Implementing Active Directory Domain Services in the AWS Cloud

Mike Pfeiffer

March 2014

Table of Contents

Abstract

This reference implementation guide includes architectural considerations and configuration steps for implementing highly available Active Directory Domain Services (AD DS) in the Amazon Web Services (AWS) cloud. We'll discuss best practices for launching the necessary AWS services, such as Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Virtual Private Cloud (Amazon VPC), in two scenarios:

- An AWS cloud-based Active Directory Domain Services deployment
- The extension of on-premises Active Directory Domain Services to the AWS cloud

We also provide links to automated AWS CloudFormation templates that you can leverage for your implementation or launch directly into your AWS account.

Amazon Web Services provides a comprehensive set of services and tools for deploying Microsoft Windows-based workloads on its reliable and secure cloud infrastructure. Active Directory Domain Services (AD DS) and Domain Name Server (DNS) are core Windows services that provide the foundation for many enterprise class Microsoft based solutions including Microsoft SharePoint, Microsoft Exchange, and .NET applications.

This guide is aimed at organizations running workloads in the AWS cloud that require secure, low latency connectivity to Active Directory Domain and DNS services. After reading this guide, IT infrastructure personnel should have a good understanding of how to design and implement a solution to launch AD DS in the AWS cloud or extend on-premises Active Directory Domain Services into the AWS cloud.

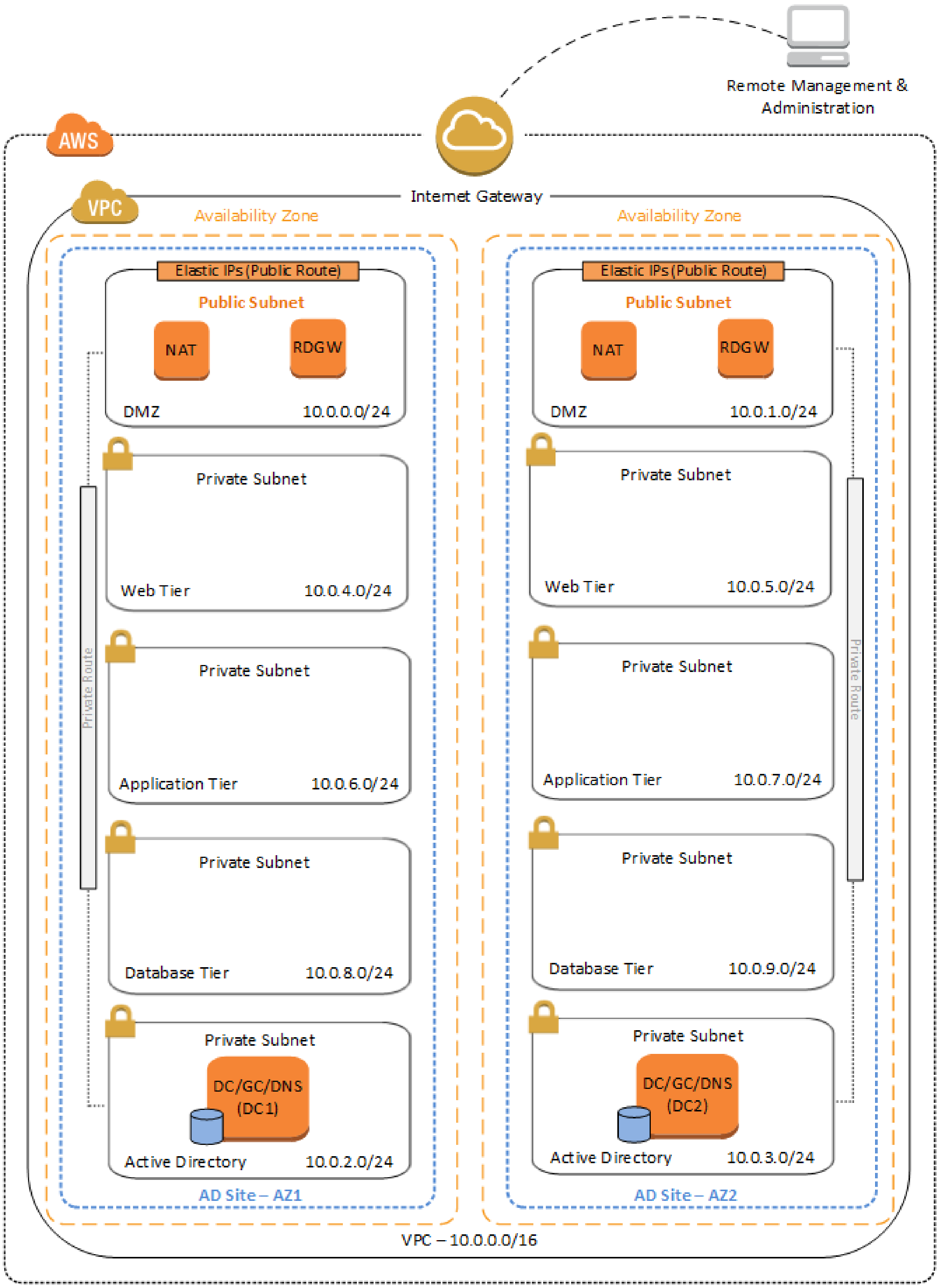


Figure 1: Reference Architecture for Highly Available AD DS in the AWS Cloud

What We'll Cover

This guide includes the following topics to help you deploy Active Directory Domain Services (AD DS) in the AWS cloud.

Architecture Considerations

Implementing a functional AD DS deployment in the AWS cloud requires a good understanding of specific AWS services. In this section, we discuss how to use Amazon VPC to define your networks in the cloud. Additionally, we cover considerations for Domain Controller placement, AD DS Sites and Services configuration, and how DNS and DHCP work in the Amazon VPC.

Sample Deployment Scenario #1: Deploy Active Directory Domain Services in the AWS Cloud

Our first deployment scenario is based on a new installation of AD DS in the AWS cloud. We provide an AWS CloudFormation template that you can use to deploy this solution which performs the following tasks:

- Set up the Amazon VPC, including subnets in two Availability Zones.
- Configure private and public routes.
- Launch Windows Server 2012 Amazon Machine Images (AMIs) and set up and configure AD DS and AD integrated DNS.
- Create empty private subnets in each Availability Zone into which you can deploy additional servers.
- Configure security groups and rules for traffic between application tiers.
- Set up and configure AD Sites and Subnets.
- Enable ingress traffic into the Amazon VPC for administrative access to Remote Desktop Gateway and NAT instances.

When the installation is complete, you will have deployed the architecture shown in Figure 1.

Considerations for Extending Existing Active Directory Domain Services into the AWS Cloud

This section outlines additional architectural considerations for leveraging existing AD DS and extending your on-premises network to the Amazon VPC.

Sample Deployment Scenario #2: Extend on-premises Active Directory Domain Services to the AWS Cloud

For our second deployment scenario, we provide an AWS CloudFormation template that will launch a base architecture performing the following tasks:

- Set up the Amazon VPC, including subnets in two Availability Zones.
- Configure private and public routes.
- Launch Windows Server 2012 Amazon Machine Images (AMIs).
- Create empty private subnets in each Availability Zone into which you can deploy additional application servers.
- Configure security groups and rules for traffic between application tiers.
- Enable ingress traffic into the VPC for administrative access to Remote Desktop Gateway and NAT instances.

This scenario will use the same base architecture shown in Figure 1. You will still need to perform several manual post-configuration tasks, such as extending your network to the Amazon VPC and promoting your Domain Controllers. These steps are discussed later in this guide.

Before You Get Started

Implementing AD DS in the AWS cloud is an advanced topic. If you are new to AWS, see the Getting Started section of the AWS documentation. In addition, familiarity with the following technologies is recommended:

- Amazon EC2
- Amazon VPC
- Windows Server 2012 or 2008 R2
- Windows Server Active Directory and DNS

This guide focuses on infrastructure configuration topics that require careful consideration when you are planning and deploying AD DS, Domain Controller instances, and DNS services in the AWS cloud. We don’t cover general Windows Server installation and software configuration tasks. For general software configuration guidance and best practices, consult the Microsoft product documentation.

