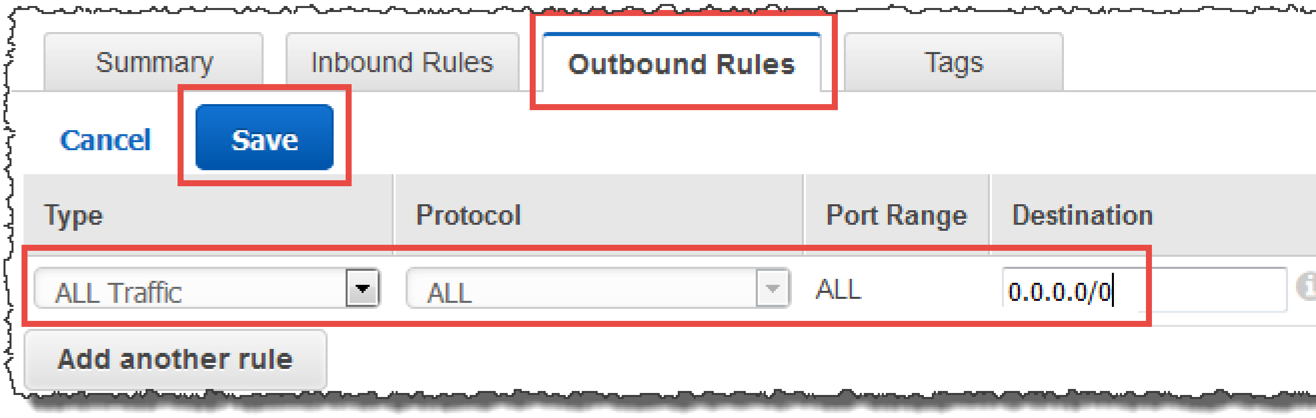
Access was denied when trying to create the trust. Either the trust password is incorrect or the remote domain's security settings do not allow a trust to be configured. To resolve this problem, try the following:

- Verify that you are using the same trust password that you used when creating the corresponding trust on the remote domain.
- Verify that your domain security settings allow for trust creation.
- Verify that your local security policy is set correctly. Specifically check `Local Security Policy > Local Policies > Security Options > Network access: Named Pipes that can be accessed anonymously` and ensure that it contains at least the following three named pipes:
  - netlogon
  - samr
  - lsarpc
- Verify that the above named pipes exist as the value(s) on the **NullSessionPipes** registry key which is in the registry path **HKLM\SYSTEM\CurrentControlSet\services\LanmanServer\Parameters**. These values must be inserted on separated rows.

  Note
  By default, `Network access: Named Pipes that can be accessed anonymously` is not set and will display `Not Defined`. This is normal, as the domain controller's effective default settings for `Network access: Named Pipes that can be accessed anonymously` is `netlogon`, `samr`, `lsarpc`.

The specified domain name does not exist or could not be contacted

To resolve this problem, ensure the security group settings for your domain and access control list (ACL) for your VPC are correct and you have accurately entered the information for your conditional forwarder. AWS configures the security group to open only the ports that are required for Active Directory communications. In the default configuration, the security group accepts traffic to these ports from any IP address. Outbound traffic is restricted to the Security group. You will need to update the outbound rule on the security group to allow traffic to your on premise network. For more information about security requirements, please see Step 2: Prepare your AWS Managed Microsoft AD (p. 112).



Type	Protocol	Port Range	Destination
ALL Traffic	ALL	ALL	0.0.0.0/0

If the DNS servers for the networks of the other directories use public (non-RFC 1918) IP addresses, you will need add an IP route on the directory from the Directory Services Console to the DNS Servers. For more information, see Create, verify, or delete a trust relationship (p. 106) and Prerequisites (p. 104).

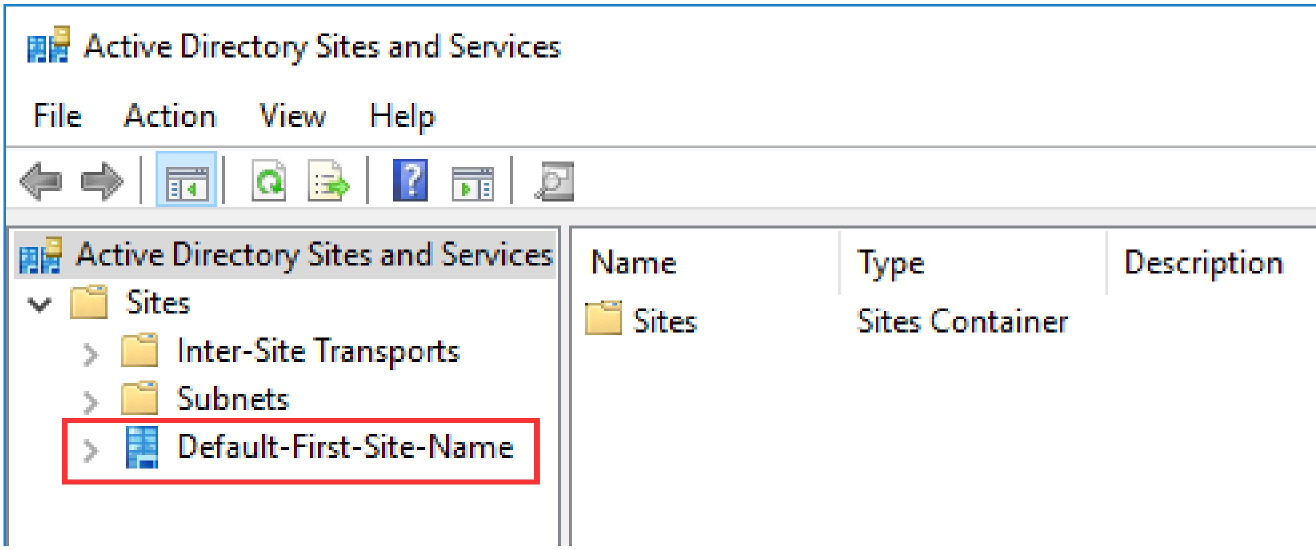
The Internet Assigned Numbers Authority (IANA) has reserved the following three blocks of the IP address space for private internets:

- 10.0.0.0 - 10.255.255.255 (10/8 prefix)
- 172.16.0.0 - 172.31.255.255 (172.16/12 prefix)
- 192.168.0.0 - 192.168.255.255 (192.168/16 prefix)

For more information, see https://tools.ietf.org/html/rfc1918.

Verify that the **Default AD Site Name** for your AWS Managed Microsoft AD matches the **Default AD Site Name** in your on-premises infrastructure. The computer determines the site name using a domain of which the computer is a member, not the user's domain. Renaming the site to match the closest on-premises ensures the DC locator will use a domain controller from the closest site. If this does not solve the issue, it is possible that information from a previously created conditional forwarder has been cached, preventing the creation of a new trust. Wait several minutes, and then try creating the trust and conditional forwarder again.

For more information about how this works, see Domain Locator Across a Forest Trust on Microsoft's website.



The operation could not be performed on this domain

To resolve this, ensure both domains / directories do not have overlapping NETBIOS name(s). If the domains / directories do have overlapping NETBIOS names, recreate one of them with a different NETBIOS name, and then try again.

Trust creation is failing because of the error "Required and valid domain name"

DNS names can contain only alphabetical characters (A-Z), numeric characters (0-9), the minus sign (-), and a period (.). Period characters are allowed only when they are used to delimit the components of domain style names. Also, consider the following:

- AWS Managed Microsoft AD does not support trusts with Single label domains. For more information, see Microsoft support for Single Label Domains.
- According to RFC 1123 (https://tools.ietf.org/html/rfc1123), the only characters that can be used in DNS labels are "A" to "Z", "a" to "z", "0" to "9", and a hyphen ("-"). A period [.] is also used in DNS names, but only between DNS labels and at the end of an FQDN.
- According to RFC 952 (https://tools.ietf.org/html/rfc952), a "name" (Net, Host, Gateway, or Domain name) is a text string up to 24 characters drawn from the alphabet (A-Z), digits (0-9), minus sign (-), and period (.). Note that periods are only allowed when they serve to delimit components of "domain style names".

For more information, see Complying with Name Restrictions for Hosts and Domains on Microsoft's website.

General tools for testing trusts

The following are tools that can be used to troubleshoot various trust related issues.

AWS Systems Manager Automation troubleshooting tool

Support Automation Workflows (SAW) leverage AWS Systems Manager Automation to provide you with a predefined runbook for AWS Directory Service. The AWSSupport-TroubleshootDirectoryTrust runbook tool helps you diagnose common trust creation issues between AWS Managed Microsoft AD and an on-premises Microsoft Active Directory.

DirectoryServicePortTest tool

The DirectoryServicePortTest testing tool can be helpful when troubleshooting trust creation issues between AWS Managed Microsoft AD and on-premises Active Directory. For an example on how the tool can be used, see Test your AD Connector (p. 191).

NETDOM and NLTEST tool

Administrators can use both the **Netdom** and **Nltest** command-line tools to find, display, create, remove and manage trusts. These tools communicate directly with the LSA authority on a domain controller. For an example on how to use these tools, see Netdom and NLTEST on Microsoft's website.

Packet capture tool

You can use the built-in Windows package capture utility to investigate and troubleshoot a potential network issue. For more information, see Capture a Network Trace without installing anything.

Active Directory Connector

AD Connector is a directory gateway with which you can redirect directory requests to your on-premises Microsoft Active Directory without caching any information in the cloud. AD Connector comes in two sizes, small and large. You can spread application loads across multiple AD Connectors to scale to your performance needs. There are no enforced user or connection limits.

AD Connector does not support AD transitive trusts. AD Connectors and your on-premises AD domains have a 1-to-1 relationship. That is, for each on-premises domain, including child domains in an AD forest that you want to authenticate against, you must create a unique AD Connector.

> **Note**
> AD Connector cannot be shared with other AWS accounts. If this is a requirement, consider using AWS Managed Microsoft AD to Share your directory (p. 63). AD Connector is also not multi-VPC aware, which means that AWS applications like WorkSpaces are required to be provisioned into the same VPC as your AD Connector.

Once set up, AD Connector offers the following benefits:

- Your end users and IT administrators can use their existing corporate credentials to log on to AWS applications such as WorkSpaces, Amazon WorkDocs, or Amazon WorkMail.
- You can manage AWS resources like Amazon EC2 instances or Amazon S3 buckets through IAM role-based access to the AWS Management Console.
- You can consistently enforce existing security policies (such as password expiration, password history, and account lockouts) whether users or IT administrators are accessing resources in your on-premises infrastructure or in the AWS Cloud.
- You can use AD Connector to enable multi-factor authentication by integrating with your existing RADIUS-based MFA infrastructure to provide an additional layer of security when users access AWS applications.

Continue reading the topics in this section to learn how to connect to a directory and make the most of AD Connector features.

Topics

Getting started with AD Connector

With AD Connector you can connect AWS Directory Service to your existing enterprise directory. When connected to your existing directory, all of your directory data remains on your domain controllers. AWS Directory Service does not replicate any of your directory data.

Topics

AD Connector prerequisites

To connect to your existing directory with AD Connector, you need the following:

VPC

Set up a VPC with the following:

- At least two subnets. Each of the subnets must be in a different Availability Zone.
- The VPC must be connected to your existing network through a virtual private network (VPN) connection or AWS Direct Connect.
- The VPC must have default hardware tenancy.

AWS Directory Service uses a two VPC structure. The EC2 instances which make up your directory run outside of your AWS account, and are managed by AWS. They have two network adapters, `ETH0` and `ETH1`. `ETH0` is the management adapter, and exists outside of your account. `ETH1` is created within your account.

The management IP range of your directory's `ETH0` network is chosen programmatically to ensure it does not conflict with the VPC where your directory is deployed. This IP range can be in either of the following pairs (as Directories run in two subnets):

- 10.0.1.0/24 & 10.0.2.0/24
- 192.168.1.0/24 & 192.168.2.0/24

We avoid conflicts by checking the first octet of the `ETH1` CIDR. If it starts with a 10, then we choose a 192.168.0.0/16 VPC with 192.168.1.0/24 and 192.168.2.0/24 subnets. If the first octet is anything else other than a 10 we choose a 10.0.0.0/16 VPC with 10.0.1.0/24 and 10.0.2.0/24 subnets.

The selection algorithm does not include routes on your VPC. It is therefore possible to have an IP routing conflict result from this scenario.

The VPC must **not** be configured with the following VPC endpoint(s):

- Route53 VPC endpoints that include DNS conditional overrides for *.amazonaws.com which resolve to non public AWS IP addresses
- CloudWatch VPC endpoint
- Systems Manager VPC endpoint
- Security Token Service VPC endpoint

For more information, see the following topics in the *Amazon VPC User Guide*:

- What is Amazon VPC?
- Subnets in your VPC
- Adding a Hardware Virtual Private Gateway to Your VPC

For more information about AWS Direct Connect, see the AWS Direct Connect User Guide.

Existing Active Directory

You'll need to connect to an existing network with an Active Directory domain.

Note
AD Connector does not support Single Label Domains.

The functional level of this domain must be `Windows Server 2003` or higher. AD Connector also supports connecting to a domain hosted on an Amazon EC2 instance.

> **Note**
> AD Connector does not support Read-only domain controllers (RODC) when used in combination with the Amazon EC2 domain-join feature.

Service account

You must have credentials for a service account in the existing directory which has been delegated the following privileges:

- Read users and groups - Required
- Join computers to the domain - Required only when using Seamless Domain Join and WorkSpaces
- Create computer objects - Required only when using Seamless Domain Join and WorkSpaces

For more information, see Delegate privileges to your service account (p. 189).

> **Note**
> AD Connector uses Kerberos for authentication and authorization of AWS applications. LDAP is only used for user and group object lookups (read operations). With the LDAP transactions, nothing is mutable and credentials are not passed in clear text. Authentication is handled by an AWS internal service, which uses Kerberos tickets to perform LDAP operations as a user.

User permissions

All Active Directory users must have permissions to read their own attributes. Specifically the following attributes:

- GivenName
- SurName
- Mail
- SamAccountName
- UserPrincipalName
- UserAccountControl
- MemberOf

By default, Active Directory users do have read permission to these attributes. However, Administrators can alter these permissions over time so you might want to verify your users have these read permissions prior to setting up AD Connector for the first time.

IP addresses

Get the IP addresses of two DNS servers or domain controllers in your existing directory.

AD Connector obtains the `_ldap._tcp.<DnsDomainName>` and `_kerberos._tcp.<DnsDomainName>` SRV records from these servers when connecting to your directory, so these servers must contain these SRV records. The AD Connector attempts to find a common domain controller that will provide both LDAP and Kerberos services, so these SRV records must include at least one common domain controller. For more information about SRV records, go to SRV Resource Records on Microsoft TechNet.

Ports for subnets

For AD Connector to redirect directory requests to your existing Active Directory domain controllers, the firewall for your existing network must have the following ports open to the CIDRs for both subnets in your Amazon VPC.

- TCP/UDP 53 - DNS
- TCP/UDP 88 - Kerberos authentication
- TCP/UDP 389 - LDAP

These are the minimum ports that are needed before AD Connector can connect to your directory. Your specific configuration may require additional ports be open.

> **Note**
> If the DNS servers or Domain Controller servers for your existing Active Directory Domain are within the VPC, the security groups associated with those servers must have the above ports open to the CIDRs for both subnets in the VPC.

For additional port requirements, see AD and AD DS Port Requirements on Microsoft TechNet.

Kerberos preauthentication

Your user accounts must have Kerberos preauthentication enabled. For detailed instructions on how to enable this setting, see Ensure that Kerberos pre-authentication is enabled (p. 110). For general information about this setting, go to Preauthentication on Microsoft TechNet.

Encryption types

AD Connector supports the following encryption types when authenticating via Kerberos to your Active Directory domain controllers:

- AES-256-HMAC
- AES-128-HMAC
- RC4-HMAC

AWS Single Sign-On prerequisites

If you plan to use AWS Single Sign-On (AWS SSO) with AD Connector, you need to ensure that the following are true:

- Your AD Connector is set up in your AWS organization's management account.
- Your instance of AWS SSO is in the same Region where your AD Connector is set up.

For more information, see AWS SSO prerequisites in the AWS Single Sign-On User Guide.

Multi-factor authentication prerequisites

To support multi-factor authentication with your AD Connector directory, you need the following:

- A Remote Authentication Dial-In User Service (RADIUS) server in your existing network that has two client endpoints. The RADIUS client endpoints have the following requirements:
  - To create the endpoints, you need the IP addresses of the AWS Directory Service servers. These IP addresses can be obtained from the **Directory IP Address** field of your directory details.
  - Both RADIUS endpoints must use the same shared secret code.
- Your existing network must allow inbound traffic over the default RADIUS server port (1812) from the AWS Directory Service servers.
- The usernames between your RADIUS server and your existing directory must be identical.

For more information about using AD Connector with MFA, see Enable multi-factor authentication for AD Connector (p. 199).

Delegate privileges to your service account

To connect to your existing directory, you must have the credentials for your AD Connector service account in the existing directory that has been delegated certain privileges. While members of the **Domain Admins** group have sufficient privileges to connect to the directory, as a best practice, you should use a service account that only has the minimum privileges necessary to connect to the directory.

The following procedure demonstrates how to create a new group called `Connectors`, delegate the necessary privileges that are needed to connect AWS Directory Service to this group, and then add a new service account to this group.

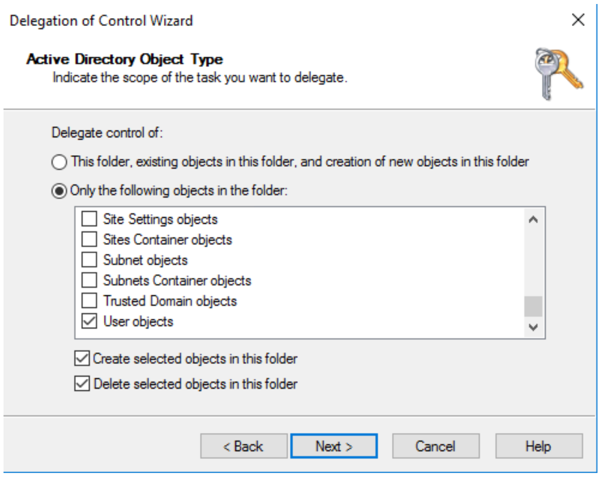
This procedure must be performed on a machine that is joined to your directory and has the **Active Directory User and Computers** MMC snap-in installed. You must also be logged in as a domain administrator.

To delegate privileges to your service account

1. Open **Active Directory User and Computers** and select your domain root in the navigation tree.
2. In the list in the left-hand pane, right-click **Users**, select **New**, and then select **Group**.
3. In the **New Object - Group** dialog box, enter the following and click **OK**.

Field	Value/Selection
Group name	`Connectors`
Group scope	**Global**
Group type	**Security**

4. In the **Active Directory User and Computers** navigation tree, select your domain root. In the menu, select **Action**, and then **Delegate Control**. If your AD Connector is connected to AWS Managed Microsoft AD, you will not have access to delegate control at the domain root level. In this case, to delegate control, select the OU under your directory OU where your computer objects will be created.
5. On the **Delegation of Control Wizard** page, click **Next**, then click **Add**.
6. In the **Select Users, Computers, or Groups** dialog box, enter `Connectors` and click **OK**. If more than one object is found, select the `Connectors` group created above. Click **Next**.
7. On the **Tasks to Delegate** page, select **Create a custom task to delegate**, and then choose **Next**.
8. Select **Only the following objects in the folder**, and then select **Computer objects** and **User objects**.
9. Select **Create selected objects in this folder** and **Delete selected objects in this folder**. Then choose **Next**.



Delegation of Control Wizard

Active Directory Object Type
Indicate the scope of the task you want to delegate.

Delegate control of:

○ This folder, existing objects in this folder, and creation of new objects in this folder

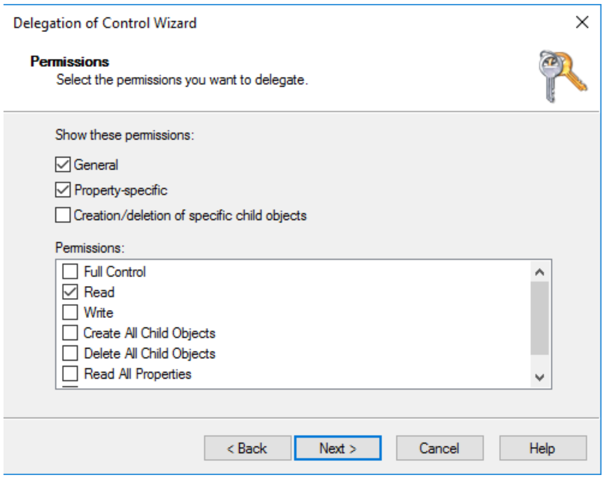
● Only the following objects in the folder:

□ Site Settings objects
□ Sites Container objects
□ Subnet objects
□ Subnets Container objects
□ Trusted Domain objects
☑ User objects

☑ Create selected objects in this folder
☑ Delete selected objects in this folder

10. Select **Read**, and then choose **Next**.

    Note
    If you will be using Seamless Domain Join or WorkSpaces, you must also enable **Write** permissions so that the Active Directory can create computer objects.



Delegation of Control Wizard

Permissions
Select the permissions you want to delegate.

Show these permissions:

☑ General
☑ Property-specific
☐ Creation/deletion of specific child objects

Permissions:

☐ Full Control
☑ Read
☐ Write
☐ Create All Child Objects
☐ Delete All Child Objects
☐ Read All Properties

11. Verify the information on the **Completing the Delegation of Control Wizard** page, and click **Finish**.
12. Create a user account with a strong password and add that user to the `Connectors` group. This user will be known as your AD Connector service account and since it is now a member of the `Connectors` group it now has sufficient privileges to connect AWS Directory Service to the directory.

Test your AD Connector

For AD Connector to connect to your existing directory, the firewall for your existing network must have certain ports open to the CIDRs for both subnets in the VPC. To test if these conditions are met, perform the following steps:

To test the connection

1. Launch a Windows instance in the VPC and connect to it over RDP. The instance must be a member of your existing domain. The remaining steps are performed on this VPC instance.
2. Download and unzip the DirectoryServicePortTest test application. The source code and Visual Studio project files are included so you can modify the test application if desired.

   Note
   This script is not supported on Windows Server 2003 or older operating systems.
3. From a Windows command prompt, run the **DirectoryServicePortTest** test application with the following options:

   Note
   The DirectoryServicePortTest test application can only be used when the domain and forest functional levels are set to Windows Server 2012 R2 and below.

   ```
   DirectoryServicePortTest.exe -d <domain_name> -ip <server_IP_address> -tcp "53,88,389"
    -udp "53,88,389"
   ```

   `<domain_name>`

   The fully qualified domain name. This is used to test the forest and domain functional levels. If you exclude the domain name, the functional levels won't be tested.

   `<server_IP_address>`

   The IP address of a domain controller in your existing domain. The ports will be tested against this IP address. If you exclude the IP address, the ports won't be tested.

This test app determines if the necessary ports are open from the VPC to your domain, and also verifies the minimum forest and domain functional levels.

The output will be similar to the following:

```
Testing forest functional level.
Forest Functional Level = Windows2008R2Forest : PASSED

Testing domain functional level.
Domain Functional Level = Windows2008R2Domain : PASSED

Testing required TCP ports to <server_IP_address>:
Checking TCP port 53: PASSED
Checking TCP port 88: PASSED
Checking TCP port 389: PASSED

Testing required UDP ports to <server_IP_address>:
Checking UDP port 53: PASSED
Checking UDP port 88: PASSED
Checking UDP port 389: PASSED
```

The following is the source code for the **DirectoryServicePortTest** application.

```
/*
   Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.

   This file is licensed under the Apache License, Version 2.0 (the "License").
   You may not use this file except in compliance with the License. A copy of
   the License is located at

    http://aws.amazon.com/apache2.0/

   This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR
   CONDITIONS OF ANY KIND, either express or implied. See the License for the
   specific language governing permissions and limitations under the License.
*/
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Net;
using System.Net.Sockets;
using System.Text;
using System.Threading.Tasks;
using System.DirectoryServices.ActiveDirectory;
using System.Threading;
using System.DirectoryServices.AccountManagement;
using System.DirectoryServices;
using System.Security.Authentication;
using System.Security.AccessControl;
using System.Security.Principal;

namespace DirectoryServicePortTest
{
    class Program
    {
        private static List<int> _tcpPorts;
        private static List<int> _udpPorts;

        private static string _domain = "";
        private static IPAddress _ipAddr = null;
        static void Main(string[] args)
        {
            if (ParseArgs(args))
            {
                try
                {
                    if (_domain.Length > 0)
                    {
                        try
                        {
                            TestForestFunctionalLevel();

                            TestDomainFunctionalLevel();
                        }
                        catch (ActiveDirectoryObjectNotFoundException)
                        {
                            Console.WriteLine("The domain {0} could not be found.\n",
_domain);
                        }
                    }

                    if (null != _ipAddr)
                    {
                        if (_tcpPorts.Count > 0)
                        {
                            TestTcpPorts(_tcpPorts);
                        }

                        if (_udpPorts.Count > 0)
                        {
                            TestUdpPorts(_udpPorts);
                        }
                    }
                }
                catch (AuthenticationException ex)
                {
                    Console.WriteLine(ex.Message);
                }
            }
            else
            {
                PrintUsage();
            }

            Console.Write("Press <enter> to continue.");
            Console.ReadLine();
        }

        static void PrintUsage()
        {
            string currentApp =
Path.GetFileName(System.Reflection.Assembly.GetExecutingAssembly().Location);
            Console.WriteLine("Usage: {0} \n-d <domain> \n-ip \"<server IP address>\"
\n[-tcp \"<tcp_port1>,<tcp_port2>,etc\"] \n[-udp \"<udp_port1>,<udp_port2>,etc\"]",
currentApp);
        }

        static bool ParseArgs(string[] args)
        {
            bool fReturn = false;
            string ipAddress = "";

            try
            {
                _tcpPorts = new List<int>();
                _udpPorts = new List<int>();

                for (int i = 0; i < args.Length; i++)
                {
                    string arg = args[i];

                    if ("-tcp" == arg | "/tcp" == arg)
                    {
                        i++;
                        string portList = args[i];
                        _tcpPorts = ParsePortList(portList);
                    }

                    if ("-udp" == arg | "/udp" == arg)
                    {
                        i++;
                        string portList = args[i];
                        _udpPorts = ParsePortList(portList);
                    }

                    if ("-d" == arg | "/d" == arg)
                    {
                        i++;
                        _domain = args[i];
                    }

                    if ("-ip" == arg | "/ip" == arg)
                    {
                        i++;
                        ipAddress = args[i];
                    }
                }
            }
            catch (ArgumentOutOfRangeException)
            {
                return false;
            }

            if (_domain.Length > 0 || ipAddress.Length > 0)
            {
                fReturn = true;
            }

            if (ipAddress.Length > 0)
            {
                _ipAddr = IPAddress.Parse(ipAddress);
            }

            return fReturn;
        }

        static List<int> ParsePortList(string portList)
        {
            List<int> ports = new List<int>();

            char[] separators = {',', ';', ':'};

            string[] portStrings = portList.Split(separators);
            foreach (string portString in portStrings)
            {
                try
                {
                    ports.Add(Convert.ToInt32(portString));
                }
                catch (FormatException)
                {
                }
            }

            return ports;
        }

        static void TestForestFunctionalLevel()
        {
            Console.WriteLine("Testing forest functional level.");

            DirectoryContext dirContext = new DirectoryContext(DirectoryContextType.Forest,
_domain, null, null);
            Forest forestContext = Forest.GetForest(dirContext);

            Console.Write("Forest Functional Level = {0} : ", forestContext.ForestMode);

            if (forestContext.ForestMode >= ForestMode.Windows2003Forest)
            {
                Console.WriteLine("PASSED");
            }
            else
            {
                Console.WriteLine("FAILED");
            }

            Console.WriteLine();
        }

        static void TestDomainFunctionalLevel()
        {
            Console.WriteLine("Testing domain functional level.");

            DirectoryContext dirContext = new DirectoryContext(DirectoryContextType.Domain,
_domain, null, null);
            Domain domainObject = Domain.GetDomain(dirContext);

            Console.Write("Domain Functional Level = {0} : ", domainObject.DomainMode);

            if (domainObject.DomainMode >= DomainMode.Windows2003Domain)
            {
                Console.WriteLine("PASSED");
            }
            else
            {
                Console.WriteLine("FAILED");
            }

            Console.WriteLine();
        }

        static List<int> TestTcpPorts(List<int> portList)
        {
            Console.WriteLine("Testing TCP ports to {0}:", _ipAddr.ToString());

            List<int> failedPorts = new List<int>();

            foreach (int port in portList)
            {
                Console.Write("Checking TCP port {0}: ", port);

                TcpClient tcpClient = new TcpClient();

                try
                {
                    tcpClient.Connect(_ipAddr, port);
                    tcpClient.Close();
                    Console.WriteLine("PASSED");
                }
                catch (SocketException)
                {
                    failedPorts.Add(port);
                    Console.WriteLine("FAILED");
                }
            }

            Console.WriteLine();

            return failedPorts;
        }

        static List<int> TestUdpPorts(List<int> portList)
        {
            Console.WriteLine("Testing UDP ports to {0}:", _ipAddr.ToString());

            List<int> failedPorts = new List<int>();

            foreach (int port in portList)
            {
                Console.Write("Checking UDP port {0}: ", port);

                UdpClient udpClient = new UdpClient();

                try
                {
                    udpClient.Connect(_ipAddr, port);
                    udpClient.Close();
                    Console.WriteLine("PASSED");
                }
                catch (SocketException)
                {
                    failedPorts.Add(port);
                    Console.WriteLine("FAILED");
                }
            }

            Console.WriteLine();

            return failedPorts;
        }
    }
}
```

Create an AD Connector

To connect to your existing directory with AD Connector, perform the following steps. Before starting this procedure, make sure you have completed the prerequisites identified in AD Connector prerequisites (p. 187).

