# Table of Contents

What Is AWS Application Discovery Service? ................................................................. 1
More About VMware Discovery .................................................................................. 1
Compare Connectors and Agents ............................................................................. 2
Assumptions .............................................................................................................. 3

## Setting Up

- Step 1: Sign Up for AWS ...................................................................................... 4
- Step 2: Create an IAM User .................................................................................. 4
- Step 3: Attach IAM Policies .................................................................................. 6

## Discovery Agent

- Data Collected by the Discovery Agent ................................................................. 17
- Prerequisites for Agent Installation ....................................................................... 19
- Agent Installation on Linux .................................................................................... 20
  - Requirements on Older Linux Platforms ............................................................... 22
  - Manage the Discovery Agent Process on Linux ..................................................... 23
  - Agent Troubleshooting on Linux .......................................................................... 24
- Agent Installation on Windows ............................................................................... 25
  - Package Signing on Windows 2003 ...................................................................... 26
  - Manage the Discovery Agent Process on Windows ............................................... 26
  - Troubleshooting on Windows ................................................................................. 27

## Start Data Collection

- Data Collected by Discovery Connector ................................................................. 29
- Download the Discovery Connector ....................................................................... 32
- Deploy the Discovery Connector ............................................................................ 33

## Configure the Discovery Connector

- Configure a static IP address for the connector ..................................................... 34
- Control Data Collection ......................................................................................... 34
- Disabling Auto-Upgrades on AWS Discovery Connector ........................................ 35

## Troubleshooting the Discovery Connector

- Fixing Unhealthy Connectors ................................................................................ 37
- Standalone ESX Host Support ................................................................................ 38
- Getting Additional Support for Connector Issues ................................................... 38

## Import

- Supported Import File Fields .................................................................................. 39
- Setting Up Your Import Permissions ...................................................................... 42
- Uploading Your Import File to Amazon S3 .............................................................. 44
- Importing Data ....................................................................................................... 45
- Tracking Your Migration Hub Import Requests ...................................................... 46

## View, Export & Explore Data

- View Collected Data .............................................................................................. 48
- Matching Logic ....................................................................................................... 48
- Export Collected Data ............................................................................................ 49

## Data Exploration in Athena

- Enabling Data Exploration in Amazon Athena ...................................................... 50
- Working with Data Exploration in Amazon Athena ............................................... 52

## Console Walkthroughs

- Main Dashboard ..................................................................................................... 59
- Navigating from the Dashboard and the Navigation Pane ....................................... 60
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. 29) and Data Collected by the Discovery Agent (p. 17).

## Compare Connectors and Agents

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

<table>
<thead>
<tr>
<th>Supported server types</th>
<th>Discovery Connector</th>
<th>Discovery Agent</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMware virtual machine</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Physical server</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Deployment</th>
<th>Discovery Connector</th>
<th>Discovery Agent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Per server</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Per vCenter</td>
<td>yes</td>
<td>no</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Collected data</th>
<th>Discovery Connector</th>
<th>Discovery Agent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static configuration data</td>
<td>yes</td>
<td>yes (Export only)</td>
</tr>
<tr>
<td>VM utilization metrics</td>
<td>yes</td>
<td>yes (Export only)</td>
</tr>
<tr>
<td>Time series performance information</td>
<td>no</td>
<td>yes (Export only)</td>
</tr>
<tr>
<td>Network inbound/outbound connections</td>
<td>no</td>
<td>yes (Export only)</td>
</tr>
<tr>
<td>Running processes</td>
<td>no</td>
<td>yes (Export only)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Supported OS</th>
<th>Discovery Connector</th>
<th>Discovery Agent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any OS running in <strong>VMware vCenter</strong> (V5.5, V6, &amp; V6.5)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Linux</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Amazon Linux 2012.03, 2015.03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Amazon Linux 2 (9/25/2018 update or later)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ubuntu 12.04, 14.04, 16.04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Red Hat Enterprise Linux 5.11, 6.9, 7.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CentOS 5.11, 6.9, 7.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUSE 11 SP4, 12 SP2</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Windows</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows Server 2003 R2 SP2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Discovery Connector</th>
<th>Discovery Agent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2008 R1 SP2, 2008 R2 SP1</td>
<td></td>
</tr>
<tr>
<td>Windows Server 2012 R1, 2012 R2</td>
<td></td>
</tr>
<tr>
<td>Windows Server 2016</td>
<td></td>
</tr>
<tr>
<td>Windows Server 2019</td>
<td></td>
</tr>
</tbody>
</table>
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 an IAM User (p. 4)
- Step 3: Provide Application Discovery Service Access to Non-Administrator Users by Attaching Policies (p. 6)

Once you have completed the three steps of Setting Up AWS Application Discovery Service (p. 4), it is recommended that you read the section Understanding and Using Service-Linked Roles for Application Discovery Service (p. 11). There is no set-up required for you to use this service-linked role as it is automatically created for you when Continuous Export is turned on by enabling Data Exploration in Amazon Athena (p. 50). However it is important to understand the concept and this section also gives instructions for deleting the service-linked role.

Step 1: Sign Up for AWS

When you sign up for Amazon Web Services (AWS), you are charged only for the services that you use. If you already have an AWS account, you can skip ahead to step 2. If you don't have an AWS account, use the following procedure to create one.

To create an AWS account

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 an IAM User

Services such as AWS Application Discovery Service require that you provide credentials when you access them. This way the service can determine whether you have permissions to access its resources. We recommend that you don't use the AWS account root user credentials to make requests. Instead, create an AWS Identity and Access Management (IAM) user, and grant that user full access. We refer to these users as having administrator-level credentials. You can use the administrator-level credentials to interact with AWS and perform tasks such as create an AWS S3 bucket, create additional IAM users, and grant permissions. For more information, see Root Account Credentials vs. IAM User Credentials in the AWS General Reference and IAM Best Practices in the IAM User Guide.

If you signed up for AWS but have not created an IAM user for yourself, you can create one using the IAM console.
To create an administrator user for yourself and add the user to an administrators group (console)

1. Use your AWS account email address and password to sign in as the AWS account root user to the IAM console at https://console.aws.amazon.com/iam/.
   