We provide links to AWS CloudFormation templates that you can leverage for your implementation or launch directly into your AWS account. For more information about using AWS CloudFormation templates, see the AWS CloudFormation User Guide.

Architecture Considerations

These considerations provide background for automation decisions and explain additional steps you may need or want to take when launching the templates or when manually configuring this architecture

Virtual Private Cloud

In this guide, we will discuss two scenarios for running Active Directory Domain Services (AD DS) in an Amazon Virtual Private Cloud (Amazon VPC): a new cloud-based deployment and the extension of an on-premises deployment into the AWS cloud. Amazon VPC lets you provision a private, isolated section of the AWS cloud where you can launch AWS resources in a virtual network that you define. With Amazon VPC, you can define a virtual network topology closely resembling a traditional network that you might operate on your own premises. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways.

An Amazon VPC can span multiple Availability Zones (AZs), allowing you to place independent infrastructure in physically separate locations. A multi-AZ deployment provides high availability and fault tolerance. In the scenarios in this guide, we will place Domain Controllers in two Availability Zones, which will provide highly available, low latency access to AD DS services in the AWS cloud.

Amazon VPC Requirements for running Highly Available Active Directory Domain Services

In order to accommodate highly available AD DS in the AWS cloud and adhere to AWS best practices, we will start with a base Amazon VPC configuration that supports the following requirements:

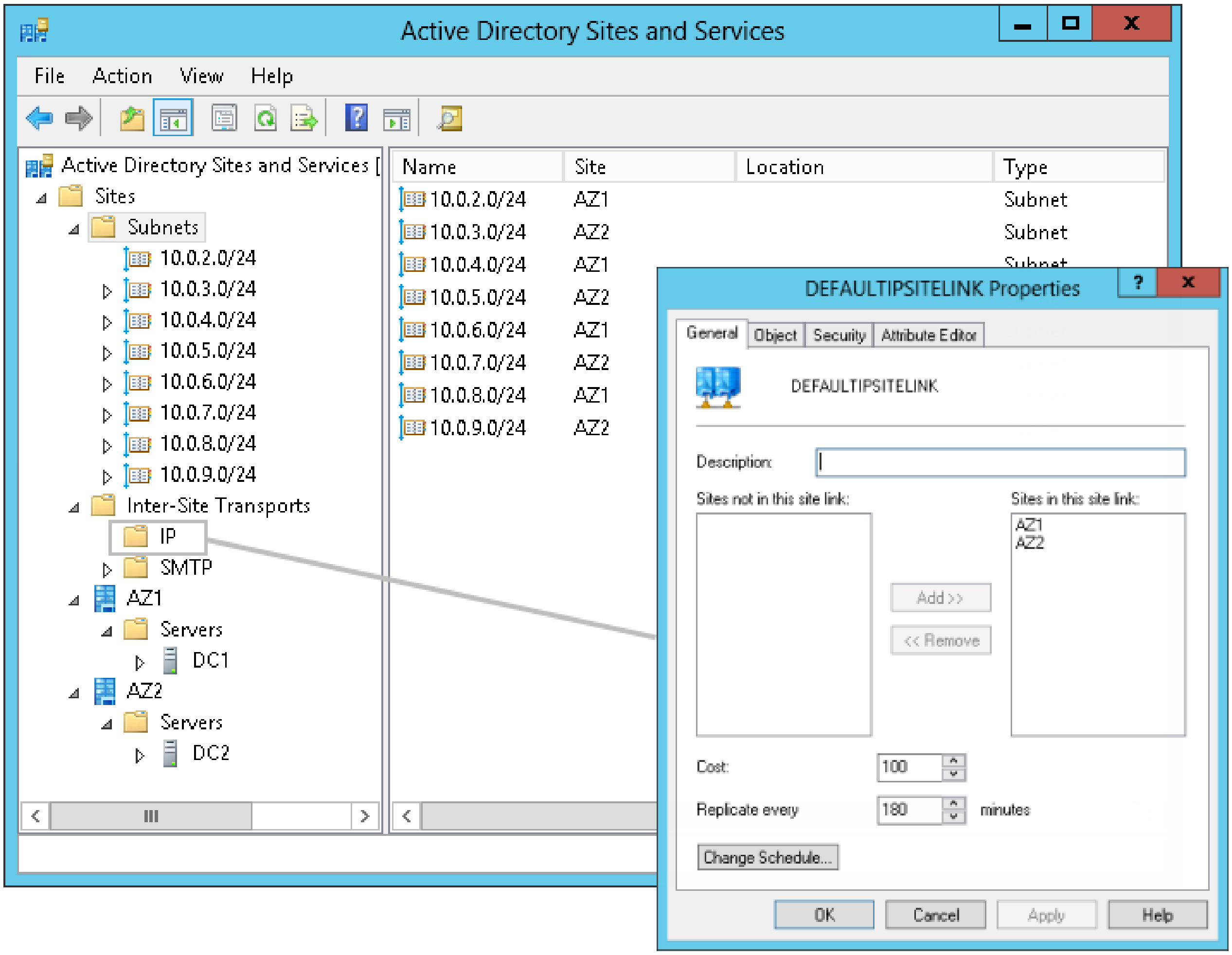
- Domain Controllers should be placed in a minimum of two Availability Zones to provide high availability.
- Instances should be placed into individual tiered groups. For example, in a SharePoint deployment, you should have separate groups for web servers, application servers, database servers, and Domain Controllers.
- Domain Controllers and other non-internet facing servers should be placed in private subnets.
- Instances launched by the deployment templates provided in this guide will require internet access to connect to the AWS CloudFormation endpoint during the bootstrapping process. To support this configuration, public subnets are used to host NAT instances for outbound internet access. Remote Desktop Gateways are also deployed into the public subnets for remote administration. Other components, such as reverse proxy servers can be placed into these public subnets, if needed.

Active Directory Design

Site Topology

Active Directory site topology allows you to logically define your physical and virtual networks. Active Directory replication sends directory changes from one Domain Controller to another, until all Domain Controllers have been updated. Site topology controls Active Directory replication between Domain Controllers in the same site and across site boundaries. Replication traffic between sites is compressed and replication is performed on a schedule based on a site link. Additionally, Domain Controllers use the site topology to provide client affinity, meaning that clients located within a specific site will prefer Domain Controllers in the same site.

Site topology is a crucial design consideration when running AD DS in the AWS cloud. A well designed site topology allows you to define subnets that can be associated with the Availability Zones within your Amazon VPC. These associations help ensure that traffic—such as directory service queries, AD DS replication, and client authentication—uses the most efficient path to a Domain Controller. They also provide you with granular control over replication traffic.



Name	Site	Location	Type
10.0.2.0/24	AZ1		Subnet
10.0.3.0/24	AZ2		Subnet
10.0.4.0/24	AZ1		Subnet
10.0.5.0/24	AZ2		
10.0.6.0/24	AZ1		
10.0.7.0/24	AZ2		
10.0.8.0/24	AZ1		
10.0.9.0/24	AZ2		

Sites in this site link:
AZ1
AZ2

Cost: 100

Replicate every 180 minutes

Figure 2: Active Directory Sites and Services Configuration

Figure 2 shows an example of site and subnet definitions for a typical AD DS architecture running within an Amazon VPC. Active Directory sites (AZ1 and AZ2) have been created in AD Sites and Services. Subnets have been defined and associated with their respective site objects.

By creating Active Directory sites that represent each Availability Zone in the Amazon VPC, subnets associated with those sites can help ensure that domain joined instances will primarily use a Domain Controller closest to them. This is also a key design configuration for maintaining a highly available AD DS deployment.