> **Note**
> You cannot create an AD Connector with a Cloud Formation template.

To connect with AD Connector

1. In the AWS Directory Service console navigation pane, choose **Directories** and then choose **Set up directory**.
2. On the **Select directory type** page, choose **AD Connector**, and then choose **Next**.
3. On the **Enter AD Connector information** page, provide the following information:

Directory size

Choose from either the **Small** or **Large** size option. For more information about sizes, see Active Directory Connector (p. 186).

Directory description

An optional description for the directory.

4. On the **Choose VPC and subnets** page, provide the following information, and then choose **Next**.

   VPC

   The VPC for the directory.

   Subnets

   Choose the subnets for the domain controllers. The two subnets must be in different Availability Zones.

5. On the **Connect to AD** page, provide the following information:

   Directory DNS name

   The fully qualified name of your existing directory, such as `corp.example.com`.

   Directory NetBIOS name

   The short name of your existing directory, such as `CORP`.

   DNS IP addresses

   The IP address of at least one DNS server in your existing directory. These servers must be accessible from each subnet specified in the next section.

   Service account username

   The user name of a user in the existing directory. For more information about this account, see the AD Connector prerequisites (p. 187).

   Service account password

   The password for the existing user account. This password is case-sensitive and must be between 8 and 128 characters in length, inclusive. It must also contain at least one character from three of the following four categories:

   - Lowercase letters (a-z)
   - Uppercase letters (A-Z)
   - Numbers (0-9)
   - Non-alphanumeric characters (~!@#$%^&*_-+=`|\(){}[]:;"'<>,.?/)

   Confirm password

   Retype the password for the existing user account.

6. On the **Review & create** page, review the directory information and make any necessary changes. When the information is correct, choose **Create directory**. It takes several minutes for the directory to be created. Once created, the **Status** value changes to **Active**.

What gets created

When you create an AD Connector, AWS Directory Service automatically creates and associates an elastic network interface (ENI) with each of your AD Connector instances. Each of these ENIs are essential for connectivity between your VPC and AWS Directory Service AD Connector and should never be deleted.

You can identify all network interfaces reserved for use with AWS Directory Service by the description: "AWS created network interface for directory *directory-id*". For more information, see Elastic Network Interfaces in the Amazon EC2 User Guide for Windows Instances.

> **Note**
> AD Connector instances are deployed across two Availability Zones in a Region by default and connected to your Amazon Virtual Private Cloud (VPC). AD Connector instances that fail are automatically replaced in the same Availability Zone using the same IP address.

When you sign in to any AWS application or service integrated with an AD Connector (AWS Single Sign-On included), the app or service forwards your authentication request to AD Connector which then forwards the request to a domain controller in your self-managed AD for authentication. If you are successfully authenticated to your self-managed AD, AD Connector then returns an authentication token to the app or service (similar to a Kerberos token). At this point, you can now access the AWS app or service.

How to administer AD Connector

This section lists all of the procedures for operating and maintaining an AD Connector environment.

Topics

Secure your AD Connector directory

This section describes considerations for securing your AD Connector environment.

Topics

Update your AD Connector service account credentials in AWS Directory Service

The AD Connector credentials you provide in AWS Directory Service represent the service account that is used to access your existing on-premises directory. You can modify the service account credentials in AWS Directory Service by performing the following steps.

> **Note**
> If single sign-on is enabled for the directory, AWS Directory Service must transfer the service principal name (SPN) from the current service account to the new service account. If the current service account does not have permission to delete the SPN or the new service account does not have permission to add the SPN, you are prompted for the credentials of a directory account that does have permission to perform both actions. These credentials are only used to transfer the SPN and are not stored by the service.

To update your AD Connector service account credentials in AWS Directory Service

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. Choose the directory ID link for your directory.
3. On the **Directory details** page, choose the **Networking & security** tab.
4. In the **Service account credentials** section, choose **Update**.
5. In the **Update service account credentials** dialog, type the new user name and password, and then choose **Update Directory**.

Enable multi-factor authentication for AD Connector

You can enable multi-factor authentication for AD Connector when you have Active Directory running on-premises or in EC2 instances. For more information about using multi-factor authentication with AWS Directory Service, see AD Connector prerequisites (p. 187).

> **Note**
> Multi-factor authentication is not available for Simple AD. However, MFA can be enabled for your AWS Managed Microsoft AD directory. For more information, see Enable multi-factor authentication for AWS Managed Microsoft AD (p. 31).

To enable multi-factor authentication for AD Connector

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. Choose the directory ID link for your AD Connector directory.
3. On the **Directory details** page, select the **Networking & security** tab.
4. In the **Multi-factor authentication** section, choose **Actions**, and then choose **Enable**.
5. On the **Enable multi-factor authentication (MFA)** page, provide the following values:

   Display label

   Provide a label name.

   RADIUS server DNS name or IP addresses

   The IP addresses of your RADIUS server endpoints, or the IP address of your RADIUS server load balancer. You can enter multiple IP addresses by separating them with a comma (e.g., `192.0.0.0,192.0.0.12`).

   > **Note**
   > RADIUS MFA is applicable only to authenticate access to the AWS Management Console, or to Amazon Enterprise applications and services such as WorkSpaces, Amazon QuickSight, or Amazon Chime. It does not provide MFA to Windows workloads running on EC2 instances, or for signing into an EC2 instance. AWS Directory Service does not support RADIUS Challenge/Response authentication.
   > Users must have their MFA code at the time they enter their user name and password. Alternatively, you must use a solution that performs MFA out-of-band such as SMS text verification for the user. In out-of-band MFA solutions, you must make sure you set the RADIUS time-out value appropriately for your solution. When using an out-of-band MFA solution, the sign-in page will prompt the user for an MFA code. In this case, the best practice is for users to enter their password in both the password field and the MFA field.

   Port

   The port that your RADIUS server is using for communications. Your on-premises network must allow inbound traffic over the default RADIUS server port (UDP:1812) from the AWS Directory Service servers.

Shared secret code

The shared secret code that was specified when your RADIUS endpoints were created.

Confirm shared secret code

Confirm the shared secret code for your RADIUS endpoints.

Protocol

Select the protocol that was specified when your RADIUS endpoints were created.

Server timeout (in seconds)

The amount of time, in seconds, to wait for the RADIUS server to respond. This must be a value between 1 and 50.

Max RADIUS request retries

The number of times that communication with the RADIUS server is attempted. This must be a value between 0 and 10.

Multi-factor authentication is available when the **RADIUS Status** changes to **Enabled**.

6. Choose **Enable**.

Enable client-side LDAPS using AD Connector

Client-side LDAPS support in AD Connector encrypts communications between Microsoft Active Directory (AD) and AWS applications. Examples of such applications include WorkSpaces, AWS SSO, Amazon QuickSight, and Amazon Chime. This encryption helps you to better protect your organization's identity data and meet your security requirements.

Topics

- Prerequisites (p. 200)
- Enable client-side LDAPS (p. 201)
- Manage client-side LDAPS (p. 203)

Prerequisites

Before you enable client-side LDAPS, you need to meet the following requirements.

Topics

- Deploy server certificates in Active Directory (p. 200)
- CA certificate requirements (p. 201)
- Networking requirements (p. 201)

Deploy server certificates in Active Directory

In order to enable client-side LDAPS, you need to obtain and install server certificates for each domain controller in Active Directory. These certificates will be used by the LDAP service to listen for and automatically accept SSL connections from LDAP clients. You can use SSL certificates that are either issued by an in-house Active Directory Certificate Services (ADCS) deployment or purchased from a commercial issuer. For more information on Active Directory server certificate requirements, see LDAP over SSL (LDAPS) Certificate on the Microsoft website.

CA certificate requirements

A certificate authority (CA) certificate, which represents the issuer of your server certificates, is required for client-side LDAPS operation. CA certificates are matched with the server certificates that are presented by your Active Directory domain controllers to encrypt LDAP communications. Note the following CA certificate requirements:

- To register a certificate, it must be more than 90 days away from expiration.
- Certificates must be in Privacy-Enhanced Mail (PEM) format. If exporting CA certificates from inside Active Directory, choose base64 encoded X.509 (.CER) as the export file format.
- A maximum of five (5) CA certificates can be stored per AD Connector directory.
- Certificates using the RSASSA-PSS signature algorithm are not supported.

Networking requirements

AWS application LDAP traffic will run exclusively on TCP port 636, with no fallback to LDAP port 389. However, Windows LDAP communications supporting replication, trusts, and more will continue using LDAP port 389 with Windows-native security. Configure AWS security groups and network firewalls to allow TCP communications on port 636 in AD Connector (outbound) and self-managed Active Directory (inbound).

Enable client-side LDAPS

To enable client-side LDAPS, you import your certificate authority (CA) certificate into AD Connector, and then enable LDAPS on your directory. Upon enabling, all LDAP traffic between AWS applications and your self-managed Active Directory will flow with Secure Sockets Layer (SSL) channel encryption.

You can use two different methods to enable client-side LDAPS for your directory. You can use either the AWS Management Console method or the AWS CLI method.

Topics

Step 1: Register certificate in AWS Directory Service

Use either of the following methods to register a certificate in AWS Directory Service.

Method 1: To register your certificate in AWS Directory Service (AWS Management Console)

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. Choose the directory ID link for your directory.
3. On the **Directory details** page, choose the **Networking & security** tab.
4. In the **Client-side LDAPS** section, select the **Actions** menu, and then select **Register certificate**.
5. In the **Register a CA certificate** dialog box, select **Browse**, and then select the certificate and choose **Open**.
6. Choose **Register certificate**.

Method 2: To register your certificate in AWS Directory Service (AWS CLI)

- Run the following command. For the certificate data, point to the location of your CA certificate file. A certificate ID will be provided in the response.

```
aws ds register-certificate --directory-id your_directory_id --certificate-data
 file://your_file_path
```

Step 2: Check registration status

To see the status of a certificate registration or a list of registered certificates, use either of the following methods.

Method 1: To check certificate registration status in AWS Directory Service (AWS Management Console)

1. Go to the **Client-side LDAPS** section on the **Directory details** page.
2. Review the current certificate registration state that is displayed under the **Registration status** column. When the registration status value changes to **Registered**, your certificate has been successfully registered.

Method 2: To check certificate registration status in AWS Directory Service (AWS CLI)

- Run the following command. If the status value returns `Registered`, your certificate has been successfully registered.

```
aws ds list-certificates --directory-id your_directory_id
```

Step 3: Enable client-side LDAPS

Use either of the following methods to enable client-side LDAPS in AWS Directory Service.

> **Note**
> You must have successfully registered at least one certificate before you can enable client-side LDAPS.

Method 1: To enable client-side LDAPS in AWS Directory Service (AWS Management Console)

1. Go to the **Client-side LDAPS** section on the **Directory details** page.
2. Choose **Enable**. If this option is not available, verify that a valid certificate has been successfully registered, and then try again.
3. In the **Enable client-side LDAPS** dialog box, choose **Enable**.

Method 2: To enable client-side LDAPS in AWS Directory Service (AWS CLI)

- Run the following command.

```
aws ds enable-ldaps --directory-id your_directory_id --type Client
```

Step 4: Check LDAPS status

Use either of the following methods to check the LDAPS status in AWS Directory Service.

Method 1: To check LDAPS status in AWS Directory Service (AWS Management Console)

1. Go to the **Client-side LDAPS** section on the **Directory details** page.

2. If the status value is displayed as **Enabled**, LDAPS has been successfully configured.

Method 2: To check LDAPS status in AWS Directory Service (AWS CLI)

- Run the following command. If the status value returns `Enabled`, LDAPS has been successfully configured.

  ```
  aws ds describe-ldaps-settings -directory-id your_directory_id
  ```

Manage client-side LDAPS

Use these commands to manage your LDAPS configuration.

You can use two different methods to manage client-side LDAPS settings. You can use either the AWS Management Console method or the AWS CLI method.

View certificate details

Use either of the following methods to see when a certificate is set to expire.

Method 1: To view certificate details in AWS Directory Service (AWS Management Console)

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. Choose the directory ID link for your directory.
3. On the **Directory details** page, choose the **Networking & security** tab.
4. In the **Client-side LDAPS** section, under **CA certificates**, information about the certificate will be displayed.

Method 2: To view certificate details in AWS Directory Service (AWS CLI)

- Run the following command. For the certificate ID, use the identifier returned by `register-certificate` or `list-certificates`.

  ```
  aws ds describe-certificate --directory-id your_directory_id --certificate-
  id your_cert_id
  ```

Deregister a certificate

Use either of the following methods to deregister a certificate.

> **Note**
> If only one certificate is registered, you must first disable LDAPS before you can deregister the certificate.

Method 1: To deregister a certificate in AWS Directory Service (AWS Management Console)

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. Choose the directory ID link for your directory.
3. On the **Directory details** page, choose the **Networking & security** tab.
4. In the **Client-side LDAPS** section, choose **Actions**, and then choose **Deregister certificate**.
5. In the **Deregister a CA certificate** dialog box, choose **Deregister**.

Method 2: To deregister a certificate in AWS Directory Service (AWS CLI)

- Run the following command. For the certificate ID, use the identifier returned by `register-certificate` or `list-certificates`.

```
aws ds deregister-certificate --directory-id your_directory_id --certificate-
id your_cert_id
```

Disable client-side LDAPS

Use either of the following methods to disable client-side LDAPS.

Method 1: To disable client-side LDAPS in AWS Directory Service (AWS Management Console)

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. Choose the directory ID link for your directory.
3. On the **Directory details** page, choose the **Networking & security** tab.
4. In the **Client-side LDAPS** section, choose **Disable**.
5. In the **Disable client-side LDAPS** dialog box, choose **Disable**.

Method 2: To disable client-side LDAPS in AWS Directory Service (AWS CLI)

- Run the following command.

```
aws ds disable-ldaps --directory-id your_directory_id --type Client
```

Enable mTLS authentication in AD Connector for use with smart cards

You can use certificate-based mutual Transport Layer Security (mTLS) authentication with smart cards to authenticate users into Amazon WorkSpaces through your self-managed Active Directory (AD) and AD Connector. When enabled, users select their smart card at the WorkSpaces login screen and enter a PIN to authenticate, instead of using a username and password. From there, the Windows or Linux virtual desktop uses the smart card to authenticate into AD from the native desktop OS.

> **Note**
> Smart card authentication in AD Connector is only available in the following regions, and only with WorkSpaces. Other AWS applications are not supported at this time.
>
> - US East (N. Virginia)
> - US West (Oregon)
> - Asia Pacific (Sydney)
> - Asia Pacific (Tokyo)
> - Europe (Ireland)
> - AWS GovCloud (US-West)

Topics

- Manage smart card authentication settings (p. 208)

Prerequisites

To enable mTLS authentication using smart cards for the Amazon WorkSpaces client, you need an operational smart card infrastructure integrated with your self-managed AD. For more information on how to set up smart card authentication with Amazon WorkSpaces and AD, see the Amazon WorkSpaces Administration Guide.

Before you enable smart card authentication for WorkSpaces, please review the following considerations:

- CA certificate requirements (p. 205)
- User certificate requirements (p. 205)
- Certificate revocation checking process (p. 205)
- Other considerations (p. 206)

CA certificate requirements

AD Connector requires a certificate authority (CA) certificate, which represents the issuer of your user certificates, for smart card authentication. AD Connector matches CA certificates with the certificates presented by your users with their smart cards. Note the following CA certificate requirements:

- Before you can register a CA certificate, it must be more than 90 days away from expiration.
- CA certificates must be in Privacy-Enhanced Mail (PEM) format. If you export CA certificates from inside Active Directory, choose Base64-encoded X.509 (.CER) as the export file format.
- All root and intermediary CA certificates that chain from an issuing CA to user certificates must be uploaded for smart card authentication to succeed.
- A maximum of 100 CA certificates can be stored per AD Connector directory
- AD Connector does not support the RSASSA-PSS signature algorithm for CA certificates.

User certificate requirements

User certificates presented to AD Connector must include the `userPrincipalName (UPN)` of the AD user in the `subjectAltName (SAN)` field of the certificate.

Certificate revocation checking process

In order to perform smart card authentication, AD Connector must check the revocation status of user certificates using Online Certificate Status Protocol (OCSP). To perform certificate revocation checking, an OCSP responder URL must be internet-accessible. If using a DNS name, an OCSP responder URL must use a top-level domain found in the Internet Assigned Numbers Authority (IANA) Root Zone Database.

AD Connector certificate revocation checking uses the following process:

- AD Connector must check the Authority Information Access (AIA) extension in the user certificate for an OCSP responder URL, then AD Connector uses the URL to check for revocation.
- If AD Connector cannot resolve the URL found in the user certificate AIA extension, or find an OCSP responder URL in the user certificate, then AD Connector uses the optional OCSP URL provided during root CA certificate registration.

  If the URL in the user certificate AIA extension resolves but is unresponsive, then user authentication fails.
- If the OCSP responder URL provided during root CA certificate registration cannot resolve, is unresponsive, or no OCSP responder URL was provided, user authentication fails.

Note
AD Connector requires an **HTTP** URL for the OCSP responder URL.

Other considerations

Before enabling smart card authentication in AD Connector, consider the following items:

- AD Connector uses certificate-based mutual Transport Layer Security authentication (mutual TLS) to authenticate users to Active Directory using hardware or software-based smart card certificates. Only common access cards (CAC) and personal identity verification (PIV) cards are supported at this time. Other types of hardware or software-based smart cards might work but have not been tested for use with the WorkSpaces Streaming Protocol.
- Smart card authentication replaces username and password authentication to WorkSpaces.

  If you have other AWS applications configured on your AD Connector directory with smart card authentication enabled, those applications still present the username and password input screen.
- Enabling smart card authentication limits the user session length to the maximum lifetime for Kerberos service tickets. You can configure this setting using a Group Policy, and is set to 10 hours by default. For more information on this setting, see Microsoft documentation.

Enable smart card authentication

To enable smart card authentication, use the following steps to import your certificate authority (CA) certificates into AD Connector using the AWS Directory Service console, API or CLI. And then enable smart card authentication for WorkSpaces on your AD Connector.

Topics

Step 1: Enable Kerberos constrained delegation for the AD Connector service account

To use smart card authentication with AD Connector, you must enable **Kerberos Constrained Delegation (KCD)** for the AD Connector Service account to the LDAP service in the self-managed AD. directory.

Kerberos Constrained Delegation is a feature in Windows Server. This feature enables administrators to specify and enforce application trust boundaries by limiting the scope where application services can act on a user's behalf. For more information, see Kerberos constrained delegation.

1. Use the `SetSpn` command to set a Service Principal Name (SPN) for the AD Connector service account in the self-managed AD. This enables the service account for delegation configuration.

   The SPN can be any service or name combination but not a duplicate of an existing SPN. The `-s` checks for duplicates.

   ```
   setspn -s my/spn service_account
   ```
2. In **AD Users and Computers**, right-click on the AD Connector service account and select **Properties.**
3. Choose the **Delegation** tab.
4. Choose the **Trust this user for delegation to specified service only** and **Use any authentication protocol** radio buttons.
5. Choose **Add** and then **Users or Computers** to locate the domain controller.

6. Choose **OK** to display a list of available services used for delegation.
7. Choose the **ldap** service type and click **OK**.
8. Click **OK** again to save the configuration.
9. Repeat this process for other domain controllers in AD. Alternatively you can automate the process using PowerShell.

Step 2: Register the CA certificate in AD Connector

Use either of the following methods to register a CA certificate for your AD Connector directory.

Method 1: To register your CA certificate in AD Connector (AWS Management Console)

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. Choose the directory ID link for your directory.
3. On the **Directory details** page, choose the **Networking & security** tab.
4. In the **Smart card authentication** section, select the **Actions** menu, and then select **Register certificate**.
5. In the **Register a certificate** dialog box, select **Choose file**, and then select a certificate and choose **Open**. You can optionally choose to perform revocation checking for this certificate by providing an Online Certificate Status Protocol (OCSP) responder URL. For more information about OCSP, see Certificate revocation checking process (p. 205).
6. Choose **Register certificate**. When you see the certificate status change to **Registered**, the registration process has completed successfully.

Method 2: To register your CA certificate in AD Connector (AWS CLI)

- Run the following command. For the certificate data, point to the location of your CA certificate file. To provide a secondary OCSP responder address, use the optional `ClientCertAuthSettings` object.

  ```
  aws ds register-certificate --directory-id your_directory_id --certificate-
  data file://your_file_path --type ClientCertAuth --client-cert-auth-settings
   OCSPUrl=http://your_OCSP_address
  ```

  If successful, the response provides a certificate ID. You can also verify your CA certificate registered successfully by running the following CLI command:

  ```
  aws ds list-certificates --directory-id your_directory_id
  ```

  If the status value returns `Registered`, you have successfully registered your certificate.

Step 3: Enable smart card authentication for supported AWS applications and services

Use either of the following methods to register a CA certificate for your AD Connector directory.

Method 1: To enable smart card authentication in AD Connector (AWS Management Console)

1. Go to the **Smart card authentication** section on the **Directory details** page, and choose **Enable**. If this option is not available, verify that a valid certificate has been successfully registered, and then try again.
2. In the **Enable smart card authentication** dialog box, select **Enable**.

Method 2: To enable smart card authentication in AD Connector (AWS CLI)

- Run the following command.

  ```
  aws ds enable-client-authentication --directory-id your_directory_id --type SmartCard
  ```

  If successful, AD Connector returns an `HTTP 200` response with an empty HTTP body.

Manage smart card authentication settings

You can use two different methods to manage smart card settings. You can use either the AWS Management Console method or the AWS CLI method.

Topics

View certificate details

Use either of the following methods to see when a certificate is set to expire.

Method 1: To view certificate details in AWS Directory Service (AWS Management Console)

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. Choose the directory ID link for your AD Connector directory.
3. On the **Directory details** page, choose the **Networking & security** tab.
4. In the **Smart card authentication** section, under **CA certificates**, choose the certificate ID to display details about that certificate.

Method 2: To view certificate details in AWS Directory Service (AWS CLI)

- Run the following command. For the certificate ID, use the identifier returned by `register-certificate` or `list-certificates`.

  ```
  aws ds describe-certificate --directory-id your_directory_id --certificate-
  id your_cert_id
  ```

Deregister a certificate

Use either of the following methods to deregister a certificate.

> **Note**
> If only one certificate is registered, you must first disable smart card authentication before you can deregister the certificate.

Method 1: To deregister a certificate in AWS Directory Service (AWS Management Console)

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. Choose the directory ID link for your AD Connector directory.
3. On the **Directory details** page, choose the **Networking & security** tab.

4. In the **Smart card authentication** section, under **CA certificates**, select the certificate you want to deregister, choose **Actions**, and then choose **Deregister certificate**.

    Important
    Ensure that the certificate you are about to deregister is not active or is currently being used as part of a CA certificate chain for smart card authentication.

5. In the **Deregister a CA certificate** dialog box, choose **Deregister**.

Method 2: To deregister a certificate in AWS Directory Service (AWS CLI)

- Run the following command. For the certificate ID, use the identifier returned by `register-certificate` or `list-certificates`.

```
aws ds deregister-certificate --directory-id your_directory_id --certificate-
id your_cert_id
```

Disable smart card authentication

Use either of the following methods to disable smart card authentication.

Method 1: To disable smart card authentication in AWS Directory Service (AWS Management Console)

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. Choose the directory ID link for your AD Connector directory.
3. On the **Directory details** page, choose the **Networking & security** tab.
4. In the **Smart card authentication** section, choose **Disable**.
5. In the **Disable smart card authentication** dialog box, choose **Disable**.

Method 2: To disable smart card authentication in AWS Directory Service (AWS CLI)

- Run the following command.

```
aws ds disable-client-authentication --directory-id your_directory_id --type SmartCard
```

Monitor your AD Connector directory

You can monitor your AD Connector directory with the following methods:

Topics

Understanding your directory status

The following are the various statuses for a directory.

Active

The directory is operating normally. No issues have been detected by the AWS Directory Service for your directory.

Creating

The directory is currently being created. Directory creation typically takes between 20 to 45 minutes but may vary depending on the system load.

Deleted

The directory has been deleted. All resources for the directory have been released. Once a directory enters this state, it cannot be recovered.

Deleting

The directory is currently being deleted. The directory will remain in this state until it has been completely deleted. Once a directory enters this state, the delete operation cannot be cancelled, and the directory cannot be recovered.

Failed

The directory could not be created. Please delete this directory. If this problem persists, please contact the AWS Support Center.

Impaired

The directory is running in a degraded state. One or more issues have been detected, and not all directory operations may be working at full operational capacity. There are many potential reasons for the directory being in this state. These include normal operational maintenance activity such as patching or EC2 instance rotation, temporary hot spotting by an application on one of your domain controllers, or changes you made to your network that inadvertently disrupt directory communications. For more information, see either Troubleshooting AWS Managed Microsoft AD (p. 175), Troubleshooting AD Connector (p. 223), Troubleshooting Simple AD (p. 272). For normal maintenance related issues, AWS resolves these issues within 40 minutes. If after reviewing the troubleshooting topic, your directory is in an Impaired state longer than 40 minutes, we recommend that you contact the AWS Support Center.

> **Important**
> Do not restore a snapshot while a directory is in an Impaired state. It is rare that snapshot restore is necessary to resolve impairments. For more information, see Snapshot or restore your directory (p. 128).

Inoperable

The directory is not functional. All directory endpoints have reported issues.

Requested

A request to create your directory is currently pending.

RestoreFailed

Restoring the directory from a snapshot failed. Please retry the restore operation. If this continues, try a different snapshot, or contact the AWS Support Center.

Restoring

The directory is currently being restored from an automatic or manual snapshot. Restoring from a snapshot typically takes several minutes, depending on the size of the directory data in the snapshot.