   **Note**
   
   We strongly recommend that you adhere to the best practice of using the Administrator IAM user below and securely lock away the root user credentials. Sign in as the root user only to perform a few account and service management tasks.

2. In the navigation pane, choose Users and then choose Add user.

3. For User name, enter Administrator.

4. Select the check box next to AWS Management Console access. Then select Custom password, and then enter your new password in the text box.

5. (Optional) By default, AWS requires the new user to create a new password when first signing in. You can clear the check box next to User must create a new password at next sign-in to allow the new user to reset their password after they sign in.

6. Choose Next: Permissions.

7. Under Set permissions, choose Add user to group.

8. Choose Create group.

9. In the Create group dialog box, for Group name enter Administrators.

10. Choose Filter policies, and then select AWS managed -job function to filter the table contents.

11. In the policy list, select the check box for AdministratorAccess. Then choose Create group.
   
   **Note**
   
   You must activate IAM user and role access to Billing before you can use the AdministratorAccess permissions to access the AWS Billing and Cost Management console. To do this, follow the instructions in step 1 of the tutorial about delegating access to the billing console.

12. Back in the list of groups, select the check box for your new group. Choose Refresh if necessary to see the group in the list.

13. Choose Next: Tags.

14. (Optional) Add metadata to the user by attaching tags as key-value pairs. For more information about using tags in IAM, see Tagging IAM Entities in the IAM User Guide.

15. Choose Next: Review to see the list of group memberships to be added to the new user. When you are ready to proceed, choose Create user.

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

**Note**

- An administrator account will by default inherit all the policies required for accessing Application Discovery Service.
- For a non-administrator user, you can manually add the policies required to access Application Discovery Service. Refer to Step 3: Provide Application Discovery Service Access to Non-Administrator Users by Attaching Policies (p. 6) for details.

To sign in as this new IAM user, first sign out of the AWS Management Console. Next, use the following URL, where your_aws_account_id is your AWS account number without the hyphens. For example, if your AWS account number is 1234-5678-9012, then your_aws_account_id is 123456789012).
Enter the IAM user name and password that you just created. When you're signed in, the navigation bar displays your_user_name@your_aws_account_id.

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

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

To verify the sign-in link for IAM users for your account, open the IAM console and check under AWS Account Alias on the dashboard.

### Step 3: Provide Application Discovery Service Access to Non-Administrator Users by Attaching Policies

Application Discovery Service uses the IAM-managed policies listed here to control access to the service or components of the service. An administrator account will by default inherit all the policies required for accessing Application Discovery Service. If your account is a non-administrative account, in order to access Application Discovery Service, you need to request your administrator to add the below policies to your account. For information about how to attach these managed policies to an IAM user account, see Working with Managed Policies in the IAM User Guide.

**AWSApplicationDiscoveryServiceFullAccess**

Grants the IAM user account access to the Application Discovery Service and Migration Hub APIs. With this policy, the user can configure Application Discovery Service, start and stop agents, start and stop agentless discovery, and query data from the AWS Discovery Service database.

**AWSApplicationDiscoveryAgentAccess**

Grants the Application Discovery Agent access to register and communicate with Application Discovery Service. 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.

**AWSAgentlessDiscoveryService**

Grants the AWS Agentless Discovery Connector that is running in your VMware vCenter Server the access to register, communicate with, and share connector health metrics with Application Discovery Service. Attach this policy to any user whose credentials are used by the connector.

**ApplicationDiscoveryServiceContinuousExportServiceRolePolicy**

This policy is automatically added to your account when you turn on Data Exploration in Amazon Athena and have the AWSApplicationDiscoveryServiceFullAccess policy assigned. It 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.

**AWSDiscoveryContinuousExportFirehosePolicy**

This 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.
An administrator needs to attach the above policies to your user. For AWSDiscoveryContinuousExportFirehosePolicy policy, the administrator needs to create a role named "AWSApplicationDiscoveryServiceFirehose" with Kinesis Data Firehose as a trusted entity and then attach the policy AWSDiscoveryContinuousExportFirehosePolicy to the role as show in the procedures below:

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.
7. Check the box beside AWSDiscoveryContinuousExportFirehosePolicy and then choose Next: Review.
8. Enter AWSApplicationDiscoveryServiceFirehose as the role name and then choose Create role.

Each of the Application Discovery Service managed policies is shown here so that you can customize them as needed.

AWSApplicationDiscoveryServiceFullAccess

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

AWSApplicationDiscoveryAgentAccess

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

AWSAgentlessDiscoveryService
Step 3: Attach IAM Policies

```json
{
    "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-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/*"
            ]
        },
        {
            "Sid": "Discovery",
            "Effect": "Allow",
            "Action": [
                "Discovery:*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "arsenal",
            "Effect": "Allow",
            "Action": [
                "arsenal:RegisterOnPremisesAgent"
            ],
            "Resource": "*"
        }
    ]
}
```
ApplicationDiscoveryServiceContinuousExportServiceRolePolicy

```json
{
    "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",
```
Step 3: Attach IAM Policies

AWSApplicationDiscoveryServiceFirehosePolicy

```
{
    "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:*"
        ]
    ]
}
```
Understanding and 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.

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*"
}
```
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](https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-files.html) 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
   ```

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](https://docs.aws.amazon.com/IAM/latest/userguide/creating-service-linked-roles.html). 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. 15).
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 Discovery Agent.

- 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, it pings the service at 15-minute intervals for configuration information. When you send a command that tells an agent to start data collection, it starts collecting data for the host or VM where it resides.

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 Prerequisites for Agent Installation (p. 19)

**Topics**
- Data Collected by the Discovery Agent (p. 17)
- Prerequisites for Agent Installation (p. 19)
- Agent Installation on Linux (p. 20)
- Agent Installation on Windows (p. 25)
- Start Discovery Agent Data Collection (p. 28)

Data Collected by the Discovery Agent

Following, you can find an inventory of the information collected by the Discovery Agent.
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).
- Data fields denoted with an asterisk (*) are only available in the .csv files produced from the agent's API export function.
- The polling period is in intervals of approximately 15 minutes.

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

These are the pre-installation tasks that should be performed to prevent errors from occurring during the actual installation of the agent. Be sure you have set a Migration Hub home region before you begin installing an agent. If you have a 1.x version of the agent installed, it must be removed before installing the latest version. Instructions for removing older versions are provided in the tasks that follow.