Highly Available Directory Domain Services

Even in the smallest AD DS deployments, we recommend implementing at least two Domain Controllers in your AWS cloud environment. This design provides fault tolerance and prevents a single Domain Controller failure from impacting the availability of the AD DS. In order to provide higher availability, we recommend that you implement Domain Controllers in at least two Availability Zones.

To further support the high availability of your architecture and mitigate the impact of a possible disaster, we also recommend placing Global Catalog (GC) and Active Directory DNS servers in each Availability Zone. GCs provide a mechanism for forest-wide searches and are required for logon authentication in forests with multiple domains. If you do not have a GC and a DNS server in each Availability Zone, AD DS queries and authentication traffic could cross availability zones. While this is not technically an issue during normal operations, full AD DS service availability could be impacted by a single Availability Zone failure.

To implement these recommendations, we suggest that you make each Domain Controller a Global Catalog and DNS server. This configuration allows AD DS in each Availability Zone to operate independently and helps ensure that AD DS availability is not impacted in the unlikely event of disaster. If an Availability Zone in this architecture becomes an island, cut off from other resources in the region, instances within the Availability Zone still have a local DC that can authenticate users, service directory lookups, and resolve DNS queries.

The requirements of a smaller environment might make a single Availability Zone more appealing. Even though a single Availability Zone AD DS design is not our recommendation, we realize that this may be the chosen architecture. In this case, we recommend that you deploy at least two Domain Controllers in your Availability Zone to provide redundancy.

The AWS CloudFormation template provided in Scenario #1 later in this guide will build out an AD DS Sites and Services configuration for you automatically that will support a highly available AD DS architecture. If you plan to deploy AD DS manually, make sure that you properly map subnets to the correct site to help ensure AD DS traffic uses the best possible path.

For detailed guidance on creating sites, adding Global Catalog servers, and creating and managing site links, see the Microsoft Active Directory Sites and Services documentation.

Read-Only and Writable Domain Controllers

Read-Only Domain Controllers (RODCs) hold a copy of the AD DS database and respond to authentication requests, but they cannot be written to by applications or other servers. RODC's are typically deployed in locations where physical security cannot be guaranteed. For example, in an on-premises scenario, you might deploy an RODC in a remote branch office where the physical server cannot be protected by a secure, locked closet or server room.

Writable Domain Controllers operate in a multi-master model; changes can be made on any writable server in the forest, and those changes are replicated to servers throughout the entire forest. Several key functions and Microsoft enterprise applications require connectivity to a writable Domain Controller.

If you are planning to deploy enterprise application servers into the AWS cloud, an RODC may not be a viable option. For example, an RODC cannot process a password reset for an end user, and Microsoft Exchange Server cannot use an RODC to perform directory look-ups. Make sure you understand the requirements of the application, the dependencies on AD DS, and compatibility before considering RODCs.

Instance Configuration

Active Directory DNS and DHCP inside the Amazon VPC

With an Amazon VPC, Dynamic Host Configuration Protocol (DHCP) services are provided by default for your instances. DHCP scopes do not need to be managed; they are created for the Amazon VPC subnets you define when you deploy your solution. These DHCP services cannot be disabled, so you'll need to use them rather than deploying your own DHCP server.

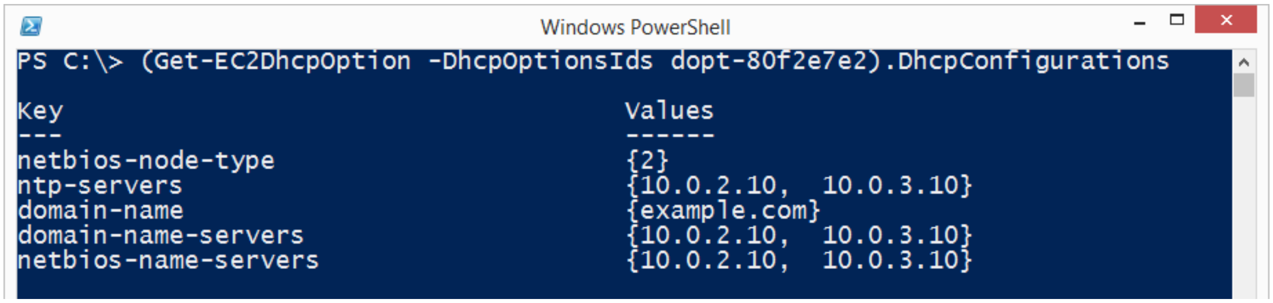
The Amazon VPC also provides an internal DNS server. This DNS provides instances with basic name resolution services for internet access and is crucial for access to AWS service endpoints such as AWS CloudFormation and Amazon Simple Storage Service (Amazon S3) during the bootstrapping process when launched via AWS CloudFormation.

Amazon provided DNS server settings will be assigned to instances launched into the VPC based on a DHCP Option Set. DHCP Option Sets are used within an Amazon VPC to define scope options, such as the domain name, or the name servers that should be handed to your instances via DHCP. Amazon-provided DNS is used only for public DNS resolution.

Since Amazon-provided DNS cannot be used to provide name resolution services for Active Directory, you'll need to ensure that domain joined Windows instances have been configured to use Active Directory DNS.

As an alternative to statically assigning Active Directory DNS server settings on Windows instances, you have the option of specifying them using a custom DHCP Option Set. This will allow you to assign your Active Directory DNS suffix and DNS server IP addresses as the name servers within the Amazon VPC via DHCP.

Figure 3 shows the configuration of a custom DHCP Option Set, where the domain-name-servers field had been set to two IP addresses (the maximum is four) of Domain Controllers running Active Directory integrated DNS in separate Availability Zones.



```
PS C:\> (Get-EC2DhcpOption -DhcpOptionsIds dopt-80f2e7e2).DhcpConfigurations

Key                                     Values
---                                     ------
netbios-node-type                       {2}
ntp-servers                             {10.0.2.10,  10.0.3.10}
domain-name                             {example.com}
domain-name-servers                     {10.0.2.10,  10.0.3.10}
netbios-name-servers                    {10.0.2.10,  10.0.3.10}
```

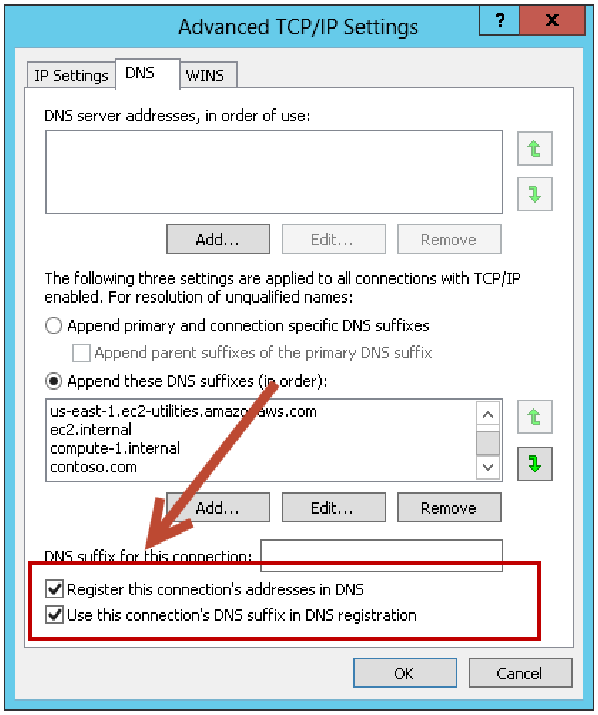
Figure 3: PowerShell Output showing DHCP Option Set Configuration

Note: The IP addresses in the domain-name-servers field are always returned in the same order. If the first DNS server in the list fails, instances should fall back to the second IP and continue to resolve host names successfully. However, during normal operations, the first DNS server listed will always handle DNS requests. If you need to ensure that the distribution of DNS queries is done evenly across multiple servers, you should consider statically configuring DNS server settings on your instances.

For details on creating a custom DHCP Option Set and associating it with your Amazon VPC, see Working with DHCP Options Sets in the Amazon VPC User Guide.