For more information, see Simple AD directory status reasons (p. 274).

Configure directory status notifications

Using Amazon Simple Notification Service (Amazon SNS), you can receive email or text (SMS) messages when the status of your directory changes. You get notified if your directory goes from an Active status to an Impaired or Inoperable status. You also receive a notification when the directory returns to an Active status.

How it works

Amazon SNS uses “topics” to collect and distribute messages. Each topic has one or more subscribers who receive the messages that have been published to that topic. Using the steps below you can add AWS Directory Service as publisher to an Amazon SNS topic. When AWS Directory Service detects a change in your directory’s status, it publishes a message to that topic, which is then sent to the topic's subscribers.

You can associate multiple directories as publishers to a single topic. You can also add directory status messages to topics that you’ve previously created in Amazon SNS. You have detailed control over who can publish to and subscribe to a topic. For complete information about Amazon SNS, see What is Amazon SNS?.

To enable SNS messaging for your directory

1. Sign in to the AWS Management Console and open the AWS Directory Service console at https://console.aws.amazon.com/directoryservicev2/.
2. On the **Directories** page, choose your directory ID.
3. Select the **Maintenance** tab.
4. In the **Directory monitoring** section, choose **Actions**, and then select **Create notification**.
5. On the **Create notification** page, select **Choose a notification type**, and then choose **Create a new notification**. Alternatively, if you already have an existing SNS topic, you can choose **Associate existing SNS topic** to send status messages from this directory to that topic.

   Note
   If you choose **Create a new notification** but then use the same topic name for an SNS topic that already exists, Amazon SNS does not create a new topic, but just adds the new subscription information to the existing topic.
   If you choose **Associate existing SNS topic**, you will only be able to choose an SNS topic that is in the same Region as the directory.
6. Choose the **Recipient type** and enter the **Recipient** contact information. If you enter a phone number for SMS, use numbers only. Do not include dashes, spaces, or parentheses.
7. (Optional) Provide a name for your topic and an SNS display name. The display name is a short name up to 10 characters that is included in all SMS messages from this topic. When using the SMS option, the display name is required.

   Note
   If you are logged in using an IAM user or role that has only the DirectoryServiceFullAccess managed policy, your topic name must start with “DirectoryMonitoring”. If you’d like to further customize your topic name you’ll need additional privileges for SNS.
8. Choose **Create**.

If you want to designate additional SNS subscribers, such as an additional email address, Amazon SQS queues or AWS Lambda, you can do this from the Amazon SNS console at https://console.aws.amazon.com/sns/v3/home.

To remove directory status messages from a topic

1. Sign in to the AWS Management Console and open the AWS Directory Service console at https://console.aws.amazon.com/directoryservicev2/.
2. On the **Directories** page, choose your directory ID.
3. Select the **Maintenance** tab.
4. In the **Directory monitoring** section, select an SNS topic name in the list, choose **Actions**, and then select **Remove**.
5. Choose **Remove**.

This removes your directory as a publisher to the selected SNS topic. If you want to delete the entire topic, you can do this from the Amazon SNS console at https://console.aws.amazon.com/sns/v3/home.

> **Note**
> Before deleting an Amazon SNS topic using the SNS console, you should ensure that a directory is not sending status messages to that topic.
> If you delete an Amazon SNS topic using the SNS console, this change will not immediately be reflected within the Directory Services console. You would only be notified the next time a directory publishes a notification to the deleted topic, in which case you would see an updated status on the directory's **Monitoring** tab indicating the topic could not be found.
> Therefore, to avoid missing important directory status messages, before deleting any topic that receives messages from AWS Directory Service, associate your directory with a different Amazon SNS topic.

Join an EC2 instance to your AD Connector directory

You can seamlessly join an EC2 instance to your directory domain when the instance is launched using AWS Systems Manager. For more information, see Seamlessly joining a Windows instance to an AWS Directory Service domain in the *Amazon EC2 User Guide for Windows Instances*.

If you need to manually join an EC2 instance to your domain, you must launch the instance in the proper Region and security group or subnet, then join the instance to the domain.

To be able to connect remotely to these instances, you must have IP connectivity to the instances from the network you are connecting from. In most cases, this requires that an internet gateway be attached to your VPC and that the instance has a public IP address.

> **Note**
> Once you join an instance to your self-managed Active Directory (on-premises), the instance communicates directly with your Active Directory and bypasses AD Connector.

Topics

Seamlessly join a Windows EC2 instance

This procedure seamlessly joins a Windows EC2 instance to your AD Connector directory.

To seamlessly join a Windows EC2 instance

1. Sign in to the AWS Management Console and open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the Region selector in the navigation bar, choose the same Region as the existing directory.
3. Choose **Launch Instance**.
4. On the **Step 1** page, choose **Select** for the appropriate AMI.
5. On the **Step 2** page, select the appropriate instance type, and then choose **Next: Configure Instance Details**.
6. On the **Step 3** page, do the following, and then choose **Next: Add Storage**:

    1. For **Network**, choose the VPC that your directory was created in.
    2. For **Subnet**, choose one of the public subnets in your VPC. The subnet that you choose must have all external traffic routed to an internet gateway. If this is not the case, you won't be able to connect to the instance remotely.
    3. For **Auto-assign Public IP**, choose **Enable**.

For more information about public and private IP addressing, see Amazon EC2 instance IP addressing in the *Amazon EC2 User Guide for Windows Instances*.

4. For **Domain join directory**, choose your domain from the list.
5. For **IAM role**, do one of the following:

   Select an IAM role that has the AWS managed policies **AmazonSSMManagedInstanceCore** and **AmazonSSMDirectoryServiceAccess** attached to it.

   -or-

   If you haven't created an IAM role that has the **AmazonSSMManagedInstanceCore** and **AmazonSSMDirectoryServiceAccess** managed policies attached to it, choose the **Create new IAM role** link, and then do the following:

   a. Choose **Create role**.
   b. Under **Select type of trusted entity**, choose **AWS service**.
   c. Under **Choose the service that this role will use**, in the full list of services, choose **EC2**.
   d. Under **Select your use case**, choose **EC2**, and the choose **Next: Permissions**.
   e. In the list of policies, select the **AmazonSSMManagedInstanceCore** and **AmazonSSMDirectoryServiceAccess** policies. (To filter the list, type `SSM` in the search box.)

      > **Note**
      > **AmazonSSMDirectoryServiceAccess** provides the permissions to join instances to an Active Directory managed by AWS Directory Service. **AmazonSSMManagedInstanceCore** provides the minimum permissions necessary to use the Systems Manager service. For more information about creating a role with these permissions, and for information about other permissions and policies you can assign to your IAM role, see Create an IAM instance profile for Systems Manager in the *AWS Systems Manager User Guide*.

   f. Choose **Next: Tags**.
   g. (Optional) Add one or more tag key-value pairs to organize, track, or control access for this role, and then choose **Next: Review**.
   h. For **Role name**, enter a name for your new role, such as `EC2DomainJoin` or another name that you prefer.
   i. (Optional) For **Role description**, enter a description.
   j. Choose **Create role**.
   k. Go back to the **Step 3** page. For **IAM role**, choose the refresh icon next to **IAM role**. Your new role should be visible in the menu. Choose it and leave the rest of the settings on this page with their default values, and then choose **Next: Add Storage**.

7. On both the **Step 4** and **Step 5** pages, leave the default settings or make changes as needed, and then choose the **Next** buttons.
8. On the **Step 6** page, select a security group for the instance that has been configured to allow remote access to the instance from your network, and then choose **Review and Launch**.
9. On the **Step 7** page, choose **Launch**, select a key pair, and then choose **Launch Instance**.

Seamlessly join a Linux EC2 instance to your AD Connector directory

This procedure seamlessly joins a Linux EC2 instance to your AD Connector directory.

The following Linux instance distributions and versions are supported:

- Amazon Linux AMI 2018.03.0

- Amazon Linux 2 (64-bit x86)
- Red Hat Enterprise Linux 8 (HVM) (64-bit x86)
- Ubuntu Server 18.04 LTS & Ubuntu Server 16.04 LTS
- CentOS 7 x86-64
- SUSE Linux Enterprise Server 15 SP1

> **Note**
> Distributions prior to Ubuntu 14 and Red Hat Enterprise Linux 7 do not support the seamless domain join feature.

Prerequisites

Before you can set up seamless domain join to a Linux EC2 instance, you need to complete the procedures in this section.

Select your seamless domain join service account

You can seamlessly join Linux computers to your on-premises Active Directory domain through AD Connector. To do that, you must create a user account with create computer account permissions to join the computers to the domain. You can use your AD Connector service account if you prefer. Or you can use any other account that has sufficient privileges to join computers to the domain. Although members of the *Domain Admins* or other groups might have sufficient privileges to join computers to the domain, we do not recommend these. As a best practice, we recommend that you use a service account that has the minimum privileges necessary to join computers to the domain.

To delegate an account with the minimum privileges necessary to join computers to the domain, you can run the following PowerShell commands. You must run these commands from a domain-joined Windows computer with the Installing the Active Directory administration tools (p. 101) installed. In addition, you must use an account that has permission to modify the permissions on your Computers OU or container. The PowerShell command sets permissions that allow the service account to create computer objects in your domain's default computers container. If you prefer using a graphical user interface (GUI) you can use the manual process that is described in Delegate privileges to your service account (p. 189).

```
$AccountName = 'awsSeamlessDomain'
# DO NOT modify anything below this comment.
# Getting Active Directory information.
Import-Module 'ActiveDirectory'
$Domain = Get-ADDomain -ErrorAction Stop
$BaseDn = $Domain.DistinguishedName
$ComputersContainer = $Domain.ComputersContainer
$SchemaNamingContext = Get-ADRootDSE | Select-Object -ExpandProperty 'schemaNamingContext'
[System.GUID]$ServicePrincipalNameGuid = (Get-ADObject -SearchBase $SchemaNamingContext -
Filter { lDAPDisplayName -eq 'Computer' } -Properties 'schemaIDGUID').schemaIDGUID
# Getting Service account Information.
$AccountProperties = Get-ADUser -Identity $AccountName
$AccountSid = New-Object -TypeName 'System.Security.Principal.SecurityIdentifier'
 $AccountProperties.SID.Value
# Getting ACL settings for the Computers container.
$ObjectAcl = Get-ACL -Path "AD:\$ComputersContainer"
# Setting ACL allowing the service account the ability to create child computer objects in
 the Computers container.
$AddAccessRule = New-Object -TypeName 'System.DirectoryServices.ActiveDirectoryAccessRule'
 $AccountSid, 'CreateChild', 'Allow', $ServicePrincipalNameGUID, 'All'
$ObjectAcl.AddAccessRule($AddAccessRule)
Set-ACL -AclObject $ObjectAcl -Path "AD:\$ComputersContainer"
```

If you prefer using a graphical user interface (GUI) you can use the manual process to described in Delegate privileges to your service account (p. 189).

Create the secrets to store the domain service account

You can use AWS Secrets Manager to store the domain service account.

To create secrets and store the domain service account information

1. Sign in to the AWS Management Console and open the AWS Secrets Manager console at https://console.aws.amazon.com/secretsmanager/.
2. Choose **Store a new secret**.
3. On the **Store a new secret** page, do the following:

    a. Under **Select secret type**, choose **Other type of secrets**.

    b. Under **Specify the key/value pairs to be stored in the secret**, do the following:

        i. In the first box, enter **awsSeamlessDomainUsername**. On the same row, in the next box, enter the user name for your service account. For example, if you used the PowerShell command previously, the service account name would be awsSeamlessDomain.

            Note
            You must enter **awsSeamlessDomainUsername** exactly as it is. Make sure there are not any leading or ending spaces. Otherwise the domain join will fail.

        ii. Choose **Add row**.

        iii. On the new row, in the first box, enter **awsSeamlessDomainPassword**. On the same row, in the next box, enter the password for your service account.

            Note
            You must enter **awsSeamlessDomainPassword** exactly as it is. Make sure there are not any leading or ending spaces. Otherwise the domain join will fail.

        iv. Under **Select the encryption key**, choose **DefaultEncryptionKey** from the menu. Secrets Manager always encrypts the secret when you choose this option, and provides it at no charge to you. You also may choose a key you created.

        v. Choose **Next**.
4. Under **Secret name**, enter a secret name that includes your directory ID using the following format:

    ```
    aws/directory-services/d-xxxxxxxxx/seamless-domain-join
    ```

    This will be used to retrieve secrets in the application.

    Note
    You must enter **aws/directory-services/*d-xxxxxxxxx*/seamless-domain-join** exactly as it is but replace *d-xxxxxxxxxxx* with your directory ID. Make sure that there are no leading or ending spaces. Otherwise the domain join will fail.
5. Leave everything else set to defaults, and then choose **Next**.
6. Under **Configure automatic rotation**, choose **Disable automatic rotation**, and then choose **Next**.
7. Review the settings, and then choose **Store** to save your changes. The Secrets Manager console returns you to the list of secrets in your account with your new secret now included in the list.
8. Choose your newly created secret name from the list, and take note of the **Secret ARN** value. You will need it in the next section.

Create the required IAM policy and role

Use the following prerequisite steps to create a custom policy that allows read-only access to your Secrets Manager seamless domain join secret (which you created earlier), and to create a new LinuxEC2DomainJoin IAM role.

Create the Secrets Manager IAM read policy

You use the IAM console to create a policy that grants read-only access to your Secrets Manager secret.

To create the Secrets Manager IAM read policy

1. Sign in to the AWS Management Console as a user that has permission to create IAM policies. Then open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Policies**.
3. Choose **Create policy**.
4. Choose the **JSON** tab and copy the text from the following JSON policy document. Then paste it into the **JSON** text box.

   Note
   Make sure you replace the `Resource` ARN with the actual ARN of the secret that you created earlier.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue",
                "secretsmanager:DescribeSecret"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:############:secret:aws/directory-
service/d-xxxxxxxxxx/seamless-domain-join-example"
            ]
        }
    ]
}
```

5. When you are finished, choose **Review policy**. The policy validator reports any syntax errors. For more information, see Validating IAM policies.
6. On the **Review policy** page, enter a policy name, such as **`SM-Secret-Linux-DJ-`*`d-xxxxxxxxxx`*`-Read`**. Review the **Summary** section to see the permissions that your policy grants. Then choose **Create policy** to save your changes. The new policy appears in the list of managed policies and is now ready to attach to an identity.

   Note
   We recommend you create one policy per secret. Doing so ensures that instances only have access to the appropriate secret and minimizes the impact if an instance is compromised.

Create the LinuxEC2DomainJoin role

You use the IAM console to create the role that you will use to domain join your Linux EC2 instance.

To create the LinuxEC2DomainJoin role

1. Sign in to the AWS Management Console as a user that has permission to create IAM policies. Then open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Roles**.
3. In the content pane, choose **Create role**.
4. Under **Select type of trusted entity**, choose **AWS service**.
5. Under **Choose a use case** choose **EC2**, and then choose **Next: Permissions**.

6. For **Filter policies**, do the following:

   a. Enter **`AmazonSSMManagedInstanceCore`**. Then select the check box for that item in the list.
   b. Enter **`AmazonSSMDirectoryServiceAccess`**. Then select the check box for that item in the list.
   c. Enter **`SM-Secret-Linux-DJ-`*`d-xxxxxxxxxx`*`-Read`** (or the name of the policy that you created in the previous procedure). Then select the check box for that item in the list.

   Note

   AmazonSSMDirectoryServiceAccess provides the permissions to join instances to an Active Directory managed by AWS Directory Service. AmazonSSMManagedInstanceCore provides the minimum permissions necessary to use the AWS Systems Manager service. For more information about creating a role with these permissions, and for information about other permissions and policies you can assign to your IAM role, see Create an IAM instance profile for Systems Manager in the *AWS Systems Manager User Guide*.

7. Choose **Next: Tags**.
8. (Optional) Add one or more tag key-value pairs to organize, track, or control access for this role. Then choose **Next: Review**.
9. Next to **Role name**, enter a name for your new role, such as **`LinuxEC2DomainJoin`** or another name that you prefer.
10. (Optional) For **Role description**, enter a description.
11. Choose **Create role**.

Seamlessly join your Linux EC2 instance

Now that you have configured all of the prerequisite tasks, you can use the following procedure to seamlessly join your Linux EC2 instance.

To seamlessly join your Linux EC2 instance

1. Sign in to the AWS Management Console and open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the Region selector in the navigation bar, choose the same Region as the existing directory.
3. Choose **Launch Instance**.
4. On the **Step 1** page, choose **Select** for the appropriate Amazon Machine Image (AMI).

   Note

   The AMI used must have AWS Systems Manager (SSM Agent) version 2.3.1644.0 or higher. To check the installed SSM Agent version in your AMI by launching an instance from that AMI, see Getting the currently installed SSM Agent version. If you need to upgrade the SSM Agent, see Installing and configuring SSM Agent on EC2 instances for Linux.

5. On the **Step 2** page, select the appropriate instance type, and then choose **Next: Configure Instance Details**.
6. On the **Step 3** page, do the following, and then choose **Next: Add Storage**:

   a. For **Network**, choose the VPC that your directory was created in.
   b. For **Subnet**, choose one of the public subnets in your VPC. The subnet that you choose must have all external traffic routed to an internet gateway. If this is not the case, you won't be able to connect to the instance remotely.
   c. For **Auto-assign Public IP**, choose **Enable**. For more information about public and private IP addressing, see Amazon EC2 instance IP addressing in the *Amazon EC2 User Guide for Windows Instances*.
   d. For **Domain join directory**, choose your domain from the list.

   e. For **IAM role**, choose the IAM role that you previously created in the prerequisites section **Step 2: Create the LinuxEC2DomainJoin Role**.

7. On both the **Step 4** and **Step 5** pages, leave the default settings or make changes as needed. Then choose **Next** on each.
8. On the **Step 6** page, select a security group that has been configured to allow remote access to the instance from your network. Then choose **Review and Launch**.
9. On the **Step 7** page, choose **Launch**, choose a key pair, and then choose **Launch Instance**.

Note
If you are performing a seamless domain join with SUSE Linux, a reboot is required before authentications will work. To reboot SUSE from the Linux terminal, type **sudo reboot**.

Maintain your AD Connector directory

This section describes how to maintain common administrative tasks for your AD Connector environment.

Topics

Delete your directory

When a Simple AD or AWS Directory Service for Microsoft Active Directory directory is deleted, all of the directory data and snapshots are deleted and cannot be recovered. After the directory is deleted, all instances that are joined to the directory remain intact. You cannot, however, use your directory credentials to log in to these instances. You need to log in to these instances with a user account that is local to the instance.

When an AD Connector directory is deleted, your on-premises directory remains intact. All instances that are joined to the directory also remain intact and remain joined to your on-premises directory. You can still use your directory credentials to log in to these instances. Any resources pointed at the now-deleted AD Connector will no longer resolve.

To delete a directory

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. Ensure that no AWS applications are enabled for the directory you intend to delete.
   a. On the **Directories** page, choose your directory ID.
   b. On the **Directory details** page, select the **Application management** tab. In the **AWS apps & services** section, you see which AWS applications are enabled for your directory.
      - To disable WorkSpaces, you must deregister the service from the directory in the WorkSpaces console. For more information, see Deregistering from a directory in the *Amazon WorkSpaces Administration Guide*.
      - To disable Amazon WorkSpaces Application Manager, you must remove all application assignments in the Amazon WAM console. For more information, see Removing all application assignments in the *Amazon WAM Administration Guide*.
      - To disable Amazon WorkDocs, you must delete the Amazon WorkDocs site in the Amazon WorkDocs console. For more information, see Delete a site in the *Amazon WorkDocs Administration Guide*.

- To disable Amazon WorkMail, you must remove the Amazon WorkMail organization in the Amazon WorkMail console. For more information, see Remove an organization in the *Amazon WorkMail Administrator Guide*.
- Disable AWS Management Console access.
- To disable Amazon Relational Database Service, you must remove the Amazon RDS instance from the domain. For more information, see Managing a DB instance in a domain in the *Amazon RDS User Guide*.
- To disable Amazon QuickSight, you must unsubscribe from Amazon QuickSight. For more information, see Closing your Amazon QuickSight account in the *Amazon QuickSight User Guide*.
- To disable Amazon Connect, you must delete the Amazon Connect Instance. For more information, see Deleting an Amazon Connect instance in the *Amazon Connect Administration Guide*.
- To disable FSx for Windows File Server, you must remove the Amazon FSx file system from the domain. For more information, see Working with Active Directory in FSx for Windows File Server in the *Amazon FSx for Windows File Server User Guide*.

> **Note**
> If you are using AWS Single Sign-On and have previously connected it to the AWS Managed Microsoft AD directory you plan to delete, you must first change the identity source before you can delete it. For more information, see Change your identity source in the *AWS SSO User Guide*.

3. In the navigation pane, choose **Directories**.
4. Select only the directory to be deleted and click **Delete**. It takes several minutes for the directory to be deleted. When the directory has been deleted, it is removed from your directory list.

To delete an inoperable AD Connector, see I cannot delete my AD Connector (p. 225).

View directory information

You can view detailed information about a directory.

To view detailed directory information

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. Click the directory ID link for your directory. Information about the directory is displayed in the **Directory details** page.

For more information about the **Status** field, see Understanding your directory status (p. 51).

Update the DNS address for your AD Connector

Use the following steps to update the DNS addresses that your AD Connector is pointing to.

> **Note**
> If you have an update in progress, you must wait until it is complete before submitting another update.

> **Note**
> If you are using WorkSpaces with your AD Connector, ensure that your WorkSpace’s DNS addresses are updated as well. For more information, see Update DNS servers for WorkSpaces.

To update your DNS settings for AD Connector

1. In the AWS Directory Service console navigation pane, choose **Directories**.
2. Choose the directory ID link for your directory.
3. On the **Directory details** page, choose **Network & security**.
4. In the **Existing DNS settings** section, choose **Update**.
5. In the **Update existing DNS addresses** dialog, type the updated DNS IP addresses, and then choose **Update**.

Best practices for AD Connector

Here are some suggestions and guidelines you should consider to avoid problems and get the most out of AD Connector.

Setting up: Prerequisites

Consider these guidelines before creating your directory.

Verify you have the right directory type

AWS Directory Service provides multiple ways to use Microsoft Active Directory with other AWS services. You can choose the directory service with the features you need at a cost that fits your budget:

- **AWS Directory Service for Microsoft Active Directory** is a feature-rich managed Microsoft Active Directory hosted on the AWS cloud. AWS Managed Microsoft AD is your best choice if you have more than 5,000 users and need a trust relationship set up between an AWS hosted directory and your on-premises directories.
- **AD Connector** simply connects your existing on-premises Active Directory to AWS. AD Connector is your best choice when you want to use your existing on-premises directory with AWS services.
- **Simple AD** is an inexpensive Active Directory–compatible service with the common directory features. In most cases, Simple AD is the least expensive option and your best choice if you have 5,000 or fewer users and don’t need the more advanced Microsoft Active Directory features.

For a more detailed comparison of AWS Directory Service options, see Which to choose (p. 1).

Ensure your VPCs and instances are configured correctly

In order to connect to, manage, and use your directories, you must properly configure the VPCs that the directories are associated with. See either AWS Managed Microsoft AD prerequisites (p. 9), AD Connector prerequisites (p. 187), or Simple AD prerequisites (p. 227) for information about the VPC security and networking requirements.

If you are adding an instance to your domain, ensure that you have connectivity and remote access to your instance as described in Join an EC2 instance to your AWS Managed Microsoft AD directory (p. 71).

Be aware of your limits

Learn about the various limits for your specific directory type. The available storage and the aggregate size of your objects are the only limitations on the number of objects you may store in your directory. See either AWS Managed Microsoft AD quotas (p. 152), AD Connector quotas (p. 222), or Simple AD quotas (p. 271) for details about your chosen directory.

Understand your directory's AWS security group configuration and use

AWS creates a security group and attaches it to your directory's elastic network interfaces that are accessible from within your peered or resized VPCs. AWS configures the security group to block unnecessary traffic to the directory and allows necessary traffic.

Modifying the directory security group

If you want to modify the security of your directories' security groups, you can do so. Make such changes only if you fully understand how security group filtering works. For more information, see Amazon EC2 security groups for Linux instances in the *Amazon EC2 User Guide*. Improper changes can result in loss of communications to intended computers and instances. AWS recommends that you do not attempt to open additional ports to your directory as this decreases the security of your directory. Please carefully review the AWS Shared Responsibility Model.

> **Warning**
> It is technically possible for you to associate the directory's security group with other EC2 instances that you create. However, AWS recommends against this practice. AWS may have reasons to modify the security group without notice to address functional or security needs of the managed directory. Such changes affect any instances with which you associate the directory security group and may disrupt operation of the associated instances. Furthermore, associating the directory security group with your EC2 instances may create a potential security risk for your EC2 instances.

Configure on-premises sites and subnets correctly when using AD Connector

If your on-premises network has Active Directory sites defined, you must make sure the subnets in the VPC where your AD Connector resides are defined in an Active Directory site, and that no conflicts exist between the subnets in your VPC and the subnets in your other sites.

To discover domain controllers, AD Connector uses the Active Directory site whose subnet IP address ranges are close to those in the VPC that contain the AD Connector. If you have a site whose subnets have the same IP address ranges as those in your VPC, AD Connector will discover the domain controllers in that site, which may not be physically close to your Region.

Understand username restrictions for AWS applications

AWS Directory Service provides support for most character formats that can be used in the construction of usernames. However, there are character restrictions that are enforced on usernames that will be used for signing in to AWS applications, such as WorkSpaces, Amazon WorkDocs, Amazon WorkMail, or Amazon QuickSight. These restrictions require that the following characters not be used:

- Spaces
- Multibyte characters
- !"#$%&'()*+,/:;<=>?@[\]^`{|}~

> **Note**
> The @ symbol is allowed as long as it precedes a UPN suffix.

Programming your applications

Before you program your applications, consider the following:

Load test before rolling out to production

Be sure to do lab testing with applications and requests that are representative of your production workload to confirm that the directory scales to the load of your application. Should you require additional capacity, spread your loads across multiple AD Connector directories.

Using your directory

Here are some suggestions to keep in mind when using your directory.

Rotate Admin credentials regularly

Change your AD Connector service account Admin password regularly, and make sure that the password is consistent with your existing Active Directory password policies. For instructions on how to change the service account password, see Update your AD Connector service account credentials in AWS Directory Service (p. 198).

Use unique AD Connectors for each domain

AD Connectors and your on-premises AD domains have a 1-to-1 relationship. That is, for each on-premises domain, including child domains in an AD forest that you want to authenticate against, you must create a unique AD Connector. Each AD Connector that you create must use a different service account, even if they are connected to the same directory.

Check for compatibility

When using AD Connector, you must ensure that your on-premises directory is and remains compatible with AWS Directory Services. For more information on your responsibilities, please see our shared responsibility model.

AD Connector quotas

The following are the default quotas for AD Connector. Each quota is per Region unless otherwise noted.

AD Connector quotas

Resource	Default quota
AD Connector directories	10
Maximum number of registered certificate authority (CA) certificates per directory	5

Application compatibility policy for AD Connector

As an alternative to AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD (p. 8)), AD Connector is an Active Directory proxy for AWS created applications and services only. You configure the proxy to use a specified Active Directory domain. When the application must look up a user or group in Active Directory, AD Connector proxies the request to the directory. Similarly, when a user logs in to the application, AD Connector proxies the authentication request to the directory. There are no third-party applications that work with AD Connector.

The following is a list of compatible AWS applications and services:

- Amazon Chime - For detailed instructions, see Connect to your Active Directory.
- Amazon Connect - For more information, see How Amazon Connect works.
- Amazon EC2 for Windows or Linux – You can use the seamless domain join feature of Amazon EC2 Windows or Linux to join your instance to your self-managed Active Directory (on-premises). Once joined, the instance communicates directly with your Active Directory and bypasses AD Connector. For more information, see Join an EC2 instance to your AD Connector directory (p. 212).
- AWS Management Console – You can use AD Connector to authenticate AWS Management Console users with their Active Directory credentials without setting up SAML infrastructure. For more information, see Enable access to the AWS Management Console with AD credentials (p. 141).
- Amazon QuickSight - For more information, see Managing user accounts in Amazon QuickSight Enterprise Edition.
- AWS Single Sign-On - For detailed instructions, see Connect AWS SSO to an on-premises Active Directory.
- AWS Transfer Family - For detailed instructions, see Working with AWS Directory Service for Microsoft Active Directory.
- AWS Client VPN - For detailed instructions, see Client authentication and authorization.
- Amazon WorkDocs - For detailed instructions, see Connecting to your on-premises directory with AD Connector.
- Amazon WorkMail - For detailed instructions, see Integrate Amazon WorkMail with an existing directory (standard setup).
- WorkSpaces - For detailed instructions, see Launch a WorkSpace using AD Connector.

> **Note**
> Amazon RDS is compatible with AWS Managed Microsoft AD only, and is not compatible with AD Connector. For more information, see the AWS Microsoft AD section in the AWS Directory Service FAQs page.

Troubleshooting AD Connector

The following can help you troubleshoot some common issues you might encounter when creating or using your directory.

Here are some common problems with AD Connector.

Seamless domain join for EC2 instances stopped working

If seamless domain join for EC2 instances was working and then stopped while the AD Connector was active, the credentials for your AD Connector service account may have expired. Expired credentials can prevent AD Connector from creating computer objects in your Active Directory.

To resolve this issue, update the service account passwords in the following order so that the passwords match:

1. Update the password for the service account in your Active Directory
2. Update the password for the service account in your AD Connector in AWS Directory Service

Updating the password only in AWS Directory Service does NOT push the password change to your existing on-premises Active Directory so it is important to do it in the order shown.

I receive a “Unable to Authenticate” error when using AWS applications to search for users or groups

You may experience errors when searching for users while using AWS applications, such as WorkSpaces or Amazon QuickSight, even while the AD Connector status was active. Expired credentials can prevent AD Connector from completing queries on objects in your Active Directory. Update the password for the service account using the ordered steps provided above.

I receive a "DNS unavailable" error when I try to connect to my on-premises directory

You receive an error message similar to the following when connecting to your on-premises directory:

```
DNS unavailable (TCP port 53) for IP: <DNS IP address>
```

AD Connector must be able to communicate with your on-premises DNS servers via TCP and UDP over port 53. Verify that your security groups and on-premises firewalls allow TCP and UDP communication over this port. For more information, see AD Connector prerequisites (p. 187).

I receive a "Connectivity issues detected" error when I try to connect to my on-premises directory

You receive an error message similar to the following when connecting to your on-premises directory:

```
Connectivity issues detected: LDAP unavailable (TCP port 389) for IP: <IP address>
Kerberos/authentication unavailable (TCP port 88) for IP: <IP address>
Please ensure that the listed ports are available and retry the operation.
```

AD Connector must be able to communicate with your on-premises domain controllers via TCP and UDP over the following ports. Verify that your security groups and on-premises firewalls allow TCP and UDP communication over these ports. For more information, see AD Connector prerequisites (p. 187).

- 88 (Kerberos)
- 389 (LDAP)

I receive an "SRV record" error when I try to connect to my on-premises directory

You receive an error message similar to one or more of the following when connecting to your on-premises directory:

```
SRV record for LDAP does not exist for IP: <DNS IP address>

SRV record for Kerberos does not exist for IP: <DNS IP address>
```

AD Connector needs to obtain the `_ldap._tcp.<DnsDomainName>` and `_kerberos._tcp.<DnsDomainName>` SRV records when connecting to your directory. You will get this error if the service cannot obtain these records from the DNS servers that you specified when connecting to your directory. For more information about these SRV records, see SRV record requirements (p. 188).

My directory is stuck in the "Requested" state

If you have a directory that has been in the "Requested" state for more than five minutes, try deleting the directory and recreating it. If this problem persists, contact the AWS Support Center.

I receive an "AZ Constrained" error when I create a directory

Some AWS accounts created before 2012 might have access to Availability Zones in the US East (N. Virginia), US West (N. California), or Asia Pacific (Tokyo) Region that do not support AWS Directory Service directories. If you receive an error such as this when creating a directory, choose a subnet in a different Availability Zone and try to create the directory again.

Some of my users cannot authenticate with my directory

Your user accounts must have Kerberos preauthentication enabled. This is the default setting for new user accounts, but it should not be modified. For more information about this setting, go to Preauthentication on Microsoft TechNet.

I receive an "Invalid Credentials" error when the service account used by AD Connector attempts to authenticate

This can occur if the hard drive on your domain controller runs out of space. Ensure that your domain controller's hard drives are not full.

I cannot delete my AD Connector

If your AD Connector switches to an inoperable state, you no longer have access to your domain controllers. We block the deletion of an AD Connector when there are still applications linked to it because one of those applications may still be using the directory. Contact AWS Support.

Simple Active Directory

Simple AD is a standalone managed directory that is powered by a Samba 4 Active Directory Compatible Server. It is available in two sizes.

- Small - Supports up to 500 users (approximately 2,000 objects including users, groups, and computers).
- Large - Supports up to 5,000 users (approximately 20,000 objects including users, groups, and computers).

Simple AD provides a subset of the features offered by AWS Managed Microsoft AD, including the ability to manage user accounts and group memberships, create and apply group policies, securely connect to Amazon EC2 instances, and provide Kerberos-based single sign-on (SSO). However, note that Simple AD does not support features such as multi-factor authentication (MFA), trust relationships with other domains, Active Directory Administrative Center, PowerShell support, Active Directory recycle bin, group managed service accounts, and schema extensions for POSIX and Microsoft applications.

Simple AD offers many advantages:

- Simple AD makes it easier to manage amazon EC2 instances running Linux and Windows and deploy Windows applications in the AWS Cloud.
- Many of the applications and tools that you use today that require Microsoft Active Directory support can be used with Simple AD.
- User accounts in Simple AD allow access to AWS applications such as WorkSpaces, Amazon WorkDocs, or Amazon WorkMail.
- You can manage AWS resources through IAM role–based access to the AWS Management Console.
- Daily automated snapshots enable point-in-time recovery.

Simple AD does not support any of the following:

- Amazon AppStream 2.0
- Amazon Chime
- Amazon RDS for SQL Server
- Amazon RDS for Oracle
- AWS Single Sign-On
- Trust relationships with other domains
- Active Directory Administrative Center
- PowerShell
- Active Directory recycle bin
- Group managed service accounts
- Schema extensions for POSIX and Microsoft applications

Continue reading the topics in this section to learn how to create your own Simple AD.

Topics

Getting started with Simple AD

Simple AD creates a fully managed, Samba-based directory in the AWS cloud. When you create a directory with Simple AD, AWS Directory Service creates two domain controllers and DNS servers on your behalf. The domain controllers are created in different subnets in a VPC; this redundancy helps ensures that your directory remains accessible even if a failure occurs.

Topics

Simple AD prerequisites

To create a Simple AD directory, you need a VPC with the following:

- At least two subnets. For Simple AD to install correctly, you must install your two domain controllers in separate subnets that must be in a different Availability Zone. In addition, the subnets must be in the same Classless Inter-Domain Routing (CIDR) range. If you want to extend or resize the VPC for your directory, then make sure to select both of the domain controller subnets for the extended VPC CIDR range.
- The VPC must have default hardware tenancy.
- The VPC must **not** be configured with the following VPC endpoint(s):
  - Route53 VPC endpoints that include DNS conditional overrides for *.amazonaws.com which resolve to non public AWS IP addresses
  - CloudWatch VPC endpoint
  - Systems Manager VPC endpoint
  - Security Token Service VPC endpoint
- If you require LDAPS support with Simple AD, we recommend that you configure it using a Network Load Balancer connected to port 389. This model enables you to use a strong certificate for the LDAPS connection, simplify access to LDAPS through a single NLB IP address, and have automatic fail-over through the NLB. Simple AD does not support the use of self-signed certificates on port 636. For more information about how to configure LDAPS with Simple AD, see How to configure an LDAPS endpoint for Simple AD in the *AWS Security Blog*.
- The following encryption types must be enabled in the directory:
  - RC4_HMAC_MD5
  - AES128_HMAC_SHA1
  - AES256_HMAC_SHA1
  - Future encryption types

    Note
    Disabling these encryption types can cause communication issues with RSAT (Remote Server Administration Tools) and impact the availability or your directory.

AWS Directory Service uses a two VPC structure. The EC2 instances which make up your directory run outside of your AWS account, and are managed by AWS. They have two network adapters, `ETH0` and `ETH1`. `ETH0` is the management adapter, and exists outside of your account. `ETH1` is created within your account.

The management IP range of your directory's `ETH0` network is chosen programmatically to ensure it does not conflict with the VPC where your directory is deployed. This IP range can be in either of the following pairs (as Directories run in two subnets):

- 10.0.1.0/24 & 10.0.2.0/24
- 192.168.1.0/24 & 192.168.2.0/24

We avoid conflicts by checking the first octet of the `ETH1` CIDR. If it starts with a 10, then we choose a 192.168.0.0/16 VPC with 192.168.1.0/24 and 192.168.2.0/24 subnets. If the first octet is anything else other than a 10 we choose a 10.0.0.0/16 VPC with 10.0.1.0/24 and 10.0.2.0/24 subnets.

The selection algorithm does not include routes on your VPC. It is therefore possible to have an IP routing conflict result from this scenario.

Create a Simple AD directory

To create a new directory, perform the following steps. Before starting this procedure, make sure you have completed the prerequisites identified in Simple AD prerequisites (p. 227).

To create a Simple AD directory

1. In the AWS Directory Service console navigation pane, choose **Directories** and then choose **Set up directory**.
2. On the **Select directory type** page, choose **Simple AD**, and then choose **Next**.
3. On the **Enter directory information** page, provide the following information:

   Directory size

      Choose from either the **Small** or **Large** size option. For more information about sizes, see Simple Active Directory (p. 226).

   Organization name

      A unique organization name for your directory that will be used to register client devices.

      This field is only available if you are creating your directory as part of launching WorkSpaces.

   Directory DNS name

      The fully qualified name for the directory, such as `corp.example.com`.

   Directory NetBIOS name

      The short name for the directory, such as `CORP`.

   Administrator password

      The password for the directory administrator. The directory creation process creates an administrator account with the user name `Administrator` and this password.

      The directory administrator password is case-sensitive and must be between 8 and 64 characters in length, inclusive. It must also contain at least one character from three of the following four categories:

      - Lowercase letters (a-z)

- Uppercase letters (A-Z)
- Numbers (0-9)
- Non-alphanumeric characters (~!@#$%^&*_-+=`|\(){}[]:;"'<>,.?/)

Confirm password

Retype the administrator password.

Directory description

An optional description for the directory.

4. On the **Choose VPC and subnets** page, provide the following information, and then choose **Next**.

   VPC

   The VPC for the directory.

   Subnets

   Choose the subnets for the domain controllers. The two subnets must be in different Availability Zones.

5. On the **Review & create** page, review the directory information and make any necessary changes. When the information is correct, choose **Create directory**. It takes several minutes for the directory to be created. Once created, the **Status** value changes to **Active**.

What gets created

When you create a directory with Simple AD, AWS Directory Service performs the following tasks on your behalf:

- Sets up a Samba-based directory within the VPC.
- Creates a directory administrator account with the user name `Administrator` and the specified password. You use this account to manage your directory.

  Important

  Be sure to save this password. AWS Directory Service does not store this password, and it cannot be retrieved. However, you can reset a password from the AWS Directory Service console or by using the ResetUserPassword API.
- Creates a security group for the directory controllers.
- Creates an account with the name `AWSAdminD-xxxxxxxx` that has domain admin privileges. This account is used by AWS Directory Service to perform automated operations for directory maintenance operations, such as taking directory snapshots and FSMO role transfers. The credentials for this account are securely stored by AWS Directory Service.
- Automatically creates and associates an elastic network interface (ENI) with each of your domain controllers. Each of these ENIs are essential for connectivity between your VPC and AWS Directory Service domain controllers and should never be deleted. You can identify all network interfaces reserved for use with AWS Directory Service by the description: "AWS created network interface for directory *directory-id*". For more information, see Elastic Network Interfaces in the Amazon EC2 User Guide for Windows Instances.

  Note

  Domain controllers are deployed across two Availability Zones in a region by default and connected to your Amazon Virtual Private Cloud (VPC). Backups are automatically taken once per day, and the Amazon Elastic Block Store (EBS) volumes are encrypted to ensure that data is secured at rest. Domain controllers that fail are automatically replaced in the same Availability Zone using the same IP address, and a full disaster recovery can be performed using the latest backup.

Configure DNS

Simple AD forwards DNS requests to the IP address of the Amazon-provided DNS servers for your VPC. These DNS servers will resolve names configured in your Route 53 private hosted zones. By pointing your on-premises computers to your Simple AD, you can now resolve DNS requests to the private hosted zone.

Note that to enable your Simple AD to respond to external DNS queries, the network access control list (ACL) for the VPC containing your Simple AD must be configured to allow traffic from outside the VPC.

If you are not using Route 53 private hosted zones, your DNS requests will be forwarded to public DNS servers.

If you're using custom DNS servers that are outside of your VPC and you want to use private DNS, you must reconfigure to use custom DNS servers on EC2 instances within your VPC. For more information, see Working with private hosted zones.

If you want your Simple AD to resolve names using both DNS servers within your VPC and private DNS servers outside of your VPC, you can do this using a DHCP options set. For a detailed example, see this article.

> **Note**
> DNS dynamic updates are not supported in Simple AD domains. You can instead make the changes directly by connecting to your directory using DNS Manager on an instance that is joined to your domain.

For more information on Route 53, see What is Route 53.

How to administer Simple AD

This section lists all of the procedures for operating and maintaining an Simple AD environment.

Topics

Manage users and groups in Simple AD

Users represent individual people or entities that have access to your directory. Groups are very useful for giving or denying privileges to groups of users, rather than having to apply those privileges to each individual user. If a user moves to a different organization, you move that user to a different group and they automatically receive the privileges needed for the new organization.

To create users and groups in an AWS Directory Service directory, you must use any instance (from either on-premises or EC2) that has been joined to your AWS Directory Service directory, and be logged in as a user that has privileges to create users and groups. You will also need to install the Active Directory Tools on your EC2 instance so you can add your users and groups with the Active Directory Users and Computers snap-in. For more information about how to set up an EC2 instance and install the necessary tools, see Seamlessly join a Linux EC2 instance to your Simple AD directory (p. 240).

Note
Your user accounts must have Kerberos preauthentication enabled. This is the default setting for new user accounts, but it should not be modified. For more information about this setting, go to Preauthentication on Microsoft TechNet.

The following topics include instructions on how to create and manage users and groups.

Topics

Installing the Active Directory administration tools

To manage your directory from an EC2 Windows instance, you need to install the Active Directory Domain Services and Active Directory Lightweight Directory Services Tools on the instance. Use the following procedure to install these tools on either Windows Server 2012, Windows Server 2016, or Windows Server 2019.

You can optionally choose to install the Active Directory administration tools using Windows PowerShell. For example, you can install the Active Directory remote administration tools from a PowerShell prompt using `Install-WindowsFeature RSAT-ADDS`. For more information, see Install-WindowsFeature on the Microsoft Website.

Install the Active Directory Administration Tools on Windows Server 2012 through Windows Server 2019

To install the Active Directory administration tools on Windows Server 2012 through Windows Server 2019

1. Open Server Manager from the Start screen by choosing **Server Manager**.
2. In the **Server Manager Dashboard**, choose **Add roles and features**,
3. In the **Add Roles and Features Wizard** choose **Installation Type**, select **Role-based or feature-based installation**, and choose **Next**.
4. Under **Server Selection**, make sure the local server is selected, and choose **Features** in the left navigation pane.
5. In the **Features** tree, open **Remote Server Administration Tools**, **Role Administration Tools**, select **AD DS and AD LDS Tools**, scroll down and select **DNS Server Tools**, and then choose **Next**.
6. Review the information and choose **Install**. When the feature installation is finished, the Active Directory Domain Services and Active Directory Lightweight Directory Services Tools are available on the Start screen in the **Administrative Tools** folder.

Create a user

Note
When using Simple AD, if you create a user account on a Linux instance with the option "Force user to change password at first login," that user will not be able to initially change their password using **kpasswd**. In order to change the password the first time, a domain administrator must update the user password using the Active Directory Management Tools.

Use the following procedure to create a user with an EC2 instance that is joined to your Simple AD directory.

To create a user

1. Open the Active Directory Users and Computers tool. There is a shortcut to this tool in the **Administrative Tools** folder.

   Tip
   You can run the following from a command prompt on the instance to open the Active Directory Users and Computers tool box directly.

   ```
   %SystemRoot%\system32\dsa.msc
   ```
2. In the directory tree, select an OU under your directory's NetBIOS name OU where you want to store your user (for example, Corp\Users). For more information about the OU structure used by directories in AWS, see What gets created (p. 12).
3. On the **Action** menu, click **New**, and then click **User** to open the new user wizard.
4. On the first page of the wizard, enter the values for the following fields, and then click **Next**.

   - **First name**
   - **Last name**
   - **User logon name**
5. On the second page of the wizard, type a temporary password in **Password** and **Confirm Password**. Make sure the **User must change password at next logon** option is selected. None of the other options should be selected. Click **Next**.
6. On the third page of the wizard, verify that the new user information is correct and click **Finish**. The new user will appear in the **Users** folder.

Delete a user

Use the following procedure to delete a user with an EC2 Windows instance that is joined to your Simple AD directory.

To delete a user

1. Open the Active Directory Users and Computers tool. There is a shortcut to this tool in the **Administrative Tools** folder.

   Tip
   You can run the following from a command prompt on the instance to open the Active Directory Users and Computers tool box directly.

   ```
   %SystemRoot%\system32\dsa.msc
   ```
2. In the directory tree, select the OU containing the user that you want to delete (for example, Corp \Users).
3. Choose **Accounts**. Then, select the user you want to delete. Right-click **Name** and choose **Delete** from the context menu.
4. Choose **Yes** in the dialog box **Are you sure you want to delete this object?** This permanently deletes the selected user.

Reset a user password

Users must adhere to password policies as defined in the directory. Sometimes this can get the best of users, including the directory admin, and they forget their password. When this happens, you can quickly reset the user's password using AWS Directory Service if the user resides in either a Simple AD or AWS Managed Microsoft AD directory.

You can reset the password for any user in your directory with the following exceptions:

- For Simple AD, you cannot reset the password for any user that is a member of either the **Domain Admins** or **Enterprise Admins** group except for the Administrator user.
- For AWS Managed Microsoft AD, you cannot reset the password for any user that is in an OU other than the OU that is based off of the NetBIOS name you typed when you created your directory. For example, you cannot reset the password for a user in the **AWS Reserved** OU. For more information about the OU structure for an AWS Managed Microsoft AD directory, see What gets created (p. 12).

You can use any of the following methods to reset a user's password.

Method 1: To reset a user password (AWS Management Console)

1. In the AWS Directory Service console navigation pane, under **Active Directory**, choose **Directories**, and then select the directory in the list where you want to reset a user's password.
2. On the **Directory details** page, choose **Reset user password**.
3. In the **Reset user password** dialog, in **Username** type the user name of the user whose password needs to change.
4. Type a password in **New password** and **Confirm password**, and then choose **Reset password**.

Method 2: To reset a user password (Windows PowerShell)

1. Open Windows PowerShell.
2. Type the following command and replace the user name "joebob" and password "P@ssw0rd" with your desired credentials. See Reset-DSUserPassword Cmdlet for more information.

```
Reset-DSUserPassword -UserName joebob -DirectoryId d-1234567890 -NewPassword
P@ssw0rd
```

Method 3: To reset a user password (AWS CLI)

1. Open the AWS CLI.
2. Type the following command and replace the user name "joebob" and password "P@ssw0rd" with your desired credentials. See reset-user-password in the *AWS CLI Command Reference* for more information.

```
aws ds reset-user-password --directory-id d-1234567890 --user-name joebob --
new-password P@ssw0rd
```

Create a group

Use the following procedure to create a security group with an EC2 instance that is joined to your Simple AD directory.

To create a group

1. Open the Active Directory Users and Computers tool. There is a shortcut to this tool in the **Administrative Tools** folder.

Tip
You can run the following from a command prompt on the instance to open the Active Directory Users and Computers tool box directly.

```
%SystemRoot%\system32\dsa.msc
```

2. In the directory tree, select an OU under your directory's NetBIOS name OU where you want to store your group (for example, Corp\Users). For more information about the OU structure used by directories in AWS, see What gets created (p. 12).
3. On the **Action** menu, click **New**, and then click **Group** to open the new group wizard.
4. Type a name for the group in **Group name**, select a **Group scope**, and select **Security** for the **Group type**.
5. Click **OK**. The new security group will appear in the **Users** folder.

Add a user to a group

Use the following procedure to add a user to a security group with an EC2 instance that is joined to your Simple AD directory.

To add a user to a group

1. Open the Active Directory Users and Computers tool. There is a shortcut to this tool in the **Administrative Tools** folder.

   Tip
   You can run the following from a command prompt on the instance to open the Active Directory Users and Computers tool box directly.

   ```
   %SystemRoot%\system32\dsa.msc
   ```
2. In the directory tree, select the OU under your directory's NetBIOS name OU where you stored your group, and select the group that you want to add a user as a member.
3. On the **Action** menu, click **Properties** to open the properties dialog box for the group.
4. Select the **Members** tab and click **Add**.
5. For **Enter the object names to select**, type the user name you want to add and click **OK**. The name will be displayed in the **Members** list. Click **OK** again to update the group membership.
6. Verify that the user is now a member of the group by selecting the user in the **Users** folder and clicking **Properties** in the **Action** menu to open the properties dialog box. Select the **Member Of** tab. You should see the name of the group in the list of groups that the user belongs to.

Monitor your Simple AD directory

You can monitor your Simple AD directory with the following methods:

Topics

Understanding your directory status

The following are the various statuses for a directory.

Active

The directory is operating normally. No issues have been detected by the AWS Directory Service for your directory.

Creating

The directory is currently being created. Directory creation typically takes between 20 to 45 minutes but may vary depending on the system load.

Deleted

The directory has been deleted. All resources for the directory have been released. Once a directory enters this state, it cannot be recovered.

Deleting

The directory is currently being deleted. The directory will remain in this state until it has been completely deleted. Once a directory enters this state, the delete operation cannot be cancelled, and the directory cannot be recovered.

Failed

The directory could not be created. Please delete this directory. If this problem persists, please contact the AWS Support Center.

Impaired

The directory is running in a degraded state. One or more issues have been detected, and not all directory operations may be working at full operational capacity. There are many potential reasons for the directory being in this state. These include normal operational maintenance activity such as patching or EC2 instance rotation, temporary hot spotting by an application on one of your domain controllers, or changes you made to your network that inadvertently disrupt directory communications. For more information, see either Troubleshooting AWS Managed Microsoft AD (p. 175), Troubleshooting AD Connector (p. 223), Troubleshooting Simple AD (p. 272). For normal maintenance related issues, AWS resolves these issues within 40 minutes. If after reviewing the troubleshooting topic, your directory is in an Impaired state longer than 40 minutes, we recommend that you contact the AWS Support Center.

> **Important**
> Do not restore a snapshot while a directory is in an Impaired state. It is rare that snapshot restore is necessary to resolve impairments. For more information, see Snapshot or restore your directory (p. 128).

Inoperable

The directory is not functional. All directory endpoints have reported issues.

Requested

A request to create your directory is currently pending.

RestoreFailed

Restoring the directory from a snapshot failed. Please retry the restore operation. If this continues, try a different snapshot, or contact the AWS Support Center.

Restoring

The directory is currently being restored from an automatic or manual snapshot. Restoring from a snapshot typically takes several minutes, depending on the size of the directory data in the snapshot.

For more information, see Simple AD directory status reasons (p. 274).

Configure directory status notifications

Using Amazon Simple Notification Service (Amazon SNS), you can receive email or text (SMS) messages when the status of your directory changes. You get notified if your directory goes from an Active status to an Impaired or Inoperable status. You also receive a notification when the directory returns to an Active status.

How it works

Amazon SNS uses “topics” to collect and distribute messages. Each topic has one or more subscribers who receive the messages that have been published to that topic. Using the steps below you can add AWS Directory Service as publisher to an Amazon SNS topic. When AWS Directory Service detects a change in your directory’s status, it publishes a message to that topic, which is then sent to the topic's subscribers.

You can associate multiple directories as publishers to a single topic. You can also add directory status messages to topics that you’ve previously created in Amazon SNS. You have detailed control over who can publish to and subscribe to a topic. For complete information about Amazon SNS, see What is Amazon SNS?.

To enable SNS messaging for your directory

1. Sign in to the AWS Management Console and open the AWS Directory Service console at https://console.aws.amazon.com/directoryservicev2/.
2. On the **Directories** page, choose your directory ID.
3. Select the **Maintenance** tab.
4. In the **Directory monitoring** section, choose **Actions**, and then select **Create notification**.
5. On the **Create notification** page, select **Choose a notification type**, and then choose **Create a new notification**. Alternatively, if you already have an existing SNS topic, you can choose **Associate existing SNS topic** to send status messages from this directory to that topic.

   > **Note**
   > If you choose **Create a new notification** but then use the same topic name for an SNS topic that already exists, Amazon SNS does not create a new topic, but just adds the new subscription information to the existing topic.
   > If you choose **Associate existing SNS topic**, you will only be able to choose an SNS topic that is in the same Region as the directory.

6. Choose the **Recipient type** and enter the **Recipient** contact information. If you enter a phone number for SMS, use numbers only. Do not include dashes, spaces, or parentheses.
7. (Optional) Provide a name for your topic and an SNS display name. The display name is a short name up to 10 characters that is included in all SMS messages from this topic. When using the SMS option, the display name is required.

   > **Note**
   > If you are logged in using an IAM user or role that has only the DirectoryServiceFullAccess managed policy, your topic name must start with “DirectoryMonitoring”. If you’d like to further customize your topic name you’ll need additional privileges for SNS.

8. Choose **Create**.

If you want to designate additional SNS subscribers, such as an additional email address, Amazon SQS queues or AWS Lambda, you can do this from the Amazon SNS console at https://console.aws.amazon.com/sns/v3/home.

To remove directory status messages from a topic

1. Sign in to the AWS Management Console and open the AWS Directory Service console at https://console.aws.amazon.com/directoryservicev2/.

2. On the **Directories** page, choose your directory ID.
3. Select the **Maintenance** tab.
4. In the **Directory monitoring** section, select an SNS topic name in the list, choose **Actions**, and then select **Remove**.
5. Choose **Remove**.

This removes your directory as a publisher to the selected SNS topic. If you want to delete the entire topic, you can do this from the Amazon SNS console at https://console.aws.amazon.com/sns/v3/home.

> **Note**
> Before deleting an Amazon SNS topic using the SNS console, you should ensure that a directory is not sending status messages to that topic.
> If you delete an Amazon SNS topic using the SNS console, this change will not immediately be reflected within the Directory Services console. You would only be notified the next time a directory publishes a notification to the deleted topic, in which case you would see an updated status on the directory's **Monitoring** tab indicating the topic could not be found.
> Therefore, to avoid missing important directory status messages, before deleting any topic that receives messages from AWS Directory Service, associate your directory with a different Amazon SNS topic.

Join an EC2 instance to your Simple AD directory

You can seamlessly join an EC2 instance to your directory domain when the instance is launched using AWS Systems Manager. For more information, see Seamlessly joining a Windows instance to an AWS Directory Service domain in the *Amazon EC2 User Guide for Windows Instances*.

If you need to manually join an EC2 instance to your domain, you must launch the instance in the proper Region and security group or subnet, then join the instance to the domain.

To be able to connect remotely to these instances, you must have IP connectivity to the instances from the network you are connecting from. In most cases, this requires that an internet gateway be attached to your VPC and that the instance has a public IP address.

Topics

Seamlessly join a Windows EC2 instance

This procedure seamlessly joins a Windows EC2 instance to your Simple AD directory.

To seamlessly join a Windows EC2 instance

1. Sign in to the AWS Management Console and open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the Region selector in the navigation bar, choose the same Region as the existing directory.
3. Choose **Launch Instance**.
4. On the **Step 1** page, choose **Select** for the appropriate AMI.

5. On the **Step 2** page, select the appropriate instance type, and then choose **Next: Configure Instance Details**.
6. On the **Step 3** page, do the following, and then choose **Next: Add Storage**:

    1. For **Network**, choose the VPC that your directory was created in.
    2. For **Subnet**, choose one of the public subnets in your VPC. The subnet that you choose must have all external traffic routed to an internet gateway. If this is not the case, you won't be able to connect to the instance remotely.
    3. For **Auto-assign Public IP**, choose **Enable**.

        For more information about public and private IP addressing, see Amazon EC2 instance IP addressing in the *Amazon EC2 User Guide for Windows Instances*.
    4. For **Domain join directory**, choose your domain from the list.
    5. For **IAM role**, do one of the following:

        Select an IAM role that has the AWS managed policies **AmazonSSMManagedInstanceCore** and **AmazonSSMDirectoryServiceAccess** attached to it.

        -or-

        If you haven't created an IAM role that has the **AmazonSSMManagedInstanceCore** and **AmazonSSMDirectoryServiceAccess** managed policies attached to it, choose the **Create new IAM role** link, and then do the following:

        a. Choose **Create role**.

        b. Under **Select type of trusted entity**, choose **AWS service**.

        c. Under **Choose the service that this role will use**, in the full list of services, choose **EC2**.

        d. Under **Select your use case**, choose **EC2**, and the choose **Next: Permissions**.

        e. In the list of policies, select the **AmazonSSMManagedInstanceCore** and **AmazonSSMDirectoryServiceAccess** policies. (To filter the list, type `SSM` in the search box.)

        > **Note**
        > **AmazonSSMDirectoryServiceAccess** provides the permissions to join instances to an Active Directory managed by AWS Directory Service. **AmazonSSMManagedInstanceCore** provides the minimum permissions necessary to use the Systems Manager service. For more information about creating a role with these permissions, and for information about other permissions and policies you can assign to your IAM role, see Create an IAM instance profile for Systems Manager in the *AWS Systems Manager User Guide*.

        f. Choose **Next: Tags**.

        g. (Optional) Add one or more tag key-value pairs to organize, track, or control access for this role, and then choose **Next: Review**.

        h. For **Role name**, enter a name for your new role, such as `EC2DomainJoin` or another name that you prefer.

        i. (Optional) For **Role description**, enter a description.

        j. Choose **Create role**.

        k. Go back to the **Step 3** page. For **IAM role**, choose the refresh icon next to **IAM role**. Your new role should be visible in the menu. Choose it and leave the rest of the settings on this page with their default values, and then choose **Next: Add Storage**.
7. On both the **Step 4** and **Step 5** pages, leave the default settings or make changes as needed, and then choose the **Next** buttons.
8. On the **Step 6** page, select a security group for the instance that has been configured to allow remote access to the instance from your network, and then choose **Review and Launch**.
9. On the **Step 7** page, choose **Launch**, select a key pair, and then choose **Launch Instance**.

Manually join a Windows instance

To manually join an existing Amazon EC2 Windows instance to a Simple AD or AWS Directory Service for Microsoft Active Directory directory, the instance must be launched as specified in Seamlessly join a Windows EC2 instance (p. 72).

To join a Windows instance to a Simple AD or AWS Managed Microsoft AD directory

1. Connect to the instance using any Remote Desktop Protocol client.
2. Open the TCP/IPv4 properties dialog box on the instance.

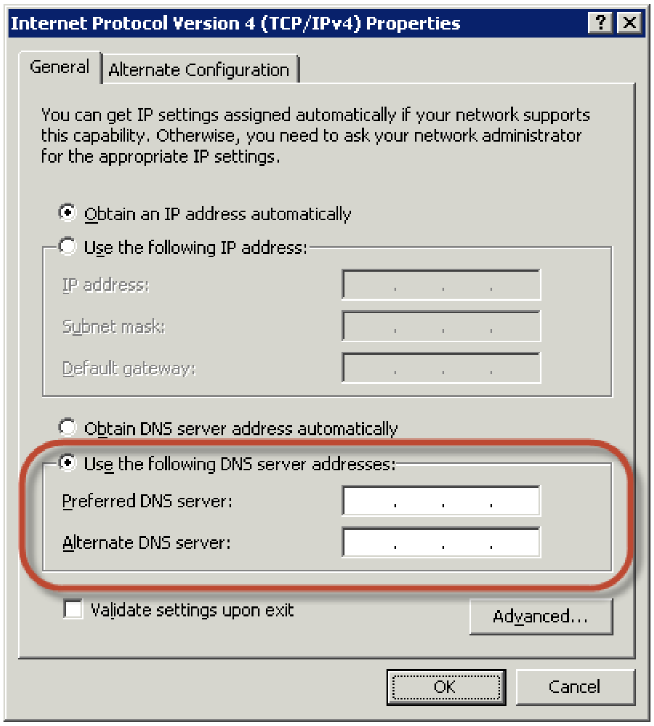
   a. Open **Network Connections**.

      Tip
      You can open **Network Connections** directly by running the following from a command prompt on the instance.

      ```
      %SystemRoot%\system32\control.exe ncpa.cpl
      ```

   b. Open the context menu (right-click) for any enabled network connection and then choose **Properties**.

   c. In the connection properties dialog box, open (double-click) **Internet Protocol Version 4**.
3. Select **Use the following DNS server addresses**, change the **Preferred DNS server** and **Alternate DNS server** addresses to the IP addresses of the AWS Directory Service-provided DNS servers, and choose **OK**.



Internet Protocol Version 4 (TCP/IPv4) Properties

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

● Obtain an IP address automatically
○ Use the following IP address:
IP address: . . .
Subnet mask: . . .
Default gateway: . . .

○ Obtain DNS server address automatically
● Use the following DNS server addresses:
Preferred DNS server: . . .
Alternate DNS server: . . .

□ Validate settings upon exit

4. Open the **System Properties** dialog box for the instance, select the **Computer Name** tab, and choose **Change**.

   Tip
   You can open the **System Properties** dialog box directly by running the following from a command prompt on the instance.