- 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
Red Hat Enterprise Linux 5.11, 6.9, 7.3
CentOS 5.11, 6.9, 7.3
SUSE 11 SP4, 12 SP2

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
- Or substitute your home region as needed for all other regions except us-west-2.
- If us-west-2 is your home region, use https://arsenal.us-west-2.amazonaws.com:443
- Access to AWS S3 in your home region is required for auto-upgrade to function.
- Create an IAM user in the IAM console and attach the existing AWSApplicationDiscoveryAgentAccess permissions policy. This policy allows the user to perform necessary agent actions on your behalf. See Step 3: Provide Application Discovery Service Access to Non-Administrator Users by Attaching Policies (p. 6) for Discovery Agent installation prerequisites.
- 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.
- Remove any previous-generation agents. If you previously installed Application Discovery Agent 1.0 for either Windows or Linux, you must uninstall it before continuing with the installation of the current agent.

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon Linux, Red Hat Enterprise Linux, CentOS</td>
<td>yum remove AwsAgent</td>
</tr>
<tr>
<td>Ubuntu Server</td>
<td>apt-get remove awsagent</td>
</tr>
<tr>
<td>Windows Server</td>
<td>Use Add/Remove Programs to uninstall AWS Agent.</td>
</tr>
</tbody>
</table>

**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. 22).
To install AWS Application Discovery Agent in your data center

1. Log 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.

   ```bash
   ```

   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:

   ```bash
   
   
   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.

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

5. Run the command to install the agent in your home region. In this example, we use the string `Your_Home_Region` to show where to insert your home region.

   ```bash
   sudo bash install -r Your_Home_Region -k <aws key id> -s <aws key secret>
   ```

   **Note**
   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 installation script.

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`
   - Or substitute your home region as needed for all other regions except us-west-2.
   - If us-west-2 is your home region, use `https://arsenal.us-west-2.amazonaws.com:443`
Note
Agents also work with transparent web proxies. However, if you need to configure a non-transparent proxy, proceed to the next step.

7. Optional: To Configure a Non-Transparent Proxy:

   a. Find the configuration file as described in Agent Troubleshooting on Linux (p. 24) and edit the file by adding the required configuration data as follows:

   ```json
   "proxyHost" : ",myproxy.mycompany.com",
   "proxyPort" : <1234>,
   "proxyUser" : ",myusername",
   "proxyPassword" : ",mypassword",
   "proxyScheme" : "http"
   ```

   b. Save the edited configuration file ensuring that you still have valid json (taking care with the quotes and the commas). If your proxy doesn't require authentication, then leave out `proxyUser` and `proxyPassword`. While most proxies use HTTP, if your proxy uses HTTPS, specify the following in the configuration file:

   ```json
   "proxyScheme" : "https"
   ```

   c. Restart the agent.

   Note
   If you encounter problems, add the following to the configuration file:

   ```json
   "enableAWSSDKLogging" : true
   ```

   Then, restart the agent again, let it run for at least 15 minutes, and contact AWS Support. They will help you troubleshoot and may ask you to send them the generated log files which can be found as described in Agent Troubleshooting on Linux (p. 24).

Topics
- Requirements on Older Linux Platforms (p. 22)
- Manage the Discovery Agent Process on Linux (p. 23)
- Agent Troubleshooting on Linux (p. 24)

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 installation script from downloading installation packages. They might also have a limited ability to find and download the platform libraries required by the agent from deprecated Linux repositories.

32-bit libc

One of the dependencies needed for the Application Discovery agent is 32-bit libc. This library must be installed on 64-bit systems that run the agent. If the installation script exits because it fails to find a suitable repository or otherwise fails to install 32-bit libc, you must manually find and install 32-bit libc before you can complete agent installation. Because 32-bit libc is a core Linux library, you must take great care in identifying a package that is compatible with your system. We recommend contacting AWS Support for assistance. After 32-bit libc is installed, run the installation script with the `-p false` parameter to skip the automated search of Linux repositories for prerequisites.
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 three installation script options can be used in any combination. In the following example command, all three have been passed to the installation script:

```
sudo bash install -r Your_Home_Region -k <aws key id> -s <aws key secret> -p false -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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verify an agent is running</td>
<td><code>sudo systemctl status aws-discovery-daemon.service</code></td>
</tr>
<tr>
<td>Start an agent</td>
<td><code>sudo systemctl start aws-discovery-daemon.service</code></td>
</tr>
<tr>
<td>Stop an agent</td>
<td><code>sudo systemctl stop aws-discovery-daemon.service</code></td>
</tr>
<tr>
<td>Restart an agent</td>
<td><code>sudo systemctl restart aws-discovery-daemon.service</code></td>
</tr>
<tr>
<td>Uninstall an agent</td>
<td><code>yum remove aws-discovery-agent</code></td>
</tr>
</tbody>
</table>

**Upstart**

Management Commands for the Application Discovery Agent

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verify a status</td>
<td><code>sudo initctl status aws-discovery-daemon</code></td>
</tr>
<tr>
<td>Start an agent</td>
<td><code>sudo initctl start aws-discovery-daemon</code></td>
</tr>
</tbody>
</table>
Agent Troubleshooting on Linux

If you encounter problems while installing or using the Application 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**

  Agent log files can be found under 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 installer.

- **Configuration files**

  Agent configuration files can be found under the following directory.

  `/var/opt/aws/discovery/`

  For instructions on how to remove older versions of the Discovery Agent, see Prerequisites for Agent Installation (p. 19).

---

**System V init**

**Management Commands for the Application Discovery Agent**

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop an agent</td>
<td><code>sudo initctl stop aws-discovery-daemon</code></td>
</tr>
<tr>
<td>Restart an agent</td>
<td><code>sudo initctl restart aws-discovery-daemon</code></td>
</tr>
<tr>
<td>Uninstall an agent</td>
<td><code>apt-get remove aws-discovery-agent</code></td>
</tr>
</tbody>
</table>

---

**Agent Troubleshooting on Linux**

If you encounter problems while installing or using the Application 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**

  Agent log files can be found under 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 installer.