DNS Settings on Windows Server Instances

To make sure that domain joined Windows instances will automatically register Host (A) and Reverse Lookup (PTR) records with Active Directory integrated DNS, set the properties of the network connection as shown in Figure 4.



DNS server addresses, in order of use:

The following three settings are applied to all connections with TCP/IP enabled. For resolution of unqualified names:

○ Append primary and connection specific DNS suffixes

□ Append parent suffixes of the primary DNS suffix

● Append these DNS suffixes (in order):

us-east-1.ec2-utilities.amazonaws.com
ec2.internal
compute-1.internal
contoso.com

DNS suffix for this connection:

☑ Register this connection's addresses in DNS

☑ Use this connection's DNS suffix in DNS registration

Figure 4: Advanced TCP/IP Settings on a domain-joined Windows instance

The default configuration for a network connection is set to automatically register the connections address in DNS. In other words, as shown in Figure 4, the "Register this connection's address in DNS" option is selected for you automatically. This takes care of Host (A) record dynamic registration. However, if you do not also select the second option, "Use this connection's DNS suffix in DNS registration," dynamic registration of PTR records will not take place.

If you have a small number of instances in the Amazon VPC, you may choose to configure the network connection manually. For larger fleets, you can push this setting out to all of your Windows instances using Active Directory Group Policy. Step-by-step instructions are available in the Microsoft TechNet article IPv4 and IPv6 Advanced DNS Tab.

Security Group Ingress Traffic

When launched, Amazon EC2 instances must be associated with a Security Group, which acts as a stateful firewall. You have complete control over the network traffic entering or leaving the Security Group, and you can build granular rules that are scoped by protocol, port number, and source/destination IP address or subnet. By default, all traffic egressing a Security Group is permitted. Ingress traffic, on the other hand, must be configured to allow the appropriate traffic to reach your instances.

In the Securing the Microsoft Platform on Amazon Web Services whitepaper, we discuss in detail the different methods for securing your AWS infrastructure. Recommendations include providing isolation between application tiers using Security Groups. We recommend that you tightly control ingress traffic in order to reduce the attack surface of your Amazon EC2 instances.

Domain Controllers and member servers will require several Security Group rules to allow traffic for services such as AD DS replication, user authentication, Windows Time services, and Distributed File System (DFS), among others. You should also consider restricting these rules to specific IP subnets that are used within your Amazon VPC.

We provide an example of how to implement these rules for each application tier later in this guide as part of an automated AWS CloudFormation template. For a detailed list of port mappings used by the AWS CloudFormation templates, see the Appendix of this guide.

Also see Microsoft's documentation for a complete list of ports in the article titled Active Directory and Active Directory Domain Services Port Requirements. Finally, for step by step guidance for implementing rules, see Adding Rules to a Security Group in the Elastic Compute Cloud documentation.

Setting up Secure Administrative Access using Remote Desktop Gateway

As we design our architecture for highly available AD DS, we should also design for highly available and secure remote access. We can do this by deploying a Remote Desktop (RD) Gateway in each Availability Zone. In case of an Availability Zone outage, this architecture allows access to the resources that may have failed over to the other Availability Zone.

After launching Windows Server instances that act as initial RD Gateways, you can create ingress Security Group rules for your public subnets that will permit access to TCP port 3389 from your network. This configuration allows you to use the RD Gateway as a "jump box" for remote administration. To finalize the configuration, you have the option of configuring the Remote Desktop Gateway Service so you can tunnel your Remote Desktop Protocol (RDP) sessions through the RD Gateways inside a secure SSL connection.

For more information about deploying RD Gateways for remote server administration, see the "Remote Server Administration" section in the Securing the Microsoft Platform on Amazon Web Services whitepaper. For detailed configuration steps, see the Deploying Remote Desktop Gateway Step-by-Step Guide.

Sample Deployment Scenario #1: Deploy Active Directory Domain Services in the AWS Cloud

Now that we've covered the key considerations for running AD DS in the AWS cloud, let's revisit the architecture introduced at the beginning of this guide:

- Domain Controllers are deployed into two Amazon VPC subnets in separate Availability Zones, making AD DS highly available.

- NAT instances are deployed to public subnets providing outbound internet access for instances in private subnets.

- Remote Desktop Gateways are deployed to each public subnet for secure remote access to instances in private subnets.

- The architecture also includes several empty subnets in each availability zone to support a typical multi-tiered server deployment.

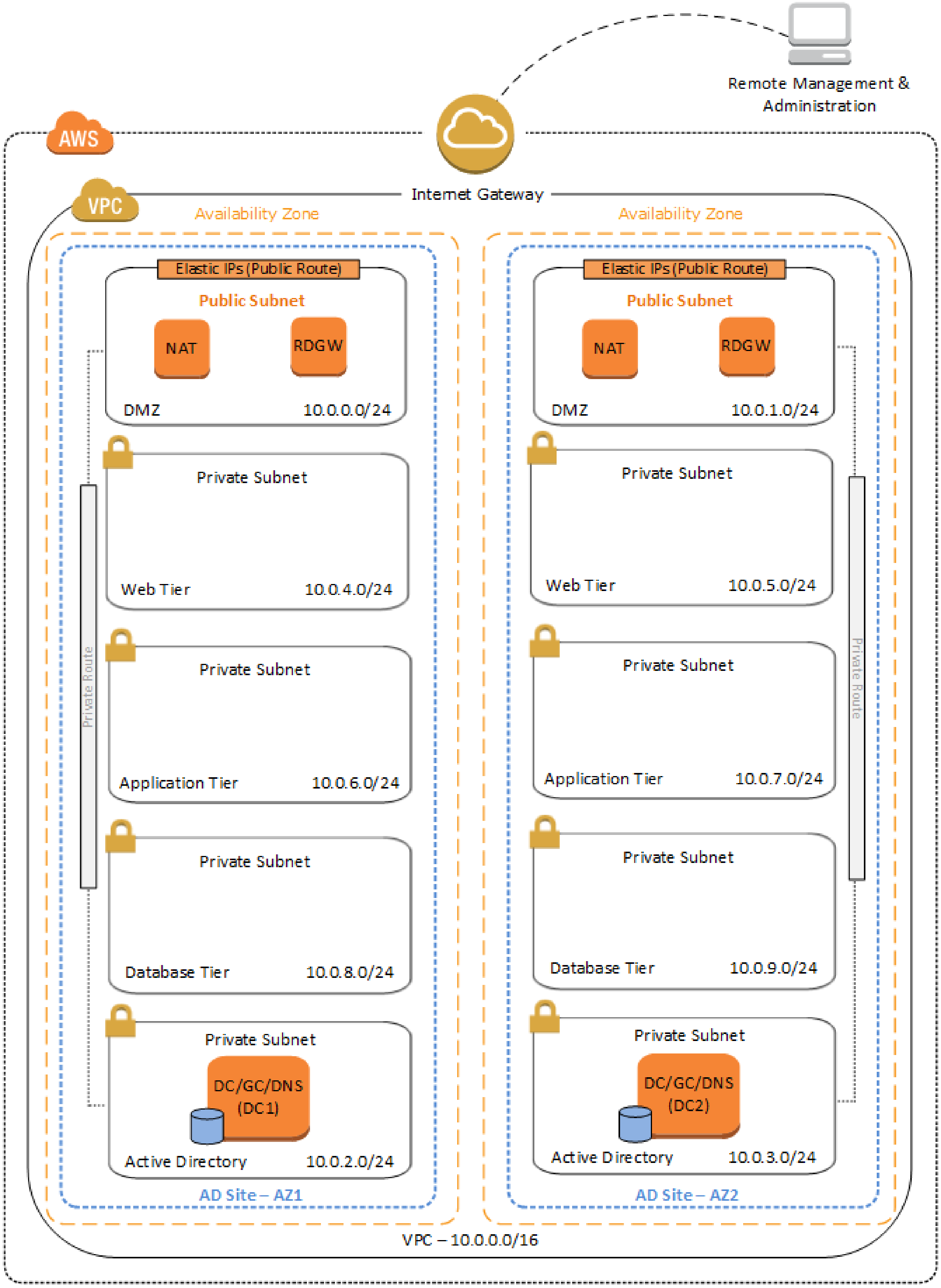


Figure 5: Reference Architecture for Highly Available AD DS in the AWS Cloud

Automated Deployment

We've created an AWS CloudFormation template that deploys the architecture shown in Figure 5. The template creates an Amazon VPC in AWS, creates security groups permitting the appropriate ingress traffic for each server tier, and launches instances to support all of the defined resources. The template is configured with several parameters which allow you to customize the CIDR blocks used for the Amazon VPC and associated subnets, the desired key pair to use for your instances, and other settings. Windows Server 2012 is used for the Remote Desktop Gateway and Domain Controller instances. The AWS CloudFormation template bootstraps each instance, deploying the required components, finalizing the configuration to create a new AD forest, and promoting instances in two Availability Zones to Active Directory Domain Controllers.

