AWS Application Discovery Service

User Guide

AWS Application Discovery Service: User Guide

Table of Contents

What Is AWS Application Discovery Service? ... 1
- More About VMware Discovery ... 1
- Compare Connectors and Agents ... 2
- Assumptions ... 2

Setting Up ... 4
- Step 1: Sign Up for AWS ... 4
- Step 2: Create IAM Users ... 4
  - Creating an IAM Administrative User ... 5
  - Creating an IAM Non-Administrative User ... 5

Discovery Agent ... 6
- Data Collected by the Discovery Agent ... 6
- Installation Prerequisites ... 8
- Agent Installation on Linux ... 9
  - Requirements on Older Linux Platforms ... 11
  - Manage the Discovery Agent Process on Linux ... 12
  - Uninstall an agent ... 13
  - Agent Troubleshooting on Linux ... 13
- Agent Installation on Windows ... 14
  - Package Signing and Automatic Upgrades ... 16
  - Manage the Discovery Agent Process on Windows ... 17
  - Troubleshooting on Windows ... 18
- Agent Data Collection ... 18

Discovery Connector ... 20
- Data Collected by Discovery Connector ... 20
- Download the Discovery Connector ... 23
- Deploy the Discovery Connector ... 23
- Configure the Discovery Connector ... 24
  - Configure a static IP address for the connector ... 25
  - Control Data Collection ... 26
  - Disabling auto-upgrades on AWS Discovery Connector ... 27
- Connector Data Collection ... 27
- Troubleshooting the Discovery Connector ... 28
  - Fixing Discovery Connector cannot reach AWS during setup ... 28
  - Fixing unhealthy connectors ... 29
  - Standalone ESX host support ... 30
  - Getting additional support for connector issues ... 30

Import ... 31
- Supported Import File Fields ... 31
- Setting Up Your Import Permissions ... 34
- Uploading Your Import File to Amazon S3 ... 36
- Importing Data ... 37
- Tracking Your Migration Hub Import Requests ... 38

View, Export & Explore Data ... 40
- View Collected Data ... 40
  - Matching Logic ... 40
- Export Collected Data ... 41
- Data Exploration in Athena ... 42
  - Enabling Data Exploration in Amazon Athena ... 43
  - Working with Data Exploration in Amazon Athena ... 44

Console Walkthroughs ... 51
- Main Dashboard ... 51
  - Main Dashboard ... 51
- Data Collection Tools ... 51
  - Starting and Stopping Data Collectors ... 52

- Viewing and Sorting Data Collectors ... 52
- View, Export & Explore Data ... 53
- Viewing and Sorting Servers ... 53
- Tagging Servers ... 54
- Exporting Server Data ... 54
- Data Exploration in Athena ... 55
- Applications ... 55
- Querying Discovered Configuration Items ... 57
- Using the `DescribeConfigurations`Action ... 57
- Using the `ListConfigurations`Action ... 59
- Security ... 69
- Identity and Access Management ... 69
- Audience ... 70
- Authenticating With Identities ... 70
- Managing Access Using Policies ... 72
- How AWS Application Discovery Service Works with IAM ... 73
- IAM Managed Policies ... 75
- Identity-Based Policy Examples ... 77
- Understanding and Using Service-Linked Roles ... 82
- IAM Troubleshooting ... 87
- Logging and monitoring in AWS Application Discovery Service ... 88
- Logging Application Discovery Service API Calls with AWS CloudTrail ... 88
- Quotas ... 91
- Troubleshooting ... 92
- Stop data collection by Data Exploration ... 92
- Remove data collected by Data Exploration ... 93
- Fix Common issues with Data Exploration in Amazon Athena ... 93
- Data Exploration in Amazon Athena Fails to Initiate Because Service-Linked Roles and Required AWS Resources Can't be Created ... 94
- New Agent Data Doesn't show Up in Amazon Athena ... 94
- You have Insufficient Permissions to Access Amazon S3, Amazon Kinesis Data Firehose, or AWS Glue ... 95
- Troubleshooting Failed Import Records ... 95
- Document History ... 98
- AWS glossary ... 100

What Is AWS Application Discovery Service?

AWS Application Discovery Service helps you plan your migration to the AWS cloud by collecting usage and configuration data about your on-premises servers. Application Discovery Service is integrated with AWS Migration Hub, which simplifies your migration tracking as it aggregates your migration status information into a single console. You can view the discovered servers, group them into applications, and then track the migration status of each application from the Migration Hub console in your home region.

All discovered data is stored in your AWS Migration Hub home region. Therefore, you must set your home region in the Migration Hub console or with CLI commands before performing any discovery and migration activities. Your data can be exported for analysis in Microsoft Excel or AWS analysis tools such as Amazon Athena and Amazon QuickSight.

Using Application Discovery Service APIs, you can export the system performance and utilization data for your discovered servers. Input this data into your cost model to compute the cost of running those servers in AWS. Additionally, you can export data about the network connections that exist between servers. This information helps you determine the network dependencies between servers and group them into applications for migration planning.

> **Note**
> Your home region must be set in AWS Migration Hub before you begin the process of discovery, because your data will be stored in your home region. For more information about working with a home region, see Home regions.

Application Discovery Service offers two ways of performing discovery and collecting data about your on-premises servers:

- **Agentless discovery** can be performed by deploying the AWS Agentless Discovery Connector (OVA file) through your VMware vCenter. After the Discovery Connector is configured, it identifies virtual machines (VMs) and hosts associated with vCenter. The Discovery Connector collects the following static configuration data: Server hostnames, IP addresses, MAC addresses, disk resource allocations. Additionally, it collects the utilization data for each VM and computes average and peak utilization for metrics such as CPU, RAM, and Disk I/O.
- **Agent-based discovery** can be performed by deploying the AWS Application Discovery Agent on each of your VMs and physical servers. The agent installer is available for Windows and Linux operating systems. It collects static configuration data, detailed time-series system-performance information, inbound and outbound network connections, and processes that are running.

Application Discovery Service integrates with application discovery solutions from AWS Partner Network (APN) partners. These third-party solutions can help you import details about your on-premises environment directly into Migration Hub, without using any discovery connector or discovery agent. Third-party application discovery tools can query AWS Application Discovery Service, and they can write to the Application Discovery Service database using the public API. In this way, you can import data into Migration Hub and view it, so that you can associate applications with servers and track migrations.

More About VMware Discovery

If you have virtual machines (VMs) that are running in the VMware vCenter environment, you can use the Discovery Connector to collect system information without having to install an agent on each VM.

Instead, you load this on-premises appliance into vCenter and allow it to discover all of its hosts and VMs.

The Discovery Connector captures system performance information and resource utilization for each VM running in the vCenter, regardless of what operating system is in use. However, it cannot “look inside” each of the VMs, and as such, cannot figure out what processes are running on each VM nor what network connections exist. Therefore, if you need this level of detail and want to take a closer look at some of your existing VMs in order to assist in planning your migration, you can install the Discovery Agent on an as-needed basis.

Also, for VMs hosted on VMware, you can use both the Discovery Connector and Discovery Agent to perform discovery simultaneously. For details regarding the exact types of data each discovery tool will collect, see Data Collected by the Discovery Connector (p. 20) and Data Collected by the Discovery Agent (p. 6).

Compare Connectors and Agents

The following table provides a quick comparison of two primary Application Discovery Service tools:

	Discovery Connector	Discovery Agent
Supported server types		
VMware virtual machine	yes	yes
Physical server	no	yes
Deployment		
Per server	no	yes
Per vCenter	yes	no
Collected data		
Static configuration data	yes	yes
VM utilization metrics	yes	no
Time series performance information	no	yes (Export only)
Network inbound/outbound connections	no	yes (Export only)
Running processes	no	yes (Export only)
Supported OS	Any OS running in **VMware vCenter** (V5.5, V6, & V6.5)	For the list of supported Linux and Windows operating systems, see Installation Prerequisites for Discovery Agent (p. 8).

Assumptions

To use Application Discovery Service, the following is assumed:

- You have signed up for AWS. For more information, see Setting Up AWS Application Discovery Service (p. 4).
- You have selected a Migration Hub home region. For more information, see the documentation regarding home regions.

Here's what to expect:

- The Migration Hub home region is the only region where Application Discovery Service stores your discovery and planning data.
- Discovery agents, connectors, and imports can be used in your selected Migration Hub home region only.
- For a list of AWS Regions where you can use Application Discovery Service, see the Amazon Web Services General Reference.

Setting Up AWS Application Discovery Service

Before you use AWS Application Discovery Service for the first time, complete the following tasks:

Step 1: Sign Up for AWS (p. 4)
Step 2: Create IAM Users (p. 4)

Step 1: Sign Up for AWS

In this section, you sign up for an AWS account. If you already have an AWS account, skip this step.

When you sign up for Amazon Web Services (AWS), your AWS account is automatically signed up for all AWS services, including AWS Application Discovery Service. You are charged only for the services that you use.

To create 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.

Note your AWS account number, because you'll need it for the next task.

Step 2: Create IAM Users

When you create an AWS account, you get a single sign-in identity that has complete access to all of the AWS services and resources in the account. This identity is called the AWS account *root user*. Signing in to the AWS Management Console using the email address and password that you used to create the account gives you complete access to all of the AWS resources in your account.

We strongly recommend that you *not* use the root user for everyday tasks, even the administrative ones. Instead, follow the security best practice Create Individual IAM Users and create an AWS Identity and Access Management (IAM) administrator user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

In addition to creating an administrative user you'll also need to create non-administrative IAM users. The following topics explain how to create both types of IAM users.

Topics

- Creating an IAM Administrative User (p. 5)
- Creating an IAM Non-Administrative User (p. 5)

Creating an IAM Administrative User

By default, an administrator account inherits all the policies required for accessing Application Discovery Service.

To create an administrator user

- Create an administrator user in your AWS account. For instructions, see Creating Your First IAM User and Administrators Group in the *IAM User Guide*.

Creating an IAM Non-Administrative User

When creating non-administrative IAM users, follow the security best practice Grant Least Privilege, granting users minimum permissions.

Use IAM managed policies to define the level of access to Application Discovery Service by non-administrative IAM users. For information about Application Discovery Service managed policies, see AWS Managed (Predefined) Policies for Application Discovery Service (p. 75).

To create a non-administrator user

- Create an administrator user in your AWS account. For instructions, see Creating Your First IAM Delegated User and Group in the *IAM User Guide*.

AWS Application Discovery Agent

The AWS Discovery Agent is AWS software that you install on on-premises servers and VMs targeted for discovery and migration. Agents capture system configuration, system performance, running processes, and details of the network connections between systems. Agents support most Linux and Windows operating systems, and you can deploy them on physical on-premises servers, Amazon EC2 instances, and virtual machines.

> **Note**
> Before you deploy the Discovery Agent, you must choose a Migration Hub home region. You must register your agent in your home region.

The Discovery Agent runs in your local environment and requires root privileges. When you start the Discovery Agent, it connects securely with your home region and registers with Application Discovery Service.

- For example, if `eu-central-1` is your home region, it registers `arsenal-discovery.`*`eu-central-1`*`.amazonaws.com` with Application Discovery Service.
- Or substitute your home region as needed for all other regions except us-west-2.
- If `us-west-2` is your home region, it registers `arsenal.us-west-2.amazonaws.com` with Application Discovery Service.

How it works

After registration, the agent starts collecting data for the host or VM where it resides. The agent pings the Application Discovery Service at 15-minute intervals for configuration information.

The collected data includes system specifications, times series utilization or performance data, network connections, and process data. You can use this information to map your IT assets and their network dependencies. All of these data points can help you determine the cost of running these servers in AWS and also plan for migration.

Data is transmitted securely by the Discovery Agents to Application Discovery Service using Transport Layer Security (TLS) encryption. Agents are configured to upgrade automatically when new versions become available. You can change this configuration setting if desired.

> **Tip**
> Before downloading and beginning Discovery Agent installation, be sure to read through all of the required prerequisites in Installation Prerequisites for Discovery Agent (p. 8)

Topics

- Data Collected by the Discovery Agent (p. 6)
- Installation Prerequisites for Discovery Agent (p. 8)
- Agent Installation on Linux (p. 9)
- Agent Installation on Windows (p. 14)
- Agent Data Collection (p. 18)

Data Collected by the Discovery Agent

AWS Application Discovery Agent is software that you install on on-premises servers and VMs. The Discovery Agent collects system configuration, times series utilization or performance data, process data, and Transmission Control Protocol (TCP) network connections. This section describes the data collected.

Table legend for Discovery Agent collected data:

- The term host refers to either a physical server or a VM.
- Collected data is in measurements of kilobytes (KB) unless stated otherwise.
- Equivalent data in the Migration Hub console is reported in megabytes (MB).
- The polling period is in intervals of approximately 15 minutes.
- Data fields denoted with an asterisk (*) are only available in the `.csv` files produced from the agent's API export function.

Data field	Description
agentAssignedProcessId*	Process ID of processes discovered by the agent
agentId	Unique ID of agent
agentProvidedTimeStamp*	Date and time of agent observation *(mm/dd/yyyy hh:mm:ss am/pm)*
cmdLine*	Process entered at the command line
cpuType	Type of CPU (central processing unit) used in host
destinationIp*	IP address of device to which packet is being sent
destinationPort*	Port number to which the data/request is to be sent
family*	Protocol of routing family
freeRAM (MB)	Free RAM and cached RAM that can be made immediately available to applications, measured in MB
gateway*	Node address of network
hostName	Name of host data was collected on
hypervisor	Type of hypervisor
ipAddress	IP address of the host
ipVersion*	IP version number
isSystem*	Boolean attribute to indicate if a process is owned by the OS
macAddress	MAC address of the host
name*	Name of the host, network, metrics, etc. data is being collected for
netMask*	IP address prefix that a network host belongs to
osName	Operating system name on host
osVersion	Operating system version on host
path	Path of the command sourced from the command line

Data field	Description
sourceIp*	IP address of the device sending the IP packet
sourcePort*	Port number from which the data/request originates from
timestamp*	Date and time of reported attribute logged by agent
totalCpuUsagePct	Percentage of CPU usage on host during polling period
totalDiskBytesReadPerSecond (Kbps)	Total amount of disk free space on host
totalDiskBytesWrittenPerSecond (Kbps)	Total size of disk on host
totalDiskFreeSize (GB)	Free disk space expressed in GB
totalDiskReadOpsPerSecond	Total number of read I/O operations per second
totalDiskSize (GB)	Total capacity of disk expressed in GB
totalDiskWriteOpsPerSecond	Total number of write I/O operations per second
totalNetworkBytesReadPerSecond (Kbps)	Total amount of throughput of bytes read per second
totalNetworkBytesWrittenPerSecond (Kbps)	Total amount of throughput of bytes written per second
totalNumCores	Total number of independent processing units within CPU
totalNumCpus	Total number of central processing units
totalNumDisks	The number of physical hard disks on a host
totalNumLogicalProcessors*	Total number of physical cores times the number of threads that can run on each core
totalNumNetworkCards	Total count of network cards on server
totalRAM (MB)	Total amount of RAM available on host
transportProtocol*	Type of transport protocol used

Installation Prerequisites for Discovery Agent

The following are the prerequisites and the tasks that you must perform before you can successfully install the AWS Application Discovery Agent (Discovery Agent).

- You must set an AWS Migration Hub home region before you begin installing Discovery Agent.
- If you have a *1.x* version of the agent installed, it must be removed before installing the latest version.
- If the host that the agent is being installed on runs Linux, then verify that the host at least supports the Intel i686 CPU architecture (also known as the P6 micro architecture).
- Verify that your operating system (OS) environment is supported:

  Linux

  Amazon Linux 2012.03, 2015.03

  Amazon Linux 2 (9/25/2018 update and later)

  Ubuntu 12.04, 14.04, 16.04, 18.04, 20.04

  Red Hat Enterprise Linux 5.11, 6.10, 7.3, 7.7, 8.1

  CentOS 5.11, 6.9, 7.3

  SUSE 11 SP4, 12 SP5

Windows

  Windows Server 2003 R2 SP2

  Windows Server 2008 R1 SP2, 2008 R2 SP1

  Windows Server 2012 R1, 2012 R2

  Windows Server 2016

  Windows Server 2019

- If outbound connections from your network are restricted, you'll need to update your firewall settings. Agents require access to `arsenal` over TCP port 443. They don't require any inbound ports to be open.

  For example, if your home region is `eu-central-1`, you'd use `https://arsenal-discovery.`*`eu-central-1`*`.amazonaws.com:443`
- Access to Amazon S3 in your home region is required for auto-upgrade to function.
- Create an AWS Identity and Access Management (IAM) user in the console and attach the existing `AWSApplicationDiscoveryAgentAccess` IAM managed policy. This policy allows the user to perform necessary agent actions on your behalf. For more information about managed policies, see AWS Managed (Predefined) Policies for Application Discovery Service (p. 75).
- Check the time skew from your Network Time Protocol (NTP) servers and correct if necessary. Incorrect time synchronization causes the agent registration call to fail.

Note
The Discovery Agent has a 32-bit agent executable, which works on 32-bit and 64-bit operating systems. The number of installation packages needed for deployment is reduced by having a single executable. This executable agent works for Linux and for Windows OS. It is addressed in their respective installation sections that follow.

Agent Installation on Linux

Complete the following procedure on Linux. Be sure that your Migration Hub home region has been set before you begin this procedure.

Note
If you are using a non-current Linux version, see Requirements on Older Linux Platforms (p. 11).

To install AWS Application Discovery Agent in your data center

1. Sign in to your Linux-based server or VM and create a new directory to contain your agent components.
2. Switch to the new directory and download the installation script from either the command line or the console.

   a. To download from the command line, run the following command.

   ```
   curl -o ./aws-discovery-agent.tar.gz https://s3-us-west-2.amazonaws.com/aws-discovery-agent.us-west-2/linux/latest/aws-discovery-agent.tar.gz
   ```

b. To download from the Migration Hub console, do the following:

   i. Open the console and go to the Discovery Tools page.

   ii. In the **Discovery Agent** box, choose **Download agent**, then choose **Linux** in the resultant list box. Your download begins immediately.

3. Verify the cryptographic signature of the installation package with the following three commands:

```
curl -o ./agent.sig https://s3.us-west-2.amazonaws.com/aws-discovery-agent.us-west-2/
linux/latest/aws-discovery-agent.tar.gz.sig
```

```
curl -o ./discovery.gpg https://s3.us-west-2.amazonaws.com/aws-discovery-agent.us-
west-2/linux/latest/discovery.gpg
```

```
gpg --no-default-keyring --keyring ./discovery.gpg --verify agent.sig aws-discovery-
agent.tar.gz
```

   The agent public key (`discovery.gpg`) fingerprint is `7638 F24C 6717 F97C 4F1B 3BC0 5133 255E 4DF4 2DA2`.

4. Extract from the tarball as shown following.

```
tar -xzf aws-discovery-agent.tar.gz
```

5. To install the agent, choose one of the following installation methods.

To...	Do this...
Install Discovery Agent	To install the agent, run the agent install command as shown in the following example. In the example, replace *`your-home-region`* with the name of your home region, *`aws-access-key-id`* with your access key id, and *`aws-secret-access-key`* with your secret access key. `sudo bash install -r your-home-region -k aws-access-key-id -s aws-secret-access-key` By default, agents automatically download and apply updates as they become available. We recommend using this default configuration. However, if you don't want agents to download and apply updates automatically, include the `–u false` parameter when running the agent install command.
(Optional) Install Discovery Agent and configure a non-transparent proxy	To configure a non-transparent proxy, add the following parameters to the agent install command: • **-e** The proxy password. • **-f** The proxy port number.

To...	Do this...
	• **-g** The proxy scheme. • **-i** The proxy username. The following is an example of the agent install command using the non-transparent proxy parameters. `sudo bash install -r your-home-region -k aws-access-key-id -s aws-secret-access-key -d myproxy.mycompany.com -e mypassword -f proxy-port-number -g https -i myusername` If your proxy doesn't require authentication, then leave out the `-e` and `-i` parameters. The example install command uses `https`, if your proxy uses HTTP, specify `http` for the `-g` parameter value.