- **Configuration files**

  Agent configuration files can be found under the following directory.

  `/var/opt/aws/discovery/`

  For instructions on how to remove older versions of the Discovery Agent, see Prerequisites for Agent Installation (p. 19).
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. Navigate to the Microsoft Download Center and choose Download to be taken to the download selection page, then on this page, select only vc_redist.x86.exe (do not select the "x64" version) regardless of the architecture of the machine you are installing on, then choose Next. Your download begins immediately.

2. Download the Windows agent installer but do not double-click and execute the installer within Windows.

   Important
   Do not double-click and execute 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.)

3. Open a command prompt as an administrator and navigate to the location where you saved the installation package.

4. To install the agent, run the following example command. The example uses the string Your_Home_Region to show where to insert the name of your home region.

   ```
   msiexec.exe /i AWSDiscoveryAgentInstaller.msi REGION="Your_Home_Region" KEY_ID="<aws key id>" KEY_SECRET="<aws key secret>" /q
   ```

   Note
   Agents automatically download and apply updates as they become available. We recommend this default configuration. To avoid downloading agents and applying updates automatically, include the following parameter when running the installation:
   AUTO_UPDATE=false

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

5. 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
   • Or substitute your home region as needed for all other regions except us-west-2.
   • If us-west-2 is your home region, use https://arsenal.us-west-2.amazonaws.com:443

   Note
   Agents also work with transparent web proxies. However, if you need to configure a non-transparent proxy, continue on with the following steps.

6. Optional: To Configure a Non-Transparent Proxy:

   a. Find the configuration file as described in Agent Troubleshooting on Windows (p. 27), and edit the file by adding the required configuration data as follows:

   ```
   "proxyHost" : "<myproxy.mycompany.com>",
   "proxyPort" : <1234>,
   ```
b. Save the edited configuration file ensuring that you still have valid json (taking care with the quotes and the commas). If your proxy doesn't require authentication, then leave out `proxyUser` and `proxyPassword`. While most proxies use http, if yours uses https, specify the following in the configuration file:

```
"proxyScheme" : "https"
```

c. Restart the agent.

**Note**
If you encounter problems, add the following to the configuration file:

```
"enableAWSSDKLogging" : true
```

Then, restart the agent again, let it run for 15 minutes, and troubleshoot what the issue may be by reading through the generated log files which can be found as described in [Agent Troubleshooting on Windows](p. 27).

### Topics
- Package Signing on Windows 2003 (p. 26)
- Manage the Discovery Agent Process on Windows (p. 26)
- Agent Troubleshooting on Windows (p. 27)

## Package Signing on Windows 2003

For Windows Server 2008 and later, Amazon cryptographically signs the Application Discovery Service agent installation package with an SHA256 certificate. However, because the SHA2 certificate family is not supported by Windows Server 2003, the installation package for that platform is signed with an SHA1 certificate. Microsoft has published [hotfixes](https://support.microsoft.com) that might allow your Windows 2003 systems to read an SHA256 certificate. If you require SHA256 in your Windows 2003 environment, contact AWS Support for assistance.

## 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.

<table>
<thead>
<tr>
<th>Task</th>
<th>Service Name</th>
<th>Service Status/Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verify that an agent is running</td>
<td>AWS Discovery Agent</td>
<td>Started</td>
</tr>
<tr>
<td></td>
<td>AWS Discovery Updater</td>
<td></td>
</tr>
<tr>
<td>Start an agent</td>
<td>AWS Discovery Agent</td>
<td>Choose <strong>Start</strong></td>
</tr>
<tr>
<td></td>
<td>AWS Discovery Updater</td>
<td></td>
</tr>
<tr>
<td>Stop an agent</td>
<td>AWS Discovery Agent</td>
<td>Choose <strong>Stop</strong></td>
</tr>
<tr>
<td></td>
<td>AWS Discovery Updater</td>
<td></td>
</tr>
</tbody>
</table>
### To uninstall a discovery agent on Windows

2. Choose Programs.
3. Choose Programs and Features.
4. Select AWS Discovery Agent.
5. Choose Uninstall.

### Agent Troubleshooting on Windows

If you encounter problems while installing or using the Application Discovery Agent on Windows, 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.