After deploying this stack, you can move on to deploying your AD DS dependent servers into the Amazon VPC.

To launch the AWS CloudFormation template into the US West (Oregon) Region, click **LaunchStack**.

Template Customization

Sample Template 1 allows for rich customization of 26 defined parameters at template launch. You can modify those parameters, change the default values, or, if you choose to edit the code of the template itself, create an entirely new set of parameters based on your specific deployment scenario. The Template 1 parameters include the following default values:

Parameter	Default	Description
KeyPairName	*<User Provided>*	Public/private key pairs allow you to connect securely to your instance after it launches.
AD1InstanceType	m3.xlarge	Amazon EC2 instance type for the first Active Directory instance.
AD2InstanceType	m3.xlarge	Amazon EC2 instance type for the second Active Directory instance.
ADServer1NetBIOSName	DC1	NetBIOS name of the first Active Directory server (up to 15 characters).
ADServer2NetBIOSName	DC2	NetBIOS name of the second Active Directory server (up to 15 characters).
ADServer1PrivateIp	10.0.2.10	Fixed private IP for the first Active Directory server located in AZ1.
ADServer2PrivateIp	10.0.3.10	Fixed private IP for the second Active Directory server located in AZ2.
NATInstanceType	m1.small	Amazon EC2 instance type for the NAT instances.
RDGWInstanceType	m3.xlarge	Amazon EC2 instance type for the Remote Desktop Gateway instances.
DomainDNSName	example.com	Fully qualified domain name (FQDN) of the forest root domain; e.g., example.com.
DomainNetBIOSName	example	NetBIOS name of the domain (up to 15 characters) for users of earlier versions of Windows; e.g., EXAMPLE.
RestoreModePassword	Password123	Password for a separate administrator account when the domain controller is in restore mode. Must be at least 8 characters containing letters, numbers, and symbols.
DomainAdminUser	StackAdmin	User name for the account that is added as domain administrator. This is separate from the default "administrator" account.
DomainAdminPassword	Password123	Password for the domain admin user. Must be at least 8 characters containing letters and numbers.
DMZ1CIDR	10.0.0.0/24	CIDR block for the Public subnet located in AZ1.
DMZ2CIDR	10.0.1.0/24	CIDR block for the Public subnet located in AZ2.
PrivSub1CIDR	10.0.2.0/24	CIDR block for the Private Subnet 1 located in AZ1.
PrivSub2CIDR	10.0.3.0/24	CIDR block for the Private Subnet 2 located in AZ2.
PrivSub3CIDR	10.0.4.0/24	CIDR block for the Private Subnet 3 located in AZ1.
PrivSub4CIDR	10.0.5.0/24	CIDR block for the Private Subnet 4 located in AZ2.
PrivSub5CIDR	10.0.6.0/24	CIDR block for the Private Subnet 5 located in AZ1.
PrivSub6CIDR	10.0.7.0/24	CIDR block for the Private Subnet 6 located in AZ2.
PrivSub7CIDR	10.0.8.0/24	CIDR block for the Private Subnet 7 located in AZ1.

PrivSub8CIDR	10.0.9.0/24	CIDR block for the Private Subnet 8 located in AZ2.
VPCCIDR	10.0.0.0/16	CIDR block for the VPC.
UserCount	25	Total number of test user accounts to create in Active Directory.

This architecture provides empty tiers for a typical web application stack. If more tiers are required, you can create additional private subnets with unique CIDR ranges to accommodate other tiers.

Considerations for Extending Existing Active Directory Domain Services into the AWS Cloud

These considerations provide background for automation decisions and explain additional steps you may need or want to take when a brand new deployment of Active Directory is not an option and existing on-premises AD DS resources must be leveraged.

Extend your on-premises network to Amazon VPC

By default, instances that you launch into a virtual private cloud can't communicate with your own network. To extend your existing AD DS into the AWS cloud, you'll need to extend your on-premises network to the Amazon VPC. We'll discuss two ways to do this in the following sections.

IPSec Tunnels over the Internet

The most common scenario for extending your on-premises network into your Amazon VPC is through IPSec Virtual Private Network (VPN) tunnels. Within the Amazon VPC, you can create a virtual private gateway which acts as a VPN concentrator on the Amazon side of the VPN tunnel. A customer gateway is the anchor on your side of that connection. The customer gateway can be a physical device or a software appliance.

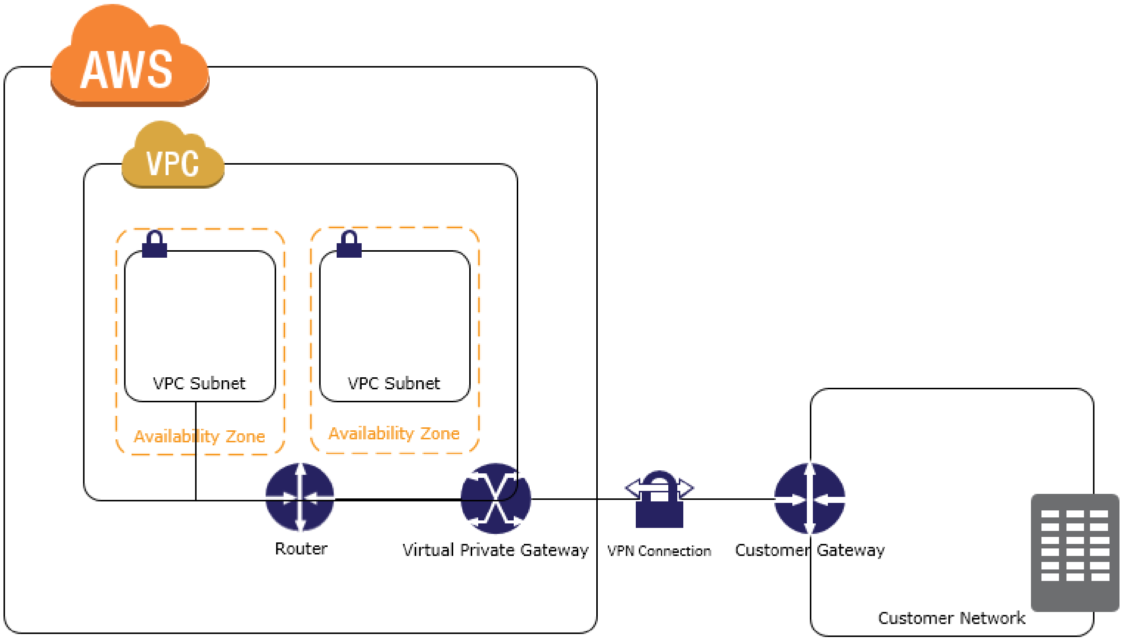


Figure 6: Single VPN Connection from Your On-Premises Network to Your Amazon VPC

Multiple VPN configuration options are available which include the ability to use multiple on-premises customer gateways and configure redundant VPN connections to provide failover. For more details, see the [VPN Configuration Examples]() topic in the *Amazon Virtual Private Cloud Network Administrator Guide*. Details about which hardware or software appliances you can use are also available in [Customer Gateway devices we've tested](), and [Requirements for your customer gateway]().