6. If outbound connections from your network are restricted, you'll need to update your firewall settings. Agents require access to `arsenal` over TCP port 443. They don't require any inbound ports to be open.

   For example, if your home region is `eu-central-1`, you'd use `https://arsenal-discovery.eu-central-1.amazonaws.com:443`

Topics

- Requirements on Older Linux Platforms (p. 11)
- Manage the Discovery Agent Process on Linux (p. 12)
- Uninstall Discovery Agent on Linux (p. 13)
- Agent Troubleshooting on Linux (p. 13)

Requirements on Older Linux Platforms

Some older Linux platforms such as SUSE 10, CentOS 5, and RHEL 5 are either at end of life or only minimally supported. These platforms can suffer from out-of-date cipher suites that prevent the agent update script from downloading installation packages.

Curl

The Application Discovery agent requires `curl` for secure communications with the AWS server. Some old versions of `curl` are not able to communicate securely with a modern web service.

To use the version of `curl` included with the Application Discovery agent for all operations, run the installation script with the `-c true` parameter.

Certificate Authority Bundle

Older Linux systems might have an out-of-date Certificate Authority (CA) bundle, which is critical to secure internet communication.

To use the CA bundle included with the Application Discovery agent for all operations, run the installation script with the `-b true` parameter.

These installation script options can be used together. In the following example command, both of the script parameters are passed to the installation script:

```
sudo bash install -r your-home_region -k aws-access-key-id -s aws-secret-access-key -c true
 -b true
```

Manage the Discovery Agent Process on Linux

You can manage the behavior of the Discovery Agent at the system level using the `systemd`, `Upstart`, or `System V init` tools. The following tabs outline the commands for the supported tasks in each of the respective tools.

systemd

Management Commands for the Application Discovery Agent

Task	Command
Verify that an agent is running	`sudo systemctl status aws-discovery-daemon.service`
Start an agent	`sudo systemctl start aws-discovery-daemon.service`
Stop an agent	`sudo systemctl stop aws-discovery-daemon.service`
Restart an agent	`sudo systemctl restart aws-discovery-daemon.service`

Upstart

Management Commands for the Application Discovery Agent

Task	Command
Verify that an agent is running	`sudo initctl status aws-discovery-daemon`
Start an agent	`sudo initctl start aws-discovery-daemon`
Stop an agent	`sudo initctl stop aws-discovery-daemon`
Restart an agent	`sudo initctl restart aws-discovery-daemon`

System V init

Management Commands for the Application Discovery Agent

Task	Command
Verify that an agent is running	`sudo /etc/init.d/aws-discovery-daemon status`

Task	Command
Start an agent	`sudo /etc/init.d/aws-discovery-daemon start`
Stop an agent	`sudo /etc/init.d/aws-discovery-daemon stop`
Restart an agent	`sudo /etc/init.d/aws-discovery-daemon restart`

Uninstall Discovery Agent on Linux

This section describes how to uninstall Discovery Agent on Linux.

To uninstall an agent if you're using the yum package manager

- Use the following command to uninstall an agent if using yum.

```
rpm -e --nodeps aws-discovery-agent
```

To uninstall an agent if you're using the apt-get package manager

- Use the following command to uninstall an agent if using apt-get.

```
apt-get remove aws-discovery-agent:i386
```

To uninstall an agent if you're using the zypper package manager

- Use the following command to uninstall an agent if using zypper.

```
zypper remove aws-discovery-agent
```

Agent Troubleshooting on Linux

If you encounter problems while installing or using the Discovery Agent on Linux, consult the following guidance about logging and configuration. When helping to troubleshoot potential issues with the agent or its connection to the Application Discovery Service, AWS Support often requests these files.

- **Log files**

  Log files for Discovery Agent are located in the following directory.

  ```
  /var/log/aws/discovery/
  ```

  Log files are named to indicate whether they are generated by the main daemon, the automatic upgrader, or the installer.

- **Configuration files**

Configuration files for Discovery Agent version 2.0.1617.0 or newer are located in the following directory.

```
/etc/opt/aws/discovery/
```

Configuration files for versions of Discovery Agent before 2.0.1617.0 are located in the following directory.

```
/var/opt/aws/discovery/
```

- For instructions on how to remove older versions of the Discovery Agent, see Installation Prerequisites for Discovery Agent (p. 8).

Agent Installation on Windows

Complete the following procedure to install an agent on Windows. Be sure that your Migration Hub home region has been set before you begin this procedure.

To install AWS Application Discovery Agent in your data center

1. Download the Windows agent installer *but do not double-click to run the installer within Windows*.

   Important
   Do not double-click to run the installer within Windows as it will fail to install. *Agent installation only works from the command prompt*. (If you already double-clicked on the installer, you must go to **Add/Remove Programs** and uninstall the agent before continuing on with the remaining installation steps.)
   If the Windows agent installer doesn't detect any version of the Visual C++ x86 runtime on the host, it automatically installs the Visual C++ x86 2015–2019 runtime before installing the agent software.

2. Open a command prompt as an administrator and navigate to the location where you saved the installation package.
3. To install the agent, choose one of the following installation methods.

To...	Do this...
Install Discovery Agent	To install the agent, run the agent install command as shown in the following example. In the example, replace *your-home-region* with the name of your home region, *aws-access-key-id* with your access key ID, and *aws-secret-access-key* with your secret access key. Optionally, you can set the agent installation location by specifying the folder path *C:\install-location* for the INSTALLLOCATION parameter. For example, INSTALLLOCATION="*C:\install-location*". The resulting folder hierarchy will be [INSTALLLOCATION path]\AWS Discovery. By default, the install location is the Program Files folder.

To...	Do this...
	Optionally, you can use `LOGANDCONFIGLOCATION` to override the default directory (ProgramData) for the agent logs folder and configuration file. The resulting folder hierarchy is `[LOGANDCONFIGLOCATION path]\AWS Discovery`. `.\AWSDiscoveryAgentInstaller.exe REGION="your-home-region" KEY_ID="aws-access-key-id" KEY_SECRET="aws-secret-access-key" /quiet` By default, agents automatically download and apply updates as they become available. We recommend using this default configuration. However, if you don't want agents to download and apply updates automatically, include the following parameter when running the agent install command: `AUTO_UPDATE=false` **Warning** Disabling auto-upgrades will prevent the latest security patches from being installed.

To...	Do this...

(Optional) Install Discovery Agent and configure a non-transparent proxy	To configure a non-transparent proxy, add the following public properties to the agent install command:
	• **PROXY_HOST** – The name of the proxy host
	• **PROXY_SCHEME** – The proxy scheme
	• **PROXY_PORT** – The proxy port number
	• **PROXY_USER** – The proxy user name
	• **PROXY_PASSWORD** – The proxy user password
	The following is an example of the agent install command using the non-transparent proxy properties.
	`.\AWSDiscoveryAgentInstaller.exe`
	` REGION="your-home-region"`
	` KEY_ID="aws-access-key-id"`
	` KEY_SECRET="aws-secret-access-key"`
	` PROXY_HOST="myproxy.mycompany.com"`
	` PROXY_SCHEME="https" PROXY_PORT="proxy-port-number" PROXY_USER="myusername"`
	` PROXY_PASSWORD="mypassword" /quiet`
	If your proxy doesn't require authentication, then omit the `PROXY_USER` and `PROXY_PASSWORD` properties. The example install command uses `https`. If your proxy uses HTTP, specify `http` for the `PROXY_SCHEME` value.

4. If outbound connections from your network are restricted, you must update your firewall settings. Agents require access to `arsenal` over TCP port 443. They don't require any inbound ports to be open.

   For example, if your home region is `eu-central-1`, you'd use the following: `https://arsenal-discovery.eu-central-1.amazonaws.com:443`

Package Signing and Automatic Upgrades

For Windows Server 2008 and later, Amazon cryptographically signs the Application Discovery Service agent installation package with an SHA256 certificate. For SHA2-signed autoupdates on Windows Server 2008 SP2, ensure that hosts have a hotfix installed to support SHA2 signature authentication. Microsoft's latest support hotfix helps support SHA2 authentication on Windows Server 2008 SP2.

> **Note**
> The hotfixes for SHA256 support for Windows 2003 are no longer publicly available from Microsoft. If these fixes are not already installed in your Windows 2003 host, manual upgrades are necessary.

To perform upgrades manually

1. Download the Windows Agent Updater.

2. Open command prompt as an administrator.
3. Navigate to the location where the updater was saved.
4. Run the following command.

```
AWSDiscoveryAgentUpdater.exe /Q
```

Manage the Discovery Agent Process on Windows

You can manage the behavior of the Discovery Agent at the system level through the Windows Server Manager Services console. The following table describes how.

Task	Service Name	Service Status/Action
Verify that an agent is running	AWS Discovery Agent AWS Discovery Updater	Started
Start an agent	AWS Discovery Agent AWS Discovery Updater	Choose **Start**
Stop an agent	AWS Discovery Agent AWS Discovery Updater	Choose **Stop**
Restart an agent	AWS Discovery Agent AWS Discovery Updater	Choose **Restart**

To uninstall a discovery agent on Windows

1. Open the Control Panel in Windows.
2. Choose **Programs**.
3. Choose **Programs and Features**.
4. Select **AWS Discovery Agent**.
5. Choose **Uninstall**.

> **Note**
> If you choose to reinstall the agent after uninstalling it, run the following command with the `/repair` and `/norestart` options.

```
.\AWSDiscoveryAgentInstaller.exe REGION="your-home-region" KEY_ID="aws-access-key-id" KEY_SECRET="aws-secret-access-key" /quiet /repair /norestart
```

To uninstall a discovery agent on Windows using the command line

1. Right-click **Start**.
2. Choose **Command Prompt**.
3. Use the following command to uninstall a discovery agent on Windows.

```
wmic product where name='AWS Discovery Agent' call uninstall
```

Agent Troubleshooting on Windows

If you encounter problems while installing or using the AWS Application Discovery Agent on Windows, consult the following guidance about logging and configuration. AWS Supportoften requests these files when helping to troubleshoot potential issues with the agent or its connection to the Application Discovery Service.

- **Installation logging**

  In some cases, the agent install command appears to fail. For example, a failure can appear with the Windows Services Manager showing that the discovery services are not being created. In this case, add **/log install.log** to the command to generate a verbose installation log.
- **Operational logging**

  On Windows Server 2008 and later, agent log files can be found under the following directory.

  ```
  C:\ProgramData\AWS\AWS Discovery\Logs
  ```

  On Windows Server 2003, agent log files can be found under the following directory.

  ```
  C:\Documents and Settings\All Users\Application Data\AWS\AWSDiscovery\Logs
  ```

  Log files are named to indicate whether generated by the main service, automatic upgrades, or the installer.

- **Configuration file**

  On Windows Server 2008 and later, the agent configuration file can be found at the following location.

  ```
  C:\ProgramData\AWS\AWS Discovery\config
  ```

  On Windows Server 2003, the agent configuration file can be found at the following location.

  ```
  C:\Documents and Settings\All Users\Application Data\AWS\AWS Discovery\config
  ```
- For instructions on how to remove earlier versions of the Discovery Agent, see Installation Prerequisites for Discovery Agent (p. 8).

Agent Data Collection

After you have deployed and configured the Discovery Agent, if data collections stops you can restart it. You can start or stop data collection through the console or by making API calls through the AWS CLI. Both of these methods are described in the following procedures.

Using the Migration Hub Console

The following procedure shows how to start or stop the Discovery Agent data collection process, on the **Data Collectors** page of the Migration Hub console.

To start or stop data collection

1. In the navigation pane, choose **Data Collectors**.
2. Choose the **Agents** tab.
3. Select the check box of the agent you want to start or stop.

   Tip
   If you installed multiple agents but only want to start or stop data collection on certain hosts, the **Hostname** column in the agent's row identifies the host the agent is installed on.
4. Choose **Start data collection** or **Stop data collection**.

Using the AWS CLI

To start or stop the Discovery Agent data collection process from the AWS CLI, you must first install the AWS CLI in your environment, and then you must set the CLI to use your selected Migration Hub home region.

To install the AWS CLI and start or stop data collection

1. If you have not already done so, install the AWS CLI appropriate to your OS type (Windows or Mac/Linux). See the AWS Command Line Interface User Guide for instructions.
2. Open the Command prompt (Windows) or Terminal (MAC/Linux).

   a. Type `aws configure` and press Enter.
   b. Enter your AWS Access Key ID and AWS Secret Access Key.
   c. Enter your home region for the Default Region Name, for example *`us-west-2`*. (We are assuming that `us-west-2` is your home region in this example.)
   d. Enter `text` for Default Output Format.
3. To find the ID of the agent you want to stop or start data collection for, type the following command:

```
aws discovery describe-agents
```

4. To start data collection by the agent, type the following command:

```
aws discovery start-data-collection-by-agent-ids --agent-ids <agent ID>
```

   To stop data collection by the agent, type the following command:

```
aws discovery stop-data-collection-by-agent-ids --agent-ids <agent ID>
```

AWS Agentless Discovery Connector

Agentless discovery uses the AWS Discovery Connector. The AWS Discovery Connector is a VMware appliance that can collect information only about VMware virtual machines (VMs). You install the Discovery Connector as a VM in your VMware vCenter Server environment using an Open Virtualization Archive (OVA) file. Because the Discovery Connector relies on VMware metadata to gather server information regardless of operating system, it minimizes the time required for initial on-premises infrastructure assessment.

Before you deploy the Discovery Connector, you must choose a Migration Hub home Region. You must register your connector in your home region. After you deploy and configure the Discovery Connector, it registers with the Application Discovery Service endpoint, and pings the service at regular intervals, approximately every 60 minutes, for configuration information.

- For example, if `eu-central-1` is your home region, it registers `arsenal-discovery.`*`eu-central-1`*`.amazonaws.com` with Application Discovery Service.
- Or substitute your home region as needed for all other regions except us-west-2.
- If `us-west-2` is your home region, it registers `arsenal.us-west-2.amazonaws.com` with Application Discovery Service.

How it works

After registration, the connector connects to VMware vCenter Server, where it collects data about all the VMs and hosts managed by this specific vCenter. The collected data is sent to the Application Discovery Service using Secure Sockets Layer (SSL) encryption. The connector is configured to upgrade automatically when new versions of the connector become available. You can change this configuration setting at any time.

Topics

- Data Collected by the Discovery Connector (p. 20)
- Download the Discovery Connector (p. 23)
- Deploy the Discovery Connector (p. 23)
- Configure the AWS Discovery Connector (p. 24)
- Discovery Connector Data Collection (p. 27)
- Troubleshooting the Discovery Connector (p. 28)

Data Collected by the Discovery Connector

The Discovery Connector collects information about your VMware vCenter Server hosts and VMs. However, you can capture this data only if VMware vCenter Server tools are installed. To make sure the AWS account you are using has the required permission for this task, see AWS Managed (Predefined) Policies for Application Discovery Service (p. 75).

Following, you can find an inventory of the information collected by the Discovery Connector.

Table legend for Discovery Connector collected data:

- Collected data is in measurements of kilobytes (KB) unless stated otherwise.

- Equivalent data in the Migration Hub console is reported in megabytes (MB).
- Data fields denoted with an asterisk (*) are only available in the .csv files produced from the connector's API export function.
- The polling period is in intervals of approximately 60 minutes.
- Data fields denoted with a double asterisk (**) currently return a *null* value.

Data field	**Description**
applicationConfigurationId*	ID of the migration application the VM is grouped under
avgCpuUsagePct	Average percentage of CPU usage over polling period
avgDiskBytesReadPerSecond	Average number of bytes read from disk over polling period
avgDiskBytesWrittenPerSecond	Average number of bytes written to disk over polling period
avgDiskReadOpsPerSecond**	Average number of read I/O operations per second null
avgDiskWriteOpsPerSecond**	Average number of write I/O operations per second
avgFreeRAM	Average free RAM expressed in MB
avgNetworkBytesReadPerSecond	Average amount of throughput of bytes read per second
avgNetworkBytesWrittenPerSecond	Average amount of throughput of bytes written per second
configId	Application Discovery Service assigned ID to the discovered VM
configType	Type of resource discovered
connectorId	ID of the Discovery Connector virtual appliance
cpuType	vCPU for a VM, actual model for a host
datacenterId	ID of the vCenter
hostId*	ID of the VM host
hostName	Name of host running the virtualization software
hypervisor	Type of hypervisor
id	ID of server
lastModifiedTimeStamp*	Latest date and time of data collection before data export
macAddress	MAC address of the VM
manufacturer	Maker of the virtualization software

Data field	Description
maxCpuUsagePct	Max. percentage of CPU usage during polling period
maxDiskBytesReadPerSecond	Max. number of bytes read from disk over polling period
maxDiskBytesWrittenPerSecond	Max. number of bytes written to disk over polling period
maxDiskReadOpsPerSecond**	Max. number of read I/O operations per second
maxDiskWriteOpsPerSecond**	Max. number of write I/O operations per second
maxNetworkBytesReadPerSecond	Max. amount of throughput of bytes read per second
maxNetworkBytesWrittenPerSecond	Max. amount of throughput of bytes written per second
memoryReservation*	Limit to avoid overcommitment of memory on VM
moRefId	Unique vCenter Managed Object Reference ID
name*	Name of VM or network (user specified)
numCores	Number of independent processing units within CPU
numCpus	Number of central processing units on VM
numDisks**	Number of disks on VM
numNetworkCards**	Number of network cards on VM
osName	Operating system name on VM
osVersion	Operating system version on VM
portGroupId*	ID of group of member ports of VLAN
portGroupName*	Name of group of member ports of VLAN
powerState*	Status of power
serverId	Application Discovery Service assigned ID to the discovered VM
smBiosId*	ID/version of the system management BIOS
state*	Status of the Discovery Connector virtual appliance
toolsStatus	Operational state of VMware tools (See Viewing and Sorting Data Collectors (p. 52) for a complete list.)
totalDiskSize	Total capacity of disk expressed in MB
totalRAM	Total amount of RAM available on VM in MB

Data field	Description
type	Type of host
vCenterId	Unique ID number of a VM
vCenterName*	Name of the vCenter host
virtualSwitchName*	Name of the virtual switch
vmFolderPath	Directory path of VM files
vmName	Name of the virtual machine

Download the Discovery Connector

Download, Set Up, and Start Collecting Data

To set up agentless discovery, you must download and deploy the Discovery Connector, which is a virtual appliance, on a VMware vCenter Server host in your on-premises environment. The Discovery Connector is an Open Virtualization Archive (OVA) file that you must install in your on-premises VMware environment.

> **Reminder**
> Discovery Connector supports VMware vCenter versions V5.5, V6.0, V6.5, V6.7, and 7.0.

Beginning with this section and those that follow on this page, you will be instructed how to download, deploy, configure, and start collecting data using the Discovery Connector.

To download the Discovery Connector OVA file and verify its checksum.

1. Sign in to vCenter as a VMware administrator and switch to the directory where you want to download the Discovery Connector OVA file.
2. Download the Discovery Connector OVA.
3. Depending on which hashing algorithm you use in your system environment, download either the MD5 or SHA256 to get the file containing the checksum value. Use this value to verify the `AWSDiscoveryConnector.ova` file downloaded in the preceding step.
4. Depending on your variation of Linux, run the version appropriate MD5 command or SHA256 command to verify that the cryptographic signature of the `AWSDiscoveryConnector.ova` file matches the value in the respective MD5/SHA256 file that you downloaded.

```
$ md5sum AWSDiscoveryConnector.ova
```

```
$ sha256sum AWSDiscoveryConnector.ova
```

Deploy the Discovery Connector

In the previous section you downloaded the AWS Agentless Discovery Connector in the Open Virtualization Archive (OVA) file. This section lists the specifications for the Discovery Connector that you downloaded and shows you how to deploy it in your VMware environment.

Discovery Connector virtual machine specifications

- **Operating System** –FreeBSD 11 (64 bit)
- **RAM** –2 GB
- **Disk storage**

  When the you deploy the OVA as virtual machine in vCenter, the vCenter client gives you the following two options to provision:

  - **Thin Provisioned** –approximately 7.8 GB
  - **Thick Provisioned** –approximately 299.0 GB (recommended option)

The following procedure steps you through deploying the Discovery Connector OVA file in your VMware environment.