```
%SystemRoot%\system32\control.exe sysdm.cpl
```

5. In the **Member of** field, select **Domain**, enter the fully qualified name of your AWS Directory Service directory, and choose **OK**.
6. When prompted for the name and password for the domain administrator, enter the user name and password of an account that has domain join privileges. For more information about delegating these privileges, see Delegate directory join privileges for AWS Managed Microsoft AD (p. 98).

   Note
   You can enter either the fully qualified name of your domain or the NetBios name, followed by a backslash (\), and then the user name.
   If using AWS Managed Microsoft AD, the user name would be **Admin**. For example, **corp.example.com\admin** or **corp\admin**.
   If using Simple AD, the user name would be **Administrator**. For example, **corp.example.com\administrator** or **corp\administrator**.
7. After you receive the message welcoming you to the domain, restart the instance to have the changes take effect.

Now that your instance has been joined to the domain, you can log into that instance remotely and install utilities to manage the directory, such as adding users and groups.

Seamlessly join a Linux EC2 instance to your Simple AD directory

This procedure seamlessly joins a Linux EC2 instance to your Simple AD directory.

The following Linux instance distributions and versions are supported:

- Amazon Linux AMI 2018.03.0
- Amazon Linux 2 (64-bit x86)
- Red Hat Enterprise Linux 8 (HVM) (64-bit x86)
- Ubuntu Server 18.04 LTS & Ubuntu Server 16.04 LTS
- CentOS 7 x86-64
- SUSE Linux Enterprise Server 15 SP1

Note
Distributions prior to Ubuntu 14 and Red Hat Enterprise Linux 7 do not support the seamless domain join feature.

Prerequisites

Before you can set up seamless domain join to a Linux EC2 instance, you need to complete the procedures in this section.

Select your seamless domain join service account

You can seamlessly join Linux computers to your Simple AD domain. To do that, you must create a user account with create computer account permissions to join the computers to the domain. Although members of the *Domain Admins* or other groups may have sufficient privileges to join computers to the domain, we do not recommend this. As a best practice, we recommend you use a service account that has the minimum privileges necessary to join the computers to the domain.

For information about how to process and delegate permissions to your service account for computer account creation, see Delegate privileges to your service account (p. 189).

Create the secrets to store the domain service account

You can use AWS Secrets Manager to store the domain service account.

To create secrets and store the domain service account information

1. Sign in to the AWS Management Console and open the AWS Secrets Manager console at https://console.aws.amazon.com/secretsmanager/.
2. Choose **Store a new secret**.
3. On the **Store a new secret** page, do the following:

    a. Under **Select secret type**, choose **Other type of secrets**.

    b. Under **Specify the key/value pairs to be stored in the secret**, do the following:

        i. In the first box, enter **awsSeamlessDomainUsername**. On the same row, in the next box, enter the user name for your service account. For example, if you used the PowerShell command previously, the service account name would be awsSeamlessDomain.

            Note
            You must enter **awsSeamlessDomainUsername** exactly as it is. Make sure there are not any leading or ending spaces. Otherwise the domain join will fail.

        ii. Choose **Add row**.

        iii. On the new row, in the first box, enter **awsSeamlessDomainPassword**. On the same row, in the next box, enter the password for your service account.

            Note
            You must enter **awsSeamlessDomainPassword** exactly as it is. Make sure there are not any leading or ending spaces. Otherwise the domain join will fail.

        iv. Under **Select the encryption key**, choose **DefaultEncryptionKey** from the menu. Secrets Manager always encrypts the secret when you choose this option, and provides it at no charge to you. You also may choose a key you created.

        v. Choose **Next**.

4. Under **Secret name**, enter a secret name that includes your directory ID using the following format:

    ```
    aws/directory-services/d-xxxxxxxxx/seamless-domain-join
    ```

    This will be used to retrieve secrets in the application.

    Note
    You must enter **aws/directory-services/*d-xxxxxxxxx*/seamless-domain-join** exactly as it is but replace *d-xxxxxxxxxxx* with your directory ID. Make sure that there are no leading or ending spaces. Otherwise the domain join will fail.

5. Leave everything else set to defaults, and then choose **Next**.
6. Under **Configure automatic rotation**, choose **Disable automatic rotation**, and then choose **Next**.
7. Review the settings, and then choose **Store** to save your changes. The Secrets Manager console returns you to the list of secrets in your account with your new secret now included in the list.
8. Choose your newly created secret name from the list, and take note of the **Secret ARN** value. You will need it in the next section.

Create the required IAM policy and role

Use the following prerequisite steps to create a custom policy that allows read-only access to your Secrets Manager seamless domain join secret (which you created earlier), and to create a new LinuxEC2DomainJoin IAM role.

Create the Secrets Manager IAM read policy

You use the IAM console to create a policy that grants read-only access to your Secrets Manager secret.

To create the Secrets Manager IAM read policy

1. Sign in to the AWS Management Console as a user that has permission to create IAM policies. Then open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Policies**.
3. Choose **Create policy**.
4. Choose the **JSON** tab and copy the text from the following JSON policy document. Then paste it into the **JSON** text box.

   Note
   Make sure you replace the `Resource` ARN with the actual ARN of the secret that you created earlier.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue",
                "secretsmanager:DescribeSecret"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:############:secret:aws/directory-
service/d-xxxxxxxxxx/seamless-domain-join-example"
            ]
        }
    ]
}
```

5. When you are finished, choose **Review policy**. The policy validator reports any syntax errors. For more information, see Validating IAM policies.
6. On the **Review policy** page, enter a policy name, such as **`SM-Secret-Linux-DJ-`*`d-xxxxxxxxxx`*`-Read`**. Review the **Summary** section to see the permissions that your policy grants. Then choose **Create policy** to save your changes. The new policy appears in the list of managed policies and is now ready to attach to an identity.

   Note
   We recommend you create one policy per secret. Doing so ensures that instances only have access to the appropriate secret and minimizes the impact if an instance is compromised.

Create the LinuxEC2DomainJoin role

You use the IAM console to create the role that you will use to domain join your Linux EC2 instance.

To create the LinuxEC2DomainJoin role

1. Sign in to the AWS Management Console as a user that has permission to create IAM policies. Then open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Roles**.
3. In the content pane, choose **Create role**.
4. Under **Select type of trusted entity**, choose **AWS service**.
5. Under **Choose a use case** choose **EC2**, and then choose **Next: Permissions**.

6. For **Filter policies**, do the following:

   a. Enter **`AmazonSSMManagedInstanceCore`**. Then select the check box for that item in the list.

   b. Enter **`AmazonSSMDirectoryServiceAccess`**. Then select the check box for that item in the list.

   c. Enter **`SM-Secret-Linux-DJ-`*`d-xxxxxxxxxx`*`-Read`** (or the name of the policy that you created in the previous procedure). Then select the check box for that item in the list.

   Note

   AmazonSSMDirectoryServiceAccess provides the permissions to join instances to an Active Directory managed by AWS Directory Service. AmazonSSMManagedInstanceCore provides the minimum permissions necessary to use the AWS Systems Manager service. For more information about creating a role with these permissions, and for information about other permissions and policies you can assign to your IAM role, see Create an IAM instance profile for Systems Manager in the *AWS Systems Manager User Guide*.

7. Choose **Next: Tags**.
8. (Optional) Add one or more tag key-value pairs to organize, track, or control access for this role. Then choose **Next: Review**.
9. Next to **Role name**, enter a name for your new role, such as **`LinuxEC2DomainJoin`** or another name that you prefer.
10. (Optional) For **Role description**, enter a description.
11. Choose **Create role**.

Seamlessly join your Linux EC2 instance

Now that you have configured all of the prerequisite tasks, you can use the following procedure to seamlessly join your Linux EC2 instance.

To seamlessly join your Linux EC2 instance

1. Sign in to the AWS Management Console and open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the Region selector in the navigation bar, choose the same Region as the existing directory.
3. Choose **Launch Instance**.
4. On the **Step 1** page, choose **Select** for the appropriate Amazon Machine Image (AMI).

   Note

   The AMI used must have AWS Systems Manager (SSM Agent) version 2.3.1644.0 or higher. To check the installed SSM Agent version in your AMI by launching an instance from that AMI, see Getting the currently installed SSM Agent version. If you need to upgrade the SSM Agent, see Installing and configuring SSM Agent on EC2 instances for Linux.

5. On the **Step 2** page, select the appropriate instance type, and then choose **Next: Configure Instance Details**.
6. On the **Step 3** page, do the following, and then choose **Next: Add Storage**:

   a. For **Network**, choose the VPC that your directory was created in.

   b. For **Subnet**, choose one of the public subnets in your VPC. The subnet that you choose must have all external traffic routed to an internet gateway. If this is not the case, you won't be able to connect to the instance remotely.

   c. For **Auto-assign Public IP**, choose **Enable**. For more information about public and private IP addressing, see Amazon EC2 instance IP addressing in the *Amazon EC2 User Guide for Windows Instances*.

   d. For **Domain join directory**, choose your domain from the list.

   e. For **IAM role**, choose the IAM role that you previously created in the prerequisites section **Step 2: Create the LinuxEC2DomainJoin Role**.

7. On both the **Step 4** and **Step 5** pages, leave the default settings or make changes as needed. Then choose **Next** on each.
8. On the **Step 6** page, select a security group that has been configured to allow remote access to the instance from your network. Then choose **Review and Launch**.
9. On the **Step 7** page, choose **Launch**, choose a key pair, and then choose **Launch Instance**.

> **Note**
> If you are performing a seamless domain join with SUSE Linux, a reboot is required before authentications will work. To reboot SUSE from the Linux terminal, type **sudo reboot**.

Manually join a Linux instance

In addition to Amazon EC2 Windows instances, you can also join certain Amazon EC2 Linux instances to your Simple AD directory. The following Linux instance distributions and versions are supported:

- Amazon Linux AMI 2018.03.0
- Amazon Linux 2 (64-bit x86)
- Red Hat Enterprise Linux 8 (HVM) (64-bit x86)
- Ubuntu Server 18.04 LTS & Ubuntu Server 16.04 LTS
- CentOS 7 x86-64
- SUSE Linux Enterprise Server 15 SP1

> **Note**
> Other Linux distributions and versions may work but have not been tested.

Join an instance to your directory

Before you can join either an Amazon Linux, CentOS, Red Hat, or Ubuntu instance to your directory, the instance must first be launched as specified in Seamlessly join a Windows EC2 instance (p. 72).

> **Important**
> Some of the following procedures, if not performed correctly, can render your instance unreachable or unusable. Therefore, we strongly suggest you make a backup or take a snapshot of your instance before performing these procedures.

To join a linux instance to your directory

Follow the steps for your specific Linux instance using one of the following tabs:

Amazon Linux

1. Connect to the instance using any SSH client.
2. Configure the Linux instance to use the DNS server IP addresses of the AWS Directory Service-provided DNS servers. You can do this either by setting it up in the DHCP Options set attached to the VPC or by setting it manually on the instance. If you want to set it manually, see How do I assign a static DNS server to a private Amazon EC2 instance in the AWS Knowledge Center for guidance on setting the persistent DNS server for your particular Linux distribution and version.
3. Make sure your Amazon Linux - 64bit instance is up to date.

```
sudo yum -y update
```

4. Install the required Amazon Linux packages on your Linux instance.

Note

Some of these packages may already be installed.

As you install the packages, you might be presented with several pop-up configuration screens. You can generally leave the fields in these screens blank.

Amazon Linux 1

```
sudo yum -y install sssd realmd krb5-workstation
```

Amazon Linux 2

```
sudo yum -y install sssd realmd krb5-workstation samba-common-tools
```

Note

For help with determining the Amazon Linux version you are using, see Identifying Amazon Linux images in the *Amazon EC2 User Guide for Linux Instances*.

5. Join the instance to the directory with the following command.

```
sudo realm join -U join_account@example.com example.com --verbose
```

join_account@example.com

An account in the *example.com* domain that has domain join privileges. Enter the password for the account when prompted. For more information about delegating these privileges, see Delegate directory join privileges for AWS Managed Microsoft AD (p. 98).

example.com

The fully qualified DNS name of your directory.

```
...
 * Successfully enrolled machine in realm
```

6. Set the SSH service to allow password authentication.
   a. Open the `/etc/ssh/sshd_config` file in a text editor.

   ```
   sudo vi /etc/ssh/sshd_config
   ```

   b. Set the `PasswordAuthentication` setting to `yes`.

   ```
   PasswordAuthentication yes
   ```

   c. Restart the SSH service.

   ```
   sudo systemctl restart sshd.service
   ```

   Alternatively:

   ```
   sudo service sshd restart
   ```

7. After the instance has restarted, connect to it with any SSH client and add the domain admins group to the sudoers list by performing the following steps:
   a. Open the `sudoers` file with the following command:

   ```
   sudo visudo
   ```

b. Add the following to the bottom of the `sudoers` file and save it.

```
## Add the "Domain Admins" group from the example.com domain.
%Domain\ Admins@example.com ALL=(ALL:ALL) ALL
```

(The above example uses "\<space>" to create the Linux space character.)

CentOS

1. Connect to the instance using any SSH client.
2. Configure the Linux instance to use the DNS server IP addresses of the AWS Directory Service-provided DNS servers. You can do this either by setting it up in the DHCP Options set attached to the VPC or by setting it manually on the instance. If you want to set it manually, see How do I assign a static DNS server to a private Amazon EC2 instance in the AWS Knowledge Center for guidance on setting the persistent DNS server for your particular Linux distribution and version.
3. Make sure your CentOS 7 instance is up to date.

```
sudo yum -y update
```

4. Install the required CentOS 7 packages on your Linux instance.

Note
Some of these packages may already be installed.
As you install the packages, you might be presented with several pop-up configuration screens. You can generally leave the fields in these screens blank.

```
sudo yum -y install sssd realmd krb5-workstation samba-common-tools
```

5. Join the instance to the directory with the following command.

```
sudo realm join -U join_account@example.com example.com --verbose
```

`join_account@example.com`

An account in the *`example.com`* domain that has domain join privileges. Enter the password for the account when prompted. For more information about delegating these privileges, see Delegate directory join privileges for AWS Managed Microsoft AD (p. 98).

`example.com`

The fully qualified DNS name of your directory.

```
...
 * Successfully enrolled machine in realm
```

6. Set the SSH service to allow password authentication.

a. Open the `/etc/ssh/sshd_config` file in a text editor.

```
sudo vi /etc/ssh/sshd_config
```

b. Set the `PasswordAuthentication` setting to `yes`.

```
PasswordAuthentication yes
```

c. Restart the SSH service.

```
sudo systemctl restart sshd.service
```

Alternatively:

```
sudo service sshd restart
```

7. After the instance has restarted, connect to it with any SSH client and add the domain admins group to the sudoers list by performing the following steps:

   a. Open the `sudoers` file with the following command:

   ```
   sudo visudo
   ```

   b. Add the following to the bottom of the `sudoers` file and save it.

   ```
   ## Add the "Domain Admins" group from the example.com domain.
   %Domain\ Admins@example.com ALL=(ALL:ALL) ALL
   ```

   (The above example uses "\<space>" to create the Linux space character.)

Red hat

1. Connect to the instance using any SSH client.
2. Configure the Linux instance to use the DNS server IP addresses of the AWS Directory Service-provided DNS servers. You can do this either by setting it up in the DHCP Options set attached to the VPC or by setting it manually on the instance. If you want to set it manually, see How do I assign a static DNS server to a private Amazon EC2 instance in the AWS Knowledge Center for guidance on setting the persistent DNS server for your particular Linux distribution and version.
3. Make sure the Red Hat - 64bit instance is up to date.

```
sudo yum -y update
```

4. Install the required Red Hat packages on your Linux instance.

   Note
   Some of these packages may already be installed.
   As you install the packages, you might be presented with several pop-up configuration screens. You can generally leave the fields in these screens blank.

```
sudo yum -y install sssd realmd krb5-workstation samba-common-tools
```

5. Join the instance to the directory with the following command.

```
sudo realm join -v -U join_account example.com --install=/
```

join_account

The **sAMAccountName** for an account in the *example.com* domain that has domain join privileges. Enter the password for the account when prompted. For more information about delegating these privileges, see Delegate directory join privileges for AWS Managed Microsoft AD (p. 98).

example.com

The fully qualified DNS name of your directory.

```
...
 * Successfully enrolled machine in realm
```

6. Set the SSH service to allow password authentication.

   a. Open the `/etc/ssh/sshd_config` file in a text editor.

   ```
   sudo vi /etc/ssh/sshd_config
   ```

   b. Set the `PasswordAuthentication` setting to `yes`.

   ```
   PasswordAuthentication yes
   ```

   c. Restart the SSH service.

   ```
   sudo systemctl restart sshd.service
   ```

   Alternatively:

   ```
   sudo service sshd restart
   ```

7. After the instance has restarted, connect to it with any SSH client and add the domain admins group to the sudoers list by performing the following steps:

   a. Open the `sudoers` file with the following command:

   ```
   sudo visudo
   ```

   b. Add the following to the bottom of the `sudoers` file and save it.

   ```
   ## Add the "Domain Admins" group from the example.com domain.
   %Domain\ Admins@example.com ALL=(ALL:ALL) ALL
   ```

   (The above example uses "\<space>" to create the Linux space character.)

Ubuntu

1. Connect to the instance using any SSH client.
2. Configure the Linux instance to use the DNS server IP addresses of the AWS Directory Service-provided DNS servers. You can do this either by setting it up in the DHCP Options set attached to the VPC or by setting it manually on the instance. If you want to set it manually, see How do I assign a static DNS server to a private Amazon EC2 instance in the AWS Knowledge Center for guidance on setting the persistent DNS server for your particular Linux distribution and version.
3. Make sure your Ubuntu - 64bit instance is up to date.

```
sudo apt-get update
sudo apt-get -y upgrade
```

4. Install the required Ubuntu packages on your Linux instance.

Note
Some of these packages may already be installed.
As you install the packages, you might be presented with several pop-up configuration screens. You can generally leave the fields in these screens blank.

```
sudo apt-get -y install sssd realmd krb5-user samba-common packagekit adcli
```

5. Disable Reverse DNS resolution and set the default realm to your domain's FQDN. Ubuntu Instances **must** be reverse-resolvable in DNS before the realm will work. Otherwise, you have to disable reverse DNS in /etc/krb5.conf as follows:

```
sudo vi /etc/krb5.conf
```

```
[libdefaults]
default_realm = EXAMPLE.COM
rdns = false
```

6. Join the instance to the directory with the following command.

```
sudo realm join -U join_account example.com --verbose
```

join_account@example.com

The **sAMAccountName** for an account in the *example.com* domain that has domain join privileges. Enter the password for the account when prompted. For more information about delegating these privileges, see Delegate directory join privileges for AWS Managed Microsoft AD (p. 98).

example.com

The fully qualified DNS name of your directory.

```
...
 * Successfully enrolled machine in realm
```

7. Set the SSH service to allow password authentication.

   a. Open the `/etc/ssh/sshd_config` file in a text editor.

   ```
   sudo vi /etc/ssh/sshd_config
   ```

   b. Set the `PasswordAuthentication` setting to `yes`.

   ```
   PasswordAuthentication yes
   ```

   c. Restart the SSH service.

   ```
   sudo systemctl restart sshd.service
   ```

   Alternatively:

   ```
   sudo service sshd restart
   ```

8. After the instance has restarted, connect to it with any SSH client and add the domain admins group to the sudoers list by performing the following steps:

   a. Open the `sudoers` file with the following command:

```
sudo visudo
```

b. Add the following to the bottom of the `sudoers` file and save it.

```
## Add the "Domain Admins" group from the example.com domain.
%Domain\ Admins@example.com ALL=(ALL:ALL) ALL
```

(The above example uses "\<space>" to create the Linux space character.)

Note
When using Simple AD, if you create a user account on a Linux instance with the option "Force user to change password at first login," that user will not be able to initially change their password using **kpasswd**. In order to change the password the first time, a domain administrator must update the user password using the Active Directory Management Tools.

Manage accounts from a Linux instance

To manage accounts in Simple AD from a Linux instance, you must update specific configuration files on your Linux instance as follows:

1. Set **krb5_use_kdcinfo** to **False** in the **/etc/sssd/sssd.conf** file. For example:

```
[domain/example.com]
    krb5_use_kdcinfo = False
```

2. In order for the configuration to take affect you need to restart the sssd service:

```
$ sudo systemctl restart sssd.service
```

Alternatively, you could use:

```
$ sudo service sssd start
```

3. If you will be managing users from a CentOS Linux instance, you must also edit the file **/etc/smb.conf** to include:

```
[global]
  workgroup = EXAMPLE.COM
  realm = EXAMPLE.COM
  netbios name = EXAMPLE
  security = ads
```

Restricting account login access

Since all accounts are defined in Active Directory, by default, all the users in the directory can log in to the instance. You can allow only specific users to log in to the instance with **ad_access_filter** in **sssd.conf**. For example:

```
ad_access_filter = (memberOf=cn=admins,ou=Testou,dc=example,dc=com)
```

`memberOf`

Indicates that users should only be allowed access to the instance if they are a member of a specific group.

`cn`

The common name of the group that should have access. In this example, the group name is *`admins`*.

`ou`

This is the organizational unit in which the above group is located. In this example, the OU is *`Testou`*.