AWS Direct Connect

AWS Direct Connect links your internal network to an AWS Direct Connect location over a standard 1 gigabit or 10 gigabit Ethernet fiber-optic cable. One end of the cable is connected to your router, the other to an AWS Direct Connect router. With this connection in place, you can create virtual interfaces directly to the AWS cloud (for example, to Amazon Elastic Compute Cloud (Amazon EC2), to Amazon Simple Storage Service (Amazon S3), and to Amazon Virtual Private Cloud (Amazon VPC), bypassing Internet service providers in your network path.

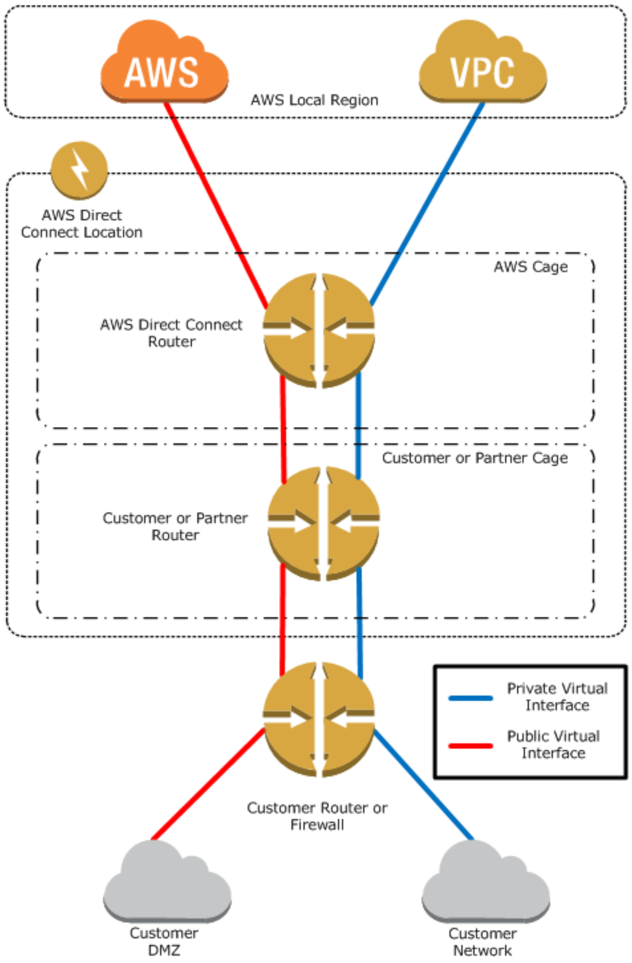

AWS
VPC
AWS Local Region
AWS Direct Connect Location
AWS Cage
AWS Direct Connect Router
Customer or Partner Cage
Customer or Partner Router
Private Virtual Interface
Public Virtual Interface
Customer Router or Firewall
Customer DMZ
Customer Network

Figure 7: How AWS Direct Connect interfaces with your network

When you choose AWS Direct Connect to extend your on-premises network to the cloud, you should consider configuring two dedicated connections for maximum redundancy. There are different configuration choices available when you provision two dedicated connections, including Active/Active (BGP multipath), and Active/Passive (failover).

In a failover configuration, only one connection link handles traffic. If it becomes unavailable, the standby connection link becomes active. We recommend that you configure both connection links as active, as this will help ensure network traffic is load balanced across uses both connections. In an active configuration, if one connection link becomes unavailable, all traffic is routed through the other.

For implementation details, see the Getting Started guide in the AWS Direct Connect documentation.

Deploy Additional Domain Controllers into the AWS Cloud

While it is possible to simply use AWS Direct Connect or a VPN connection to provide access to on-premises resources from the Amazon VPC, we recommend that you also add Domain Controllers into the AWS cloud. Additional Domain Controllers provide a reliable, low-latency network connection for resources in AWS that need access to your AD DS. They can also maintain availability for AD DS in the AWS cloud in the event of an on-premises infrastructure outage.

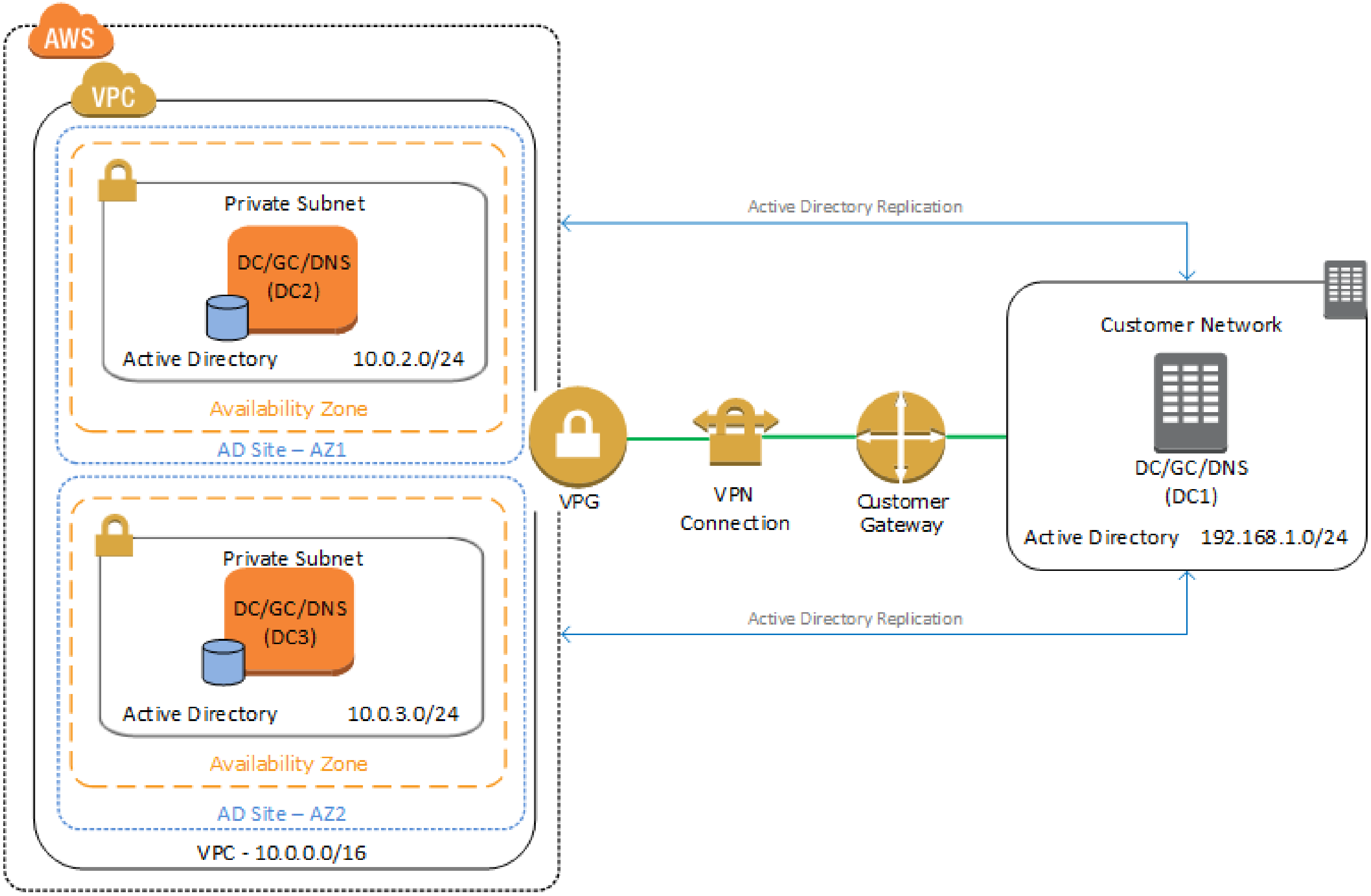


Figure 8: Single AD forest with a Domain Controller on-premises and in an Amazon VPC

In the architecture shown in Figure 8, a single Active Directory forest has been extended from an on-premises deployment into an Amazon VPC using a VPN connection. Within the Amazon VPC, additional Domain Controllers configured as Global Catalog and DNS servers are deployed in the existing Active Directory forest.