To deploy the Discovery Connector

1. Sign in to vCenter as a VMware administrator.
2. Choose **File**, **Deploy OVF Template**, select the OVA file you downloaded in the previous section, and complete the wizard.
3. On the **Disk Format** page, select one of the thick provision disk types. We recommend that you choose **Thick Provision Eager Zeroed**, because it has the best performance and reliability. However, it requires several hours to zero out the disk. Do not choose **Thin Provision**. This option makes deployment faster but significantly reduces disk performance. For more information, see Types of supported virtual disks in the VMware documentation.
4. Locate and open the context (right-click) menu for the newly deployed template in the vSphere client inventory tree and choose **Power**, **Power On**.
5. Open the context (right-click) menu for the template again and choose **Open Console**. The console displays the IP address of the connector console. Make note of the IP address as you'll need it in order to complete the connector setup process.

Configure the AWS Discovery Connector

To finish the setup process, complete the following procedure and the optional connector configuration tasks as needed.

> **Reminder**
> Before starting the procedure select a Migration Hub home region, if you haven't already done so.

To configure the connector using the console

1. In a web browser, type the following URL in the address bar: **`https://`***`<ip_address>`***`/`**, where *ip_address* is the IP address of the connector console that you saved earlier.
2. Choose **Get started now** and then follow the directions to complete the following setup pages: **License Agreement**, **Create a Password**, and **Network Info.**
3. On the **Log Uploads and Upgrades** page, we recommend that you select **Upload logs automatically**. When your logs are made available through automatic uploads, AWS can better help you troubleshoot connector issues.

   The **AWS Agentless Discovery Connector auto-upgrade** feature is enabled by default. Running the latest version of the connector ensures that the latest security patches are installed. You can disable auto-upgrades at any time, see Disabling auto-upgrades on AWS Discovery Connector (p. 27).

4. On the **Discovery Connector Set Up** page, perform the following:

   a. Under **Configure vCenter credentials**:

      i. For **vCenter Host**, enter the hostname or IP address of your VMware vCenter Server host.

      ii. For **vCenter Username**, enter the name of a local or domain user that the connector uses to communicate with vCenter. For domain users, use the form *domain*\username or *username@domain*.

      iii. For **vCenter Password**, enter the local or domain user password.

      iv. Choose **Ignore security certificate** to bypass SSL certificate validation with vCenter.

   b. Under **Configure AWS credentials**, enter the credentials for the IAM user who is assigned the `AWSAgentlessDiscoveryService` IAM managed policy. For more information about managed policies, see AWS Managed (Predefined) Policies for Application Discovery Service (p. 75).

      Then choose **Next**.

   c. Under **Configure where to publish data**, select to publish to a local file or to a specific AWS Regional endpoint. If you select to publish to a local file, your Discovery Connector will not send data about your on-premise servers to AWS. However, the Discovery Connector will continue to send data about the connector itself to AWS.

      Then choose **Next** to go back to the AWS Agentless Discovery Connector console.

The following topics describe optional connector configuration tasks.

Topics

- Configure a static IP address for the connector (p. 25)
- Control the scope of data collection (p. 26)
- Disabling auto-upgrades on AWS Discovery Connector (p. 27)

Configure a static IP address for the connector

Follow this procedure if your environment requires that you use a static IP address.

To Configure a static IP address for the connector

1. Open the connector's virtual machine console and log in as **`ec2-user`** with the password **`ec2pass`**. Supply a new password if prompted.
2. Run the command **`sudo setup.rb`** and enter the password for *ec2-user* when prompted to display the configuration menu.
3. Enter **2** to select **Reconfigure network settings**. This displays current network information and a submenu for making changes to the network settings.
4. In the submenu generated from the previous step, enter **2** to select **Set up a static IP**. This will display a form to supply network settings:

   - For each field, provide an appropriate value and press Enter. You should see output similar to the following where *nnn.nnn.nnn.nnn* is populated with the address numbers you entered for each field:

```
Setting up static IP:
     1. Enter IP address: <nnn.nnn.nnn.nnn>
     2. Enter netmask: <nnn.nnn.nnn.nnn>
     3. Enter gateway: <nnn.nnn.nnn.nnn>
     4. Enter DNS 1: <nnn.nnn.nnn.nnn>
     5. Enter DNS 2: <nnn.nnn.nnn.nnn>

Static IP address configured.
```

Control the scope of data collection

The vCenter user requires read-only permissions on each ESX host or VM to inventory using Application Discovery Service. Using the permission settings, you can control which hosts and VMs are included in the data collection. You can either allow all hosts and VMs under the current vCenter to be inventoried, or grant permissions on a case-by-case basis.

> **Note**
> As a security best practice, we recommend against granting additional, unneeded permissions to the vCenter user of the Discovery Connector.

The following procedures describe configuration scenarios ordered from least granular to most granular.

To discover data about all ESX hosts and VMs under the current vCenter

1. In your VMware vSphere client, choose **vCenter** and then choose either **Hosts and Clusters** or **VMs and Templates**.
2. Choose **Manage**, **Permissions**.
3. Select the vCenter user, open the context (right-click) menu, and choose **Change Role**.
4. In the **Assigned Role** pane, choose **Read-only**.
5. Choose **Propagate to children**, and then choose **OK**.

To discover data about a specific ESX host and all of its child objects

1. In your VMware vSphere client, choose **vCenter** and then choose either **Hosts and Clusters** or **VMs and Templates**.
2. Choose **Related Objects**, **Hosts**.
3. Open the context (right-click) menu for the host name and choose **All vCenter Actions**, **Add Permission**.
4. Under **Add Permission**, add the vCenter user to the host. For **Assigned Role**, choose **Read-only**.
5. Choose **Propagate to children**, and then choose **OK**.

Discover data about a specific ESX host or child VM

1. In your VMware vSphere client, choose **vCenter** and then choose either **Hosts and Clusters** or **VMs and Templates**.
2. Choose **Related Objects**.
3. Choose **Hosts** (showing a list of ESX hosts known to vCenter) or **Virtual Machines** (showing a list of VMs across all ESX hosts).
4. Open the context (right-click) menu for the host or VM name and choose **All vCenter Actions**, **Add Permission**.
5. Under **Add Permission**, add the vCenter user to the host or VM. For **Assigned Role**, choose **Read-only**, .
6. Choose **OK**.

> **Note**
> If you chose **Propagate to children**, you can still remove the read-only permission from ESX hosts and VMs on a case-by-case basis. This option has no effect on inherited permissions applying to other ESX hosts and VMs.

Disabling auto-upgrades on AWS Discovery Connector

To ensure that you are running the latest version of AWS Discovery Connector, the auto-upgrade feature is enabled by default upon installation. However, you may disable the auto-upgrade feature as shown below.

To disable auto-upgrades

1. In a web browser, type the following URL in the address bar: **`https://<ip_address>/`**, where *ip_address* is the IP address of the AWS Discovery Connector.
2. In the Discovery Connector console, under **Actions**, choose **Disable Auto-Upgrade**.

> **Warning**
> Disabling auto-upgrades will prevent the latest security patches from being installed.

Discovery Connector Data Collection

After you have deployed and configured the Discovery Connector in your VMware environment, if data collections stops you can restart it. You can start or stop data collection through the console or by making API calls through the AWS CLI. Both of these methods are described in the following procedures.

Using the Migration Hub Console

The following procedure shows how to start or stop the Discovery Connector data collection process, on the **Data Collectors** page of the Migration Hub console.

To start or stop data collection

1. In the navigation pane, choose **Data Collectors**.
2. Choose the **Connectors** tab.
3. Select the check box of the connector you want to start or stop.
4. Choose **Start data collection** or **Stop data collection**.

> **Note**
> If you don't see inventory information after starting data collection with the connector, confirm that you have registered the connector with your vCenter Server.

Using the AWS CLI

To start the Discovery Connector data collection process from the AWS CLI, the AWS CLI must first be installed in your environment, and then you must set the CLI to use your selected Migration Hub home region.

To install the AWS CLI and start data collection

1. Install the AWS CLI for your operating system (Linux, macOS, or Windows). See the AWS Command Line Interface User Guide for instructions.

2. Open the Command prompt (Windows) or Terminal (Linux or macOS).

   a. Type `aws configure` and press Enter.
   b. Enter your AWS Access Key ID and AWS Secret Access Key.
   c. Enter your home region, for example `us-west-2`, for the Default Region Name.
   d. Enter `text` for Default Output Format.

3. To find the ID of the connector you want to start or stop data collection for, type the following command to see the connector's ID:

```
aws discovery describe-agents --filters
 condition=EQUALS,name=hostName,values=connector
```

4. To start data collection by the connector, type the following command:

```
aws discovery start-data-collection-by-agent-ids --agent-ids <connector ID>
```

   Note
   If you don’t see inventory information after starting data collection with the connector, confirm that you have registered the connector with your vCenter Server.

   To stop data collection by the connector, type the following command:

```
aws discovery stop-data-collection-by-agent-ids --agent-ids <connector ID>
```

Troubleshooting the Discovery Connector

This section contains topics that can help you troubleshoot known issues with Application Discovery Service.

Fixing Discovery Connector cannot reach AWS during setup

When configuring the AWS Agentless Discovery Connector in the console you can get the following error message:

> **Could Not Reach AWS**
> AWS cannot be reached (connection reset). Please verify network and proxy settings.

This error occurs because of a failed attempt by the Discovery Connector to establish an HTTPS connection to `ec2.amazonaws.com` during the setup process. The Discovery Connector configuration fails if a connection can't be established.

To fix the connection to AWS

1. Check if your firewall is blocking egress traffic to `ec2.amazonaws.com`. If it is, unblock it. After you update the firewall, reconfigure the Discovery Connector.
2. If updating the firewall does not resolve the connection issue, check to make sure that the connector virtual machine has outbound network connectivity. If the virtual machine has outbound connectivity, test the connection to `aws.amazon.com` and `ec2.amazonaws.com` by running **telnet** on ports 80 and 443 as shown in the following example.

```
telnet ec2.amazonaws.com 80
```

3. If outbound connectivity from the virtual machine is enabled, you must contact AWS Support for further troubleshooting.

Fixing unhealthy connectors

Health information for every Discovery Connector can be found in the Data Collectors page of the Migration Hub console. You can identify connectors with problems by finding any connectors with a **Health** status of **Unhealthy**. The following procedure outlines how to access the connector console to identify health issues.

Access a connector console

1. Open the Migration Hub console in a web browser, and choose **Data Collectors** from the left hand navigation.
2. From the **Connectors** tab, make a note of the **IP address** for each connector that has a health status of **Unhealthy**.
3. Open a browser on any computer that can connect to the connector virtual machine, and enter the URL of the connector console, `https://`*`ip_address_of_connector`*, where `ip_address_of_connector` is the IP address of an unhealthy connector.
4. Enter the connector management console password, which was set up when the connector was configured.

Once you've accessed the connector console, you can take actions to resolve an unhealthy status. Here you can choose **View Info** for **vCenter connectivity**, and you'll get a dialog box with a diagnostic message. The **View Info** link is only available on connectors that are version 1.0.3.12 or later.

After correcting the health issues, the connector will re-establish connectivity with vCenter server, and the connector's status will change to the **HEALTHY** state. If the issues persist, contact AWS Support.

The most common causes for unhealthy connectors are IP address issues and credentials issues. The following sections can help you resolve these issues and return a connector to a healthy state.

Topics

- IP address issues (p. 29)
- Credentials issues (p. 30)

IP address issues

A connector can go into an unhealthy state if the vCenter endpoint provided during connector setup is malformed, invalid, or if the vCenter server is currently down and not reachable. In this case, when you choose **View Info** for **vCenter connectivity** you'll get a dialog box with the message "Confirm the operational status of your vCenter server, or choose Edit Settings to update the vCenter endpoint."

The following procedure can help you resolve IP address issues.

1. From the connector console (`https://`*`ip_address_of_connector`*), choose **Edit Settings**.
2. From the left-side navigation, choose **Step 5: Discovery Connector Set Up**.
3. From **Configure vCenter credentials**, make a note of the **vCenter Host** IP address.
4. Using a separate command line tool like `ping` or `traceroute`, validate that the associated vCenter server is active and the IP is reachable from the connector VM.

   - If the IP address is incorrect and the vCenter service is active, then update the IP address in the connector console, and choose **Next**.

- If the IP address is correct but the vCenter server is inactive, activate it.
- If the IP address is correct and the vCenter server is active, check if it is blocking ingress network connections due to firewall issues. If yes, update your firewall settings to allow incoming connections from the connector VM.

Credentials issues

Connectors can go into an unhealthy state if the vCenter user credentials provided during connector setup, are invalid, or do not have vCenter read and view account privileges. In this case, when you choose **View Info** for **vCenter connectivity** you'll get a dialog box with the message "Choose Edit Settings to update your vCenter username and password for your account with read and view privileges."

The following procedure can help you resolve credentials issues. As a prerequisite, ensure that you have created a vCenter user that has read and view account permissions on vCenter server.

1. From the connector console (`https://`*`ip_address_of_connector`*), choose **Edit Settings**.
2. From the left-side navigation, choose **Step 5: Discovery Connector Set Up**.
3. From **Configure vCenter credentials**, update the **vCenter Username** and **vCenter Password** by providing the credentials for a vCenter user with read and view permissions.
4. Choose **Next** to complete setup.

Standalone ESX host support

The Discovery Connector does not support a standalone ESX host. The ESX host must be part of the vCenter Server instance.

Getting additional support for connector issues

If you encounter problems and need help, contact AWS Support. You will be contacted and may be asked to send the connector logs. To obtain the logs, do the following:

- Log back in to the AWS Agentless Discovery Connector console (as you did during configuration (p. 24)) and choose **Download log bundle**.
- Once the log bundle has finished downloading, send it as instructed by AWS Support.

Migration Hub Import

Migration Hub import allows you to import details of your on-premises environment directly into Migration Hub without using the Discovery Connector or Discovery Agent, so you can perform migration assessment and planning directly from your imported data. You also can group your devices as applications and track their migration status.

To initiate an import request

- Download the specially-formatted, comma separated value (CSV) import template.
- Populate it with your existing on-premises server data.
- Upload it to Migration Hub using the Migration Hub console, AWS CLI or one of the AWS SDKs.

You can submit multiple import requests. Each request is processed sequentially. You can check the status of your import requests at any time, through the console or import APIs.

After an import request is complete, you can view the details of individual imported records. View utilization data, tags, and application mappings directly from within the Migration Hub console. If errors were encountered during the import, you can review the count of successful and failed records, and you can see the error details for each failed record.

Handling errors: A link is provided to download the error log and failed records files as CSV files in a compressed archive. Use these files to resubmit your import request after correcting the errors.

Limits apply to the number of imported records, imported servers, and deleted records you can keep. For more information, see AWS Application Discovery Service Quotas (p. 91).

Supported Import File Fields

Migration Hub import allows you to import data from any source. The data provided must be in the supported format for a CSV file, and the data must contain only the supported fields with the supported ranges for those fields.

An asterisk next to an import field name in the following table denotes that it is a required field. Each record of your import file must have at least one or more of those required fields populated to uniquely identify a server or application. Otherwise, a record without any of the required fields will fail to be imported.

Note
If you're using either VMware.MoRefId or VMWare.VCenterId, to identify a record, you must have both fields in the same record.

Import Field Name	Description	Examples
ExternalId*	A custom identifier that allows you to mark each record as unique. For example, **ExternalId** can be the inventory ID for the server in your data center.	Inventory Id 1 Server 2 CMBD Id 3
SMBiosId	System management BIOS (SMBIOS) ID.	

Import Field Name	Description	Examples
IPAddress*	A comma-delimited list of IP addresses of the server, in quotes.	192.0.0.2 "10.12.31.233, 10.12.32.11"
MACAddress*	A comma-delimited list of MAC address of the server, in quotes.	00:1B:44:11:3A:B7 "00-15-E9-2B-99-3C, 00-14-22-01-23-45"
HostName*	The host name of the server. We recommend using the fully qualified domain name (FQDN) for this value.	ip-1-2-3-4 localhost.domain
VMware.MoRefId*	The managed object reference ID. Must be provided with a VMware.VCenterId.	
VMware.VCenterId*	Virtual machine unique identifier. Must be provided with a VMware.MoRefId.	
CPU.NumberOfProcessors	The number of CPUs.	4
CPU.NumberOfCores	The total number of physical cores.	8
CPU.NumberOfLogicalCores	The total number of threads that can run concurrently on all CPUs in a server. Some CPUs support multiple threads to run concurrently on a single CPU core. In those cases, this number will be larger than the number of physical (or virtual) cores.	16
OS.Name	The name of the operating system.	Linux Windows.Hat
OS.Version	The version of the operating system.	16.04.3 NT 6.2.8
VMware.VMName	The name of the virtual machine.	Corp1
RAM.TotalSizeInMB	The total RAM available on the server, in MB.	64 128
RAM.UsedSizeInMB.Avg	The average amount of used RAM on the server, in MB.	64 128
RAM.UsedSizeInMB.Max	The maximum amount of used RAM available on the server, in MB.	64 128

Import Field Name	Description	Examples
CPU.UsagePct.Avg	The average CPU utilization when the discovery tool was collecting data.	45 23.9
CPU.UsagePct.Max	The maximum CPU utilization when the discovery tool was collecting data.	55.34 24
DiskReadsPerSecondInKB.Avg	The average number of disk reads per second, in KB.	1159 84506
DiskWritesPerSecondInKB.Avg	The average number of disk writes per second, in KB.	199 6197
DiskReadsPerSecondInKB.Max	The maximum number of disk reads per second, in KB.	37892 869962
DiskWritesPerSecondInKB.Max	The maximum number of disk writes per second, in KB.	18436 1808
DiskReadsOpsPerSecond.Avg	The average number of disk read operations per second.	45 28
DiskWritesOpsPerSecond.Avg	The average number of disk write operations per second.	8 3
DiskReadsOpsPerSecond.Max	The maximum number of disk read operations per second.	1083 176
DiskWritesOpsPerSecond.Max	The maximum number of disk write operations per second.	535 71
NetworkReadsPerSecondInKB.Avg	The average number of network read operations per second, in KB.	45 28
NetworkWritesPerSecondInKB.Avg	The average number of network write operations per second, in KB.	8 3
NetworkReadsPerSecondInKB.Max	The maximum number of network read operations per second, in KB.	1083 176
NetworkWritesPerSecondInKB.Max	The maximum number of network write operations per second, in KB.	535 71

Import Field Name	Description	Examples
Applications	A comma-delimited list of applications that include this server, in quotes. This value can include existing applications and/or new applications that are created upon import.	Application1 "Application2, Application3"
Tags	A comma-delimited list of tags formatted as name:value. **Important** Do not store sensitive information (like personal data) in tags.	"zone:1, critical:yes" "zone:3, critical:no, zone:1"

You can import data even if you don't have data populated for all the fields defined in the import template, so long as each record has at least one of the required fields within it. Duplicates are managed across multiple import requests by using either an external or internal matching key. If you populate your own matching key, `External ID`, this field is used to uniquely identify and import the records. If no matching key is specified, import uses an internally generated matching key that is derived from some of the columns in the import template. For more information on this matching, see Matching Logic for Discovered Servers and Applications (p. 40).

> **Note**
> Migration Hub import does not support any fields outside of those defined in the import template. Any custom fields supplied will be ignored and will not be imported.

Setting Up Your Import Permissions

Before you can import your data, ensure that your IAM user has the necessary Amazon S3 permissions to upload (`s3:PutObject`) your import file to Amazon S3, and to read the object (`s3:GetObject`). You also must establish programmatic access (for the AWS CLI) or console access, by creating an IAM policy and attaching it to the IAM user that performs imports in your AWS account.

Console Permissions

Use the following procedure to edit the permissions policy for the IAM user that will make import requests in your AWS account using the console.