- **Installation logging**

  In some cases, the `msiexec` command described preceding 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 `/L*V 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
  ```

  Logs files are named to indicate whether generated by the main service, automatic upgrader, or 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 older versions of the Discovery Agent, see Prerequisites for Agent Installation (p. 19).

Start Discovery Agent Data Collection

Now that you have deployed and configured the Discovery Agent, you must complete the final step of actually turning on its data collection process. You can turn on data collection through the console or by making API calls through the AWS CLI. Instructions are provided below for both ways, by expanding your method of choice:

Using the Migration Hub Console

Start the Discovery Agent data collection process on the Data Collectors page of the Migration Hub console. Be sure you've selected a Migration Hub home region before you start data collection. All of your data is stored in your home region.

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

Using the AWS CLI

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

**To install the AWS CLI and start 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. Type the following command:

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

   • If you don’t know the ID of the agent you want to start, enter the following command to see the agent's ID:

   ```bash
   aws discovery describe-agents
   ```
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

When you start the connector’s data collecting process, it connects to VMware vCenter Server where it collects information 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. 29)
- Download the Discovery Connector (p. 32)
- Deploy the Discovery Connector (p. 33)
- Configure the AWS Discovery Connector (p. 33)
- Start Discovery Connector Data Collection (p. 36)
- Troubleshooting the Discovery Connector (p. 37)

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. See Step 3: Provide Application Discovery Service Access to Non-Administrator Users by Attaching Policies (p. 6) for Discovery Connector installation prerequisites.

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.

<table>
<thead>
<tr>
<th>Data field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>applicationConfigurationId</td>
<td>ID of the migration application the VM is grouped under</td>
</tr>
<tr>
<td>avgCpuUsagePct</td>
<td>Average percentage of CPU usage over polling period</td>
</tr>
<tr>
<td>avgDiskBytesReadPerSecond</td>
<td>Average number of bytes read from disk over polling period</td>
</tr>
<tr>
<td>avgDiskBytesWrittenPerSecond</td>
<td>Average number of bytes written to disk over polling period</td>
</tr>
<tr>
<td>avgDiskReadOpsPerSecond**</td>
<td>Average number of read I/O operations per second null</td>
</tr>
<tr>
<td>avgDiskWriteOpsPerSecond**</td>
<td>Average number of write I/O operations per second</td>
</tr>
<tr>
<td>avgFreeRAM</td>
<td>Average free RAM expressed in MB</td>
</tr>
<tr>
<td>avgNetworkBytesReadPerSecond</td>
<td>Average amount of throughput of bytes read per second</td>
</tr>
<tr>
<td>avgNetworkBytesWrittenPerSecond</td>
<td>Average amount of throughput of bytes written per second</td>
</tr>
<tr>
<td>configId</td>
<td>Application Discovery Service assigned ID to the discovered VM</td>
</tr>
<tr>
<td>configType</td>
<td>Type of resource discovered</td>
</tr>
<tr>
<td>connectorId</td>
<td>ID of the Discovery Connector virtual appliance</td>
</tr>
<tr>
<td>cpuType</td>
<td>vCPU for a VM, actual model for a host</td>
</tr>
<tr>
<td>datacenterId</td>
<td>ID of the vCenter</td>
</tr>
<tr>
<td>hostId*</td>
<td>ID of the VM host</td>
</tr>
<tr>
<td>hostName</td>
<td>Name of host running the virtualization software</td>
</tr>
<tr>
<td>hypervisor</td>
<td>Type of hypervisor</td>
</tr>
<tr>
<td>id</td>
<td>ID of server</td>
</tr>
<tr>
<td>lastModifiedTimeStamp*</td>
<td>Latest date and time of data collection before data export</td>
</tr>
<tr>
<td>macAddress</td>
<td>MAC address of the VM</td>
</tr>
<tr>
<td>manufacturer</td>
<td>Maker of the virtualization software</td>
</tr>
<tr>
<td>Data field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>------------------------------------------------------------</td>
</tr>
<tr>
<td>maxCpuUsagePct</td>
<td>Max. percentage of CPU usage during polling period</td>
</tr>
<tr>
<td>maxDiskBytesReadPerSecond</td>
<td>Max. number of bytes read from disk over polling period</td>
</tr>
<tr>
<td>maxDiskBytesWrittenPerSecond</td>
<td>Max. number of bytes written to disk over polling period</td>
</tr>
<tr>
<td>maxDiskReadOpsPerSecond</td>
<td>Max. number of read I/O operations per second</td>
</tr>
<tr>
<td>maxDiskWriteOpsPerSecond</td>
<td>Max. number of write I/O operations per second</td>
</tr>
<tr>
<td>maxNetworkBytesReadPerSecond</td>
<td>Max. amount of throughput of bytes read per second</td>
</tr>
<tr>
<td>maxNetworkBytesWrittenPerSecond</td>
<td>Max. amount of throughput of bytes written per second</td>
</tr>
<tr>
<td>memoryReservation</td>
<td>Limit to avoid overcommitment of memory on VM</td>
</tr>
<tr>
<td>moRefId</td>
<td>Unique vCenter Managed Object Reference ID</td>
</tr>
<tr>
<td>name</td>
<td>Name of VM or network (user specified)</td>
</tr>
<tr>
<td>numCores</td>
<td>Number of independent processing units within CPU</td>
</tr>
<tr>
<td>numCpus</td>
<td>Number of central processing units on VM</td>
</tr>
<tr>
<td>numDisks**</td>
<td>Number of disks on VM</td>
</tr>
<tr>
<td>numNetworkCards**</td>
<td>Number of network cards on VM</td>
</tr>
<tr>
<td>osName</td>
<td>Operating system name on VM</td>
</tr>
<tr>
<td>osVersion</td>
<td>Operating system version on VM</td>
</tr>
<tr>
<td>portGroupId</td>
<td>ID of group of member ports of VLAN</td>
</tr>
<tr>
<td>portGroupName</td>
<td>Name of group of member ports of VLAN</td>
</tr>
<tr>
<td>powerState</td>
<td>Status of power</td>
</tr>
<tr>
<td>serverId</td>
<td>Application Discovery Service assigned ID to the discovered VM</td>
</tr>
<tr>
<td>smBiosId</td>
<td>ID/version of the system management BIOS</td>
</tr>
<tr>
<td>state</td>
<td>Status of the Discovery Connector virtual appliance</td>
</tr>
<tr>
<td>tagKey</td>
<td>User-defined key to store custom data or metadata about servers</td>
</tr>
<tr>
<td>tagValue</td>
<td>User-defined value to further define a key's custom data or metadata about servers</td>
</tr>
</tbody>
</table>
Download the Discovery Connector

Data field | Description
--- | ---
toolsStatus | Operational state of VMware tools (See Viewing and Sorting Data Collectors (p. 62) for a complete list.)
totalDiskSize | Total capacity of disk expressed in MB
totalRAM | Total amount of RAM available on VM in MB
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, 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, and V6.5.

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.