In this type of environment, the customer network will already be defined in Active Directory Sites and Services. For example, there will already be a site definition that corresponds to the on-premises network, along with a subnet definition for the 192.168.1.0/24 network. The next step is to configure Active Directory Sites and Services to support the network components located in the Amazon VPC.

Additional Active Directory sites should be created to reference the Availability Zones in AWS. The 10.0.2.0/24 and 10.0.3.0/24 CIDR blocks used by the Amazon VPC subnets should be added to Active Directory Sites and Services. The subnets can then be associated with the AD DS site definition for each AWS Availability Zone. Additional subnets for Web, Application, and Database tiers in the Amazon VPC can be mapped to each AWS site object. Both the on-premises site and the site in the AWS cloud can be mapped to a site link, which can be configured to replicate at custom intervals or during a specific time of day, if needed.

By properly configuring Active Directory Sites and Services, you can help ensure the AD DS queries and authentication requests that originate from the Amazon VPC are serviced by a local Domain Controller in the same AWS Availability Zone. This configuration reduces network latency and minimizes traffic that may otherwise need to travel across the VPN back to the on-premises infrastructure.

Initial DNS Configuration

After you've created an Amazon VPC and established connectivity to your on-premises network using AWS Direct Connect or a VPN connection, your next step is to launch Windows instances to act as Domain Controllers. In order to join the on-premises Active Directory domain and promote your Windows instances to Domain Controllers, you'll need to ensure that DNS resolution is configured appropriately.

As discussed previously, instances launched into the Amazon VPC by default will be assigned Amazon provided DNS server, which will not provide DNS resolution for your on-premises infrastructure. To address this, you can do one of two things:

- Manually assign DNS server settings on the Windows instances. This static DNS setting would initially point to the on-premises Active Directory DNS server. After promoting the instance to a Domain Controller, you could modify the setting to use a cloud based Active Directory DNS server IP address to prevent subsequent DNS queries from traversing the link back to the on-premises environment.

- Initially configure the Amazon VPC DHCP Option Set to assign your on-premises Active Directory DNS server IP address to your instances launched into the Amazon VPC. After the Windows instances have been joined to the domain and promoted to Domain Controllers, you can create a new DHCP option set to assign the IP address of the Active Directory DNS server instances running in AWS.

Sample Deployment Scenario #2: Extend On-premises Active Directory Domain Services to the AWS Cloud

The following scenario provides an example of using an Amazon VPC and a virtual private gateway to enable communication with your own network over an IPsec VPN tunnel. Active Directory is deployed in the customer datacenter, and Windows Servers are deployed into two Amazon VPC subnets. After implementing the VPN connection, you can promote the Windows instances to Domain Controllers in the on-premises Active Directory forest, making AD DS highly available in the AWS cloud.

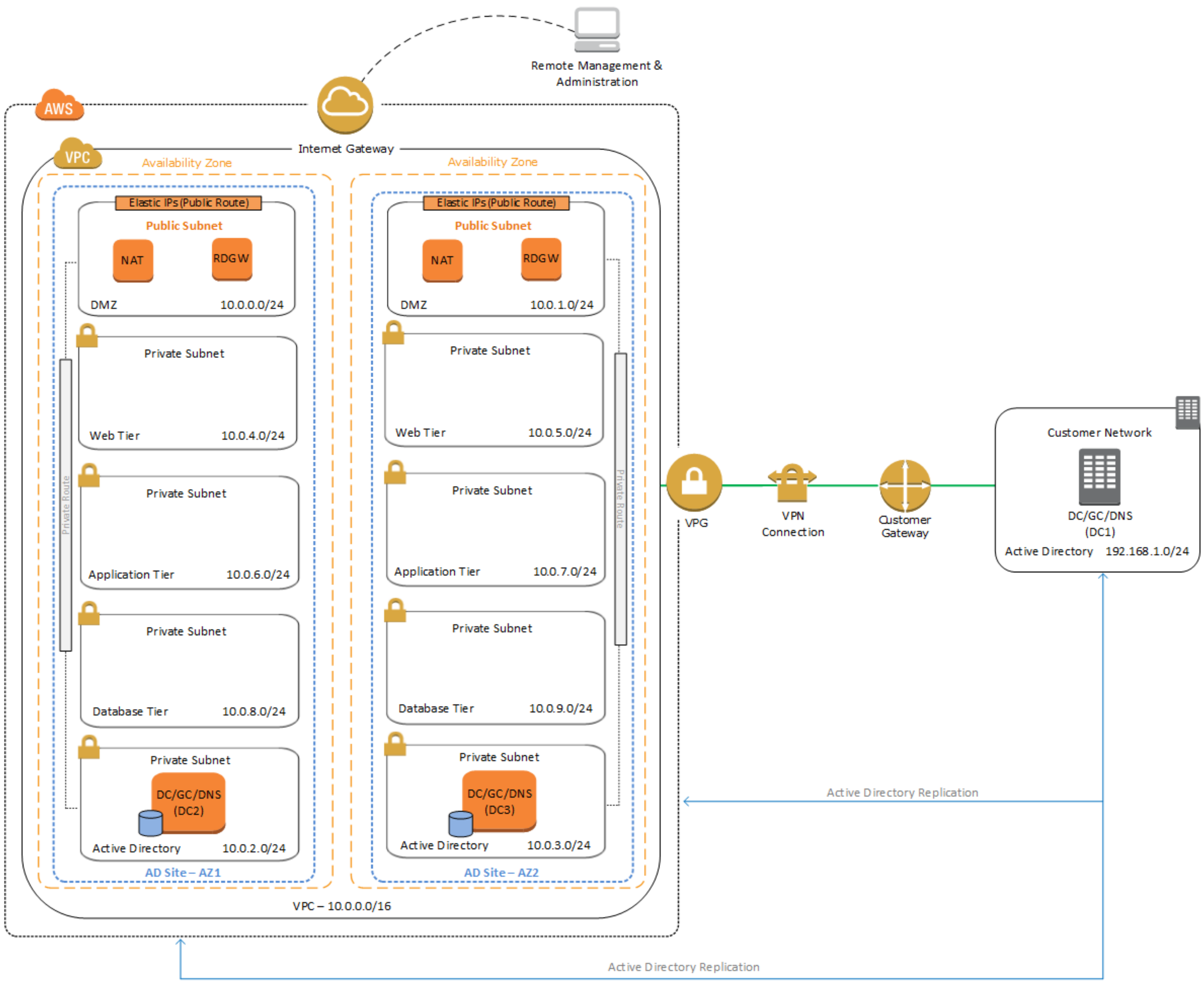


Figure 9: Reference Architecture for an Amazon VPC extended to an on-premises network

After implementing the VPN connection and promoting your servers to Domain Controllers, additional instances can be launched into the empty Amazon VPC subnets in the Web, Application, or Database tiers. These instances will have access to cloud based Domain Controllers for secure, low-latency directory services and DNS. All network traffic, including AD DS communication, authentication requests, and Active Directory replication, is secured either within the private subnets or across the VPN tunnel.

Partially Automated Deployment

We've created an AWS CloudFormation template that deploys the architecture shown in Figure 9, with the exception of the virtual private gateway and VPN connection, which you can create manually. The template creates an Amazon VPC in AWS, creates security groups permitting the appropriate ingress traffic for each server tier, and launches instances to support all of the defined resources. The template is configured with several parameters which allow you to customize the CIDR blocks used for the Amazon VPC and associated subnets, the desired key pair to use for your instances, and other settings.

Note: The default CIDR ranges in this template are provided as an example to get you started and can be modified to meet your specific requirements. Keep in mind that the provided CIDR blocks may overlap with your on-premises networks. If this is the case, you'll need use unique CIDR ranges to help ensure that you can successfully implement a VPN connection.