`dc`

This is the domain component of your domain. In this example, *`example`*.

`dc`

This is an additional domain component. In this example, *`com`*.

You must manually add **ad_access_filter** to your **/etc/sssd/sssd.conf**.

Open the **/etc/sssd/sssd.conf** file in a text editor.

```
sudo vi /etc/sssd/sssd.conf
```

After you do this, your **sssd.conf** might look like this:

```
[sssd]
domains = example.com
config_file_version = 2
services = nss, pam

[domain/example.com]
ad_domain = example.com
krb5_realm = EXAMPLE.COM
realmd_tags = manages-system joined-with-samba
cache_credentials = True
id_provider = ad
krb5_store_password_if_offline = True
default_shell = /bin/bash
ldap_id_mapping = True
use_fully_qualified_names = True
fallback_homedir = /home/%u@%d
access_provider = ad
ad_access_filter = (memberOf=cn=admins,ou=Testou,dc=example,dc=com)
```

In order for the configuration to take effect, you need to restart the sssd service:

```
sudo systemctl restart sssd.service
```

Alternatively, you could use:

```
sudo service sssd restart
```

Connect to the instance

When a user connects to the instance using an SSH client, they are prompted for their user name. The user can enter the user name in either the `username@example.com` or `EXAMPLE\username` format. The response will appear similar to the following, depending on which linux distribution you are using:

Amazon Linux, Red Hat Enterprise Linux, and CentOS Linux

```
login as: johndoe@example.com
johndoe@example.com's password:
Last login: Thu Jun 25 16:26:28 2015 from XX.XX.XX.XX
```

SUSE Linux

```
SUSE Linux Enterprise Server 15 SP1 x86_64 (64-bit)

As "root" (sudo or sudo -i) use the:
  - zypper command for package management
  - yast command for configuration management

Management and Config: https://www.suse.com/suse-in-the-cloud-basics
Documentation: https://www.suse.com/documentation/sles-15/
Forum: https://forums.suse.com/forumdisplay.php?93-SUSE-Public-Cloud

Have a lot of fun...
```

Ubuntu Linux

```
login as: admin@example.com
admin@example.com@10.24.34.0's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-1057-aws x86_64)

* Documentation:  https://help.ubuntu.com
* Management:     https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

  System information as of Sat Apr 18 22:03:35 UTC 2020

  System load:  0.01              Processes:           102
  Usage of /:   18.6% of 7.69GB   Users logged in:     2
  Memory usage: 16%               IP address for eth0: 10.24.34.1
  Swap usage:   0%
```

Delegate directory join privileges for Simple AD

To join a computer to your directory, you need an account that has privileges to join computers to the directory.

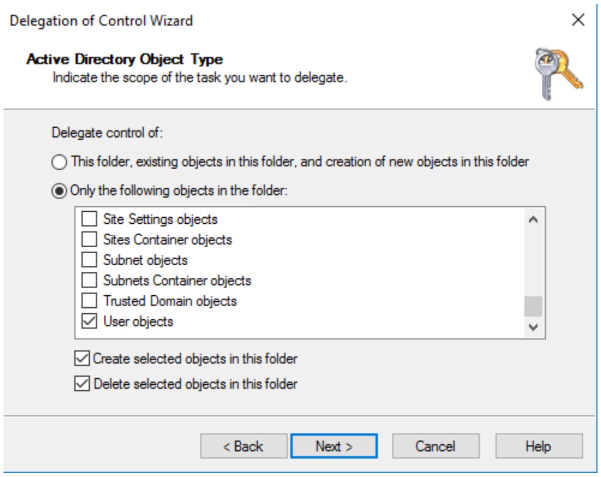
With Simple AD, members of the **Domain Admins** group have sufficient privileges to join computers to the directory.

However, as a best practice, you should use an account that has only the minimum privileges necessary. The following procedure demonstrates how to create a new group called `Joiners` and delegate the privileges to this group that are needed to join computers to the directory.

You must perform this procedure on a computer that is joined to your directory and has the **Active Directory User and Computers** MMC snap-in installed. You must also be logged in as a domain administrator.

To delegate join privileges for Simple AD

1. Open **Active Directory User and Computers** and select your domain root in the navigation tree.
2. In the navigation tree on the left, open the context menu (right-click) for **Users**, choose **New**, and then choose **Group**.
3. In the **New Object - Group** box, type the following and choose **OK**.
   - For **Group name**, type `Joiners`.
   - For **Group scope**, choose **Global**.
   - For **Group type**, choose **Security**.
4. In the navigation tree, select your domain root. From the **Action** menu, choose **Delegate Control**.
5. On the **Delegation of Control Wizard** page, choose **Next**, and then choose **Add**.
6. In the **Select Users, Computers, or Groups** box, type `Joiners` and choose **OK**. If more than one object is found, select the `Joiners` group created above. Choose **Next**.
7. On the **Tasks to Delegate** page, select **Create a custom task to delegate**, and then choose **Next**.
8. Select **Only the following objects in the folder**, and then select **Computer objects**.
9. Select **Create selected objects in this folder** and **Delete selected objects in this folder**. Then choose **Next**.



Delegation of Control Wizard

Active Directory Object Type
Indicate the scope of the task you want to delegate.

Delegate control of:

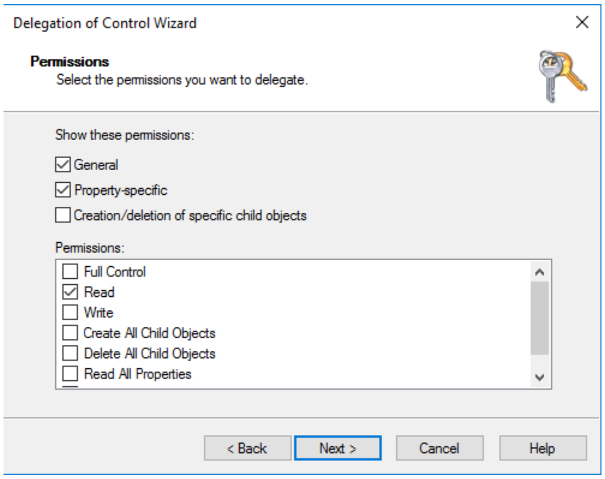
○ This folder, existing objects in this folder, and creation of new objects in this folder

● Only the following objects in the folder:

☐ Site Settings objects
☐ Sites Container objects
☐ Subnet objects
☐ Subnets Container objects
☐ Trusted Domain objects
☑ User objects

☑ Create selected objects in this folder
☑ Delete selected objects in this folder

10. Select **Read** and **Write**, and then choose **Next**.



Delegation of Control Wizard

Permissions
Select the permissions you want to delegate.

Show these permissions:

☑ General
☑ Property-specific
☐ Creation/deletion of specific child objects

Permissions:

☐ Full Control
☑ Read
☐ Write
☐ Create All Child Objects
☐ Delete All Child Objects
☐ Read All Properties

11. Verify the information on the **Completing the Delegation of Control Wizard** page and choose **Finish**.

12. Create a user with a strong password and add that user to the `Joiners` group. The user will then have sufficient privileges to connect AWS Directory Service to the directory.

Create a DHCP options set

AWS recommends that you create a DHCP options set for your AWS Directory Service directory and assign the DHCP options set to the VPC that your directory is in. This allows any instances in that VPC to point to the specified domain and DNS servers to resolve their domain names.

For more information about DHCP options sets, see DHCP options sets in the *Amazon VPC User Guide*.

To create a DHCP options set for your directory

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **DHCP Options Sets**, and then choose **Create DHCP options set**.
3. On the **Create DHCP options set** page, enter the following values for your directory:

   Name

      An optional tag for the options set.

   Domain name

      The fully qualified name of your directory, such as `corp.example.com`.

   Domain name servers

      The IP addresses of your AWS-provided directory's DNS servers.

      > **Note**
      > You can find these addresses by going to the AWS Directory Service console navigation pane, selecting **Directories** and then choosing the correct directory ID.

   NTP servers

      Leave this field blank.

   NetBIOS name servers

      Leave this field blank.

   NetBIOS node type

      Leave this field blank.

4. Choose **Create DHCP options set**. The new set of DHCP options appears in your list of DHCP options.
5. Make a note of the ID of the new set of DHCP options (dopt-*xxxxxxxx*). You use it to associate the new options set with your VPC.

To change the DHCP options set associated with a VPC

After you create a set of DHCP options, you can't modify them. If you want your VPC to use a different set of DHCP options, you must create a new set and associate them with your VPC. You can also set up your VPC to use no DHCP options at all.

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Your VPCs**
3. Select the VPC, and then choose **Actions**, **Edit DHCP options set**.
4. For **DHCP options set**, select an options set or choose **No DHCP options set**, and then choose **Save**.

Maintain your Simple AD directory

This section describes how to maintain common administrative tasks for your Simple AD environment.

Topics

Delete your directory

When a Simple AD or AWS Directory Service for Microsoft Active Directory directory is deleted, all of the directory data and snapshots are deleted and cannot be recovered. After the directory is deleted, all instances that are joined to the directory remain intact. You cannot, however, use your directory credentials to log in to these instances. You need to log in to these instances with a user account that is local to the instance.

When an AD Connector directory is deleted, your on-premises directory remains intact. All instances that are joined to the directory also remain intact and remain joined to your on-premises directory. You can still use your directory credentials to log in to these instances.

To delete a directory

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. Ensure that no AWS applications are enabled for the directory you intend to delete.

    a. On the **Directories** page, choose your directory ID.

    b. On the **Directory details** page, select the **Application management** tab. In the **AWS apps & services** section, you see which AWS applications are enabled for your directory.

        - To disable WorkSpaces, you must deregister the service from the directory in the WorkSpaces console. For more information, see Deregistering from a directory in the *Amazon WorkSpaces Administration Guide*.
        - To disable Amazon WorkSpaces Application Manager, you must remove all application assignments in the Amazon WAM console. For more information, see Removing all application assignments in the *Amazon WAM Administration Guide*.
        - To disable Amazon WorkDocs, you must delete the Amazon WorkDocs site in the Amazon WorkDocs console. For more information, see Delete a site in the *Amazon WorkDocs Administration Guide*.
        - To disable Amazon WorkMail, you must remove the Amazon WorkMail organization in the Amazon WorkMail console. For more information, see Remove an organization in the *Amazon WorkMail Administrator Guide*.
        - Disable AWS Management Console access.
        - To disable Amazon Relational Database Service, you must remove the Amazon RDS instance from the domain. For more information, see Managing a DB instance in a domain in the *Amazon RDS User Guide*.
        - To disable Amazon QuickSight, you must unsubscribe from Amazon QuickSight. For more information, see Closing your Amazon QuickSight account in the *Amazon QuickSight User Guide*.
        - To disable Amazon Connect, you must delete the Amazon Connect Instance. For more information, see Deleting an Amazon Connect instance in the *Amazon Connect Administration Guide*.

- To disable FSx for Windows File Server, you must remove the Amazon FSx file system from the domain. For more information, see Working with Active Directory in FSx for Windows File Server in the *Amazon FSx for Windows File Server User Guide*.

> **Note**
> If you are using AWS Single Sign-On and have previously connected it to the AWS Managed Microsoft AD directory you plan to delete, you must first change the identity source before you can delete it. For more information, see Change your identity source in the *AWS SSO User Guide*.

3. In the navigation pane, choose **Directories**.
4. Select only the directory to be deleted and click **Delete**. It takes several minutes for the directory to be deleted. When the directory has been deleted, it is removed from your directory list.

Snapshot or restore your directory

AWS Directory Service provides the ability to take manual snapshots of data for your Simple AD directory. These snapshots can be used to perform a point-in-time restore for your directory. You cannot take snapshots of AD Connector directories.

Topics

Creating a snapshot of your directory

A snapshot can be used to restore your directory to what it was at the point in time that the snapshot was taken. To create a manual snapshot of your directory, perform the following steps.

> **Note**
> You are limited to 5 manual snapshots for each directory. If you have already reached this limit, you must delete one of your existing manual snapshots before you can create another.

To create a manual snapshot

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. On the **Directories** page, choose your directory ID.
3. On the **Directory details** page, select the **Maintenance** tab.
4. In the **Snapshots** section, choose **Actions**, and then select **Create snapshot**.
5. In the **Create directory snapshot** dialog box, provide a name for the snapshot, if desired. When ready, choose **Create**.

Depending on the size of your directory, it may take several minutes to create the snapshot. When the snapshot is ready, the **Status** value changes to `Completed`.

Restoring your directory from a snapshot

Restoring a directory from a snapshot is equivalent to moving the directory back in time. Directory snapshots are unique to the directory they were created from. A snapshot can only be restored to the directory from which it was created. In addition, the maximum supported age of a manual snapshot is

180 days. For more information, see Useful shelf life of a system-state backup of Active Directory on the Microsoft website.

> **Warning**
> We recommend that you contact the AWS Support Center before any snapshot restore; we may be able to help you avoid the need to do a snapshot restore. Any restore from snapshot can result in data loss as they are a point in time. It is important you understand that all of the DCs and DNS servers associated with the directory will be offline until the restore operation has been completed.

To restore your directory from a snapshot, perform the following steps.

To restore a directory from a snapshot

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. On the **Directories** page, choose your directory ID.
3. On the **Directory details** page, select the **Maintenance** tab.
4. In the **Snapshots** section, select a snapshot in the list, choose **Actions**, and then select **Restore snapshot**.
5. Review the information in the **Restore directory snapshot** dialog box, and choose **Restore**.

For a Simple AD directory, it may take several minutes for the directory to be restored. When it has been successfully restored, the **Status** value of the directory changes to `Active`. Any changes made to the directory after the snapshot date are overwritten.

Deleting a snapshot

To delete a snapshot

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. On the **Directories** page, choose your directory ID.
3. On the **Directory details** page, select the **Maintenance** tab.
4. In the **Snapshots** section, choose **Actions**, and then select **Delete snapshot**.
5. Verify that you want to delete the snapshot, and then choose **Delete**.

View directory information

You can view detailed information about a directory.

To view detailed directory information

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. Click the directory ID link for your directory. Information about the directory is displayed in the **Directory details** page.

For more information about the **Status** field, see Understanding your directory status (p. 51).

Enable access to AWS applications and services

AWS Directory Service can give other AWS applications and services, such as Amazon WorkSpaces, access to your directory users. The following AWS applications and services can be enabled or disabled to work with AWS Directory Service.

AWS application / service	More information...
Amazon Chime	For more information, see the Amazon Chime Administration Guide.
Amazon Connect	For more information, see the Amazon Connect Administration Guide.
FSx for Windows File Server	For more information, see Using Amazon FSx with AWS Directory Service for Microsoft Active Directory in the *FSx for Windows File Server User Guide*.
Amazon QuickSight	For more information, see the Amazon QuickSight User Guide.
Amazon Relational Database Service	For more information, see the Amazon RDS User Guide.
AWS Transfer Family	For more information, see the AWS Transfer Family User Guide.
Amazon WorkDocs	For more information, see the Amazon WorkDocs Administration Guide.
Amazon WorkMail	For more information, see the Amazon WorkMail Administrator Guide.
WorkSpaces	You can create a Simple AD, AWS Managed Microsoft AD, or AD Connector directly from WorkSpaces. Simply launch **Advanced Setup** when creating your Workspace. For more information, see the Amazon WorkSpaces Administration Guide.
Amazon WorkSpaces Application Manager	For more information, see the Amazon WAM Administration Guide.
AWS Management Console	For more information, see Enable access to the AWS Management Console with AD credentials (p. 141).

Once enabled, you manage access to your directories in the console of the application or service that you want to give access to your directory. To find the AWS applications and services links described above in the AWS Directory Service console, perform the following steps.

To display the applications and services for a directory

1. In the AWS Directory Service console navigation pane, choose **Directories**.
2. On the **Directories** page, choose your directory ID.
3. On the **Directory details** page, select the **Application management** tab.
4. Review the list under the **AWS apps & services** section.

Topics

- Single sign-on (p. 259)

Creating an access URL

An access URL is used with AWS applications and services, such as Amazon WorkDocs, to reach a login page that is associated with your directory. The URL must be unique globally. You can create an access URL for your directory by performing the following steps.

> **Warning**
> Once you create an application access URL for this directory, it cannot be changed. After an access URL is created, it cannot be used by others. If you delete your directory, the access URL is also deleted and can then be used by any other account.

To create an access URL

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. On the **Directories** page, choose your directory ID.
3. On the **Directory details** page, select the **Application management** tab.
4. In the **Application access URL** section, if an access URL has not been assigned to the directory, the **Create** button is displayed. Enter a directory alias and choose **Create**. If an **Entity Already Exists** error is returned, the specified directory alias has already been allocated. Choose another alias and repeat this procedure.

   Your access URL is displayed in the format *<alias>*.awsapps.com.

Single sign-on

AWS Directory Service provides the ability to allow your users to access Amazon WorkDocs from a computer joined to the directory without having to enter their credentials separately.

Before you enable single sign-on, you need to take additional steps to enable your users web browsers to support single sign-on. Users may need to modify their web browser settings to enable single sign-on.

> **Note**
> Single sign-on only works when used on a computer that is joined to the AWS Directory Service directory. It cannot be used on computers that are not joined to the directory.

If your directory is an AD Connector directory and the AD Connector service account does not have the permission to add or remove its service principal name attribute, then for Steps 5 and 6 below, you have two options:

1. You can proceed and will be prompted for the user name and password for a directory user that has this permission to add or remove the service principal name attribute on the AD Connector service account. These credentials are only used to enable single sign-on and are not stored by the service. The AD Connector service account permissions are not changed.
2. You can delegate permissions to allow the AD Connector service account to add or remove the service principal name attribute on itself, you can run the below PowerShell commands from a domain joined computer using an account that has permissions to modify the permissions on the AD Connector service account. The below command will give the AD Connector service account the ability to add and remove a service principal name attribute only for itself.

```
$AccountName = 'ConnectorAccountName'
# DO NOT modify anything below this comment.
# Getting Active Directory information.
Import-Module 'ActiveDirectory'
$RootDse = Get-ADRootDSE
```

```
[System.GUID]$ServicePrincipalNameGuid = (Get-ADObject -SearchBase
 $RootDse.SchemaNamingContext -Filter { lDAPDisplayName -eq 'servicePrincipalName' } -
Properties 'schemaIDGUID').schemaIDGUID
# Getting AD Connector service account Information.
$AccountProperties = Get-ADUser -Identity $AccountName
$AclPath = $AccountProperties.DistinguishedName
$AccountSid = New-Object -TypeName 'System.Security.Principal.SecurityIdentifier'
 $AccountProperties.SID.Value
# Getting ACL settings for AD Connector service account.
$ObjectAcl = Get-ACL -Path "AD:\$AclPath"
# Setting ACL allowing the AD Connector service account the ability to add and remove a
 Service Principal Name (SPN) to itself
$AddAccessRule = New-Object -TypeName 'System.DirectoryServices.ActiveDirectoryAccessRule'
 $AccountSid, 'WriteProperty', 'Allow', $ServicePrincipalNameGUID, 'None'
$ObjectAcl.AddAccessRule($AddAccessRule)
Set-ACL -AclObject $ObjectAcl -Path "AD:\$AclPath"
```

To enable or disable single sign-on with Amazon WorkDocs

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. On the **Directories** page, choose your directory ID.
3. On the **Directory details** page, select the **Application management** tab.
4. In the **Application access URL** section, choose **Enable** to enable single sign-on for Amazon WorkDocs.

   If you do not see the **Enable** button, you may need to first create an Access URL before this option will be displayed. For more information about how to create an access URL, see Creating an access URL (p. 135).
5. In the **Enable Single Sign-On for this directory** dialog box, choose **Enable**. Single sign-on is enabled for the directory.
6. If you later want to disable single sign-on with Amazon WorkDocs, choose **Disable**, and then in the **Disable Single Sign-On for this directory** dialog box, choose **Disable** again.

Topics

Single sign-on for IE and Chrome

To allow Microsoft Internet Explorer (IE) and Google Chrome browsers to support single sign-on, the following tasks must be performed on the client computer:

- Add your access URL (e.g., https://*<alias>*.awsapps.com) to the list of approved sites for single sign-on.
- Enable active scripting (JavaScript).
- Allow automatic logon.
- Enable integrated authentication.

You or your users can perform these tasks manually, or you can change these settings using Group Policy settings.

Topics

- Group policy settings for single sign-on (p. 138)

Manual update for single sign-on on Windows

To manually enable single sign-on on a Windows computer, perform the following steps on the client computer. Some of these settings may already be set correctly.

To manually enable single sign-on for Internet Explorer and Chrome on Windows

1. To open the **Internet Properties** dialog box, choose the **Start** menu, type `Internet Options` in the search box, and choose **Internet Options**.
2. Add your access URL to the list of approved sites for single sign-on by performing the following steps:

   a. In the **Internet Properties** dialog box, select the **Security** tab.

   b. Select **Local intranet** and choose **Sites**.

   c. In the **Local intranet** dialog box, choose **Advanced**.

   d. Add your access URL to the list of websites and choose **Close**.

   e. In the **Local intranet** dialog box, choose **OK**.
3. To enable active scripting, perform the following steps:

   a. In the **Security** tab of the **Internet Properties** dialog box, choose **Custom level**.

   b. In the **Security Settings - Local Intranet Zone** dialog box, scroll down to **Scripting** and select **Enable** under **Active scripting**.

   c. In the **Security Settings - Local Intranet Zone** dialog box, choose **OK**.
4. To enable automatic logon, perform the following steps:

   a. In the **Security** tab of the **Internet Properties** dialog box, choose **Custom level**.

   b. In the **Security Settings - Local Intranet Zone** dialog box, scroll down to **User Authentication** and select **Automatic logon only in Intranet zone** under **Logon**.

   c. In the **Security Settings - Local Intranet Zone** dialog box, choose **OK**.

   d. In the **Security Settings - Local Intranet Zone** dialog box, choose **OK**.
5. To enable integrated authentication, perform the following steps:

   a. In the **Internet Properties** dialog box, select the **Advanced** tab.

   b. Scroll down to **Security** and select **Enable Integrated Windows Authentication**.

   c. In the **Internet Properties** dialog box, choose **OK**.
6. Close and re-open your browser to have these changes take effect.

Manual update for single sign-on on OS X

To manually enable single sign-on for Chrome on OS X, perform the following steps on the client computer. You will need administrator rights on your computer to complete these steps.

To manually enable single sign-on for Chrome on OS X

1. Add your access URL to the AuthServerWhitelist policy by running the following command:

```
defaults write com.google.Chrome AuthServerWhitelist "https://<alias>.awsapps.com"
```

2. Open **System Preferences**, go to the **Profiles** panel, and delete the `Chrome Kerberos Configuration` profile.
3. Restart Chrome and open chrome://policy in Chrome to confirm that the new settings are in place.

Group policy settings for single sign-on

The domain administrator can implement Group Policy settings to make the single sign-on changes on client computers that are joined to the domain.

> **Note**
> If you manage the Chrome web browsers on the computers in your domain with Chrome policies, you must add your access URL to the AuthServerWhitelist policy. For more information about setting Chrome policies, go to Policy Settings in Chrome.

To enable single sign-on for Internet Explorer and Chrome using Group Policy settings

1. Create a new Group Policy object by performing the following steps:

   a. Open the Group Policy Management tool, navigate to your domain and select **Group Policy Objects**.

   b. From the main menu, choose **Action** and select **New**.

   c. In the **New GPO** dialog box, enter a descriptive name for the Group Policy object, such as `SSO Policy`, and leave **Source Starter GPO** set to **(none)**. Click **OK**.

2. Add the access URL to the list of approved sites for single sign-on by performing the following steps:

   a. In the Group Policy Management tool, navigate to your domain, select **Group Policy Objects**, open the context (right-click) menu for your SSO policy, and choose **Edit**.

   b. In the policy tree, navigate to **User Configuration** > **Preferences** > **Windows Settings**.

   c. In the **Windows Settings** list, open the context (right-click) menu for **Registry** and choose **New registry item**.

   d. In the **New Registry Properties** dialog box, enter the following settings and choose **OK**:

   Action

   `Update`

   Hive

   `HKEY_CURRENT_USER`

   Path

   `Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\Domains\awsapps.com\<alias>`

   The value for *`<alias>`* is derived from your access URL. If your access URL is `https://examplecorp.awsapps.com`, the alias is `examplecorp`, and the registry key will be `Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\Domains\awsapps.com\examplecorp`.

   Value name

   `https`

   Value type

   `REG_DWORD`

   Value data

   `1`

3. To enable active scripting, perform the following steps:

   a. In the Group Policy Management tool, navigate to your domain, select **Group Policy Objects**, open the context (right-click) menu for your SSO policy, and choose **Edit**.

   b. In the policy tree, navigate to **Computer Configuration** > **Policies** > **Administrative Templates** > **Windows Components** > **Internet Explorer** > **Internet Control Panel** > **Security Page** > **Intranet Zone**.

   c. In the **Intranet Zone** list, open the context (right-click) menu for **Allow active scripting** and choose **Edit**.

   d. In the **Allow active scripting** dialog box, enter the following settings and choose **OK**:

      - Select the **Enabled** radio button.
      - Under **Options** set **Allow active scripting** to **Enable**.

4. To enable automatic logon, perform the following steps:

   a. In the Group Policy Management tool, navigate to your domain, select Group Policy Objects, open the context (right-click) menu for your SSO policy, and choose **Edit**.

   b. In the policy tree, navigate to **Computer Configuration** > **Policies** > **Administrative Templates** > **Windows Components** > **Internet Explorer** > **Internet Control Panel** > **Security Page** > **Intranet Zone**.

   c. In the **Intranet Zone** list, open the context (right-click) menu for **Logon options** and choose **Edit**.

   d. In the **Logon options** dialog box, enter the following settings and choose **OK**:

      - Select the **Enabled** radio button.
      - Under **Options** set **Logon options** to **Automatic logon only in Intranet zone**.

5. To enable integrated authentication, perform the following steps:

   a. In the Group Policy Management tool, navigate to your domain, select **Group Policy Objects**, open the context (right-click) menu for your SSO policy, and choose **Edit**.

   b. In the policy tree, navigate to **User Configuration** > **Preferences** > **Windows Settings**.

   c. In the **Windows Settings** list, open the context (right-click) menu for **Registry** and choose **New registry item**.

   d. In the **New Registry Properties** dialog box, enter the following settings and choose **OK**:

      Action

      ```
      Update
      ```

      Hive

      ```
      HKEY_CURRENT_USER
      ```

      Path

      ```
      Software\Microsoft\Windows\CurrentVersion\Internet Settings
      ```

      Value name

      ```
      EnableNegotiate
      ```

      Value type

      ```
      REG_DWORD
      ```

      Value data

      ```
      1
      ```

6. Close the **Group Policy Management Editor** window if it is still open.

7. Assign the new policy to your domain by following these steps:

   a. In the Group Policy Management tree, open the context (right-click) menu for your domain and choose **Link an Existing GPO**.

   b. In the **Group Policy Objects** list, select your SSO policy and choose **OK**.

These changes will take effect after the next Group Policy update on the client, or the next time the user logs in.

Single sign-on for Firefox

To allow Mozilla Firefox browser to support single sign-on, add your access URL (e.g., https://*<alias>*.awsapps.com) to the list of approved sites for single sign-on. This can be done manually, or automated with a script.

Topics

Manual update for single sign-on

To manually add your access URL to the list of approved sites in Firefox, perform the following steps on the client computer.

To manually add your access URL to the list of approved sites in Firefox

1. Open Firefox and open the `about:config` page.
2. Open the `network.negotiate-auth.trusted-uris` preference and add your access URL to the list of sites. Use a comma (,) to separate multiple entries.

Automatic update for single sign-on

As a domain administrator, you can use a script to add your access URL to the Firefox `network.negotiate-auth.trusted-uris` user preference on all computers on your network. For more information, go to https://support.mozilla.org/en-US/questions/939037.