To edit a user's attached managed policies

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Users**.
3. Choose the name of the user whose permissions policy you want to change.
4. Choose the **Permissions** tab and choose **Add permissions**.
5. Choose **Attach existing policies directly**, and then choose **Create policy**.

   a. In the **Create policy** page that opens, choose **JSON**, and paste in the following policy. Remember to replace the name of your bucket with the actual name of the bucket that the IAM user will upload the import files into.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:ListAllMyBuckets"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": ["s3:ListBucket"],
      "Resource": ["arn:aws:s3:::importBucket"]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:DeleteObject"
      ],
      "Resource": ["arn:aws:s3:::importBucket/*"]
    }
  ]
}
```

   b. Choose **Review policy**.

   c. Give your policy a new **Name** and optional description, before reviewing the summary of the policy.

   d. Choose **Create policy**.

6. Return to the **Grant permissions** IAM console page for the user that will make import requests in your AWS account.
7. Refresh the table of policies, and search for the name of the policy you just created.
8. Choose **Next: Review**.
9. Choose **Add permissions**.

Now that you've added the policy to your IAM user, you're ready to start the import process.

AWS CLI Permissions

Use the following procedure to create the managed policies necessary to give an IAM user the permissions to make import data requests using the AWS CLI.

To create and attach the managed policies

1. Use the `aws iam create-policy` AWS CLI command to create an IAM policy with the following permissions. Remember to replace the name of your bucket with the actual name of the bucket that the IAM user will upload the import files into.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:ListBucket"],
      "Resource": ["arn:aws:s3:::importBucket"]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:DeleteObject"
      ],
      "Resource": ["arn:aws:s3:::importBucket/*"]
    }
  ]
}
```

For more information on using this command, see create-policy in the *AWS CLI Command Reference*.

2. Use the `aws iam create-policy` AWS CLI command to create an additional IAM policy with the following permissions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "discovery:ListConfigurations",
                "discovery:CreateApplication",
                "discovery:UpdateApplication",
                "discovery:AssociateConfigurationItemsToApplication",
                "discovery:DisassociateConfigurationItemsFromApplication",
                "discovery:GetDiscoverySummary",
                "discovery:StartImportTask",
                "discovery:DescribeImportTasks",
                "discovery:BatchDeleteImportData"
            ],
            "Resource": "*"
        }
    ]
}
```

3. Use the `aws iam attach-user-policy` AWS CLI command to attach the policies you created in the previous two steps to the IAM user that will be performing import requests in your AWS account using the AWS CLI. For more information on using this command, see attach-user-policy in the *AWS CLI Command Reference*.

Now that you've added the policies to your IAM user, you're ready to start the import process.

Remember that when the IAM user uploads objects to the Amazon S3 bucket that you specified, they must leave the default permissions for the objects set so that the user can read the object.

Uploading Your Import File to Amazon S3

Next, you must upload your CSV formatted import file into Amazon S3 so it can be imported. Before you begin, you should have an Amazon S3 bucket that will house your import file created and/or chosen ahead of time.

Console S3 Upload

To upload your import file to Amazon S3

1. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.

2. In the **Bucket name** list, choose the name of the bucket that you want to upload your object to.
3. Choose **Upload**.
4. In the **Upload** dialog box, choose **Add files** to choose the file to upload.
5. Choose a file to upload, and then choose **Open.**
6. Choose **Upload**.
7. Once your file has been uploaded, choose the name of your data file object from your bucket dashboard.
8. From the **Overview** tab of the object details page, copy the **Object URL**. You'll need this when you create your import request.
9. Go to the **Import** page in the Migration Hub console as described in Importing Data (p. 37). Then, paste the object URL in the **Amazon S3 Object URL** field.

AWS CLI S3 Upload

To upload your import file to Amazon S3

1. Open a terminal window and navigate to the directory that your import file is saved to.
2. Enter the following command:

```
aws s3 cp ImportFile.csv s3://BucketName/ImportFile.csv
```

3. This returns the following results:

```
upload: .\ImportFile.csv to s3://BucketName/ImportFile.csv
```

4. Copy the full Amazon S3 object path that was returned. You will need this when you create your import request.

Importing Data

After you download the import template from the Migration Hub console and populate it with your existing on-premises server data, you're ready to start importing the data into Migration Hub. The following instructions describe two ways to do this, either by using the console or by making API calls through the AWS CLI.

Console Import

Start data import on the **Tools** page of the Migration Hub console.

To start data import

1. In the navigation pane, under **Discover**, choose **Tools**.
2. If you don't already have an import template filled out, you can download the template by choosing **import template** in the **Import** box. Open the downloaded template and populate it with your existing on-premises server data. You can also download the import template from our Amazon S3 bucket at https://s3.us-west-2.amazonaws.com/templates-7cffcf56-bd96-4b1c-b45b-a5b42f282e46/import_template.csv
3. To open the **Import** page, choose **Import** in the **Import** box.
4. Under **Import name**, specify a name for the import.
5. Fill out the **Amazon S3 Object URL** field. To do this step, you'll need to upload your import data file to Amazon S3. For more information, see Uploading Your Import File to Amazon S3 (p. 36).

6. Choose **Import** in the lower-right area. This will open the **Imports** page where you can see your import and its status listed in the table.

After following the preceding procedure to start your data import, the **Imports** page will show details of each import request including its progress status, completion time, and the number of successful or failed records with the ability to download those records. From this screen, you can also navigate to the **Servers** page under **Discover** to see the actual imported data.

On the **Servers** page, you can see a list of all the servers (devices) that are discovered along with the import name. When you navigate from the **Imports** (import history) page by selecting the name of the import listed in the **Name** column, you are taken to the **Servers** page where a filter is applied based on the selected import's data set. Then, you only see data belonging to that particular import.

The archive is in a .zip format, and contains two files; `errors-file` and `failed-entries-file`. The errors file contains a list of error messages associated with each failed line and associated column name from your data file that failed the import. You can use this file to quickly identify where problems occurred. The failed entries file includes each line and all the provided columns that failed. You can make the changes called out in the errors file in this file and attempt to import the file again with the corrected information.

AWS CLI Import

To start the data import process from the AWS CLI, the AWS CLI must first be installed in your environment. For more information, see Installing the AWS Command Line Interface in the *AWS Command Line Interface User Guide*.

> **Note**
> If you don't already have an import template filled out, you can download the import template from our Amazon S3 bucket here: https://s3.us-west-2.amazonaws.com/templates-7cffcf56-bd96-4b1c-b45b-a5b42f282e46/import_template.csv

To start data import

1. Open a terminal window, and type the following command:

```
aws discovery start-import-task --import-url s3://BucketName/ImportFile.csv --
name ImportName
```

2. This will create your import task, and return the following status information:

```
{
    "task": {
        "status": "IMPORT_IN_PROGRESS",
        "applicationImportSuccess": 0,
        "serverImportFailure": 0,
        "serverImportSuccess": 0,
        "name": "ImportName",
        "importRequestTime": 1547682819.801,
        "applicationImportFailure": 0,
        "clientRequestToken": "EXAMPLE1-abcd-1234-abcd-EXAMPLE1234",
        "importUrl": "s3://BucketName/ImportFile.csv",
        "importTaskId": "import-task-EXAMPLE1229949eabfEXAMPLE03862c0"
    }
}
```

Tracking Your Migration Hub Import Requests

You can track the status of your Migration Hub import requests using the console, AWS CLI, or one of the AWS SDKs.

Console Tracking

From the **Imports** dashboard in the Migration Hub console, you'll find the following elements.

- **Name** – The name of the import request.
- **Import ID** – The unique ID of the import request.
- **Import time** – The date and time that the import request was created.
- **Import status** – The status for the import request. This can be one of the following values:
  - **Importing** – This data file is currently being imported.
  - **Imported** – The entire data file was successfully imported.
  - **Imported with errors** – One or more of the records in the data file failed to import. To resolve your failed records, choose **Download failed records** for your import task and resolve the errors in the failed entries csv file, and do the import again.
  - **Import Failed** – None of the records in the data file where imported. To resolve your failed records, choose **Download failed records** for your import task and resolve the errors in the failed entries csv file, and do the import again.
- **Imported records** – The number of records in a specific data file that were successfully imported.
- **Failed records** – The number records in a specific data file that weren't imported.

CLI Tracking

You can track the status of your import tasks with the `aws discovery describe-import-tasks` AWS CLI command.

1. Open a terminal window, and type the following command:

```
aws discovery describe-import-tasks
```

2. This will return a list of all your import tasks in JSON format, complete with status and other relevant information. Optionally, you can filter results to return a subset of your import tasks.

When tracking your import tasks, you may find that the `serverImportFailure` value returned is greater than zero. When this happens, your import file had one or more entries that couldn't be imported. This can be resolved by downloading your failed records archive, reviewing the files within, and doing another import request with the modified failed-entries.csv file.

After creating your import task, you can perform additional actions to help manage and track your data migration. For example, you can download an archive of failed records for a specific request. For information on using the failed records archive to resolve import issues, see Troubleshooting Failed Import Records (p. 95).

View, Export, and Explore Discovered Data

The AWS Discovery Connector and AWS Discovery Agent both provide system performance data based on average and peak utilization. You can use the system performance data collected to perform a high-level TCO (total cost of ownership). Discovery Agents collect more detailed data including time series data for system performance information, inbound and outbound network connections, and processes running on the server. You can use this data to understand network dependencies between servers and group the related servers as applications for migration planning.

In this section you'll find instructions on how to view and work with data discovered by Discovery Connectors and Discovery Agents from both the console and the AWS CLI.

Topics

- View Collected Data Using the Console (p. 40)
- Export collected data (p. 41)
- Data Exploration in Amazon Athena (p. 42)

View Collected Data Using the Console

For both the Discovery Connector and Discovery Agent, after the data collection process starts, you can use the console to view their collected data about your servers and VMs. Data appears in the console approximately 15 minutes after data collection starts. This data can also be viewed in a csv format by exporting the collected data by making API calls through the AWS CLI. Exporting collected data is covered in the next section Export collected data (p. 41).

To view collected data about discovered servers

1. In the console's navigation pane, choose **Servers**. The discovered servers appear in the servers list.
2. For details comprised of the collected data, choose the server name link in the **Server info** column. Doing so displays a screen that describes detail information such as system information, performance metrics, and more.

To learn more about using the console to view, sort, and tag servers discovered by your Discovery Connectors or Discovery Agents, see AWS Application Discovery Service Console Walkthroughs (p. 51).

Matching Logic for Discovered Servers and Applications

AWS Application Discovery Service has built-in matching logic that identifies when servers that it discovers match existing entries. When this logic finds a match, it updates the information for the already-existing discovered server with new values. This matching logic handles duplicate servers from multiple sources including Migration Hub import, Discovery Connector, Discovery Agent, and other migration tools. For more information about Migration Hub import, see AWS Migration Hub Import.

When server discovery occurs, each entry is cross-checked with previously imported records to ensure that the imported server does not already exist. If no match is found, a new record is created and a new unique server identifier is assigned. If a match is found, then a new entry is still created, but it's assigned the same unique server identifier as the existing server. When viewing this server in the Migration Hub console, you only find one unique entry for the server.

Server attributes associated with this entry are merged to show attribute values from a previously available record as well as the newly imported record. If there is more than one value for a given server attribute from multiple sources, e.g., two different values within for `Total RAM` associated with a given server discovered using import and also by the Discovery Agent, then the value that was most recently updated is shown in the matched record for the server.

Matching Fields

The following fields are used to match servers when discovery tools are used.

- **ExternalId** – This is the primary field used to match servers. If the value in this field is identical to another `ExternalId` in another entry, then Application Discovery Service matches the two entries, regardless of whether the other fields match or not.
- **IPAddress**
- **HostName**
- **MacAddress**
- **VMware.MoRefId** and **VMware.vCenterId** – Both of these values must be identical to the respective fields in another entry for Application Discovery Service to perform a match.

Export collected data

For both the Discovery Connector and Discovery Agent, after the data collection process starts, you can export their collected data about your servers and VMs. This data can be exported either by interacting with the console or by making API calls through the AWS CLI, depending on which discovery tool you used to collect data.

- **Discovery Agent**, you can export the collected data by using the console or the AWS CLI.
- **Discovery Connector**, you can only export the collected data by using the AWS CLI.

Instructions are provided below for both ways by expanding your method of choice:

Export data collected for all servers

The data collected from all the Discovery Connectors and Discovery Agents running on your hosts and VMs can be bulk exported using the AWS CLI. If not already installed, the AWS CLI must first be installed in your environment.

To install the AWS CLI and export collected data

1. If you have not already done so, install the AWS CLI appropriate to your OS type (Windows or Mac/Linux). See the AWS Command Line Interface User Guide for instructions.
2. Open the Command prompt (Windows) or Terminal (MAC/Linux).
   a. Type `aws configure` and press Enter.
   b. Enter your AWS Access Key Id and AWS Secret Access Key.
   c. Enter `us-west-2` for the Default Region Name.
   d. Enter `text` for Default Output Format.
3. Type the following command to generate an export ID:

```
aws discovery start-export-task
```

4. Using the export ID generated in the previous step, type the following command to generate an S3 URL as a value for the parameter "`configurationsDownloadUrl`":

```
aws discovery describe-export-tasks --export-ids <export ID>
```

5. Copy the URL generated in the previous step and paste it in a browser to download the zip file with collected data of the discovered servers.

Export agent collected data using the console

Exporting agent collected data from the console is limited to one agent, when you are on the detail page for a specific server. On the detail page, you can find the server's export jobs listed at the bottom of the screen, underneath **Exports**. If no export jobs exist, the table is empty. You can run up to five exports of server data at a time.

To export collected data about a discovered server

1. In the navigation pane, choose **Servers**.
2. In the **Server info** column, choose the link for the server that you want to export data for.
3. In the **Exports** section at the bottom of the screen, choose **Export server details**.
4. For **Export server details**, fill in **Start date** and **Time**.

   Note
   The start time can't be more than 72 hours prior from the current time.
5. Choose **Export** to start the job. The initial status is **In-progress**; to update the status, click the refresh icon for the **Exports** section.
6. When the export job is complete, choose **Download** and save the .zip file.
7. Unzip the saved file. A set of .csv files contains the export data.

   You can open the .csv files in Microsoft Excel and review the exported server data.

   Among the files, you can find a JSON file containing data about the export task and its results.

Note
For information on generating and exporting Amazon EC2 instance recommendations in the AWS Migration Hub console, see Amazon EC2 Instance Recommendations in the *AWS Migration Hub User Guide*.

Data Exploration in Amazon Athena

Data Exploration in Amazon Athena allows you to analyze the data collected from all the discovered on-premises servers by Discovery Agents in one place. Once Data Exploration in Amazon Athena is enabled from the Migration Hub console (or by using the StartContinousExport API) and the data collection for agents is turned on, data collected by agents will automatically get stored in your S3 bucket at regular intervals.

You can then visit Amazon Athena to run pre-defined queries to analyze the time-series system performance for each server, the type of processes that are running on each server and the network dependencies between different servers. In addition, you can write your own custom queries using

Amazon Athena, upload additional existing data sources such as configuration management database (CMDB) exports, and associate the discovered servers with the actual business applications. You can also integrate the Athena database with Amazon QuickSight to visualize the query outputs and perform additional analysis

Steps

1. Enabling Data Exploration in Amazon Athena (p. 43)
2. Working with Data Exploration in Amazon Athena (p. 44)

Enabling Data Exploration in Amazon Athena

Data Exploration in Amazon Athena is enabled by turning on Continuous Export using the Migration Hub console or an API call from the AWS CLI. You must turn on data exploration before you can see and start exploring your discovered data in Amazon Athena,

When you turn on Continuous Export the service-linked role `AWSServiceRoleForApplicationDiscoveryServiceContinuousExport` is automatically used by your account. For more information about this service-linked role, see Service-Linked Role Permissions for Application Discovery Service (p. 83).

The following instructions show how to enable Data Exploration in Amazon Athena by using the console and the AWS CLI.

Enable with the console

> Data Exploration in Amazon Athena is enabled by Continuous Export implicitly being turned on when you choose "Start data collection", or click the toggle labeled, "Data exploration in Amazon Athena" on the **Data Collectors** page of the Migration Hub console.
>
> **To enable Data Exploration in Amazon Athena from the console**
>
> 1. In the navigation pane, choose **Data Collectors**.
> 2. Choose the **Agents** tab.
> 3. Choose **Start data collection**, or if you already have data collection turned on, click the **Data exploration in Amazon Athena** toggle.
> 4. In the dialog box generated from the previous step, click the checkbox agreeing to associated costs and choose **Continue** or **Enable**.
>
> **Note**
> Your agents are now running in "continuous export" mode which will enable you to see and work with your discovered data in Amazon Athena. The first time this is enable it may take up to 30 minutes for your data to appear in Amazon Athena.

Enable with the AWS CLI

> Data Exploration in Amazon Athena is enabled by Continuous Export explicitly being turned on through an API call from the AWS CLI. To do this, the AWS CLI must first be installed in your environment.
>
> **To install the AWS CLI and enable Data Exploration in Amazon Athena**
>
> 1. Install the AWS CLI for your operating system (Linux, macOS, or Windows). See the AWS Command Line Interface User Guide for instructions.
> 2. Open the Command prompt (Windows) or Terminal (Linux or macOS).

   a. Type `aws configure` and press Enter.
   b. Enter your AWS Access Key Id and AWS Secret Access Key.
   c. Enter `us-west-2` for the Default Region Name.
   d. Enter `text` for Default Output Format.

3. Type the following command:

   ```
   aws discovery start-continuous-export
   ```

   Note
   Your agents are now running in "continuous export" mode which will enable you to see and work with your discovered data in Amazon Athena. The first time this is enable it may take up to 30 minutes for your data to appear in Amazon Athena.

Working with Data Exploration in Amazon Athena

After you enable Data Exploration in Amazon Athena, you can begin exploring and working with detailed current data that was discovered by your agents by querying the data directly in Athena. You can use the data to generate spreadsheets, run a cost analysis, port the query to a visualization program to diagram network dependencies, and more.

The topics in this section describe the ways that you can work with your data in Athena to assess and plan for migrating your local environment to AWS.

Topics

- Exploring Data Directly in Amazon Athena (p. 44)
- Visualizing Amazon Athena Data (p. 45)
- Predefined Queries to use in Athena (p. 45)

Exploring Data Directly in Amazon Athena

The following instructions explain how to explore your agent data directly in the Athena console. If you don’t have any data in Athena or have not enabled Data Exploration in Amazon Athena, you will be prompted by a dialog box to enable Data Exploration in Amazon Athena, as explained in Enabling Data Exploration in Amazon Athena (p. 43).

To explore agent-discovered data directly in Athena

1. In the AWS Migration Hub console, choose **Servers** in the navigation pane.
2. To open the Amazon Athena console, choose **Explore data in Amazon Athena**.
3. On the **Query Editor** page, in the navigation pane under **Database**, make sure that **application_discovery_service_database** is selected.

   Note
   Under **Tables** the following tables represent the datasets grouped by the agents.

   - **os_info_agent**
   - **network_interface_agent**
   - **sys_performance_agent**
   - **processes_agent**
   - **inbound_connection_agent**

- **outbound_connection_agent**
- **id_mapping_agent**

4. Query the data in the Amazon Athena console by writing and running SQL queries in the Athena Query Editor. For example, you can use the following query to see all of the discovered server IP addresses.

```
SELECT * FROM network_interface_agent;
```

For more example queries, see Predefined Queries to use in Athena (p. 45).

Visualizing Amazon Athena Data

To visualize your data, a query can be ported to a visualization program such as Amazon QuickSight or other open-source visualization tools such as Cytoscape, yEd, or Gelphi. Use these tools to render network diagrams, summary charts, and other graphical representations. When this method is used, you connect to Athena through the visualization program so that it can access your collected data as a source to produce the visualization.

To visualize your Amazon Athena data using Amazon QuickSight

1. Sign in to Amazon QuickSight.
2. Choose **Connect to another data source or upload a file**.
3. Choose **Athena**. The **New Athena data source** dialog box displays.
4. Enter a name in the **Data source name** field.
5. Choose **Create data source**.
6. Select the **Agents-servers-os** table in the **Choose your table** dialog box and choose **Select**.
7. In the **Finish dataset creation** dialog box, select **Import to SPICE for quicker analytics**, and choose **Visualize**.

   Your visualization is rendered.

Predefined Queries to use in Athena

This section contains a set of predefined queries that perform typical use cases, such as TCO analysis and network visualization. You can use these queries as is or modify them to suit your needs.

To use a predefined query

1. In the AWS Migration Hub console, choose **Servers** in the navigation pane.
2. To open the Amazon Athena console, choose **Explore data in Amazon Athena**.
3. On the **Query Editor** page, in the navigation pane under **Database**, make sure that **application_discovery_service_database** is selected.
4. Choose the plus **(+)** sign in the Query Editor to create a tab for a new query.
5. Copy one of the queries from Predefined Queries (p. 45).
6. Paste the query into the query pane of the new query tab you just created.
7. Choose **Run Query**.

Predefined Queries

Choose a title to see information about the query.

Obtain IP Addresses and Hostnames for Servers

This view helper function retrieves IP addresses and hostnames for a given server. You can use this view in other queries. For information about how to create a view, see CREATE VIEW in the *Amazon Athena User Guide*.