To launch the AWS CloudFormation template into the US West (Oregon) Region, click **LaunchStack**.

Template Customization

Sample Template 2 allows for rich customization of 20 defined parameters at template launch. You can modify those parameters, change the default values, or, if you choose to edit the code of the template itself, create an entirely new set of parameters based on your specific deployment scenario. The Template 2 parameters include the following default values:

Parameter	Default	Description
KeyPairName	*<User Provided>*	Public/private key pairs allow you to connect securely to your instance after it launches.
AD1InstanceType	m3.xlarge	Amazon EC2 instance type for the first Active Directory instance.
AD2InstanceType	m3.xlarge	Amazon EC2 instance type for the second Active Directory instance.
ADServer1NetBIOSName	DC1	NetBIOS name of the first Active Directory server (up to 15 characters).
ADServer2NetBIOSName	DC2	NetBIOS name of the second Active Directory server (up to 15 characters).
ADServer1PrivateIp	10.0.2.10	Fixed private IP for the first Active Directory server located in AZ1.
ADServer2PrivateIp	10.0.3.10	Fixed private IP for the second Active Directory server located in AZ2.
NATInstanceType	m1.small	Amazon EC2 instance type for the NAT instances.
RDGWInstanceType	m3.xlarge	Amazon EC2 instance type for the Remote Desktop Gateway instances.
DMZ1CIDR	10.0.0.0/24	CIDR block for the Public subnet located in AZ1.
DMZ2CIDR	10.0.1.0/24	CIDR block for the Public subnet located in AZ2.
PrivSub1CIDR	10.0.2.0/24	CIDR block for the Private Subnet 1 located in AZ1.
PrivSub2CIDR	10.0.3.0/24	CIDR block for the Private Subnet 2 located in AZ2.
PrivSub3CIDR	10.0.4.0/24	CIDR block for the Private Subnet 3 located in AZ1.
PrivSub4CIDR	10.0.5.0/24	CIDR block for the Private Subnet 4 located in AZ2.
PrivSub5CIDR	10.0.6.0/24	CIDR block for the Private Subnet 5 located in AZ1.
PrivSub6CIDR	10.0.7.0/24	CIDR block for the Private Subnet 6 located in AZ2.
PrivSub7CIDR	10.0.8.0/24	CIDR block for the Private Subnet 7 located in AZ1.
PrivSub8CIDR	10.0.9.0/24	CIDR block for the Private Subnet 8 located in AZ2.
VPCCIDR	10.0.0.0/16	CIDR block for the VPC.

Post-Configuration Tasks

After the stack has been created successfully, you'll need to perform the following tasks manually:

1. Connect your on-premises network to the Amazon VPC using AWS Direct Connect or a VPN connection.

2. Properly configure your on-premises Active Directory Sites and Services to include sites and subnets that represent the availability zones within your Amazon VPC.

3. Promote the Windows Server instances in the Private Subnet 1 and Private Subnet 2 to Domain Controllers in your Active Directory Domain.

4. Ensure that instances can resolve names via AD DNS using one of the following methods:

    - Statically assign AD DNS servers on Windows Instances.

    - Set the domain-name-servers field in a new DHCP Option Set in your Amazon VPC to include your AWS based Domain Controllers hosting Active Directory DNS.

Further Reading

- Microsoft on AWS:
    - http://aws.amazon.com/microsoft/
- Amazon EC2 Windows Guide:
    - http://docs.amazonwebservices.com/AWSEC2/latest/WindowsGuide/Welcome.html?r=7870
- Secure Microsoft Applications on AWS
    - http://media.amazonwebservices.com/AWS_Microsoft_Platform_Security.pdf
- Getting Started with AWS Direct Connect
    - http://docs.aws.amazon.com/directconnect/latest/UserGuide/getstarted.html
- Scenarios for Amazon VPC
    - http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Scenarios.html
- Amazon Virtual Private Cloud Network Administrator Guide
    - http://docs.aws.amazon.com/AmazonVPC/latest/NetworkAdminGuide/Welcome.html

Appendix

Amazon EC2 Security Group Configuration

AWS provides a set of building blocks, including Amazon EC2 and Amazon VPC that you can use to provision infrastructure for your applications. In this model, some security capabilities such as physical security are the responsibility of AWS and are highlighted in the AWS security whitepaper. Other capabilities, such as controlling access to applications, are the responsibility of the application developer and the tools provided in the Microsoft platform.

If you have followed the automated deployment options in this guide, the necessary security groups are configured for you by the provided AWS CloudFormation Templates and are listed here for your reference:

Subsystem Port Mappings

Subsystem	Associated With	Inbound Interface	Port(s)
DomainControllerSG1	DC1	DomainMemberSG	TCP5985, UDP123, TCP135, UDP138, TCP445, UDP445, TCP464, UDP464, TCP49152-65535, UDP49152-65535, TCP389, UDP389, TCP636, TCP3268, TCP3269, TCP54, UDP53, TCP88, UDP67, UDP2535
		PrivSubnet2CIDR (subnet where the second DC is deployed into)	UDP123, TCP135, UDP138, TCP445, UDP445, TCP464, UDP464, TCP49152-65535, UDP49152-65535, TCP389, UDP389, TCP636, TCP3268, TCP3269, TCP54, UDP53, TCP88, UDP67, UDP2535, UDP5355, UDP137, TCP139, TCP5722
		DMZ1CIDR (subnet where the Remote Desktop Gateway is deployed in AZ1)	TCP3389, (ICMP -1)
		DMZ2CIDR (subnet where the Remote Desktop Gateway is deployed in AZ2)	TCP3389, (ICMP -1)
DomainControllerSG2	DC2	DomainMemberSG	TCP5985, UDP123, TCP135, UDP138, TCP445, UDP445, TCP464, UDP464, TCP49152-65535, UDP49152-65535, TCP389, UDP389, TCP636, TCP3268, TCP3269, TCP54, UDP53, TCP88, UDP67, UDP2535
		PrivSubnet1CIDR (Security Group where the first DC is deployed into)	UDP123, TCP135, UDP138, TCP445, UDP445, TCP464, UDP464, TCP49152-65535, UDP49152-65535, TCP389, UDP389, TCP636, TCP3268, TCP3269, TCP54, UDP53, TCP88, UDP67, UDP2535, UDP5355, UDP137, TCP139, TCP5722
		DMZ1CIDR (subnet where the Remote Desktop Gateway is deployed in AZ2)	TCP3389, (ICMP -1)
		DMZ2CIDR (subnet where the Remote Desktop Gateway is deployed in AZ2)	TCP3389, (ICMP -1)
DomainMemberSG	RDGW1, RDGW2	PrivSubnet1CIDR (subnet where Primary DC is deployed into)	TCP53, UDP53, TCP49152-65535, UDP49152-65535
		PrivSubnet2CIDR (subnet where Secondary DC is deployed into)	TCP53, UDP53, TCP49152-65535, UDP49152-65535

		PrivSubnet1CIDR (subnet where the Remote Desktop Gateway is deployed in AZ1)	TCP3389
		PublicSubnet2CIDR (subnet where the Remote Desktop Gateway is deployed in AZ2)	TCP3389
NATSecurityGroup	NAT1, NAT2	0.0.0.0/0	TCP22
		PrivSubnet1CIDR, PrivSubnet2CIDR, PrivSubnet3CIDR, PrivSubnet4CIDR	ALL1-65535
RDGWSecurityGroup	RDGW1, RDGW2	0.0.0.0/0 *	TCP3389

* **NOTE:** It is important that **RDP never be opened up to the entire Internet**—not even for testing purposes or temporarily. For more information, see Amazon Security Bulletin. Always restrict ports and source traffic to the minimum necessary to support the functionality of the application. For a further discussion about securing Remote Desktop Gateway, see the Securing the Microsoft Platform on Amazon Web Services whitepaper.