Enable access to the AWS Management Console with AD credentials

AWS Directory Service allows you to grant members of your directory access to the AWS Management Console. By default, your directory members do not have access to any AWS resources. You assign IAM roles to your directory members to give them access to the various AWS services and resources. The IAM role defines the services, resources, and level of access that your directory members have.

Before you can grant console access to your directory members, your directory must have an access URL. For more information about how to view directory details and get your access URL, see View directory information (p. 130). For more information about how to create an access URL, see Creating an access URL (p. 135).

For more information about how to create and assign IAM roles to your directory members, see Grant users and groups access to AWS resources (p. 130).

Topics

- Enable AWS Management Console access (p. 265)
- Disable AWS Management Console access (p. 265)
- Set login session length (p. 265)

Related AWS Security Blog Article

- How to Access the AWS Management Console Using AWS Managed Microsoft AD and Your On-Premises Credentials

Enable AWS Management Console access

By default, console access is not enabled for any directory. To enable console access for your directory users and groups, perform the following steps:

To enable console access

1. In the AWS Directory Service console navigation pane, choose **Directories**.
2. On the **Directories** page, choose your directory ID.
3. On the **Directory details** page, select the **Application management** tab.
4. Under the **AWS Management Console** section, choose **Enable**. Console access is now enabled for your directory.

   Before users can sign-in to the console with your access URL, you must first add your users to the role. For general information about assigning users to IAM roles, see Assigning users or groups to an existing role (p. 132). After the IAM roles have been assigned, users can then access the console using your access URL. For example, if your directory access URL is example-corp.awsapps.com, the URL to access the console is https://example-corp.awsapps.com/console/.

Disable AWS Management Console access

To disable console access for your directory users and groups, perform the following steps:

To disable console access

1. In the AWS Directory Service console navigation pane, choose **Directories**.
2. On the **Directories** page, choose your directory ID.
3. On the **Directory details** page, select the **Application management** tab.
4. Under the **AWS Management Console** section, choose **Disable**. Console access is now disabled for your directory.
5. If any IAM roles have been assigned to users or groups in the directory, the **Disable** button may be unavailable. In this case, you must remove all IAM role assignments for the directory before proceeding, including assignments for users or groups in your directory that have been deleted, which will show as **Deleted User** or **Deleted Group**.

   After all IAM role assignments have been removed, repeat the steps above.

Set login session length

By default, users have 1 hour to use their session after successfully signing in to the console before they are logged out. After that, users must sign in again to start the next 1 hour session before being logged off again. You can use the following procedure to change the length of time to up to 12 hours per session.

To set login session length

1. In the AWS Directory Service console navigation pane, choose **Directories**.
2. On the **Directories** page, choose your directory ID.
3. On the **Directory details** page, select the **Application management** tab.
4. Under the **AWS apps & services** section, choose **AWS Management Console**.
5. In the **Manage Access to AWS Resource** dialog box, choose **Continue**.
6. In the **Assign users and groups to IAM roles** page, under **Set login session length**, edit the numbered value, and then choose **Save**.

Tutorial: Create a Simple AD directory

The following tutorial walks you through all of the steps necessary to set up an AWS Directory Service Simple AD directory. It is intended to get you started with AWS Directory Service quickly and easily, but is not intended to be used in a large-scale production environment.

Topics

Prerequisites

This tutorial assumes the following:

- You have an active AWS account.
- Your account has not reached its limit of VPCs for the Region in which you want to use AWS Directory Service. For more information about Amazon VPC, see What is Amazon VPC? and Subnets in your VPC in the *Amazon VPC User Guide*.
- You do not have an existing VPC in the Region with a CIDR of 10.0.0.0/16.

Step 1: Create and configure your VPC

The following sections demonstrate how to create and configure a VPC for use with AWS Directory Service.

Topics

Create a new VPC

This tutorial uses one of the VPC creation wizards to create the following:

- The VPC
- One of the subnets

- An Internet gateway

To create your VPC using the VPC wizard

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. On the **VPC Dashboard**, choose **Launch VPC Wizard**.
3. Under **Step 1: Select a VPC Configuration**, on **VPC with a Single Public Subnet**, choose **Select**.
4. Enter the following information into the wizard and choose **Create VPC**.

   IP CIDR block

   `10.0.0.0/16`

   VPC name

   `ADS VPC`

   Public subnet

   `10.0.0.0/24`

   Availability Zone

   No Preference

   Subnet name

   `ADS Subnet 1`

   Enable DNS hostnames

   Leave default selection

   Hardware tenancy

   Default
5. It takes several minutes for the VPC to be created. After the VPC is created, proceed to the following section to add a second subnet.

Add a second subnet

AWS Directory Service requires two subnets in your VPC, and each subnet must be in a different Availability Zone. The VPC wizard only creates one subnet, so you must manually create the second subnet, and specify a different Availability Zone than the first subnet. Create the second subnet by performing the following steps.

To create a subnet

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, select **Subnets**, select the subnet with the name `ADS Subnet 1`, and select the **Summary** tab at the bottom of the page. Make a note of the Availability Zone of this subnet.
3. Click **Create Subnet** and enter the following information in the **Create Subnet** dialog box and click **Yes, Create**.

   Name tag

   `ADS Subnet 2`

   VPC

   Select your VPC. This is the VPC with the name `ADS VPC`.

Availability Zone

Select any Availability Zone other than the one noted in step 2. The two subnets used by AWS Directory Service must reside in different Availability Zones.

CIDR Block

```
10.0.1.0/24
```

Step 2: Create your Simple AD directory

To create a new directory, perform the following steps. Before starting this procedure, make sure you have completed the prerequisites identified in Simple AD prerequisites (p. 227).

To create a Simple AD directory

1. In the AWS Directory Service console navigation pane, choose **Directories** and then choose **Set up directory**.
2. On the **Select directory type** page, choose **Simple AD**, and then choose **Next**.
3. On the **Enter directory information** page, provide the following information:

   Directory size

   Choose from either the **Small** or **Large** size option. For more information about sizes, see Simple Active Directory (p. 226).

   Organization name

   A unique organization name for your directory that will be used to register client devices.

   This field is only available if you are creating your directory as part of launching WorkSpaces.

   Directory DNS name

   The fully qualified name for the directory, such as `corp.example.com`.

   Directory NetBIOS name

   The short name for the directory, such as `CORP`.

   Administrator password

   The password for the directory administrator. The directory creation process creates an administrator account with the user name `Administrator` and this password.

   The directory administrator password is case-sensitive and must be between 8 and 64 characters in length, inclusive. It must also contain at least one character from three of the following four categories:

   - Lowercase letters (a-z)
   - Uppercase letters (A-Z)
   - Numbers (0-9)
   - Non-alphanumeric characters (~!@#$%^&*_-+=`|\(){}[]:;"'<>,.?/)

   Confirm password

   Retype the administrator password.

   Directory description

   An optional description for the directory.
4. On the **Choose VPC and subnets** page, provide the following information, and then choose **Next**.

VPC

The VPC for the directory.

Subnets

Choose the subnets for the domain controllers. The two subnets must be in different Availability Zones.

5. On the **Review & create** page, review the directory information and make any necessary changes. When the information is correct, choose **Create directory**. It takes several minutes for the directory to be created. Once created, the **Status** value changes to **Active**.

Best practices for Simple AD

Here are some suggestions and guidelines you should consider to avoid problems and get the most out of AWS Managed Microsoft AD.

Setting up: Prerequisites

Consider these guidelines before creating your directory.

Verify you have the right directory type

AWS Directory Service provides multiple ways to use Microsoft Active Directory with other AWS services. You can choose the directory service with the features you need at a cost that fits your budget:

- **AWS Directory Service for Microsoft Active Directory** is a feature-rich managed Microsoft Active Directory hosted on the AWS cloud. AWS Managed Microsoft AD is your best choice if you have more than 5,000 users and need a trust relationship set up between an AWS hosted directory and your on-premises directories.
- **AD Connector** simply connects your existing on-premises Active Directory to AWS. AD Connector is your best choice when you want to use your existing on-premises directory with AWS services.
- **Simple AD** is an inexpensive Active Directory–compatible service with the common directory features. In most cases, Simple AD is the least expensive option and your best choice if you have 5,000 or fewer users and don’t need the more advanced Microsoft Active Directory features.

For a more detailed comparison of AWS Directory Service options, see Which to choose (p. 1).

Ensure your VPCs and instances are configured correctly

In order to connect to, manage, and use your directories, you must properly configure the VPCs that the directories are associated with. See either AWS Managed Microsoft AD prerequisites (p. 9), AD Connector prerequisites (p. 187), or Simple AD prerequisites (p. 227) for information about the VPC security and networking requirements.

If you are adding an instance to your domain, ensure that you have connectivity and remote access to your instance as described in Join an EC2 instance to your AWS Managed Microsoft AD directory (p. 71).

Be aware of your limits

Learn about the various limits for your specific directory type. The available storage and the aggregate size of your objects are the only limitations on the number of objects you may store in your directory. See either AWS Managed Microsoft AD quotas (p. 152), AD Connector quotas (p. 222), or Simple AD quotas (p. 271) for details about your chosen directory.

Understand your directory’s AWS security group configuration and use

AWS creates a security group and attaches it to your directory’s domain controller elastic network interfaces. AWS configures the security group to block unnecessary traffic to the directory and allows necessary traffic.

Modifying the directory security group

If you want to modify the security of your directories’ security groups, you can do so. Make such changes only if you fully understand how security group filtering works. For more information, see Amazon EC2 security groups for Linux instances in the *Amazon EC2 User Guide*. Improper changes can result in loss of communications to intended computers and instances. AWS recommends that you do not attempt to open additional ports to your directory as this decreases the security of your directory. Please carefully review the AWS Shared Responsibility Model.

> **Warning**
> It is technically possible for you to associate the directory’s security group with other EC2 instances that you create. However, AWS recommends against this practice. AWS may have reasons to modify the security group without notice to address functional or security needs of the managed directory. Such changes affect any instances with which you associate the directory security group and may disrupt operation of the associated instances. Furthermore, associating the directory security group with your EC2 instances may create a potential security risk for your EC2 instances.

Use AWS Managed Microsoft AD if trusts are required

Simple AD does not support trust relationships. If you need to establish a trust between your AWS Directory Service directory and another directory, you should use AWS Directory Service for Microsoft Active Directory.

Setting up: Creating your directory

Here are some suggestions to consider as you create your directory.

Remember your administrator ID and password

When you set up your directory, you provide a password for the administrator account. That account ID is *Administrator* for Simple AD. Remember the password that you create for this account; otherwise you will not be able to add objects to your directory.

Understand username restrictions for AWS applications

AWS Directory Service provides support for most character formats that can be used in the construction of usernames. However, there are character restrictions that are enforced on usernames that will be used for signing in to AWS applications, such as WorkSpaces, Amazon WorkDocs, Amazon WorkMail, or Amazon QuickSight. These restrictions require that the following characters not be used:

- Spaces
- Multibyte characters
- !"#$%&'()*+,/:;<=>?@[\]^`{|}~

> **Note**
> The @ symbol is allowed as long as it precedes a UPN suffix.

Programming your applications

Before you program your applications, consider the following:

Use the Windows DC locator service

When developing applications, use the Windows DC locator service or use the Dynamic DNS (DDNS) service of your AWS Managed Microsoft AD to locate domain controllers (DCs). Do not hard code applications with the address of a DC. The DC locator service helps ensure directory load is distributed and enables you to take advantage of horizontal scaling by adding domain controllers to your deployment. If you bind your application to a fixed DC and the DC undergoes patching or recovery, your application will lose access to the DC instead of using one of the remaining DCs. Furthermore, hard coding of the DC can result in hot spotting on a single DC. In severe cases, hot spotting may cause your DC to become unresponsive. Such cases may also cause AWS directory automation to flag the directory as impaired and may trigger recovery processes that replace the unresponsive DC.

Load test before rolling out to production

Be sure to do lab testing with objects and requests that are representative of your production workload to confirm that the directory scales to the load of your application. Should you require additional capacity, you should use AWS Directory Service for Microsoft Active Directory, which enables you to add domain controllers for high performance. For more information, see Deploy additional domain controllers (p. 143).

Use efficient LDAP queries

Broad LDAP queries to a domain controller across thousands of objects can consume significant CPU cycles in a single DC, resulting in hot spotting. This may affect applications that share the same DC during the query.

Simple AD quotas

Generally, you should not add more than 500 users to a Small Simple AD directory and no more than 5,000 users to a Large Simple AD directory. For more flexible scaling options and additional Active Directory features, consider using AWS Directory Service for Microsoft Active Directory (Standard Edition or Enterprise Edition) instead.

The following are the default quotas for Simple AD. Each quota is per Region unless otherwise noted.

Simple AD quotas

Resource	Default quota
Simple AD directories	10
Manual snapshots *	5 per Simple AD

* The manual snapshot quota cannot be changed.

> **Note**
> You cannot attach a public IP address to your AWS elastic network interface (ENI).

Application compatibility policy for Simple AD

Simple AD is an implementation of Samba that provides many of the basic features of Active Directory. Due to the magnitude of custom and commercial off-the-shelf applications that use Active Directory, AWS does not and cannot perform formal or broad verification of third-party application compatibility with Simple AD. Although AWS works with customers in an attempt to overcome any potential application installation challenges they might encounter, we are unable to guarantee that any application is or will continue to be compatible with Simple AD.

The following third-party applications are compatible with Simple AD:

- Microsoft Internet Information Services (IIS) on the following platforms:
  - Windows Server 2003 R2
  - Windows Server 2008 R1
  - Windows Server 2008 R2
  - Windows Server 2012
  - Windows Server 2012 R2
- Microsoft SQL Server:
  - SQL Server 2005 R2 (Express, Web, and Standard editions)
  - SQL Server 2008 R2 (Express, Web, and Standard editions)
  - SQL Server 2012 (Express, Web, and Standard editions)
  - SQL Server 2014 (Express, Web, and Standard editions)
- Microsoft SharePoint:
  - SharePoint 2010 Foundation
  - SharePoint 2010 Enterprise
  - SharePoint 2013 Enterprise

Customers can choose to use AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD (p. 8)) for a higher level of compatibility based on actual Active Directory.

Troubleshooting Simple AD

The following can help you troubleshoot some common issues you might encounter when creating or using your directory.

Topics

Password recovery

If a user forgets a password or is having trouble signing in to either your Simple AD or AWS Managed Microsoft AD directory, you can reset their password using either the AWS Management Console, Windows PowerShell or the AWS CLI.

For more information, see Reset a user password (p. 232).

I receive a "KDC can't fulfill requested option" error when adding a user to Simple AD

This can occur when the Samba CLI client does not correctly send the 'net' commands to all domain controllers. If you see this error message when using the 'net ads' command to add a user to your Simple AD directory, use the -S argument and specify the IP address of one of your domain controllers. If you still see the error, try the other domain controller. You can also use the Active Directory Administration Tools to add users to your directory. For more information, see Installing the Active Directory administration tools (p. 231).

I am not able to update the DNS name or IP address of an instance joined to my domain (DNS dynamic update)

DNS dynamic updates are not supported in Simple AD domains. You can instead make the changes directly by connecting to your directory using DNS Manager on an instance that is joined to your domain.

I cannot log onto SQL Server using a SQL Server account

You might receive an error if you attempt to use SQL Server Management Studio (SSMS) with a SQL Server account to log into SQL Server running on a Windows 2012 R2 EC2 instance or in Amazon RDS. The issue occurs when SSMS is run as a domain user and can result in the error "Login failed for user," even when valid credentials are provided. This is a known issue and AWS is actively working to resolve it.

To work around the issue, you can log into SQL Server with Windows Authentication instead of SQL Authentication. Or launch SSMS as a local user instead of a Simple AD domain user.

My directory is stuck in the "requested" state

If you have a directory that has been in the "Requested" state for more than five minutes, try deleting the directory and recreating it. If this problem persists, contact the AWS Support Center.

I receive an "AZ constrained" error when I create a directory

Some AWS accounts created before 2012 might have access to Availability Zones in the US East (N. Virginia), US West (N. California), or Asia Pacific (Tokyo) Region that do not support AWS Directory Service directories. If you receive an error such as this when creating a directory, choose a subnet in a different Availability Zone and try to create the directory again.

Some of my users cannot authenticate with my directory

Your user accounts must have Kerberos preauthentication enabled. This is the default setting for new user accounts, and it should not be modified. For more information about this setting, go to Preauthentication on Microsoft TechNet.

Topics

Simple AD directory status reasons

When a directory is impaired or inoperable, the directory status message contains additional information. The status message is displayed in the AWS Directory Service console, or returned in the `DirectoryDescription.StageReason` member by the `DescribeDirectories` API. For more information about the directory status, see Understanding your directory status (p. 51).

The following are the status messages for a Simple AD directory:

Topics

The directory service's elastic network interface is not attached

Description

The critical elastic network interface (ENI) that was created on your behalf during directory creation to establish network connectivity with your VPC is not attached to the directory instance. AWS applications backed by this directory will not be functional. Your directory cannot connect to your on-premises network.

Troubleshooting

If the ENI is detached but still exists, contact AWS Support. If the ENI is deleted, there is no way to resolve the issue and your directory is permanently unusable. You must delete the directory and create a new one.

Issue(s) detected by instance

Description

An internal error was detected by the instance. This usually signifies that the monitoring service is actively attempting to recover the impaired instances.

Troubleshooting

In most cases, this is a transient issue, and the directory eventually returns to the Active state. If the problem persists, contact AWS Support for more assistance.

The critical AWS Directory Service reserved user is missing from the directory

Description

When a Simple AD is created, AWS Directory Service creates a service account in the directory with the name `AWSAdminD-xxxxxxxxxx`. This error is received when this service account cannot be found. Without this account, AWS Directory Service cannot perform administrative functions on the directory, rendering the directory unusable.

Troubleshooting

To correct this issue, restore the directory to a previous snapshot that was created before the service account was deleted. Automatic snapshots are taken of your Simple AD directory one time a day. If it has been more than five days after this account was deleted, you may not be able to restore the directory to a state where this account exists. If you are not able to restore the directory from a snapshot where this account exists, your directory may become permanently unusable. If this is the case, you must delete your directory and create a new one.

The critical AWS Directory Service reserved user needs to belong to the Domain Admins group

Description

When a Simple AD is created, AWS Directory Service creates a service account in the directory with the name `AWSAdminD-xxxxxxxxxx`. This error is received when this service account is not a member of the `Domain Admins` group. Membership in this group is needed to give AWS Directory Service the privileges it needs to perform maintenance and recovery operations, such as transferring FSMO roles, domain joining new directory controllers, and restoring from snapshots.

Troubleshooting

Use the Active Directory Users and Computers tool to re-add the service account to the `Domain Admins` group.

The critical AWS Directory Service reserved user is disabled

Description

When a Simple AD is created, AWS Directory Service creates a service account in the directory with the name `AWSAdminD-xxxxxxxxxx`. This error is received when this service account is disabled. This account must be enabled so that AWS Directory Service can perform maintenance and recovery operations on the directory.

Troubleshooting

Use the Active Directory Users and Computers tool to re-enable the service account.

The main domain controller does not have all FSMO roles

Description

All the FSMO roles are not owned by the Simple AD directory controller. AWS Directory Service cannot guarantee certain behavior and functionality if the FSMO roles do not belong to the correct Simple AD directory controller.

Troubleshooting

Use Active Directory tools to move the FSMO roles back to the original working directory controller. For more information about moving the FSMO roles, go to https://docs.microsoft.com/troubleshoot/windows-server/identity/transfer-or-seize-fsmo-roles-in-ad-ds. If this does not correct the problem, please contact AWS Support for more assistance.

Domain controller replication failures

Description

The Simple AD directory controllers are failing to replicate with one another. This can be caused by one or more of the following issues:

- The security groups for the directory controllers does not have the correct ports open.
- The network ACLs are too restrictive.
- The VPC route table is not routing network traffic between the directory controllers correctly.
- Another instance has been promoted to a domain controller in the directory.

Troubleshooting

For more information about your VPC network requirements, see either AWS Managed Microsoft AD AWS Managed Microsoft AD prerequisites (p. 9), AD Connector AD Connector prerequisites (p. 187), or Simple AD Simple AD prerequisites (p. 227). If there is an unknown domain controller in your directory, you must demote it. If your VPC network setup is correct, but the error persists, please contact AWS Support for more assistance.

Security in AWS Directory Service

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS compliance programs. To learn about the compliance programs that apply to AWS Directory Service, see AWS Services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using AWS Directory Service. The following topics show you how to configure AWS Directory Service to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your AWS Directory Service resources.

Security topics

The following security topics can be found in this section:

Additional security topics

The following additional security topics can be found in this guide:

Accounts, trusts, and AWS resource access

Secure your directory

Logging and monitoring

- Monitor your AWS Managed Microsoft AD (p. 51)
- Monitor your AD Connector directory (p. 209)

Resilience

- Patching and maintenance for AWS Managed Microsoft AD (p. 20)

Identity and access management for AWS Directory Service

Access to AWS Directory Service requires credentials that AWS can use to authenticate your requests. Those credentials must have permissions to access AWS resources, such as an AWS Directory Service directory. The following sections provide details on how you can use AWS Identity and Access Management (IAM) and AWS Directory Service to help secure your resources by controlling who can access them:

- Authentication (p. 278)
- Access control (p. 279)

Authentication

You can access AWS as any of the following types of identities:

- **AWS account root user** – When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.
- **IAM user** – An IAM user is an identity within your AWS account that has specific custom permissions (for example, permissions to create a directory in AWS Managed Microsoft AD). You can use an IAM user name and password to sign in to secure AWS webpages like the AWS Management Console, AWS Discussion Forums, or the AWS Support Center.

  In addition to a user name and password, you can also generate access keys for each user. You can use these keys when you access AWS services programmatically, either through one of the several SDKs or by using the AWS Command Line Interface (CLI). The SDK and CLI tools use the access keys to cryptographically sign your request. If you don't use AWS tools, you must sign the request yourself. AWS Managed Microsoft AD supports *Signature Version 4*, a protocol for authenticating inbound API requests. For more information about authenticating requests, see Signature Version 4 signing process in the *AWS General Reference*.
- **IAM role** – An IAM role is an IAM identity that you can create in your account that has specific permissions. An IAM role is similar to an IAM user in that it is an AWS identity with permissions policies that determine what the identity can and cannot do in AWS. However, instead of being uniquely associated with one person, a role is intended to be assumable by anyone who needs it. Also, a role does not have standard long-term credentials such as a password or access keys associated with it. Instead, when you assume a role, it provides you with temporary security credentials for your role session. IAM roles with temporary credentials are useful in the following situations:

- **Federated user access** – Instead of creating an IAM user, you can use existing identities from AWS Directory Service, your enterprise user directory, or a web identity provider. These are known as *federated users*. AWS assigns a role to a federated user when access is requested through an identity provider. For more information about federated users, see Federated users and roles in the *IAM User Guide*.
- **AWS service access** – A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Creating a role to delegate permissions to an AWS service in the *IAM User Guide*.
- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see Using an IAM role to grant permissions to applications running on Amazon EC2 instances in the *IAM User Guide*.

Access control

You can have valid credentials to authenticate your requests, but unless you have permissions you cannot create or access AWS Directory Service resources. For example, you must have permissions to create an AWS Directory Service directory or to create a directory snapshot.

The following sections describe how to manage permissions for AWS Directory Service. We recommend that you read the overview first.

Overview of managing access permissions to your AWS Directory Service resources

Every AWS resource is owned by an AWS account, and permissions to create or access the resources are governed by permissions policies. An account administrator can attach permissions policies to IAM identities (that is, users, groups, and roles), and some services (such as AWS Lambda) also support attaching permissions policies to resources.

> **Note**
> An *account administrator* (or administrator user) is a user with administrator privileges. For more information, see IAM best practices in the *IAM User Guide*.

When granting permissions, you decide who is getting the permissions, the resources they get permissions for, and the specific actions that you want to allow on those resources.

Topics

AWS Directory Service resources and operations

In AWS Directory Service, the primary resource is a *directory*. AWS Directory Service supports directory snapshot resources as well. However, you can create snapshots only in the context of an existing directory. Therefore, a snapshot is referred to as a *subresource*.

These resources have unique Amazon Resource Names (ARNs) associated with them as shown in the following table.

Resource Type	ARN Format
Directory	`arn:aws:ds:region:account-id:directory/external-directory-id`
Snapshot	`arn:aws:ds:region:account-id:snapshot/external-snapshot-id`

AWS Directory Service provides a set of operations to work with the appropriate resources. For a list of available operations, see Directory Service Actions.

Understanding resource ownership

A *resource owner* is the AWS account that created a resource. That is, the resource owner is the AWS account of the *principal entity* (the root account, an IAM user, or an IAM role) that authenticates the request that creates the resource. The following examples illustrate how this works:

- If you use the root account credentials of your AWS account to create an AWS Directory Service resource, such as a directory, your AWS account is the owner of that resource.
- If you create an IAM user in your AWS account and grant permissions to create AWS Directory Service resources to that user, the user can also create AWS Directory Service resources. However, your AWS account, to which the user belongs, owns the resources.
- If you create an IAM role in your AWS account with permissions to create AWS Directory Service resources, anyone who can assume the role can create AWS Directory Service resources. Your AWS account, to which the role belongs, owns the AWS Directory Service resources.

Managing access to resources

A *permissions policy* describes who has access to what. The following section explains the available options for creating permissions policies.

> **Note**
> This section discusses using IAM in the context of AWS Directory Service. It doesn't provide detailed information about the IAM service. For complete IAM documentation, see What is IAM? in the *IAM User Guide*. For information about IAM policy syntax and descriptions, see IAM JSON policy reference in the *IAM User Guide*.

Policies attached to an IAM identity are referred to as *identity-based* policies (IAM polices) and policies attached to a resource are referred to as *resource-based* policies. AWS Directory Service supports only identity-based policies (IAM policies).

Topics

Identity-based policies (IAM policies)

You can attach policies to IAM identities. For example, you can do the following:

- **Attach a permissions policy to a user or a group in your account** – An account administrator can use a permissions policy that is associated with a particular user to grant permissions for that user to create an AWS Directory Service resource, such as a new directory.
- **Attach a permissions policy to a role (grant cross-account permissions)** – You can attach an identity-based permissions policy to an IAM role to grant cross-account permissions. For example, the administrator in Account A can create a role to grant cross-account permissions to another AWS account (for example, Account B) or an AWS service as follows:

  1. Account A administrator creates an IAM role and attaches a permissions policy to the role that grants permissions on resources in Account A.
  2. Account A administrator attaches a trust policy to the role identifying Account B as the principal who can assume the role.
  3. Account B administrator can then delegate permissions to assume the role to any users in Account B. Doing this allows users in Account B to create or access resources in Account A. The principal in the trust policy can also be an AWS service principal if you want to grant an AWS service permissions to assume the role.

  For more information about using IAM to delegate permissions, see Access management in the *IAM User Guide*.