```
CREATE OR REPLACE VIEW hostname_ip_helper AS
SELECT DISTINCT
  "os"."host_name"
, "nic"."agent_id"
, "nic"."ip_address"
FROM
  os_info_agent os
, network_interface_agent nic
WHERE ("os"."agent_id" = "nic"."agent_id");
```

Identify Servers With or Without Agents

This query can help you perform data validation. If you've deployed agents on a number of servers in your network, you can use this query to understand if there are other servers in your network without agents deployed on them. In this query, we look into the inbound and outbound network traffic, and filter the traffic for private IP addresses only. That is, IP addresses starting with `192`, `10`, or `172`.

```
SELECT DISTINCT "destination_ip" "IP Address" ,
         (CASE
    WHEN (
    (SELECT "count"(*)
    FROM network_interface_agent
    WHERE ("ip_address" = "destination_ip") ) = 0) THEN
        'no'
        WHEN (
        (SELECT "count"(*)
        FROM network_interface_agent
        WHERE ("ip_address" = "destination_ip") ) > 0) THEN
            'yes' END) "agent_running"
    FROM outbound_connection_agent
WHERE ((("destination_ip" LIKE '192.%')
        OR ("destination_ip" LIKE '10.%'))
        OR ("destination_ip" LIKE '172.%'))
UNION
SELECT DISTINCT "source_ip" "IP ADDRESS" ,
         (CASE
    WHEN (
    (SELECT "count"(*)
    FROM network_interface_agent
    WHERE ("ip_address" = "source_ip") ) = 0) THEN
        'no'
        WHEN (
        (SELECT "count"(*)
        FROM network_interface_agent
        WHERE ("ip_address" = "source_ip") ) > 0) THEN
            'yes' END) "agent_running"
    FROM inbound_connection_agent
WHERE ((("source_ip" LIKE '192.%')
        OR ("source_ip" LIKE '10.%'))
        OR ("source_ip" LIKE '172.%'));
```

Analyze System Performance Data for Servers With Agents

You can use this query to analyze system performance and utilization pattern data for your on-premises servers that have agents installed on them. The query combines the `system_performance_agent` table with the `os_info_agent` table to identify the hostname for each server. This query returns the time series utilization data (in 15 minute intervals) for all the servers where agents are running.

```
SELECT "OS"."os_name" "OS Name" ,
    "OS"."os_version" "OS Version" ,
    "OS"."host_name" "Host Name" ,
     "SP"."agent_id" ,
     "SP"."total_num_cores" "Number of Cores" ,
     "SP"."total_num_cpus" "Number of CPU" ,
     "SP"."total_cpu_usage_pct" "CPU Percentage" ,
     "SP"."total_disk_size_in_gb" "Total Storage (GB)" ,
     "SP"."total_disk_free_size_in_gb" "Free Storage (GB)" ,
     ("SP"."total_disk_size_in_gb" - "SP"."total_disk_free_size_in_gb") "Used Storage" ,
     "SP"."total_ram_in_mb" "Total RAM (MB)" ,
     ("SP"."total_ram_in_mb" - "SP"."free_ram_in_mb") "Used RAM (MB)" ,
     "SP"."free_ram_in_mb" "Free RAM (MB)" ,
     "SP"."total_disk_read_ops_per_sec" "Disk Read IOPS" ,
     "SP"."total_disk_bytes_written_per_sec_in_kbps" "Disk Write IOPS" ,
     "SP"."total_network_bytes_read_per_sec_in_kbps" "Network Reads (kbps)" ,
     "SP"."total_network_bytes_written_per_sec_in_kbps" "Network Write (kbps)"
FROM "sys_performance_agent" "SP" , "OS_INFO_agent" "OS"
WHERE ("SP"."agent_id" = "OS"."agent_id") limit 10;
```

Track Outbound Communication Between Servers Based On Port Number and Process Details

This query gets the details on the outbound traffic for each service, along with the port number and process details.

Before running the query, if you have not already done so, you must create the `iana_service_ports_import` table that contains the IANA port registry database downloaded from IANA. For information about how to create this table, see Creating the IANA Port Registry Import Table (p. 50).

After the `iana_service_ports_import` table is created, create two view helper functions for tracking outbound traffic. For information about how to create a view, see CREATE VIEW in the *Amazon Athena User Guide*.

To create outbound tracking helper functions

1. Open the Athena console at https://console.aws.amazon.com/athena/.
2. Create the `valid_outbound_ips_helper` view, using the following helper function that lists all distinct outbound destination IP addresses.

```
CREATE OR REPLACE VIEW valid_outbound_ips_helper AS
SELECT DISTINCT "destination_ip"
FROM outbound_connection_agent;
```

3. Create the `outbound_query_helper` view, using the following helper function that determines the frequency of communication for outbound traffic.

```
CREATE OR REPLACE VIEW outbound_query_helper AS
SELECT "agent_id" ,
         "source_ip" ,
         "destination_ip" ,
         "destination_port" ,
         "agent_assigned_process_id" ,
         "count"(*) "frequency"
FROM outbound_connection_agent
WHERE (("ip_version" = 'IPv4')
        AND ("destination_ip" IN
    (SELECT *
    FROM valid_outbound_ips_helper )))
GROUP BY  "agent_id", "source_ip", "destination_ip", "destination_port",
 "agent_assigned_process_id";
```

4. After you create the `iana_service_ports_import` table and your two helper functions, you can run the following query to get the details on the outbound traffic for each service, along with the port number and process details.

```
SELECT hip1.host_name "Source Host Name",
         outbound_connections_results0.source_ip "Source IP Address",
         hip2.host_name "Destination Host Name",
         outbound_connections_results0.destination_ip "Destination IP Address",
         outbound_connections_results0.frequency "Connection Frequency",
         outbound_connections_results0.destination_port "Destination Communication
 Port",
         outbound_connections_results0.servicename "Process Service Name",
         outbound_connections_results0.description "Process Service Description"
FROM
    (SELECT DISTINCT o.source_ip,
         o.destination_ip,
         o.frequency,
         o.destination_port,
         ianap.servicename,
         ianap.description
    FROM outbound_query_helper o, iana_service_ports_import ianap
    WHERE o.destination_port = TRY_CAST(ianap.portnumber AS integer)) AS
 outbound_connections_results0 LEFT OUTER
JOIN hostname_ip_helper hip1
    ON outbound_connections_results0.source_ip = hip1.ip_address LEFT OUTER
JOIN hostname_ip_helper hip2
    ON outbound_connections_results0.destination_ip = hip2.ip_address
```

Track Inbound Communication Between Servers Based On Port Number and Process Details

This query gets information about inbound traffic for each service, along with the port number and process details.

Before running this query, if you have not already done so, you must create the `iana_service_ports_import` table that contains the IANA port registry database downloaded from IANA. For information about how to create this table, see Creating the IANA Port Registry Import Table (p. 50).

After the `iana_service_ports_import` table is created, create two view helper functions for tracking inbound traffic. For information about how to create a view, see CREATE VIEW in the *Amazon Athena User Guide*.

To create import tracking helper functions

1. Open the Athena console at https://console.aws.amazon.com/athena/.
2. Create the `valid_inbound_ips_helper` view, using the following helper function that lists all distinct inbound source IP addresses.

```
CREATE OR REPLACE VIEW valid_inbound_ips_helper AS
SELECT DISTINCT "source_ip"
FROM inbound_connection_agent;
```

3. Create the `inbound_query_helper` view, using the following helper function that determines the frequency of communication for inbound traffic.

```
CREATE OR REPLACE VIEW inbound_query_helper AS
```

```
SELECT "agent_id" ,
         "source_ip" ,
         "destination_ip" ,
         "destination_port" ,
         "agent_assigned_process_id" ,
         "count"(*) "frequency"
FROM inbound_connection_agent
WHERE (("ip_version" = 'IPv4')
        AND ("source_ip" IN
    (SELECT *
    FROM valid_inbound_ips_helper )))
GROUP BY  "agent_id", "source_ip", "destination_ip", "destination_port",
 "agent_assigned_process_id";
```

4. After you create the `iana_service_ports_import` table and your two helper functions, you can run the following query to get the details on the inbound traffic for each service, along with the port number and process details.

```
SELECT hip1.host_name "Source Host Name",
         inbound_connections_results0.source_ip "Source IP Address",
         hip2.host_name "Destination Host Name",
         inbound_connections_results0.destination_ip "Destination IP Address",
         inbound_connections_results0.frequency "Connection Frequency",
         inbound_connections_results0.destination_port "Destination Communication
 Port",
         inbound_connections_results0.servicename "Process Service Name",
         inbound_connections_results0.description "Process Service Description"
FROM
    (SELECT DISTINCT i.source_ip,
         i.destination_ip,
         i.frequency,
         i.destination_port,
         ianap.servicename,
         ianap.description
    FROM inbound_query_helper i, iana_service_ports_import ianap
    WHERE i.destination_port = TRY_CAST(ianap.portnumber AS integer)) AS
 inbound_connections_results0 LEFT OUTER
JOIN hostname_ip_helper hip1
    ON inbound_connections_results0.source_ip = hip1.ip_address LEFT OUTER
JOIN hostname_ip_helper hip2
    ON inbound_connections_results0.destination_ip = hip2.ip_address
```

Identify Running Software From Port Number

This query identifies the running software based on port numbers.

Before running this query, if you have not already done so, you must create the `iana_service_ports_import` table that contains the IANA port registry database downloaded from IANA. For information about how to create this table, see Creating the IANA Port Registry Import Table (p. 50).

Run the following query to identify the running software based on port numbers.

```
SELECT o.host_name "Host Name",
       ianap.servicename "Service",
       ianap.description "Description",
       con.destination_port,
       con.cnt_dest_port "Destination Port Count"
FROM   (SELECT agent_id,
               destination_ip,
               destination_port,
               Count(destination_port) cnt_dest_port
```

```
        FROM   inbound_connection_agent
        GROUP  BY agent_id,
                  destination_ip,
                  destination_port) con,
       (SELECT agent_id,
               host_name,
               Max("timestamp")
        FROM   os_info_agent
        GROUP  BY agent_id,
                  host_name) o,
       iana_service_ports_import ianap
WHERE  ianap.transportprotocol = 'tcp'
       AND con.destination_ip NOT LIKE '172%'
       AND con.destination_port = ianap.portnumber
       AND con.agent_id = o.agent_id
ORDER BY cnt_dest_port DESC;
```

Creating the IANA Port Registry Import Table

Some of the predefined queries require a table named `iana_service_ports_import` that contains information downloaded from Internet Assigned Numbers Authority (IANA).

To create the iana_service_ports_import table

1. Download the IANA port registry database **CSV** file from Service Name and Transport Protocol Port Number Registry on *iana.org*.
2. Upload the file to Amazon S3. For more information, see How Do I Upload Files and Folders to an S3 Bucket?.
3. Create a new table in Athena named `iana_service_ports_import`. For instructions, see Create a Table in the *Amazon Athena User Guide*. In the following example, you need to replace `my_bucket_name` with the name of the S3 bucket that you uploaded the CSV file to in the previous step.