```bash
$ md5sum AWSDiscoveryConnector.ova
$ sha256sum AWSDiscoveryConnector.ova
```
Deploy the Discovery Connector

Deploy the downloaded OVA file of the Discovery Connector 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 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 it's 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, select **Upload logs automatically** if you want to automatically upload your logs to make them available to AWS for troubleshooting 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. 35)**.
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 policy that you created in **Provide Application**
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, 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, 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.

---

**Start Discovery Connector Data Collection**

After you have deployed and configured the Discovery Connector in your VMware environment, you must complete the final step of actually turning on its data collection process. You can turn on data collection through the console or by making API calls through the AWS CLI. Instructions are provided below for both methods.

**Using the Migration Hub Console**

You start the Discovery Connector data collection process on the **Data Collectors** page of the Migration Hub console.

**To start 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.
4. Choose **Start 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.

**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](https://docs.aws.amazon.com/cli/latest/userguide/) 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. Type the following command:

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

   • If you don't know the ID of the connector you want to start, enter the following command exactly as shown to see the connector's ID:

   ```bash
   aws discovery describe-agents --filters condition=EQUALS,name=hostName,values=connector
   ```
Troubleshooting the Discovery Connector

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

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 VM, 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. 37)
- Credentials Issues (p. 38)

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 commandline 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. 33)) 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 Limits (p. 84).

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.

<table>
<thead>
<tr>
<th>Import Field Name</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
</table>
| 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 |
<p>| SMBiosId          | System management BIOS (SMBIOS) ID. | |</p>
<table>
<thead>
<tr>
<th>Import Field Name</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
</table>
| 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 execute 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  
<pre><code>              |             | 128 |
</code></pre>
<table>
<thead>
<tr>
<th>Import Field Name</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU.UsagePct.Avg</td>
<td>The average CPU utilization when the discovery tool was collecting data.</td>
<td>45 23.9</td>
</tr>
<tr>
<td>CPU.UsagePct.Max</td>
<td>The maximum CPU utilization when the discovery tool was collecting data.</td>
<td>55.34 24</td>
</tr>
<tr>
<td>DiskReadsPerSecondInKB.Avg</td>
<td>The average number of disk reads per second, in KB.</td>
<td>1159 84506</td>
</tr>
<tr>
<td>DiskWritesPerSecondInKB.Avg</td>
<td>The average number of disk writes per second, in KB.</td>
<td>199 6197</td>
</tr>
<tr>
<td>DiskReadsPerSecondInKB.Max</td>
<td>The maximum number of disk reads per second, in KB.</td>
<td>37892 869962</td>
</tr>
<tr>
<td>DiskWritesPerSecondInKB.Max</td>
<td>The maximum number of disk writes per second, in KB.</td>
<td>18436 1808</td>
</tr>
<tr>
<td>DiskReadsOpsPerSecond.Avg</td>
<td>The average number of disk read operations per second.</td>
<td>45 28</td>
</tr>
<tr>
<td>DiskWritesOpsPerSecond.Avg</td>
<td>The average number of disk write operations per second.</td>
<td>8 3</td>
</tr>
<tr>
<td>DiskReadsOpsPerSecond.Max</td>
<td>The maximum number of disk read operations per second.</td>
<td>1083 176</td>
</tr>
<tr>
<td>DiskWritesOpsPerSecond.Max</td>
<td>The maximum number of disk write operations per second.</td>
<td>535 71</td>
</tr>
<tr>
<td>NetworkReadsPerSecondInKB.Avg</td>
<td>The average number of network read operations per second, in KB.</td>
<td>45 28</td>
</tr>
<tr>
<td>NetworkWritesPerSecondInKB.Avg</td>
<td>The average number of network write operations per second, in KB.</td>
<td>8 3</td>
</tr>
<tr>
<td>NetworkReadsPerSecondInKB.Max</td>
<td>The maximum number of network read operations per second, in KB.</td>
<td>1083 176</td>
</tr>
<tr>
<td>NetworkWritesPerSecondInKB.Max</td>
<td>The maximum number of network write operations per second, in KB.</td>
<td>535 71</td>
</tr>
</tbody>
</table>
### 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, 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.

1. Sign in to the AWS Management Console and open the IAM console at [https://console.aws.amazon.com/iam/](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.

   ```json
   {
   }
   ```

---

<table>
<thead>
<tr>
<th>Import Field Name</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applications</td>
<td>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.</td>
<td>Application1, Application2, Application3</td>
</tr>
<tr>
<td>Tags</td>
<td>A comma-delimited list of tags formatted as name:value. Important: Do not store sensitive information (like personal data) in tags.</td>
<td>“zone:1, critical:yes”, “zone:3, critical:no, zone:1”</td>
</tr>
</tbody>
</table>

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. 48).

**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.
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**.

**AWS CLI 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 AWS CLI.

**To edit a user's attached 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.

   ```json
   {   "Version": "2012-10-17",
       "Statement": [   
           {   "Effect": "Allow",
               "Action": [   
                   "s3:ListBucket",
                   "s3:PutObject",
                   "s3:GetObject",
                   "s3:DeleteObject"
               ],
               "Resource": ["arn:aws:s3:::importBucket/""
           }   
       ]   
   }
   ```
For more information on using this command, see \texttt{create-policy} in the \textit{AWS CLI Command Reference}.

2. Use the \texttt{aws iam attach-user-policy} AWS CLI command to attach the policy you created in the last step 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 \texttt{attach-user-policy} in the \textit{AWS CLI Command Reference}.

Now that you've added the policy to your IAM user, you're ready to start the import process. Remember that when your user uploads object to the Amazon S3 bucket that you specified, that they leave the default permissions for the objects set so that the user can read the object.

For more information on how to upload your import file to Amazon S3, read on.

### 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

1. Sign in to the AWS Management Console and open the Amazon S3 console at \url{https://console.aws.amazon.com/s3/}.
2. In the \textbf{Bucket name} list, choose the name of the bucket that you want to upload your object to.
3. Choose \textbf{Upload}.
4. In the \textbf{Upload} dialog box, choose \textbf{Add files} to choose the file to upload.
5. Choose a file to upload, and then choose \textbf{Open}.
6. Choose \textbf{Upload}.
7. Once your file has been uploaded, choose the name of your data file object from your bucket dashboard.
8. From the \textbf{Overview} tab of the object details page, copy the \textbf{Object URL}. You'll need this when you create your import request.
9. Return to the and paste it in the \textbf{Data file link on S3} field on the \textbf{Start new import} page.

#### AWS CLI S3 Upload

1. Open a terminal window, and navigate to the directory that you've saved your import file to.
2. Type the following command:

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

3. This will return the following results:
4. Copy the full Amazon S3 object path that was returned. You’ll need this when you create your import request.

**Importing Data**

After you have downloaded the import template from the Migration Hub console and have populated it with your existing on-premises server data, you are ready to start importing the data into Migration Hub. There are two ways to do this: Through the console or by making API calls through the AWS CLI. Instructions are provided below for both ways.

**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-a5b42f82e46/import_template.csv
3. Choose the **Import** button in the **Import** box, which will take you to the **Import** page under **Tools**.
4. Choose **Start new import**.
5. In the next screen, specify a name for the import in the **Import name** field.
6. Fill out the **Data file link on S3** 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. 44).
7. 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 and 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.
To start data import

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

   ```bash
   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:

   ```json
   {
     "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:

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

   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. 81).
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. 48)
- Export Collected Data (p. 49)
- Data Exploration in Amazon Athena (p. 50)

View Collected Data Using the Console

After starting the data collection process of your Discovery Connector or Discovery Agent, you can use the console to view their collected data about your servers and VMs. Data appears in the console approximately 15 minutes after turning on data collection. 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. 49).

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. 59).

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**

After starting the data collection process of your Discovery Connector or Discovery Agent, 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 either from the console or from the AWS CLI.
- **Discovery Connector**, you can only export the collected data from the AWS CLI.

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

**Export System Performance Data for All Servers**

Collected data from all the Discovery Connectors and Discovery Agents running on your hosts and VMs can be bulk exported from 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).
   - Type `aws configure` and press Enter.
   - Enter your AWS Access Key Id and AWS Secret Access Key.
   - Enter `us-west-2` for the Default Region Name.
   - Enter `text` for Default Output Format.
3. Type the following command to generate an export ID:

   ```bash
   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":

   ```bash
   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. There, you can find the server’s export jobs listed at the bottom of the screen, underneath Exports. If no export jobs yet exist, the table is empty. You can execute up to five exports of server data at a time.

**To export collected data about a discovered server**

1. In the navigation pane, choose Server.
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, 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.

   **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 StartContinuousExport 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. 51)
2. Working with Data Exploration in Amazon Athena (p. 52)

Enabling Data Exploration in Amazon Athena

Before you can actually see and start exploring your discovered data in Amazon Athena, Data Exploration in Amazon Athena must first be 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. Data Exploration in Amazon Athena can also be enabled by Continuous Export explicitly being turned on through an API call from the AWS CLI. Instructions are provided below for both ways by expanding your method of choice:

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 current, detailed data that was discovered by your agents in Amazon Athena. You can query this data directly in Athena. With the data, you can generate spreadsheets, run a cost analysis, port the query to a visualization program to diagram network dependencies, and more.

The topics in this section provide instructions about the various ways you can work with your data in Amazon Athena to assess and plan for migrating your local environment to AWS.

**Topics**
- Explore Data Directly in Amazon Athena (p. 52)
- Visualize Amazon Athena Data (p. 53)
- Predefined Queries to use in Athena (p. 53)

## Explore Data Directly in Amazon Athena

These instructions guide you to all 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 [here](p. 51).

**To explore agent discovered data directly in Athena**

1. In the navigation pane, choose **Servers**.
2. Choose the **Explore data in Amazon Athena** link.

You will be taken to the Amazon Athena console where you will see:

- The **Query Editor** window
- In the navigation pane:
  - Database list box, which will have the default database pre-listed as `application_discovery_service_database`
  - Tables list consisting of seven tables representing the data sets grouped by the agents.
    - `os_info_agent`
    - `network_interface_agent`
    - `sys_performance_agent`
    - `processes_agent`
3. Query the data in the Amazon Athena console by writing and running your own SQL queries in the Athena Query Editor. Analyze details about your on-premises servers.

**Visualize 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** which will produce the **New Athena data source** dialog box.
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 data set creation** dialog box, select **Import to SPICE for quicker analytics** and choose **Visualize**.

Your visualization will be rendered.

**Predefined Queries to use in Athena**

Here you will find a set of predefined queries of typical use cases, such as TCO analysis and network visualization. You can use these queries as is or modify them to suit your needs. Simply expand the query you want to use and follow these instructions:

**To use a predefined query**

1. In the navigation pane, choose **Servers**.
2. Choose the **Explore data in Amazon Athena** link to be taken to your data in the Amazon Athena console.
3. Expand one of the predefined queries listed below and copy it.
4. Paste the query in the Athena's Query Editor window.
5. Choose **Run Query**.

**Obtain IP Addresses and Hostnames for Servers**

This helper function retrieves IP addresses and hostnames for a given server. This view can be used in other queries.

```sql
CREATE OR REPLACE VIEW hostname_ip_helper AS
SELECT DISTINCT
"oa"."host_name",
"nic"."agent_id",
"nic"."ip_address"
FROM
```
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.

```sql
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 `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.

```sql
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",
```

54
Track Outbound Communication Between Servers Based On Port Number and Process Details

Before running this query, perform the following procedures. Note that if you've already created the iana_service_ports_import table, you can skip the first procedure.

To create the iana_service_ports_import table

1. Download the IANA Port 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 and specify the Amazon S3 path to the object you uploaded in the previous step. For more information, see Step 2: Create a Table in the Amazon Athena User Guide.

Now that the iana_service_ports_import table has been created, you create two help functions for tracking outbound traffic. For information on how to create a view, see CREATE VIEW in the Amazon Athena User Guide.

To create outbound tracking helper functions

2. Create the view valid_outbound_ips_helper. This helper function gives the list of all distinct outbound source ip addresses and is as follows:

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

3. Create the view outbound_query_helper. This helper function determines the frequency of communication for outbound traffic and is as follows:

```sql
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";
```
Now that you have 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.

```sql
SELECT DISTINCT
  "hin1"."host_name" "Source Host Name",
  "hin2"."host_name" "Destination Host Name",
  "o"."source_ip" "Source IP Address",
  "o"."destination_ip" "Destination IP Address",
  "o"."frequency" "Connection Frequency",
  "o"."destination_port" "Destination Communication Port",
  "p"."name" "Process Name",
  "ianap"."service name" "Process Service Name",
  "ianap"."description" "Process Service Description"
FROM
  outbound_query_helper o,
  hostname_ip_helper hin1,
  hostname_ip_helper hin2,
  processes_agent p,
  iana_service_ports_import ianap
WHERE ((("o"."source_ip" = "hin1"."ip_address") AND ("o"."destination_ip" = "hin2"."ip_address")) AND ("p"."agent_assigned_process_id" = "o"."agent_assigned_process_id")) AND ("hin1"."host_name" <> "hin2"."host_name")
AND ((("o"."destination_port" = TRY_CAST("ianap"."port number" AS integer)) AND ("ianap"."transport protocol" = 'tcp')))
ORDER BY "hin1"."host_name" ASC, "o"."frequency" DESC;
```

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

Before running this query, perform the following procedures. Note that if you've already created the `iana_service_ports_import` table, you can skip the first procedure.