The following permissions policy grants permissions to a user to run all of the actions that begin with `Describe`. These actions show information about an AWS Directory Service resource, such as a directory or snapshot. Note that the wildcard character (*) in the `Resource` element indicates that the actions are allowed for all AWS Directory Service resources owned by the account.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"ds:Describe*",
         "Resource":"*"
      }
   ]
}
```

For more information about using identity-based policies with AWS Directory Service, see Using identity-based policies (IAM policies) for AWS Directory Service (p. 282). For more information about users, groups, roles, and permissions, see Identities (users, groups, and roles) in the *IAM User Guide*.

Resource-based policies

Other services, such as Amazon S3, also support resource-based permissions policies. For example, you can attach a policy to an S3 bucket to manage access permissions to that bucket. AWS Directory Service doesn't support resource-based policies.

Specifying policy elements: Actions, effects, resources, and principals

For each AWS Directory Service resource, the service defines a set of API operations. For more information, see AWS Directory Service resources and operations (p. 280). For a list of available API operations, see Directory Service Actions.

To grant permissions for these API operations, AWS Directory Service defines a set of actions that you can specify in a policy. Note that performing an API operation can require permissions for more than one action.

The following are the basic policy elements:

- **Resource** – In a policy, you use an Amazon Resource Name (ARN) to identify the resource to which the policy applies. For AWS Directory Service resources, you always use the wildcard character (*) in IAM policies. For more information, see AWS Directory Service resources and operations (p. 280).
- **Action** – You use action keywords to identify resource operations that you want to allow or deny. For example, the `ds:DescribeDirectories` permission allows the user permissions to perform the AWS Directory Service `DescribeDirectories` operation.
- **Effect** – You specify the effect when the user requests the specific action. This can be either allow or deny. If you don't explicitly grant access to (allow) a resource, access is implicitly denied. You can also explicitly deny access to a resource, which you might do to make sure that a user cannot access it, even if a different policy grants access.
- **Principal** – In identity-based policies (IAM policies), the user that the policy is attached to is the implicit principal. For resource-based policies, you specify the user, account, service, or other entity that you want to receive permissions (applies to resource-based policies only). AWS Directory Service doesn't support resource-based policies.

To learn more about IAM policy syntax and descriptions, see IAM JSON policy reference in the *IAM User Guide*.

For a table showing all of the AWS Directory Service API actions and the resources that they apply to, see AWS Directory Service API permissions: Actions, resources, and conditions reference (p. 288).

Specifying conditions in a policy

When you grant permissions, you can use the access policy language to specify the conditions when a policy should take effect. For example, you might want a policy to be applied only after a specific date. For more information about specifying conditions in a policy language, see Condition in the *IAM User Guide*.

To express conditions, you use predefined condition keys. There are no condition keys specific to AWS Directory Service. However, there are AWS condition keys that you can use as appropriate. For a complete list of AWS keys, see Available global condition keys in the *IAM User Guide*.

Using identity-based policies (IAM policies) for AWS Directory Service

This topic provides examples of identity-based policies in which an account administrator can attach permissions policies to IAM identities (that is, users, groups, and roles).

> **Important**
> We recommend that you first review the introductory topics that explain the basic concepts and options available for you to manage access to your AWS Directory Service resources. For more information, see Overview of managing access permissions to your AWS Directory Service resources (p. 279).

The sections in this topic cover the following:

- Permissions required to use the AWS Directory Service console (p. 284)
- AWS managed (predefined) policies for AWS Directory Service (p. 284)

- Customer managed policy examples (p. 285)
- Using tags with IAM policies (p. 286)

The following shows an example of a permissions policy.

```
{
   "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowDsEc2IamGetRole",
            "Effect": "Allow",
            "Action": [
                "ds:CreateDirectory",
                "ec2:RevokeSecurityGroupIngress",
                "ec2:CreateNetworkInterface",
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeVpcs",
                "ec2:CreateSecurityGroup",
                "ec2:RevokeSecurityGroupEgress",
                "ec2:DeleteSecurityGroup",
                "ec2:DeleteNetworkInterface",
                "ec2:DescribeSubnets",
                "iam:GetRole"
            ],
            "Resource": "*"
        },
        {
            "Sid": "WarningAllowsCreatingRolesWithDirSvcPrefix",
            "Effect": "Allow",
            "Action": [
                "iam:CreateRole",
                "iam:PutRolePolicy"
            ],
            "Resource": "arn:aws:iam::111122223333:role/DirSvc*"
        },
        {
            "Sid": "AllowPassRole",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "cloudwatch.amazonaws.com"
                }
            }
        }
    ]
}
```

The policy includes the following:

- The first statement grants permission to create a AWS Directory Service directory. AWS Directory Service doesn't support permissions for this particular action at the resource-level. Therefore, the policy specifies a wildcard character (*) as the `Resource` value.
- The second statement grants permissions to certain IAM actions. The access to IAM actions is needed so that AWS Directory Service can read and create IAM roles on your behalf. The wildcard character (*) at the end of the `Resource` value means that the statement allows permission for the IAM actions on any IAM role. To limit this permission to a specific role, replace the wildcard character (*) in the resource ARN with the specific role name. For more information, see IAM Actions.

- The third statement grants permissions to a specific set of Amazon EC2 resources that are necessary to allow AWS Directory Service to create, configure, and destroy its directories. The wildcard character (*) at the end of the `Resource` value means that the statement allows permission for the EC2 actions on any EC2 resource or subresource. To limit this permission to a specific role, replace the wildcard character (*) in the resource ARN with the specific resource or subresource. For more information, see Amazon EC2 Actions

The policy doesn't specify the `Principal` element because in an identity-based policy you don't specify the principal who gets the permission. When you attach policy to a user, the user is the implicit principal. When you attach a permission policy to an IAM role, the principal identified in the role's trust policy gets the permissions.

For a table showing all of the AWS Directory Service API actions and the resources that they apply to, see AWS Directory Service API permissions: Actions, resources, and conditions reference (p. 288).

Permissions required to use the AWS Directory Service console

For a user to work with the AWS Directory Service console, that user must have permissions listed in the preceding policy or the permissions granted by the Directory Service Full Access Role or Directory Service Read Only role, described in AWS managed (predefined) policies for AWS Directory Service (p. 284).

If you create an IAM policy that is more restrictive than the minimum required permissions, the console won't function as intended for users with that IAM policy.

AWS managed (predefined) policies for AWS Directory Service

AWS addresses many common use cases by providing standalone IAM policies that are created and administered by AWS. Managed policies grant necessary permissions for common use cases so you can avoid having to investigate what permissions are needed. For more information, see AWS managed policies in the *IAM User Guide*.

The following AWS managed policies, which you can attach to users in your account, are specific to AWS Directory Service:

- **AWSDirectoryServiceReadOnlyAccess** – Grants a user or group read-only access to all AWS Directory Service resources, EC2 subnets, EC2 network interfaces, and Amazon Simple Notification Service (Amazon SNS) topics and subscriptions for the root AWS account. For more information, see Using AWS managed policies with AWS Directory Service (p. 133).
- **AWSDirectoryServiceFullAccess** – Grants a user or group the following:
  - Full access to AWS Directory Service
  - Access to key Amazon EC2 services required to use AWS Directory Service
  - Ability to list Amazon SNS topics
  - Ability to create, manage, and delete Amazon SNS topics with a name beginning with “DirectoryMonitoring”

  For more information, see Using AWS managed policies with AWS Directory Service (p. 133).

In addition, there are other AWS managed policies that are suitable for use with other IAM roles. These policies are assigned to the roles that are associated with users in your AWS Directory Service directory. These policies are required for those users to have access to other AWS resources, such as Amazon EC2. For more information, see Grant users and groups access to AWS resources (p. 130).

You can also create custom IAM policies that allow users to access the required API actions and resources. You can attach these custom policies to the IAM users or groups that require those permissions.

Customer managed policy examples

In this section, you can find example user policies that grant permissions for various AWS Directory Service actions.

> **Note**
> All examples use the US West (Oregon) Region (`us-west-2`) and contain fictitious account IDs.

Examples

Example 1: Allow a user to perform any Describe action on any AWS Directory Service resource

The following permissions policy grants permissions to a user to run all of the actions that begin with `Describe`. These actions show information about an AWS Directory Service resource, such as a directory or snapshot. Note that the wildcard character (*) in the `Resource` element indicates that the actions are allowed for all AWS Directory Service resources owned by the account.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"ds:Describe*",
         "Resource":"*"
      }
   ]
}
```

Example 2: Allow a user to create a directory

The following permissions policy grants permissions to allow a user to create a directory and all other related resources, such as snapshots and trusts. In order to do so, permissions to certain Amazon EC2 services are also required.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action": [
                "ds:Create*",
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CreateNetworkInterface",
                "ec2:CreateSecurityGroup",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteSecurityGroup",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:RevokeSecurityGroupEgress",
                "ec2:RevokeSecurityGroupIngress"
                  ],
         "Resource":"*"
```

```
            ]
         }
      ]
}
```

Using tags with IAM policies

You can apply tag-based resource-level permissions in the IAM policies you use for most AWS Directory Service API actions. This gives you better control over what resources a user can create, modify, or use. You use the `Condition` element (also called the `Condition` block) with the following condition context keys and values in an IAM policy to control user access (permissions) based on a resource's tags:

- Use `aws:ResourceTag`/**`tag-key`**: **`tag-value`** to allow or deny user actions on resources with specific tags.
- Use `aws:ResourceTag`/**`tag-key`**: **`tag-value`** to require that a specific tag be used (or not used) when making an API request to create or modify a resource that allows tags.
- Use `aws:TagKeys`: [**`tag-key`**, ...] to require that a specific set of tag keys be used (or not used) when making an API request to create or modify a resource that allows tags.

> **Note**
> The condition context keys and values in an IAM policy apply only to those AWS Directory Service actions where an identifier for a resource capable of being tagged is a required parameter.

Controlling access using tags in the *IAM User Guide* has additional information on using tags. The IAM JSON policy reference section of that guide has detailed syntax, descriptions, and examples of the elements, variables, and evaluation logic of JSON policies in IAM.

The following tag policy example allows all `ds` calls as long as it contains the tag key-value pair "`fooKey`":"`fooValue`".

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"VisualEditor0",
         "Effect":"Allow",
         "Action":[
            "ds:*"
         ],
         "Resource":"*",
         "Condition":{
            "StringEquals":{
               "aws:ResourceTag/fooKey":"fooValue"
            }
         }
      },
      {
         "Effect":"Allow",
         "Action":[
            "ec2:*"
         ],
         "Resource":"*"
      }
   ]
}
```

The following resource policy example allows all `ds` calls as long as the resource contains the directory ID "`d-1234567890`".

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"VisualEditor0",
         "Effect":"Allow",
         "Action":[
            "ds:*"
         ],
         "Resource":"arn:aws:ds:us-east-1:123456789012:directory/d-1234567890"
      },
      {
         "Effect":"Allow",
         "Action":[
            "ec2:*"
         ],
         "Resource":"*"
      }
   ]
}
```

For more information about ARNs, see Amazon Resource Names (ARNs) and AWS Service Namespaces.

The following list of AWS Directory Service API operations support tag-based resource-level permissions:

- AcceptSharedDirectory
- AddIpRoutes
- AddTagsToResource
- CancelSchemaExtension
- CreateAlias
- CreateComputer
- CreateConditionalForwarder
- CreateSnapshot
- CreateLogSubscription
- CreateTrust
- DeleteConditionalForwarder
- DeleteDirectory
- DeleteLogSubscription
- DeleteSnapshot
- DeleteTrust
- DeregisterEventTopic
- DescribeConditionalForwarders
- DescribeDomainControllers
- DescribeEventTopics
- DescribeSharedDirectories
- DescribeSnapshots
- DescribeTrusts
- DisableRadius
- DisableSso
- EnableRadius

- EnableSso
- GetSnapshotLimits
- ListIpRoutes
- ListSchemaExtensions
- ListTagsForResource
- RegisterEventTopic
- RejectSharedDirectory
- RemoveIpRoutes
- RemoveTagsFromResource
- ResetUserPassword
- RestoreFromSnapshot
- ShareDirectory
- StartSchemaExtension
- UnshareDirectory
- UpdateConditionalForwarder
- UpdateNumberOfDomainControllers
- UpdateRadius
- UpdateTrust
- VerifyTrust

AWS Directory Service API permissions: Actions, resources, and conditions reference

When you are setting up Access control (p. 279) and writing permissions policies that you can attach to an IAM identity (identity-based policies), you can use the following table as a reference. The list includes the following:

- Each AWS Directory Service API operation
- The corresponding actions for which you can grant permissions to perform the action
- The AWS resource for which you can grant the permissions

You specify the actions in the policy's `Action` field and the resource value in the policy's `Resource` field.

> **Note**
> Some AWS applications may require use of nonpublic AWS Directory Service API operations such as `ds:AuthorizeApplication`, `ds:CheckAlias`, `ds:CreateIdentityPoolDirectory`, and `ds:UnauthorizeApplication` in their policies.

You can use AWS global condition keys in your AWS Directory Service policies to express conditions. For a complete list of AWS keys, see Available Global Condition Keys in the *IAM User Guide*.

> **Note**
> To specify an action, use the `ds:` prefix followed by the API operation name (for example, `ds:CreateDirectory`).

Related Topics

- Access control (p. 279)

Logging and monitoring in AWS Directory Service

As a best practice, monitor your organization to ensure that changes are logged. This helps you to ensure that any unexpected change can be investigated and unwanted changes can be rolled back. AWS Directory Service currently supports the following two AWS services so that you can monitor your organization and the activity that happens within it.

- Amazon CloudWatch - You can use CloudWatch Events with the AWS Managed Microsoft AD directory type. For more information, see Enable log forwarding (p. 55). Additionally, you can use CloudWatch Metrics to monitor domain controller performance. For more information, see Use Amazon CloudWatch metrics to determine when to add domain controllers (p. 145).
- AWS CloudTrail - You can use CloudTrail with all AWS Directory Service directory types. For more information, see Logging AWS Directory Service API calls with CloudTrail.

Compliance validation for AWS Directory Service

Third-party auditors assess the security and compliance of AWS Directory Service as part of multiple AWS compliance programs. With AWS Managed Microsoft AD, these include SOC, PCI, FedRAMP, HIPAA, and others. For more information, see Manage compliance for AWS Managed Microsoft AD (p. 41).

For a list of AWS services in scope of specific compliance programs, see AWS services in scope by compliance program. For general information, see AWS compliance programs.

You can use AWS Artifact to download third-party audit reports. For more information, see Downloading reports in AWS Artifact.

When you use AWS Directory Service, your compliance responsibility is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- Security and compliance quick start guides – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- Architecting for HIPAA security and compliance whitepaper – This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.
- AWS compliance resources – This collection of workbooks and guides might apply to your industry and location.
- AWS Config – This AWS service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.

Resilience in AWS Directory Service

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between Availability Zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see AWS global infrastructure.

In addition to the AWS global infrastructure, AWS Directory Service offers the ability to take manual snapshots of data at any point in time to help support your data resiliency and backup needs. For more information, see Snapshot or restore your directory (p. 128).

Infrastructure security in AWS Directory Service

As a managed service, AWS Directory Service is protected by the AWS global network security procedures that are described in the Amazon Web Services: Overview of security processes whitepaper.

You use AWS published API calls to access AWS Directory Service through the network. Clients must support Transport Layer Security (TLS). We recommend TLS 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal information processing standard (FIPS) 140-2.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

Cross-service confused deputy prevention

The confused deputy problem is a security issue where an entity that doesn't have permission to perform an action can coerce a more-privileged entity to perform the action. In AWS, cross-service impersonation can result in the confused deputy problem. Cross-service impersonation can occur when one service (the *calling service*) calls another service (the *called service*). The calling service can be manipulated to use its permissions to act on another customer's resources in a way it should not otherwise have permission to access. To prevent this, AWS provides tools that help you protect your data for all services with service principals that have been given access to resources in your account.

We recommend using the `aws:SourceArn` and `aws:SourceAccount` global condition context keys in resource policies to limit the permissions that AWS Directory Service for Microsoft Active Directory gives another service to the resource. If the `aws:SourceArn` value does not contain the account ID, such as an Amazon S3 bucket ARN, you must use both global condition context keys to limit permissions. If you use both global condition context keys and the `aws:SourceArn` value contains the account ID, the `aws:SourceAccount` value and the account in the `aws:SourceArn` value must use the same account ID when used in the same policy statement. Use `aws:SourceArn` if you want only one resource to be associated with the cross-service access. Use `aws:SourceAccount` if you want to allow any resource in that account to be associated with the cross-service use.

For the following example, the value of `aws:SourceArn` must be a CloudWatch log group.

The most effective way to protect against the confused deputy problem is to use the `aws:SourceArn` global condition context key with the full ARN of the resource. If you don't know the full ARN of the resource or if you are specifying multiple resources, use the `aws:SourceArn` global context condition key with wildcards (`*`) for the unknown portions of the ARN. For example, `arn:aws:servicename:*:123456789012:*`.

The following example shows how you can use the `aws:SourceArn` and `aws:SourceAccount` global condition context keys in AWS Managed Microsoft AD to prevent the confused deputy problem.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "ds.amazonaws.com"
    },
    "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
    "Resource": [
      "arn:aws:logs:YOUR_REGION:YOUR_ACCOUNT_NUMBER:log-group:/aws/
directoryservice/YOUR_LOG_GROUP:*"
    ],
    "Condition": {
      "StringEquals": {
        "aws:SourceArn":
 "arn:aws:ds:YOUR_REGION:YOUR_ACCOUNT_NUMBER:directory/YOUR_DIRECTORY_ID",
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

For the following example, the value of `aws:SourceArn` must be a SNS topic in your account. For example, you can use something like `arn:aws:sns:ap-southeast-1:123456789012:DirectoryMonitoring_d-966739499f` where “ap-southeast-1” is your region, "123456789012" is your customer id and "DirectoryMonitoring_d-966739499f" is the Amazon SNS topic name that you created.

The most effective way to protect against the confused deputy problem is to use the `aws:SourceArn` global condition context key with the full ARN of the resource. If you don't know the full ARN of the resource or if you are specifying multiple resources, use the `aws:SourceArn` global context condition key with wildcards (`*`) for the unknown portions of the ARN. For example, `arn:aws:servicename:*:123456789012:*`.

The following example shows how you can use the `aws:SourceArn` and `aws:SourceAccount` global condition context keys in AWS Managed Microsoft AD to prevent the confused deputy problem.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "ds.amazonaws.com"
    },
    "Action": ["SNS:GetTopicAttributes",
     "SNS:SetTopicAttributes",
        "SNS:AddPermission",
     "SNS:RemovePermission",
     "SNS:DeleteTopic",
     "SNS:Subscribe",
     "SNS:ListSubscriptionsByTopic",
     "SNS:Publish"],
    "Resource": [
      "arn:aws:sns:YOUR_REGION:YOUR_ACCOUNT_NUMBER:YOUR_SNS_TOPIC_NAME"
    ],
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:sns:YOUR_REGION:YOUR_ACCOUNT_NUMBER:directory/
YOUR_EXTERNAL_DIRECTORY_ID"
```

```
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

Service level agreement for AWS Directory Service

AWS Directory Service is a highly available service, and is built on AWS-managed infrastructure. It is backed by a service level agreement that defines our service availability policy.

For more information, see Service level agreement for AWS Directory Service.

Region availability for AWS Directory Service

The following table provides a list describing which Region-specific endpoints are supported by directory type.

Region name	Region	Endpoint	Protocol	AWS Managed Microsoft AD	AD Connecto	Simple AD
US East (Ohio)	us-east-2	ds.us-east-2.amazonaws.com	HTTPS	✓Yes	✓Yes	⊗No
US East (N. Virginia)	us-east-1	ds.us-east-1.amazonaws.com	HTTPS	✓Yes	✓Yes	✓Yes
US West (N. California)	us-west-1	ds.us-west-1.amazonaws.com	HTTPS	✓Yes	✓Yes	⊗No
US West (Oregon)	us-west-2	ds.us-west-2.amazonaws.com	HTTPS	✓Yes	✓Yes	✓Yes
Africa (Cape Town)	af-south-1	ds.af-south-1.amazonaws.com	HTTPS	✓Yes	✓Yes	⊗No
Asia Pacific (Hong Kong)	ap-east-1	ds.ap-east-1.amazonaws.com	HTTPS	✓Yes	✓Yes	⊗No
Asia Pacific (Mumbai)	ap-south-1	ds.ap-south-1.amazonaws.com	HTTPS	✓Yes	✓Yes	⊗No
Asia Pacific (Osaka)	ap-northeast-3	ds.ap-northeast-3.amazonaws.com	HTTPS	✓Yes	✓Yes	⊗No
Asia Pacific (Seoul)	ap-northeast-2	ds.ap-northeast-2.amazonaws.com	HTTPS	✓Yes	✓Yes	⊗No
Asia Pacific (Singapore)	ap-southeast-1	ds.ap-southeast-1.amazonaws.com	HTTPS	✓Yes	✓Yes	✓Yes
Asia Pacific (Sydney)	ap-southeast-2	ds.ap-southeast-2.amazonaws.com	HTTPS	✓Yes	✓Yes	✓Yes

Region name	Region	Endpoint	Protocol	AWS Managed Microsoft AD	AD Connecto	Simple AD
Asia Pacific (Tokyo)	ap-northeast-1	ds.ap-northeast-1.amazonaws.com	HTTPS	✓Yes	✓Yes	✓Yes
Canada (Central)	ca-central-1	ds.ca-central-1.amazonaws.com	HTTPS	✓Yes	✓Yes	⊗No
China (Beijing)	cn-north-1	ds.cn-north-1.amazonaws.com.cn	HTTPS	✓Yes	✓Yes	⊗No
China (Ningxia)	cn-northwest-1	ds.cn-northwest-1.amazonaws.com.cn	HTTPS	✓Yes	✓Yes	⊗No
Europe (Frankfurt)	eu-central-1	ds.eu-central-1.amazonaws.com	HTTPS	✓Yes	✓Yes	⊗No
Europe (Ireland)	eu-west-1	ds.eu-west-1.amazonaws.com	HTTPS	✓Yes	✓Yes	✓Yes
Europe (London)	eu-west-2	ds.eu-west-2.amazonaws.com	HTTPS	✓Yes	✓Yes	⊗No
Europe (Paris)	eu-west-3	ds.eu-west-3.amazonaws.com	HTTPS	✓Yes	✓Yes	⊗No
Europe (Stockholm)	eu-north-1	ds.eu-north-1.amazonaws.com	HTTPS	✓Yes	✓Yes	⊗No
Europe (Milan)	eu-south-1	ds.eu-south-1.amazonaws.com	HTTPS	✓Yes	✓Yes	⊗No
Middle East (Bahrain)	me-south-1	ds.me-south-1.amazonaws.com	HTTPS	✓Yes	✓Yes	⊗No
South America (São Paulo)	sa-east-1	ds.sa-east-1.amazonaws.com	HTTPS	✓Yes	✓Yes	⊗No
AWS GovCloud (US-West)	us-gov-west-1	ds.us-gov-west-1.amazonaws.com	HTTPS	✓Yes	✓Yes	⊗No
AWS GovCloud (US-East)	us-gov-east-1	ds.us-gov-east-1.amazonaws.com	HTTPS	✓Yes	✓Yes	⊗No

For information about using AWS Directory Service in the AWS GovCloud (US-West) Region and AWS GovCloud (US-East) Region, see Service endpoints.

For information about using AWS Directory Service in the Beijing and Ningxia Regions, see Endpoints and ARNs for Amazon Web Services in China.

Browser compatibility

AWS applications and services such as WorkSpaces, Amazon WorkMail, Amazon Connect, Amazon Chime, Amazon WorkDocs, and AWS Single Sign-On (AWS SSO) all require valid sign-in credentials from a compatible browser before you can access them. The following table describes only the browsers and browser versions that are compatible for sign-ins.

Browser	Version	Compatibility
Microsoft Internet Explorer	Desktop IE versions 7 and below	Not compatible
	Desktop IE versions 8, 9, and 10	Compatible only when running Windows 7 or newer and TLS 1.1 enabled. See What is TLS? (p. 298) for more information.
	Desktop IE versions 11 and above	Compatible
	Mobile IE versions 10 and below	Not compatible
	Mobile IE versions 11 and above	Compatible
Microsoft Edge	All versions	Compatible
Mozilla Firefox	Firefox 23 and below	Not compatible
	Firefox 24 to 26	Compatible, but not by default.
	Firefox 27 and above	Compatible
Google Chrome	Google Chrome 21 and below	Not compatible
	Google Chrome 22 to 37	Compatible, but not by default.
	Google Chrome 38 and above	Compatible
Apple Safari	Desktop Safari versions 6 and below for OS X 10.8 (Mountain Lion) and below	Not compatible
	Desktop Safari versions 7 and higher for OS X 10.9 (Mavericks) and higher	Compatible
	Mobile Safari for iOS 4 and below	Not compatible
	Mobile Safari versions 5 and higher for iOS 5 and higher	Compatible

Now that you've verified you are using a supported version of your browser, we recommend that you also review the section below to verify your browser has been configured to use the Transport Layer Security (TLS) setting required by AWS.

What is TLS?

TLS is a protocol web browsers and other applications use to exchange data securely over a network. TLS ensures that a connection to a remote endpoint is the intended endpoint through encryption and endpoint identity verification. The versions of TLS, to date, are TLS 1.0, 1.1, 1.2 and 1.3.

Which TLS versions are supported by AWS SSO

AWS applications and services support TLS 1.1, 1.2 and 1.3 for secure sign-ins. As of October 30th 2019, TLS 1.0 is no longer supported so it is important that all browsers are configured to support TLS 1.1 or above. This means, you will not be able to sign-in to AWS applications and services if you access them while TLS 1.0 is enabled. For assistance making this change, contact your admin.

How do I enable supported TLS versions in my browser

It depends on your browser. Usually you can find this setting under the advanced settings area in your browser settings. For example, in Internet Explorer you’ll find various TLS options under **Internet Properties**, the **Advanced** tab, and then under the **Security** section. Check your browser manufacturers Help web site for specific instructions.

Document history

The following table describes the important changes since the last release of the *AWS Directory Service Administrator Guide*.

- **Latest documentation update:** August 25, 2021

update-history-change	update-history-description	update-history-date
Simple AD VPC Endpoints	Added content regarding which VPC endpoints should not be configured.	August 25, 2021
AD Connector VPC Endpoints	Added content regarding which VPC endpoints should not be configured.	August 25, 2021
Smart card support	Added content on support for smart cards and Amazon WorkSpaces Application Manager in AWS GovCloud (US-West) Region	December 1, 2020
Password reset	Added content on how to reset user passwords using the AWS Management Console, Windows PowerShell and AWS CLI.	January 2, 2019
Directory sharing	Added documentation for using directory sharing with AWS Managed Microsoft AD.	September 25, 2018
Migrated content to new Amazon Cloud Directory Developer Guide	The Amazon Cloud Directory content previously in this guide has been moved to the new *Amazon Cloud Directory Developer Guide*.	June 21, 2018
Complete overhaul of the admin guide TOC	Reorganized the content to more directly map to customer needs and added new content where needed.	April 5, 2018
AWS delegated groups	Added list of AWS delegated groups that can be assigned to on-premises users.	March 8, 2018
Fine-grained password policies	Added new password policies content.	July 5, 2017
Additional domain controllers	Added information for adding more domain controllers to your AWS Managed Microsoft AD.	June 30, 2017

Tutorials	Added new tutorials for testing a AWS Managed Microsoft AD lab environment.	June 21, 2017
MFA with AWS Managed Microsoft AD	Added documentation for using MFA with AWS Managed Microsoft AD.	February 13, 2017
Amazon Cloud Directory	New directory type introduced.	January 26, 2017
Schema extensions	Added documentation for schema extensions with AWS Directory Service for Microsoft Active Directory.	November 14, 2016
Major reorganization of the Directory Service admin guide	Reorganized the content to more directly map to customer needs.	November 14, 2016
SNS notifications	Added documentation for SNS notifications.	February 25, 2016
Authorization and authentication	Added additional documentation for using IAM with Directory Services.	February 25, 2016
AWS Managed Microsoft AD	Added documentation for AWS Managed Microsoft AD, and combined guides into a single guide.	November 17, 2015
Allow Linux instances to be joined to a Simple AD directory	Added documentation for joining a Linux instance to a Simple AD directory.	July 23, 2015
Guide separation	The *AWS Directory Service administration guide* is split into separate guides, the *Simple AD guide* and the *AD Connector guide*.	July 14, 2015
Single sign-on support	Added single sign-on documentation.	March 31, 2015
New guide	This is the first release of the *AWS Directory Service Administration Guide*.	October 21, 2014