```
CREATE EXTERNAL TABLE IF NOT EXISTS iana_service_ports_import (
         ServiceName STRING,
         PortNumber INT,
         TransportProtocol STRING,
         Description STRING,
         Assignee STRING,
         Contact STRING,
         RegistrationDate STRING,
         ModificationDate STRING,
         Reference STRING,
         ServiceCode STRING,
         UnauthorizedUseReported STRING,
         AssignmentNotes STRING
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'
WITH SERDEPROPERTIES (
  'serialization.format' = ',',
  'quoteChar' = '"',
  'field.delim' = ','
) LOCATION 's3://my_bucket_name/'
TBLPROPERTIES ('has_encrypted_data'='false',"skip.header.line.count"="1");
```

AWS Application Discovery Service Console Walkthroughs

AWS Application Discovery Service is integrated with AWS Migration Hub and customers can view and manage their data collectors, servers, and applications within Migration Hub. When you use the Application Discovery Service console, you are redirected to the Migration Hub console. Working with the Migration Hub console requires no extra steps or setup on your part.

In this section, you can find how to manage and monitor your Discovery Connectors and Discovery Agents using the console.

Topics

- Main Dashboard (p. 51)
- Data Collection Tools (p. 51)
- View, Export, and Explore Server Data (p. 53)

Main Dashboard

To view the the main dashboard, choose **Dashboard** from the AWS Migration Hub console navigation pane. In Migration Hub's main dashboard, you can view high-level statistics about servers, applications, and data collectors such as Discovery Connectors and Discovery Agents.

Topics

- Main Dashboard (p. 51)

Main Dashboard

The main dashboard gathers data from the **Discover** and **Migrate** dashboards in a central location. It has four status and information panes and a list of links for quick access. Using the panes, you can see a summary status of your most recently updated applications. You can also get quick access to any of your applications, get an overview of applications in different states, and track the migration progress over time.

To view the main dashboard, choose **Dashboard** from the navigation pane, which is on the left side of the Migration Hub console homepage.

Data Collection Tools

The Discovery Connector and Discovery Agent are the data collection tools that Application Discovery Service uses to help you discover your existing infrastructure. You can download and deploy discovery connectors and discovery agents as explained in AWS Agentless Discovery Connector (p. 20) and AWS Application Discovery Agent (p. 6).

These data collection tools store their data in the Application Discovery Service's repository, providing details about each server and the processes running on them. When either of these tools is deployed, you can start, stop, and view the collected data from the AWS Migration Hub console.

Topics

- Starting and Stopping Data Collectors (p. 52)
- Viewing and Sorting Data Collectors (p. 52)

Starting and Stopping Data Collectors

Whether you deployed a Discovery Connector or a Discovery Agent, you can start or stop their data collection process on the **Data Collectors** page of the AWS Migration Hub console.

To start or stop data collection tools

1. In the Migration Hub console navigation pane under **Discover**, choose **Data Collectors**.
2. Choose either the **Connectors** or **Agents** tab.
3. Select the check box of the collection tool you want to start or stop.
4. Choose **Start data collection** or **Stop data collection**.

Viewing and Sorting Data Collectors

If you deployed many data collectors, you can sort the Discovery Connectors or Discovery Agents that are returned to the **Data Collectors** page of the console. You can do this by applying filters in the search bar. You can search and filter on most of the criteria specified in the **Data Collectors** list.

The following table shows the search criteria that you can use, including operators, values, and a definition of the values.

Search Criterion	Operato	Value: Definition
Collection status	== !=	Started: Data is being collected and sent to Application Discovery S Start scheduled: Data collection is scheduled to start. Data will be s on next ping, and status will change to **Started**. Stopped: Data is not being collected or sent to Application Discove Stop scheduled: Data collection is scheduled to stop. Data will stop Service on next ping, and status will change to **Stopped**.
Health		Healthy: Data collection isn't turned on. The tool is functioning nor Unhealthy: The tool is in an error state. Data isn't being collected o Unknown: No connection established in over an hour. Shutdown: The tool last communicated "shutting down" due to a sy down. If a reboot or tool upgrade occurred, status will change to ar cycle. Running: Data collection is turned on. The tool is functioning norm
Hostname		For agents, any host name selected from the pre-populated list of l For connectors, not applicable.
IP address		Any IP address selected from the pre-populated list where a collect
Connector/Agent ID	==	Any connector or agent ID selected from the pre-populated list fro

To sort data collectors by applying search filters

1. In the Migration Hub console navigation pane under **Discover**, choose **Data Collectors**.
2. Choose either the **Connectors** or **Agents** tab.
3. Click inside the search bar and choose a search criterion from the list.
4. Choose an operator from the next list.
5. Choose a value from the last list.

View, Export, and Explore Server Data

The **Servers** page provides system configuration and performance data about each server instance known to the data collection tools. You can view server information, sort servers with filters, tag servers with key-value pairs, and export detailed server and system information.

Topics

- Viewing and Sorting Servers (p. 53)
- Tagging Servers (p. 54)
- Exporting Server Data (p. 54)
- Data Exploration in Athena (p. 55)
- Applications (p. 55)

Viewing and Sorting Servers

You can view information about the servers discovered by the data collection tools, and you can sort through the servers using filters.

Viewing Servers

You can get a general view and a detailed view of the servers discovered by the data collection tools.

To view discovered servers

1. In the Migration Hub console navigation pane under **Discover**, choose **Servers**. The discovered servers appear in the servers list.
2. For more detail about a server, choose its server link in the **Server info** column. Doing so displays a screen that describes the server.

The server's detail screen displays system information and performance metrics. You can also find a button to export network dependencies and processes information. To export detailed server information, see Exporting Server Data (p. 54).

Sorting Servers with Search Filters

To easily find specific servers, apply search filters to sort through all the servers discovered by the collection tools. You can search and filter on numerous criteria.

To sort servers by applying search filters

1. In the Migration Hub console navigation pane under **Discover**, choose **Servers**.
2. Click inside the search bar, and choose a search criterion from the list.

3. Choose an operator from the next list.
4. Type in a case-sensitive value for the search criterion you selected, and press Enter.
5. Multiple filters can be applied by repeating steps 2 - 4.

Tagging Servers

To assist migration planning and help stay organized, you can create multiple tags for each server. *Tags* are user-defined key-value pairs that can store any custom data or metadata about servers. You can tag an individual server or multiple servers in a single operation. Application Discovery Service tags are similar to AWS tags, but the two types of tag cannot be used interchangeably.

You can add or remove multiple tags for one or more servers from the main **Servers** page. On a server's detail page, you can add or remove one or more tags for the selected server. You can do any type of tagging task involving multiple servers or tags in a single operation. You can also remove tags.

To add tags to one or more servers

1. In the Migration Hub console navigation pane under **Discover**, choose **Servers**.
2. In the **Server info** column, choose the server link for the server that you want to add tags for. To add tags to more than one server at a time, click inside the check boxes of multiple servers.
3. Choose **Add tags**, and then choose **Add new tag**.
4. In the dialog box, type a key in the **Key** field, and optionally a value in the **Value** field.

   Add more tags by choosing **Add new tag** and adding more information.
5. Choose **Save**.

To remove tags from one or more servers

1. In the Migration Hub console navigation pane under **Discover**, choose **Servers**.
2. In the **Server info** column, choose the server link for the server that you want to remove tags from. Select the check boxes of multiple servers to remove tags from more than one server at a time.
3. Choose **Remove tags**.
4. Select each tag that you want to remove.
5. Choose **Confirm**.

Exporting Server Data

To export network dependencies and process information for one server at a time, you can use a server's detail screen. You can find the export jobs for a server in a table located in the **Exports** section of the server's detail screen. If no export jobs yet exist, the table is empty. You can simultaneously export up to five collections of data.

> **Note**
> Exporting server data from the console is only available for data collected by an agent running on that server. If you want to download data collected by a connector, see Export data collected for all servers (p. 41). Or, if you want to bulk export data for all servers where agents have been installed, see Data Exploration in Amazon Athena (p. 42).

To export detailed server data

1. In the Migration Hub console navigation pane under **Discover**, choose **Servers**.
2. In the **Server info** column, choose the ID of the server for which you want to export data.

3. In the **Exports** section at the bottom of the screen, choose **Export server details**.
4. For **Export server details**, fill in **Start date** and **Time**.

   Note
   The start time can't be more than 72 hours before the current time.
5. Choose **Export** to start the job. The initial status is **In-progress**; to update the status, click the refresh icon for the **Exports** section.
6. When the export job is complete, choose **Download** and save the .zip file.
7. Unzip the saved file. A set of .csv files contains the export data, similar to the following:

   - *<AWS account ID>*_destinationProcessConnection.csv
   - *<AWS account ID>*_networkInterface.csv
   - *<AWS account ID>*_osInfo.csv
   - *<AWS account ID>*_process.csv
   - *<AWS account ID>*_sourceProcessConnection.csv
   - *<AWS account ID>*_systemPerformance.csv

   You can open the .csv files in Microsoft Excel and review the exported server data.

   Among the files, you can find a JSON file containing data about the export task and its results.

Data Exploration in Athena

Data Exploration in Amazon Athena allows you to analyze the data collected from all the discovered on-premises servers by Discovery Agents in one place. Once Data Exploration in Amazon Athena is enabled from the Migration Hub console (or by using the StartContinousExport API) and the data collection for agents is turned on, data collected by agents will automatically get stored in your S3 bucket at regular intervals. For more information, see Data Exploration in Amazon Athena (p. 42).

Applications

Some of your discovered servers might need to be migrated together to remain functional. In this case, you can logically define and group discovered servers into applications.

As part of the grouping process, you can search, filter, and add tags.

To group servers into a new or existing application

1. In the Migration Hub console navigation pane under **Discover**, choose **Servers**.
2. In the servers list, select each server that you want to group into a new or existing application.

   To help choose servers for your group, you can search and filter on any criteria that you specify in the server list. Click inside the search bar and choose an item from the list, choose an operator from the next list, and then type in your criteria.
3. Optional: For each selected server, choose **Add tag**, type a value for **Key**, and then optionally type a value for **Value**.
4. Choose **Group as application** to create your application, or add to an existing one.
5. In the **Group as application** dialog box, choose **Group as a new application** or **Add to an existing application**.

   a. If you chose **Group as a new application**, type a name for **Application name**. Optionally, you can type a description for **Application description**.

   b. If you chose **Add to an existing application**, select the name of the application to add to in the list.

6. Choose **Save**.

Querying Discovered Configuration Items

A *configuration item* is an IT asset that was discovered in your data center by an agent or the connector. When you use Application Discovery Service, you can specify filters and query specific configuration items for server, application, process, and connection assets.

The tables in the following sections list the available input filters and output sorting options for two Application Discovery Service actions:

- `DescribeConfigurations`
- `ListConfigurations`

The filtering and sorting options are organized by the type of asset to which apply (server, application, process, or connection).

Using the `DescribeConfigurations`Action

The `DescribeConfigurations`action retrieves attributes for a list of configuration IDs. All the supplied IDs must be for the same asset type (server, application, process, or connection). Output fields are specific to the asset type selected. For example, the output for a server configuration item includes a list of attributes about the server, such as host name, operating system, and number of network cards. For more information about command syntax, see DescribeConfigurations.

The `DescribeConfigurations`action does not support filtering.

Output fields for `DescribeConfigurations`

The following tables, organized by asset type, list the supported output fields of the `DescribeConfigurations`action. The ones marked as mandatory are always present in the output.

Server assets

Field	Mandatory
`server.agentId`	
`server.applications`	
`server.applications.hasMoreValues`	
`server.configurationId`	x
`server.cpuType`	
`server.hostName`	
`server.hypervisor`	
`server.networkInterfaceInfo`	
`server.networkInterfaceInfo.hasMoreValues`	

Field	Mandatory
`server.osName`	
`server.osVersion`	
`server.tags`	
`server.tags.hasMoreValues`	
`server.timeOfCreation`	x
`server.type`	
`server.performance.avgCpuUsagePct`	
`server.performance.avgDiskReadIOPS`	
`server.performance.avgDiskReadsPerSecondInKB`	
`server.performance.avgDiskWriteIOPS`	
`server.performance.avgDiskWritesPerSecondInKB`	
`server.performance.avgFreeRAMInKB`	
`server.performance.avgNetworkReadsPerSecondInKB`	
`server.performance.avgNetworkWritesPerSecondInKB`	
`server.performance.maxCpuUsagePct`	
`server.performance.maxDiskReadIOPS`	
`server.performance.maxDiskReadsPerSecondInKB`	
`server.performance.maxDiskWriteIOPS`	
`server.performance.maxDiskWritesPerSecondInKB`	
`server.performance.maxNetworkReadsPerSecondInKB`	
`server.performance.maxNetworkWritesPerSecondInKB`	
`server.performance.minFreeRAMInKB`	
`server.performance.numCores`	
`server.performance.numCpus`	
`server.performance.numDisks`	
`server.performance.numNetworkCards`	
`server.performance.totalRAMInKB`	

Process assets

Field	Mandatory
`process.commandLine`	

Field	Mandatory
process.configurationId	x
process.name	
process.path	
process.timeOfCreation	x

Application assets

Field	Mandatory
application.configurationId	x
application.description	
application.lastModifiedTime	x
application.name	x
application.serverCount	x
application.timeOfCreation	x

Using the ListConfigurationsAction

The ListConfigurationsaction retrieves a list of configuration items according to the criteria that you specify in a filter. For more information about command syntax, see ListConfigurations.

Output fields for ListConfigurations

The following tables, organized by asset type, list the supported output fields of the ListConfigurationsaction. The ones marked as mandatory are always present in the output.

Server assets

Field	Mandatory
server.configurationId	x
server.agentId	
server.hostName	
server.osName	
server.osVersion	
server.timeOfCreation	x
server.type	

Process assets

Field	Mandatory
process.commandLine	
process.configurationId	x
process.name	
process.path	
process.timeOfCreation	x
server.agentId	
server.configurationId	x

Application assets

Field	Mandatory
application.configurationId	x
application.description	
application.name	x
application.serverCount	x
application.timeOfCreation	x
application.lastModifiedTime	x

Connection assets

Field	Mandatory
connection.destinationIp	x
connection.destinationPort	x
connection.ipVersion	x
connection.latestTimestamp	x
connection.occurrence	x
connection.sourceIp	x
connection.transportProtocol	
destinationProcess.configurationId	
destinationProcess.name	
destinationServer.configurationId	
destinationServer.hostName	

Field	Mandatory
`sourceProcess.configurationId`	
`sourceProcess.name`	
`sourceServer.configurationId`	
`sourceServer.hostName`	

Supported filters for `ListConfigurations`

The following tables, organized by asset type, list the supported filters for the `ListConfigurations`action. Filters and values are in a key/value relationship defined by one of the supported logical conditions. You can sort the output of the indicated filters.

Server assets

Filter	Supported conditions	Supported values	Supported sorting
`server.configurationId`	• EQUALS • NOT_EQUALS • EQ • NE	• Any valid server configuration ID	None
`server.hostName`	• EQUALS • NOT_EQUALS • EQ • NE • CONTAINS • NOT_CONTAINS	• String	• ASC • DESC
`server.osName`	• EQUALS • NOT_EQUALS • EQ • NE • CONTAINS • NOT_CONTAINS	• String	• ASC • DESC
`server.osVersion`	• EQUALS • NOT_EQUALS • EQ • NE • CONTAINS • NOT_CONTAINS	• String	• ASC • DESC
`server.agentId`	• EQUALS • NOT_EQUALS • EQ • NE	• String	None
`server.connectorId`	• EQUALS	• String	None

Filter	Supported conditions	Supported values	Supported sorting
	• NOT_EQUALS • EQ • NE		
`server.type`	• EQUALS • NOT_EQUALS • EQ • NE	String with one of the following values: • EC2 • OTHER • VMWARE_VM • VMWARE_HOST • VMWARE_VM_TEMPLATE	None
`server.vmWareInfo.moRefId`	• EQUALS • NOT_EQUALS • EQ • NE • CONTAINS • NOT_CONTAINS	• String	None
`server.vmWareInfo.vcenterId`	• EQUALS • NOT_EQUALS • EQ • NE • CONTAINS • NOT_CONTAINS	• String	None
`server.vmWareInfo.hostId`	• EQUALS • NOT_EQUALS • EQ • NE • CONTAINS • NOT_CONTAINS	• String	None
`server.networkInterfaceInfo.portGroupId`	• EQUALS • NOT_EQUALS • EQ • NE • CONTAINS • NOT_CONTAINS	• String	None
`server.networkInterfaceInfo.portGroupName`	• EQUALS • NOT_EQUALS • EQ • NE • CONTAINS • NOT_CONTAINS	• String	None

Filter	Supported conditions	Supported values	Supported sorting
`server.networkInterfaceInfo.virtualSwitchName`	• EQUALS • NOT_EQUALS • EQ • NE • CONTAINS • NOT_CONTAINS	• String	None
`server.networkInterfaceInfo.ipAddress`	• EQUALS • NOT_EQUALS • EQ • NE • CONTAINS • NOT_CONTAINS	• String	None
`server.networkInterfaceInfo.macAddress`	• EQUALS • NOT_EQUALS • EQ • NE • CONTAINS • NOT_CONTAINS	• String	None
`server.performance.avgCpuUsagePct`	• GE • LE • GT • LT	• Percentage	None
`server.performance.totalDiskFreeSizeInKB`	• GE • LE • GT • LT	• Double	None
`server.performance.avgFreeRAMInKB`	• GE • LE • GT • LT	• Double	None
`server.tag.value`	• EQUALS • NOT_EQUALS • EQ • NE • CONTAINS • NOT_CONTAINS	• String	None

Filter	Supported conditions	Supported values	Supported sorting
server.tag.key	• EQUALS • NOT_EQUALS • EQ • NE • CONTAINS • NOT_CONTAINS	• String	None
server.application.name	• EQUALS • NOT_EQUALS • EQ • NE • CONTAINS • NOT_CONTAINS	• String	None
server.application.description	• EQUALS • NOT_EQUALS • EQ • NE • CONTAINS • NOT_CONTAINS	• String	None
server.application.configurationId	• EQUALS • NOT_EQUALS • EQ • NE	• Any valid application configuration ID	None
server.process.configurationId	• EQUALS • NOT_EQUALS • EQ • NE	• ProcessId	None
server.process.name	• EQUALS • NOT_EQUALS • EQ • NE • CONTAINS • NOT_CONTAINS	• String	None
server.process.commandLine	• EQUALS • NOT_EQUALS • EQ • NE • CONTAINS • NOT_CONTAINS	• String	None

Application assets

Filter	Supported conditions	Supported values	Supported sorting
`application.configurationId`	• EQUALS • NOT_EQUALS • EQ • NE	• ApplicationId	None
`application.name`	• EQUALS • NOT_EQUALS • EQ • NE • CONTAINS • NOT_CONTAINS	• String	• ASC • DESC
`application.description`	• EQUALS • NOT_EQUALS • EQ • NE • CONTAINS • NOT_CONTAINS	• String	• ASC • DESC
`application.serverCount`	Filtering not supported.	Filtering not supported.	• ASC • DESC
`application.timeOfCreation`	Filtering not supported.	Filtering not supported.	• ASC • DESC
`application.lastModifiedTime`	Filtering not supported.	Filtering not supported	• ASC • DESC
`server.configurationId`	• EQUALS • NOT_EQUALS • EQ • NE	• ServerId	None

Process assets

Filter	Supported conditions	Supported values	Supported sorting
`process.configurationId`	• EQUALS • NOT_EQUALS • EQ • NE	• ProcessId	

Filter	Supported conditions	Supported values	Supported sorting
`process.name`	• EQUALS • NOT_EQUALS • EQ • NE • CONTAINS • NOT_CONTAINS	• String	• ASC • DESC
`process.commandLine`	• EQUALS • NOT_EQUALS • EQ • NE • CONTAINS • NOT_CONTAINS	• String	• ASC • DESC
`server.configurationId`	• EQUALS • NOT_EQUALS • EQ • NE	• ServerId	
`server.hostName`	• EQUALS • NOT_EQUALS • EQ • NE • CONTAINS • NOT_CONTAINS	• String	• ASC • DESC
`server.osName`	• EQUALS • NOT_EQUALS • EQ • NE • CONTAINS • NOT_CONTAINS	• String	• ASC • DESC
`server.osVersion`	• EQUALS • NOT_EQUALS • EQ • NE • CONTAINS • NOT_CONTAINS	• String	• ASC • DESC
`server.agentId`	• EQUALS • NOT_EQUALS • EQ • NE • CONTAINS • NOT_CONTAINS	• String	

Connection assets

Filter	Supported conditions	Supported values	Supported sorting
`connection.sourceIp`	- EQUALS - NOT_EQUALS - EQ - NE - CONTAINS - NOT_CONTAINS	- IP	- ASC - DESC
`connection.destinationIp`	- EQUALS - NOT_EQUALS - EQ - NE - CONTAINS - NOT_CONTAINS	- IP	- ASC - DESC
`connection.destinationPort`	- EQUALS - NOT_EQUALS - EQ - NE	- Integer	- ASC - DESC
`sourceServer.configurationId`	- EQUALS - NOT_EQUALS - EQ - NE	- ServerId	
`sourceServer.hostName`	- EQUALS - NOT_EQUALS - EQ - NE - CONTAINS - NOT_CONTAINS	- String	- ASC - DESC
`destinationServer.osName`	- EQUALS - NOT_EQUALS - EQ - NE - CONTAINS - NOT_CONTAINS	- String	- ASC - DESC
`destinationServer.osVersion`	- EQUALS - NOT_EQUALS - EQ - NE - CONTAINS - NOT_CONTAINS	- String	- ASC - DESC
`destinationServer.agentId`	- EQUALS	- String	

Filter	Supported conditions	Supported values	Supported sorting
	- NOT_EQUALS - EQ - NE - CONTAINS - NOT_CONTAINS		
sourceProcess.configurationId	- EQUALS - NOT_EQUALS - EQ - NE	- ProcessId	
sourceProcess.name	- EQUALS - NOT_EQUALS - EQ - NE - CONTAINS - NOT_CONTAINS	- String	- ASC - DESC
sourceProcess.commandLine	- EQUALS - NOT_EQUALS - EQ - NE - CONTAINS - NOT_CONTAINS	- String	- ASC - DESC
destinationProcess.configurationId	- EQUALS - NOT_EQUALS - EQ - NE	- ProcessId	
destinationProcess.name	- EQUALS - NOT_EQUALS - EQ - NE - CONTAINS - NOT_CONTAINS	- String	- ASC - DESC
destinationprocess.commandLine	- EQUALS - NOT_EQUALS - EQ - NE - CONTAINS - NOT_CONTAINS	- String	- ASC - DESC

Security in AWS Application Discovery Service

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that are 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. The effectiveness of our security is regularly tested and verified by third-party auditors as part of the AWS compliance programs.
- **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 organization's requirements, and applicable laws and regulations.

This documentation will help you understand how to apply the shared responsibility model when using Application Discovery Service. The following topics show you how to configure Application Discovery Service to meet your security and compliance objectives. You'll also learn how to use other AWS services that can help you to monitor and secure your Application Discovery Service resources.

Topics

- Identity and Access Management for AWS Application Discovery Service (p. 69)
- Logging and monitoring in AWS Application Discovery Service (p. 88)

Identity and Access Management for AWS Application Discovery Service

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use Application Discovery Service resources. IAM is an AWS service that you can use with no additional charge.

Topics

- Audience (p. 70)
- Authenticating With Identities (p. 70)
- Managing Access Using Policies (p. 72)
- How AWS Application Discovery Service Works with IAM (p. 73)
- AWS Managed (Predefined) Policies for Application Discovery Service (p. 75)
- AWS Application Discovery Service Identity-Based Policy Examples (p. 77)
- Using Service-Linked Roles for Application Discovery Service (p. 82)
- Troubleshooting AWS Application Discovery Service Identity and Access (p. 87)

Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work that you do in Application Discovery Service.

Service user – If you use the Application Discovery Service service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more Application Discovery Service features to do your work, you might need additional permissions. Understanding how access is managed can help you request the right permissions from your administrator. If you cannot access a feature in Application Discovery Service, see Troubleshooting AWS Application Discovery Service Identity and Access (p. 87).

Service administrator – If you're in charge of Application Discovery Service resources at your company, you probably have full access to Application Discovery Service. It's your job to determine which Application Discovery Service features and resources your employees should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with Application Discovery Service, see How AWS Application Discovery Service Works with IAM (p. 73).

IAM administrator – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to Application Discovery Service. To view example Application Discovery Service identity-based policies that you can use in IAM, see AWS Application Discovery Service Identity-Based Policy Examples (p. 77).

Authenticating With Identities

Authentication is how you sign in to AWS using your identity credentials. For more information about signing in using the AWS Management Console, see Signing in to the AWS Management Console as an IAM user or root user in the *IAM User Guide*.

You must be *authenticated* (signed in to AWS) as the AWS account root user, an IAM user, or by assuming an IAM role. You can also use your company's single sign-on authentication or even sign in using Google or Facebook. In these cases, your administrator previously set up identity federation using IAM roles. When you access AWS using credentials from another company, you are assuming a role indirectly.

To sign in directly to the AWS Management Console, use your password with your root user email address or your IAM user name. You can access AWS programmatically using your root user or IAM users access keys. AWS provides SDK and command line tools to cryptographically sign your request using your credentials. If you don't use AWS tools, you must sign the request yourself. Do this using *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*.

Regardless of the authentication method that you use, you might also be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see Using multi-factor authentication (MFA) in AWS in the *IAM User Guide*.

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 Users and Groups

An *IAM user* is an identity within your AWS account that has specific permissions for a single person or application. An IAM user can have long-term credentials such as a user name and password or a set of access keys. To learn how to generate access keys, see Managing access keys for IAM users in the *IAM User Guide*. When you generate access keys for an IAM user, make sure you view and securely save the key pair. You cannot recover the secret access key in the future. Instead, you must generate a new access key pair.

An *IAM group* is an identity that specifies a collection of IAM users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier to manage for large sets of users. For example, you could have a group named *IAMAdmins* and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. To learn more, see When to create an IAM user (instead of a role) in the *IAM User Guide*.

IAM Roles

An *IAM role* is an identity within your AWS account that has specific permissions. It is similar to an IAM user, but is not associated with a specific person. You can temporarily assume an IAM role in the AWS Management Console by switching roles. You can assume a role by calling an AWS CLI or AWS API operation or by using a custom URL. For more information about methods for using roles, see Using IAM roles in the *IAM User Guide*.

IAM roles with temporary credentials are useful in the following situations:

- **Temporary IAM user permissions** – An IAM user can assume an IAM role to temporarily take on different permissions for a specific task.
- **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*.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see How IAM roles differ from resource-based policies in the *IAM User Guide*.
- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.
  - **Principal permissions** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. Policies grant permissions to a principal. When you use some services, you might perform an action that then triggers another action in a different service. In this case, you must have permissions to perform both actions. To see whether an action requires additional dependent actions in a policy, see Actions, Resources, and Condition Keys for AWS Application Discovery Service in the *Service Authorization Reference*.
  - **Service role** – 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*.
  - **Service-linked role** – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear

in your IAM account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

- **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*.

To learn whether to use IAM roles or IAM users, see When to create an IAM role (instead of a user) in the *IAM User Guide*.

Managing Access Using Policies

You control access in AWS by creating policies and attaching them to IAM identities or AWS resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. You can sign in as the root user or an IAM user, or you can assume an IAM role. When you then make a request, AWS evaluates the related identity-based or resource-based policies. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

Every IAM entity (user or role) starts with no permissions. In other words, by default, users can do nothing, not even change their own password. To give a user permission to do something, an administrator must attach a permissions policy to a user. Or the administrator can add the user to a group that has the intended permissions. When an administrator gives permissions to a group, all users in that group are granted those permissions.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, suppose that you have a policy that allows the `iam:GetRole` action. A user with that policy can get role information from the AWS Management Console, the AWS CLI, or the AWS API.

Identity-Based Policies

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Creating IAM policies in the *IAM User Guide*.

Identity-based policies can be further categorized as *inline policies* or *managed policies*. Inline policies are embedded directly into a single user, group, or role. Managed policies are standalone policies that you can attach to multiple users, groups, and roles in your AWS account. Managed policies include AWS managed policies and customer managed policies. To learn how to choose between a managed policy or an inline policy, see Choosing between managed policies and inline policies in the *IAM User Guide*.

Resource-Based Policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

Access Control Lists (ACLs)

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

Amazon S3, AWS WAF, and Amazon VPC are examples of services that support ACLs. To learn more about ACLs, see Access control list (ACL) overview in the *Amazon Simple Storage Service Developer Guide*.

Other Policy Types

AWS supports additional, less-common policy types. These policy types can set the maximum permissions granted to you by the more common policy types.

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the `Principal` field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see Permissions boundaries for IAM entities in the *IAM User Guide*.
- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see How SCPs work in the *AWS Organizations User Guide*.
- **Session policies** – Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. For more information, see Session policies in the *IAM User Guide*.

Multiple Policy Types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see Policy evaluation logic in the *IAM User Guide*.

How AWS Application Discovery Service Works with IAM

Before you use IAM to manage access to Application Discovery Service, you should understand what IAM features are available to use with Application Discovery Service. To get a high-level view of how Application Discovery Service and other AWS services work with IAM, see AWS Services That Work with IAM in the *IAM User Guide*.

Topics

- Application Discovery Service Identity-Based Policies (p. 74)
- Application Discovery Service Resource-Based Policies (p. 75)
- Authorization Based on Application Discovery Service Tags (p. 75)
- Application Discovery Service IAM Roles (p. 75)

Application Discovery Service Identity-Based Policies

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. Application Discovery Service supports specific actions, resources, and condition keys. To learn about all of the elements that you use in a JSON policy, see IAM JSON Policy Elements Reference in the *IAM User Guide*.

Actions

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Policy actions usually have the same name as the associated AWS API operation. There are some exceptions, such as *permission-only actions* that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called *dependent actions*.

Include actions in a policy to grant permissions to perform the associated operation.

Policy actions in Application Discovery Service use the following prefix before the action: `discovery:`. Policy statements must include either an `Action` or `NotAction` element. Application Discovery Service defines its own set of actions that describe tasks that you can perform with this service.

To specify multiple actions in a single statement, separate them with commas as follows:

```
"Action": [
      "discovery:action1",
      "discovery:action2"
```

You can specify multiple actions using wildcards (*). For example, to specify all actions that begin with the word `Describe`, include the following action:

```
"Action": "discovery:Describe*"
```

To see a list of Application Discovery Service actions, see Actions Defined by AWS Application Discovery Service in the *IAM User Guide*.

Resources

Application Discovery Service does not support specifying resource ARNs in a policy.

Condition Keys

Application Discovery Service does not provide any service-specific condition keys, but it does support using some global condition keys. To see all AWS global condition keys, see AWS Global Condition Context Keys in the *IAM User Guide*.

Examples

To view examples of Application Discovery Service identity-based policies, see AWS Application Discovery Service Identity-Based Policy Examples (p. 77).

Application Discovery Service Resource-Based Policies

Application Discovery Service does not support resource-based policies.

Authorization Based on Application Discovery Service Tags

Application Discovery Service does not support tagging resources or controlling access based on tags.

Application Discovery Service IAM Roles

An IAM role is an entity within your AWS account that has specific permissions.

Using Temporary Credentials with Application Discovery Service

Application Discovery Service does not support using temporary credentials.

Service-Linked Roles

Service-linked roles allow AWS services to access resources in other services to complete an action on your behalf. Service-linked roles appear in your IAM account and are owned by the service. An IAM administrator can view but not edit the permissions for service-linked roles.

Application Discovery Service supports service-linked roles. For details about creating or managing Application Discovery Service service-linked roles, see Using Service-Linked Roles for Application Discovery Service (p. 82).

Service Roles

This feature allows a service to assume a service role on your behalf. This role allows the service to access resources in other services to complete an action on your behalf. Service roles appear in your IAM account and are owned by the account. This means that an IAM administrator can change the permissions for this role. However, doing so might break the functionality of the service.

Application Discovery Service supports service roles.

AWS Managed (Predefined) Policies for Application Discovery Service

An *AWS managed policy* is an AWS Identity and Access Management (IAM) policy that is created and administered by AWS. Managed policies are predefined to provide permissions for common use cases. For more information about managed policies, see Managed Policies and Inline Policies in the *IAM User Guide*.

The AWS managed policies listed in this topic are used to control access to AWS Application Discovery Service. An administrator AWS account by default inherits all the policies required for accessing Application Discovery Service.

If your account is a non-administrative account, to access Application Discovery Service, you need to request that your administrator add one or more of the following managed policies to your IAM user account. For information about how to attach managed policies to an account, see Creating an IAM Non-Administrative User (p. 5).

AWSApplicationDiscoveryServiceFullAccess

The `AWSApplicationDiscoveryServiceFullAccess` policy grants an IAM user account access to the Application Discovery Service and Migration Hub APIs.

An IAM user account with this policy attached can configure Application Discovery Service, start and stop agents, start and stop agentless discovery, and query data from the AWS Discovery Service database. For an example of this policy, see Granting Full Access to Application Discovery Service (p. 78).

AWSApplicationDiscoveryAgentAccess

The `AWSApplicationDiscoveryAgentAccess` policy grants the Application Discovery Agent access to register and communicate with Application Discovery Service.

You attach this policy to any user whose credentials are used by Application Discovery Agent.

This policy also grants the user access to Arsenal. Arsenal is an agent service that is managed and hosted by AWS. Arsenal forwards data to Application Discovery Service in the cloud. For an example of this policy, see Granting Access to Discovery Agents (p. 78).

AWSAgentlessDiscoveryService

The `AWSAgentlessDiscoveryService` policy grants the AWS Agentless Discovery Connector that is running in your VMware vCenter Server access to register, communicate with, and share connector health metrics with Application Discovery Service.

You attach this policy to any user whose credentials are used by the connector.

For an example of this policy, see Granting AWS Agentless Discovery Connector Access (p. 79).

ApplicationDiscoveryServiceContinuousExportServiceRolePolicy

If your IAM account has the `AWSApplicationDiscoveryServiceFullAccess` policy attached, `ApplicationDiscoveryServiceContinuousExportServiceRolePolicy` is automatically attached to your account when you turn on Data Exploration in Amazon Athena.

This policy allows AWS Application Discovery Service to create Amazon Kinesis Data Firehose streams to transform and deliver data collected by AWS Application Discovery Service agents to an Amazon S3 bucket in your AWS account.

In addition, this policy creates an AWS Glue Data Catalog with a new database called *application_discovery_service_database* and table schemas for mapping data collected by the agents. For an example of this policy, see Granting permissions for Agent Data Collection (p. 80).

AWSDiscoveryContinuousExportFirehosePolicy

The `AWSDiscoveryContinuousExportFirehosePolicy` policy is required to use Data Exploration in Amazon Athena. It allows Amazon Kinesis Data Firehose to write data collected from Application Discovery Service to Amazon S3. For information about using this policy, see Creating the AWSApplicationDiscoveryServiceFirehose Role (p. 76). For an example of this policy, see Granting Permissions for Data Collection (p. 81).

Creating the AWSApplicationDiscoveryServiceFirehose Role

An administrator attaches managed policies to your IAM user account. When using the `AWSDiscoveryContinuousExportFirehosePolicy` policy, the administrator must first create a role named **AWSApplicationDiscoveryServiceFirehose** with Kinesis Data Firehose as a trusted entity and then attach the `AWSDiscoveryContinuousExportFirehosePolicy` policy to the role, as shown in the following procedure.

To create the AWSApplicationDiscoveryServiceFirehose IAM role

1. In the IAM console, choose **Roles** on the navigation pane.
2. Choose **Create Role**.
3. Choose **Kinesis**.
4. Choose **Kinesis Firehose** as your use case.
5. Choose **Next: Permissions**.
6. Under **Filter Policies** search for **AWSDiscoveryContinuousExportFirehosePolicy**.
7. Select the box beside **AWSDiscoveryContinuousExportFirehosePolicy**, and then choose **Next: Review**.
8. Enter **AWSApplicationDiscoveryServiceFirehose** as the role name, and then choose **Create role**.

AWS Application Discovery Service Identity-Based Policy Examples

By default, IAM users and roles don't have permission to create or modify Application Discovery Service resources. They also can't perform tasks using the AWS Management Console, AWS CLI, or AWS API. An IAM administrator must create IAM policies that grant users and roles permission to perform specific API operations on the specified resources they need. The administrator must then attach those policies to the IAM users or groups that require those permissions.

To learn how to create an IAM identity-based policy using these example JSON policy documents, see Creating Policies on the JSON Tab in the *IAM User Guide*.

Topics

- Policy Best Practices (p. 77)
- Granting Full Access to Application Discovery Service (p. 78)
- Granting Access to Discovery Agents (p. 78)
- Granting AWS Agentless Discovery Connector Access (p. 79)
- Granting permissions for Agent Data Collection (p. 80)
- Granting Permissions for Data Collection (p. 81)

Policy Best Practices

Identity-based policies are very powerful. They determine whether someone can create, access, or delete Application Discovery Service resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started using AWS managed policies** – To start using Application Discovery Service quickly, use AWS managed policies to give your employees the permissions they need. These policies are already available in your account and are maintained and updated by AWS. For more information, see Get started using permissions with AWS managed policies in the *IAM User Guide*.
- **Grant least privilege** – When you create custom policies, grant only the permissions required to perform a task. Start with a minimum set of permissions and grant additional permissions as necessary. Doing so is more secure than starting with permissions that are too lenient and then trying to tighten them later. For more information, see Grant least privilege in the *IAM User Guide*.

- **Enable MFA for sensitive operations** – For extra security, require IAM users to use multi-factor authentication (MFA) to access sensitive resources or API operations. For more information, see Using multi-factor authentication (MFA) in AWS in the *IAM User Guide*.
- **Use policy conditions for extra security** – To the extent that it's practical, define the conditions under which your identity-based policies allow access to a resource. For example, you can write conditions to specify a range of allowable IP addresses that a request must come from. You can also write conditions to allow requests only within a specified date or time range, or to require the use of SSL or MFA. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.

Granting Full Access to Application Discovery Service

The AWSApplicationDiscoveryServiceFullAccess managed policy grants the IAM user account access to the Application Discovery Service and Migration Hub APIs.

An IAM user with this policy attached to their account can configure Application Discovery Service, start and stop agents, start and stop agentless discovery, and query data from the AWS Discovery Service database. For more information about this policy, see AWS Managed (Predefined) Policies for Application Discovery Service (p. 75).

Example AWSApplicationDiscoveryServiceFullAccess Policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "mgh:*",
                "discovery:*"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "iam:GetRole"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Granting Access to Discovery Agents

The AWSApplicationDiscoveryAgentAccess managed policy grants the Application Discovery Agent access to register and communicate with Application Discovery Service. For more information about this policy, see AWS Managed (Predefined) Policies for Application Discovery Service (p. 75).

Attach this policy to any user whose credentials are used by Application Discovery Agent.

This policy also grants the user access to Arsenal. Arsenal is an agent service that is managed and hosted by AWS. Arsenal forwards data to Application Discovery Service in the cloud.

Example AWSApplicationDiscoveryAgentAccess Policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "arsenal:RegisterOnPremisesAgent"
            ],
            "Resource": "*"
        }
    ]
}
```

Granting AWS Agentless Discovery Connector Access

The AWSAgentlessDiscoveryService managed policy grants the AWS Agentless Discovery Connector that is running in a VMware vCenter Server the access to register, communicate with, and share connector health metrics with Application Discovery Service. For more information about this policy, see AWS Managed (Predefined) Policies for Application Discovery Service (p. 75).

Example AWSAgentlessDiscoveryService Policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "awsconnector:RegisterConnector",
                "awsconnector:GetConnectorHealth"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:GetUser",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::connector-platform-upgrade-info/*",
                "arn:aws:s3:::connector-platform-upgrade-info",
                "arn:aws:s3:::connector-platform-upgrade-bundles/*",
                "arn:aws:s3:::connector-platform-upgrade-bundles",
                "arn:aws:s3:::connector-platform-release-notes/*",
                "arn:aws:s3:::connector-platform-release-notes",
                "arn:aws:s3:::prod.agentless.discovery.connector.upgrade/*",
                "arn:aws:s3:::prod.agentless.discovery.connector.upgrade"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": [
                "arn:aws:s3:::import-to-ec2-connector-debug-logs/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "SNS:Publish"
            ],
            "Resource": "arn:aws:sns:*:*:metrics-sns-topic-for-*"
        },
        {
            "Sid": "Discovery",
            "Effect": "Allow",
            "Action": [
                "Discovery:*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "arsenal",
            "Effect": "Allow",
            "Action": [
                "arsenal:RegisterOnPremisesAgent"
            ],
            "Resource": "*"
        }
    ]

}
```

Granting permissions for Agent Data Collection

The ApplicationDiscoveryServiceContinuousExportServiceRolePolicy managed policy allows AWS Application Discovery Service to create Amazon Kinesis Data Firehose streams to transform and deliver data collected by Application Discovery Service agents to an Amazon S3 bucket in your AWS account.

In addition, this policy creates an AWS Glue Data Catalog with a new database called `application_discovery_service_database` and table schemas for mapping data collected by the agents.

For information about using this policy, see AWS Managed (Predefined) Policies for Application Discovery Service (p. 75).

Example ApplicationDiscoveryServiceContinuousExportServiceRolePolicy

```
{
   "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "glue:CreateDatabase",
                "glue:UpdateDatabase",
                "glue:CreateTable",
                "glue:UpdateTable",
                "firehose:CreateDeliveryStream",
                "firehose:DescribeDeliveryStream",
                "logs:CreateLogGroup"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "firehose:DeleteDeliveryStream",
                "firehose:PutRecord",
                "firehose:PutRecordBatch",
                "firehose:UpdateDestination"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:firehose:*:*:deliverystream/aws-application-discovery-
service*"
        },
        {
            "Action": [
                "s3:CreateBucket",
                "s3:ListBucket",
                "s3:PutBucketLogging",
                "s3:PutEncryptionConfiguration"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::aws-application-discovery-service*"
        },
        {
            "Action": [
                "s3:GetObject"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::aws-application-discovery-service*/*"
        },
        {
            "Action": [
                "logs:CreateLogStream",
                "logs:PutRetentionPolicy"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:logs:*:*:log-group:/aws/application-discovery-service/
firehose*"
        },
        {
            "Action": [
                "iam:PassRole"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:iam::*:role/AWSApplicationDiscoveryServiceFirehose",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "firehose.amazonaws.com"
                }
            }
        },
        {
            "Action": [
                "iam:PassRole"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:iam::*:role/service-role/
AWSApplicationDiscoveryServiceFirehose",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "firehose.amazonaws.com"
                }
            }
        }
    ]
}
```

Granting Permissions for Data Collection

The AWSDiscoveryContinuousExportFirehosePolicy policy is required to use Data Exploration in Amazon Athena. It allows Amazon Kinesis Data Firehose to write data collected from Application Discovery Service to Amazon S3. For information about using this policy, see Creating the AWSApplicationDiscoveryServiceFirehose Role (p. 76).

Example AWSDiscoveryContinuousExportFirehosePolicy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "glue:GetTableVersions"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:AbortMultipartUpload",
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::aws-application-discovery-service-*",
                "arn:aws:s3:::aws-application-discovery-service-*/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:*:*:log-group:/aws/application-discovery-service/
firehose:log-stream:*"
            ]
        }
    ]
}
```

Using Service-Linked Roles for Application Discovery Service

AWS Application Discovery Service uses AWS Identity and Access Management (IAM) service-linked roles. A service-linked role is a unique type of IAM role that is linked directly to Application Discovery Service. Service-linked roles are predefined by Application Discovery Service and include all the permissions that the service requires to call other AWS services on your behalf.

A service-linked role makes setting up Application Discovery Service easier because you don't have to manually add the necessary permissions. Application Discovery Service defines the permissions of its service-linked roles, and unless defined otherwise, only Application Discovery Service can assume its roles. The defined permissions include the trust policy and the permissions policy, and that permissions policy cannot be attached to any other IAM entity.

You can delete a service-linked role only after first deleting their related resources. This protects your Application Discovery Service resources because you can't inadvertently remove permission to access the resources.

Topics

- Service-Linked Role Permissions for Application Discovery Service (p. 83)

- Creating a Service-Linked Role for Application Discovery Service (p. 85)
- Deleting a Service-Linked Role for Application Discovery Service (p. 86)

For information about other services that support service-linked roles, see AWS Services That Work with IAM and look for the services that have **Yes** in the **Service-Linked Role** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

Service-Linked Role Permissions for Application Discovery Service

Application Discovery Service uses the service-linked role named **AWSServiceRoleForApplicationDiscoveryServiceContinuousExport** – Enables access to AWS Services and Resources used or managed by AWS Application Discovery Service.

The AWSServiceRoleForApplicationDiscoveryServiceContinuousExport service-linked role trusts the following services to assume the role:

- `continuousexport.discovery.amazonaws.com`

The role permissions policy allows Application Discovery Service to complete the following actions:

glue

`CreateDatabase`

`UpdateDatabase`

`CreateTable`

`UpdateTable`

firehose

`CreateDeliveryStream`

`DeleteDeliveryStream`

`DescribeDeliveryStream`

`PutRecord`

`PutRecordBatch`

`UpdateDestination`

s3

`CreateBucket`

`ListBucket`

`GetObject`

logs

`CreateLogGroup`

`CreateLogStream`

`PutRetentionPolicy`

iam

`PassRole`

This is the full policy showing which resources the above actions apply to:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "glue:CreateDatabase",
                "glue:UpdateDatabase",
                "glue:CreateTable",
                "glue:UpdateTable",
                "firehose:CreateDeliveryStream",
                "firehose:DescribeDeliveryStream",
                "logs:CreateLogGroup"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "firehose:DeleteDeliveryStream",
                "firehose:PutRecord",
                "firehose:PutRecordBatch",
                "firehose:UpdateDestination"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:firehose:*:*:deliverystream/aws-application-discovery-
service*"
        },
        {
            "Action": [
                "s3:CreateBucket",
                "s3:ListBucket",
                "s3:PutBucketLogging",
                "s3:PutEncryptionConfiguration"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::aws-application-discovery-service*"
        },
        {
            "Action": [
                "s3:GetObject"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::aws-application-discovery-service*/*"
        },
        {
            "Action": [
                "logs:CreateLogStream",
                "logs:PutRetentionPolicy"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:logs:*:*:log-group:/aws/application-discovery-service/
firehose*"
        },
        {
            "Action": [
                "iam:PassRole"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:iam::*:role/AWSApplicationDiscoveryServiceFirehose",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "firehose.amazonaws.com"
                }
            }
        },
        {
            "Action": [
                "iam:PassRole"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:iam::*:role/service-role/
AWSApplicationDiscoveryServiceFirehose",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "firehose.amazonaws.com"
                }
            }
        }
    ]
}
```

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. For more information, see Service-Linked Role Permissions in the *IAM User Guide*.

Creating a Service-Linked Role for Application Discovery Service

You don't need to manually create a service-linked role. The AWSServiceRoleForApplicationDiscoveryServiceContinuousExport service-linked role is automatically created when Continuous Export is implicitly turned on by a) confirming options in the dialog box presented from the Data Collectors page after you choose “Start data collection”, or click the slider labeled, “Data exploration in Athena”, or b) when you call the StartContinuousExport API using the AWS CLI.

> **Important**
> This service-linked role can appear in your account if you completed an action in another service that uses the features supported by this role. To learn more, see A New Role Appeared in My IAM Account.

Creating the Service-Linked Role from the Migration Hub Console

You can use the Migration Hub console to create the AWSServiceRoleForApplicationDiscoveryServiceContinuousExport service-linked role.

To create the service-linked role (console)

1. In the navigation pane, choose **Data Collectors**.
2. Choose the **Agents** tab.
3. Toggle the **Data exploration in Athena** slider to the On position.
4. In the dialog box generated from the previous step, click the checkbox agreeing to associated costs and choose **Continue** or **Enable**.

Creating the Service-Linked Role from the AWS CLI

You can use Application Discovery Service commands from the AWS Command Line Interface to create the AWSServiceRoleForApplicationDiscoveryServiceContinuousExport service-linked role.

This service-linked role is automatically created when you start Continuous Export from the AWS CLI (the AWS CLI must first be installed in your environment).

To create the service-linked role (CLI) by starting Continuous Export from the AWS CLI

1. Install the AWS CLI for your operating system (Linux, macOS, or Windows). See the AWS Command Line Interface User Guide for instructions.
2. Open the Command prompt (Windows) or Terminal (Linux or macOS).
   1. Type `aws configure` and press Enter.
   2. Enter your AWS Access Key Id and AWS Secret Access Key.
   3. Enter `us-west-2` for the Default Region Name.
   4. Enter `text` for Default Output Format.
3. Type the following command:

```
aws discovery start-continuous-export
```

You can also use the IAM console to create a service-linked role with the **Discovery Service - Continuous Export** use case. In the IAM CLI or the IAM API, create a service-linked role with the `continuousexport.discovery.amazonaws.com` service name. For more information, see Creating a Service-Linked Role in the *IAM User Guide*. If you delete this service-linked role, you can use this same process to create the role again.

Deleting a Service-Linked Role for Application Discovery Service

If you no longer need to use a feature or service that requires a service-linked role, we recommend that you delete that role. That way you don’t have an unused entity that is not actively monitored or maintained. However, you must clean up your service-linked role before you can manually delete it.

Cleaning Up the Service-Linked Role

Before you can use IAM to delete a service-linked role, you must first delete any resources used by the role.

> **Note**
> If Application Discovery Service is using the role when you try to delete the resources, then the deletion might fail. If that happens, wait for a few minutes and try the operation again.

To delete Application Discovery Service resources used by the AWSServiceRoleForApplicationDiscoveryServiceContinuousExport service-linked role from the Migration Hub Console

1. In the navigation pane, choose **Data Collectors**.
2. Choose the **Agents** tab.
3. Toggle the **Data exploration in Athena** slider to the Off position.

To delete Application Discovery Service resources used by the AWSServiceRoleForApplicationDiscoveryServiceContinuousExport service-linked role from the AWS CLI

1. Install the AWS CLI for your operating system (Linux, macOS, or Windows). See the AWS Command Line Interface User Guide for instructions.
2. Open the Command prompt (Windows) or Terminal (Linux or macOS).

   a. Type `aws configure` and press Enter.
   b. Enter your AWS Access Key Id and AWS Secret Access Key.
   c. Enter `us-west-2` for the Default Region Name.
   d. Enter `text` for Default Output Format.

3. Type the following command:

```
aws discovery stop-continuous-export --export-id <export ID>
```

   - If you don't know the export-ID of the continuous export you want to stop, enter the following command to see the continuous export's ID:

   ```
   aws discovery describe-continuous-exports
   ```

4. Enter the follow command to ensure that Continuous Export has stopped by verifying its return status is "INACTIVE":

```
aws discovery describe-continuous-export
```

Manually Delete the Service-Linked Role

You can delete the AWSServiceRoleForApplicationDiscoveryServiceContinuousExport service-linked role by using the IAM console, the IAM CLI, or the IAM API. If you no longer need to use the Discovery Service - Continuous Export features that require this service-linked role, we recommend that you delete that role. That way you don't have an unused entity that is not actively monitored or maintained. For more information, see Deleting a Service-Linked Role in the *IAM User Guide*.

> **Note**
> You must first clean up your service-linked role before you can delete it. See Cleaning Up the Service-Linked Role (p. 86).

Troubleshooting AWS Application Discovery Service Identity and Access

Use the following information to help you diagnose and fix common issues that you might encounter when working with Application Discovery Service and IAM.

Topics

- I Am Not Authorized to Perform iam:PassRole (p. 87)

I Am Not Authorized to Perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, then you must contact your administrator for assistance. Your administrator is the person that provided you with your user name and password. Ask that person to update your policies to allow you to pass a role to Application Discovery Service.

Some AWS services allow you to pass an existing role to that service, instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named `marymajor` tries to use the console to perform an action in Application Discovery Service. However, the action requires the service to have permissions granted by a service role. Mary does not have permissions to pass the role to the service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform: iam:PassRole
```

In this case, Mary asks her administrator to update her policies to allow her to perform the `iam:PassRole` action.

Logging and monitoring in AWS Application Discovery Service

AWS Application Discovery Service is integrated with AWS CloudTrail. You can use CloudTrail to log, continuously monitor, and retain account activity for troubleshooting and auditing purposes. CloudTrail provides an event history of your AWS account activity, including actions taken through the AWS Management Console, AWS SDKs, and command line tools. The topic in this section explains how to use CloudTrail with Application Discovery Service.

Topics

- Logging Application Discovery Service API Calls with AWS CloudTrail (p. 88)

Logging Application Discovery Service API Calls with AWS CloudTrail

AWS Application Discovery Service is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Application Discovery Service. CloudTrail captures all API calls for Application Discovery Service as events. The calls captured include calls from the Application Discovery Service console and code calls to the Application Discovery Service API operations.

If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Application Discovery Service. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to Application Discovery Service, the IP address from which the request was made, who made the request, when it was made, and additional details.

To learn more about CloudTrail, see the AWS CloudTrail User Guide.

Application Discovery Service Information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in Application Discovery Service, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing Events with CloudTrail Event History.

For an ongoing record of events in your AWS account, including events for Application Discovery Service, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- Overview for Creating a Trail

- CloudTrail Supported Services and Integrations
- Configuring Amazon SNS Notifications for CloudTrail
- Receiving CloudTrail Log Files from Multiple Regions and Receiving CloudTrail Log Files from Multiple Accounts

All Application Discovery Service actions are logged by CloudTrail and are documented in the Application Discovery Service API Reference. For example, calls to the `CreateTags`, `DescribeTags`, and `GetDiscoverySummary` actions generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or AWS Identity and Access Management (IAM) user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the CloudTrail userIdentity Element.

Understanding Application Discovery Service Log File Entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following example shows a CloudTrail log entry that demonstrates the `DescribeTags` action.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAJBHMC4H6EKEXAMPLE:sample-user",
        "arn": "arn:aws:sts::444455556666:assumed-role/ReadOnly/sample-user",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AIDAJQABLZS4A3QDU576Q",
                "arn": "arn:aws:iam::444455556666:role/ReadOnly",
                "accountId": "444455556666",
                "userName": "sampleAdmin"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2020-05-05T15:19:03Z"
            }
        }
    },
    "eventTime": "2020-05-05T17:02:40Z",
    "eventSource": "discovery.amazonaws.com",
    "eventName": "DescribeTags",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "20.22.33.44",
    "userAgent": "Coral/Netty4",
    "requestParameters": {
        "maxResults": 0,
        "filters": [
            {
                "values": [
                    "d-server-0315rfdjreyqsq"
                ],
                "name": "configurationId"
            }
        ]
    },
    "responseElements": null,
    "requestID": "mgh-console-eb1cf315-e2b4-4696-93e5-b3a3b9346b4b",
    "eventID": "7b32b778-91c9-4c75-9cb0-6c852791b2eb",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

AWS Application Discovery Service Quotas

The Service Quotas console provides information about AWS Application Discovery Service quotas. You can use the Service Quotas console to view the default service quotas or to request quota increases for adjustable quotas.

Currently, the only quota that can be increased is **imported servers per account**.

Application Discovery Service has the following default quotas:

- 1,000 applications per account.

  If you reach this quota, and want to import new applications, you can delete existing applications with the `DeleteApplications` API action. For more information, see DeleteApplications in the *Application Discovery Service API Reference*.
- Each import file can have a maximum file size of 10 MB.
- 25,000 imported server records per account.
- 25,000 deletions of import records per day.
- 10,000 imported servers per account (you can request to increase this quota).
- 1,000 active agents, which are collecting and sending data to Application Discovery Service.
- 10,000 inactive agents, which are responsive but not collecting data.
- 400 servers per application.
- 30 tags per server.

Troubleshooting Data Exploration in Amazon Athena

In this section, you can find information about how to fix common issues with your AWS Application Discovery Service.

Topics

- Stop data collection by Data Exploration (p. 92)
- Remove data collected by Data Exploration (p. 93)
- Fix Common issues with Data Exploration in Amazon Athena (p. 93)
- Troubleshooting Failed Import Records (p. 95)

Stop data collection by Data Exploration

To stop Data Exploration, you can either switch off the toggle switch in the Migration Hub console under Discover > Data Collectors > Agents tab, or invoke the `StopContinuousExport` API. It can take up to 30 minutes to stop the data collection, and during this stage, the toggle switch on the console and the `DescribeContinuousExport` API invocation will show the Data Exploration state as "Stop In Progress".

> **Note**
> If after refreshing the console page, the toggle does not switch off and an error message is thrown or the `DescribeContinuousExport` API returns "Stop_Failed" state, you can try again by switching the toggle switch off or calling the `StopContinuousExport` API. If the "Data Exploration" still shows error and fails to successfully stop, please reach out to AWS support.

Alternatively, you can manually stop data collection as described in the following steps.

Option 1: Stop Agent Data collection

If you have already completed your discovery using ADS agents and no longer want to collect additional data in the ADS database repository:

1. From the Migration Hub console choose Discover > Data Collectors > Agents tab.
2. Select all existing running agents and choose **Stop Data Collection**.

   This will ensure that no new data is being collected by the agents in both the ADS data repository and your S3 bucket. Your existing data remains accessible.

Option 2: Delete Data Exploration's Amazon Kinesis Data Streams

If you want to continue collecting data by agents in ADS data repository, but don't want to collect data in your Amazon S3 bucket using Data Exploration, you can manually delete the Amazon Kinesis Data Firehose streams created by Data Exploration:

1. Log in to Amazon Kinesis from the AWS console and choose **Data Firehose** from the navigation pane.
2. Delete the following streams created by the Data Exploration feature:

   - `aws-application-discovery-service-id_mapping_agent`
   - `aws-application-discovery-service-inbound_connection_agent`
   - `aws-application-discovery-service-network_interface_agent`

- `aws-application-discovery-service-os_info_agent`
- `aws-application-discovery-service-outbound_connection_agent`
- `aws-application-discovery-service-processes_agent`
- `aws-application-discovery-service-sys_performance_agent`

Remove data collected by Data Exploration

To remove data collected by Data Exploration

1. Remove the discovery agent data stored in Amazon S3.

   Data collected by Application Discovery Service (ADS) will be stored in an S3 bucket named `aws-application-discover-discovery-service-`*`uniqueid`*.

   > **Note**
   > Deleting the Amazon S3 bucket or any of the objects in it while Data Exploration in Amazon Athena is enabled will cause an error. It will continuing to send new discovery agent data to S3. The deleted data will no longer be accessible in Athena as well.

2. Remove AWS Glue Data Catalog.

   When Data Exploration in Amazon Athena is turned on, it creates an Amazon S3 bucket in your account to store the data collected by ADS agents at regular time intervals. In addition, it also creates an AWS Glue Data Catalog to allow you to query the data stored in a Amazon S3 bucket from Amazon Athena. When you turn off Data Exploration in Amazon Athena, no new data is stored in your Amazon S3 bucket, but data that was collected previously will persist. If you no longer need this data and want to return your account to the state before Data Exploration in Amazon Athena was turned on

   a. Visit Amazon S3 from the AWS console and manually delete the bucket with the name "aws-application-discover-discovery-service-uniqueid"

   b. You can manually remove the Data Exploration AWS Glue Data Catalog by deleting the *application-discovery-service-database* database and all of these tables:

      - `os_info_agent`
      - `network_interface_agent`
      - `sys_performance_agent`
      - `processes_agent`
      - `inbound_connection_agent`
      - `outbound_connection_agent`
      - `id_mapping_agent`

Removing your data from AWS Application Discovery Service

To have all your data removed from Application Discovery Service, contact AWS Support and request full data deletion.

Fix Common issues with Data Exploration in Amazon Athena

In this section, you can find information about how to fix common issues with Data Exploration in Amazon Athena.

Topics

- Data Exploration in Amazon Athena Fails to Initiate Because Service-Linked Roles and Required AWS Resources Can't be Created (p. 94)
- New Agent Data Doesn't show Up in Amazon Athena (p. 94)
- You have Insufficient Permissions to Access Amazon S3, Amazon Kinesis Data Firehose, or AWS Glue (p. 95)

Data Exploration in Amazon Athena Fails to Initiate Because Service-Linked Roles and Required AWS Resources Can't be Created

When you turn on Data Exploration in Amazon Athena, it creates a service-linked-role, `AWSServiceRoleForApplicationDiscoveryServiceContinuousExport`, in your account that allows it to create the required AWS resources for making the agent collected data accessible in Amazon Athena including an Amazon S3 bucket, Amazon Kinesis streams, and AWS Glue Data Catalog. If your account does not have the right permissions for Data Exploration in Amazon Athena to create this role, it will fail to initialize. Refer to AWS Managed (Predefined) Policies for Application Discovery Service (p. 75).

New Agent Data Doesn't show Up in Amazon Athena

If new data does not flow into Athena, it has been more than 30 minutes since an agent started, and Data Exploration status is Active, check the solutions listed below:

- AWS Discovery Agents

  Ensure that your agent's **Collection** status is marked as **Started** and the **Health** status is marked as **Running**.

- Kinesis Role

  Ensure that you have the `AWSApplicationDiscoveryServiceFirehose` role in your account.

- Kinesis Data Firehose Status

  Ensure that the following Kinesis Data Firehose delivery streams are working correctly:
  - `aws-application-discovery-service/os_info_agent`
  - `aws-application-discovery-service-network_interface_agent`
  - `aws-application-discovery-service-sys_performance_agent`
  - `aws-application-discovery-service-processes_agent`
  - `aws-application-discovery-service-inbound_connection_agent`
  - `aws-application-discovery-service-outbound_connection_agent`
  - `aws-application-discovery-service-id_mapping_agent`

- AWS Glue Data Catalog

  Ensure that the `application-discovery-service-database` database is in AWS Glue. Make sure that the following tables are present in AWS Glue:
  - `os_info_agent`
  - `network_interface_agent`

  - `sys_performance_agent`
  - `processes_agent`
  - `inbound_connection_agent`
  - `outbound_connection_agent`
  - `id_mapping_agent`

- Amazon S3 Bucket

  Ensure that you have an Amazon S3 bucket named `aws-application-discovery-service-`*`uniqueid`* in your account. If objects in the bucket have been moved or deleted, they will not show up properly in Athena.

- Your on-premises servers

  Ensure that your servers are running so that your agents can collect and send data to AWS Application Discovery Service.

You have Insufficient Permissions to Access Amazon S3, Amazon Kinesis Data Firehose, or AWS Glue

If you are using AWS Organizations, and initialization for Data Exploration in Amazon Athena fails, it can be because you don't have permissions to access Amazon S3, Amazon Kinesis Data Firehose, Athena or AWS Glue.

You will need an IAM user with administrator permissions to grant you access to these services. An administrator can use their account to grant this access. See AWS Managed (Predefined) Policies for Application Discovery Service (p. 75).

To ensure that Data Exploration in Amazon Athena works correctly, do not modify or delete the AWS resources created by Data Exploration in Amazon Athena including the Amazon S3 bucket, Amazon Kinesis Data Firehose Streams, and AWS Glue Data Catalog. If you accidentally delete or modify these resources, please stop and start Data Exploration and it will automatically create these resources again. If you delete the Amazon S3 bucket created by Data Exploration, you may lose the data that was collected in the bucket.

Troubleshooting Failed Import Records

Migration Hub import allows you to import details of your on-premises environment directly into Migration Hub without using the Discovery Connector or Discovery Agent. This gives you the option to perform migration assessment and planning directly from your imported data. You can also group your devices as applications and track their migration status.

When importing data, it's possible that you'll encounter errors. Typically, these errors occur for one of the following reasons:

- **An import-related quota was reached** – There is a quota associated with import tasks. If you make an import task request that would exceeds the quotas, then the request will fail and return an error. For more information, see

  > The Service Quotas console provides information about AWS Application Discovery Service quotas. You can use the Service Quotas console to view the default service quotas or to request quota increases for adjustable quotas.

Currently, the only quota that can be increased is **imported servers per account**.

Application Discovery Service has the following default quotas:

- 1,000 applications per account.

  If you reach this quota, and want to import new applications, you can delete existing applications with the `DeleteApplications` API action. For more information, see DeleteApplications in the *Application Discovery Service API Reference*.
- Each import file can have a maximum file size of 10 MB.
- 25,000 imported server records per account.
- 25,000 deletions of import records per day.
- 10,000 imported servers per account (you can request to increase this quota).
- 1,000 active agents, which are collecting and sending data to Application Discovery Service.
- 10,000 inactive agents, which are responsive but not collecting data.
- 400 servers per application.
- 30 tags per server.

(p. 91).

- **An extra comma (,) was inserted into the import file** – Commas in .CSV files are used to differentiate one field from the next. Having a comma appear within a field is unsupported, because it will always split a field. This can cause a cascade of formatting errors. Be sure that commas are only used between fields, and are not otherwise used in your import files.
- **A field has a value outside of its supported range** – Some fields, like `CPU.NumberOfCores` must have a range of values they support. If you have more or less than this supported range, then the record will fail to be imported.

If any errors occur with your import request, you can resolve them by downloading your failed records for your import task, and resolve the errors in the failed entries CSV file, and do the import again.

Console

To download your failed records archive

1. Sign into the AWS Management Console, and open the Migration Hub console at https://console.aws.amazon.com/migrationhub.
2. From the left-side navigation, under **Discover**, choose **Tools**.
3. From **Discovery Tools**, choose **view imports**.
4. From the **Imports** dashboard, choose the radio button associated an import request with some number of **Failed records**.
5. Choose **Download failed records** from above the table on the dashboard. This will open your browser's download dialog box for downloading the archive file.

AWS CLI

To download your failed records archive

1. Open a terminal window, and type the following command, where *`ImportName`* `is the name of the import task with the failed entries that you want to correct.:`

```
aws discovery describe-import-tasks - -name ImportName
```

2. From the output, copy the entire contents of the value returned for `errorsAndFailedEntriesZip`, without the surrounding quotes.
3. Open a web browser, and paste in the contents into the URL text box and press `ENTER`. This will download the failed records archive, compressed in a .zip format.

Now that you've downloaded your failed records archive, you can extract the two files within and correct the errors. Note that if your errors are tied to service-based limits, you'll either need to request a limit increase, or delete enough of the associated resources to get your account under the limit. The archive has the following files:

- **errors-file.csv** – This file is your error log, and it tracks the line, column name, `ExternalId`, and a descriptive error message for each failed record of each failed entry.
- **failed-entries-file.csv** – This file contains only the failed entries from your original import file.

To correct the non-limit-based errors you've encountered, use the `errors-file.csv` to correct the issues in the `failed-entries-file.csv` file, and then import that file. For instructions on importing files, see Importing Data (p. 37).

Document History for AWS Application Discovery Service

- **API version**: 2015-11-01
- **Latest User Guide documentation update**: November 14, 2019

The following table describes important changes to the *AWS Migration Hub User Guide* after January 18 2019. For notifications about documentation updates, you can subscribe to the RSS feed.

update-history-change	update-history-description	update-history-date
Introducing the Home Region (p. 98)	The Migration Hub home region provides a single repository of discovery and migration planning information for your entire portfolio, and a single view of migrations into multiple AWS Regions.	November 20, 2019
Introducing the Migration Hub import feature (p. 98)	Migration Hub import allows you to import information about your on-premises servers and applications into Migration Hub, including server specifications and utilization data. You can also use this data to track the status of application migrations. For more information, see Migration Hub Import.	January 18, 2019

The following table describes documentation releases for the *AWS Migration Hub User Guide* before January 18, 2019:

Change	Description	Date
New Feature	Updated docs to support Data Exploration in Amazon Athena and added Troubleshooting chapter.	August 09, 2018
Major revision	Rewrites to usage & output details; entire document restructured.	May 25, 2018
Discovery Agent 2.0	A new and improved Application Discovery agent was released.	October 19, 2017
Console	The AWS Management Console was added.	December 19, 2016

Change	Description	Date
Agentless discovery	This release describes how to set up and configure agentless discovery.	July 28, 2016
New details for Microsoft Windows Server and command issue fixes	This update adds details about Microsoft Windows Server. It also documents fixes to various command issues.	May 20, 2016
Initial publication	This is the first release of the *Application Discovery Service User Guide.*	May 12, 2016

AWS glossary

For the latest AWS terminology, see the AWS glossary in the *AWS General Reference*.