**To create the `iana_service_ports_import` table**

1. Download the IANA Port database CSV file from Service Name and Transport Protocol Port Number Registry on [iana.org](http://iana.org).
2. Upload the file to Amazon S3. For more information, see [How Do I Upload Files and Folders to an S3 Bucket?](https://docs.aws.amazon.com/AmazonS3/latest/userguide/Upload.htm).
3. Create a new table in Athena named `iana_service_ports_import` and specify the Amazon S3 path to the object you uploaded in the previous step. For more information, see [Step 2: Create a Table](https://docs.aws.amazon.com/AmazonAthena/latest/ug/Creating-the-first-table.html) in the Amazon Athena User Guide.

Now that the `iana_service_ports_import` table has been created, you create two help functions for tracking inbound traffic. For information on how to create a view, see [CREATE VIEW](https://docs.aws.amazon.com/AmazonAthena/latest/ug/Creating-view.html) in the Amazon Athena User Guide.

**To create import tracking helper functions**

2. Create the view `valid_inbound_ips_helper`. This helper function gives the list of all distinct inbound source ip addresses and is as follows:

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

3. Create the view `inbound_query_helper`. This helper function determines the frequency of communication for inbound traffic and is as follows:
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";

Now that you have 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 DISTINCT
  "hin1"."host_name" "Source Host Name",
  "hin2"."host_name" "Destination Host Name",
  "i"."source_ip" "Source IP Address",
  "i"."destination_ip" "Destination IP Address",
  "i"."frequency" "Connection Frequency",
  "i"."destination_port" "Destination Communication Port",
  "p"."name" "Process Name",
  "ianap"."service name" "Process Service Name",
  "ianap"."description" "Process Service Description"
FROM
  inbound_query_helper i,
  hostname_ip_helper hin1,
  hostname_ip_helper hin2,
  processes_agent p,
  iana_service_ports_import ianap
WHERE ((("i"."source_ip" = "hin1"."ip_address") AND ("i"."destination_ip" = "hin2"."ip_address")) AND ("p"."agent_assigned_process_id" = "i"."agent_assigned_process_id")) AND ("hin1"."host_name" <> "hin2"."host_name")
  AND ((("i"."destination_port" = TRY_CAST(ianap."port number" AS integer)) AND (ianap."transport protocol" = 'tcp')))
ORDER BY "hin1"."host_name" ASC, "i"."frequency" DESC;
```

### Identify Running Software From Port Number

This query can be used to identify the running software based on port numbers. Note that this query requires you to download the IANA Port database, which can be downloaded from the IANA website at the following URL: [https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.csv](https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.csv).

Before running this query, you'll need to create the `iana_service_ports_import` table. Note that if you've already created the table, you can skip this procedure.

### To create the `iana_service_ports_import` table

1. Download the IANA Port database CSV file from [Service Name and Transport Protocol Port Number Registry](https://iana.org) on [iana.org](https://iana.org).
2. Upload the file to Amazon S3. For more information, see [How Do I Upload Files and Folders to an S3 Bucket](https://aws.amazon.com/s3/).
3. Create a new table in Athena named `iana_service_ports_import` and specify the Amazon S3 path to the object you uploaded in the previous step. For more information, see Step 2: Create a Table in the Amazon Athena User Guide.

```
SELECT DISTINCT
    "o"."host_name" "Host Name",
    "ianap"."service name" "Service",
    "ianap"."description" "Description",
    "con"."destination_port",
    "count"("con"."destination_port") "Destination Port Count"
FROM
    inbound_connection_agent con,
    os_info_agent o,
    iana_service_ports_import ianap,
    network_interface_agent n
WHERE ((("con"."destination_ip" = "ni"."ip_address") AND (NOT ("con"."destination_ip"
LIKE '172%'))) AND ((("con"."destination_port" = "ianap"."port number") AND
("ianap"."transport protocol" = 'tcp')))) AND ("con"."agent_id" = "o"."agent_id") AND
("o"."agent_id" = "ni"."agent_id")
GROUP BY "o"."host_name", "ianap"."service name", "ianap"."description",
"con"."destination_port"
ORDER BY "Destination Port Count" DESC
```
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. 59)
- Data Collection Tools (p. 61)
- View, Export, and Explore Server Data (p. 63)

## Main Dashboard

The main dashboard is selected by default on the homepage of the AWS Migration Hub console, or, by choosing **Dashboard** in the 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. 59)
- Navigating from the Dashboard and the Navigation Pane (p. 60)

## 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 reach the main dashboard, choose **Dashboard** from the navigation pane, which is on the left side of the Migration Hub homepage.
Navigating from the Dashboard and the Navigation Pane

After you view dashboard data summaries, you might want to retrieve more detail. To do this, navigate directly from the relevant status or information box on the main dashboard.

In the table following, you can find instructions on how to navigate from a dashboard to the information you want to see. You can also find instructions on how to get to this information by using the navigation pane, which is on the left side of the Migration Hub homepage.

<table>
<thead>
<tr>
<th>To See</th>
<th>Do This in the Dashboard</th>
<th>Do This in the Navigation Pane</th>
</tr>
</thead>
<tbody>
<tr>
<td>All servers</td>
<td>In the main dashboard, in the</td>
<td>1. In the navigation pane, choose</td>
</tr>
<tr>
<td></td>
<td>Discovery summary box, choose Servers.</td>
<td>Servers.</td>
</tr>
<tr>
<td>All agents</td>
<td>In the main dashboard, in the</td>
<td>1. In the navigation pane, choose</td>
</tr>
<tr>
<td></td>
<td>Discovery summary box, choose Agents.</td>
<td>Data Collectors.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Choose the Agents tab.</td>
</tr>
</tbody>
</table>
### 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](#) and [AWS Application Discovery Agent](#).

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 Migration Hub console.

**Topics**
- [Starting and Stopping Data Collectors](#)
- [Viewing and Sorting Data Collectors](#)

## 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 Migration Hub console.

**To start or stop data collection tools**

1. In the navigation pane, 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.

<table>
<thead>
<tr>
<th>Search Criterion</th>
<th>Operator</th>
<th>Value: Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collection status</td>
<td></td>
<td>Started: Data is being collected and sent to Application Discovery Service.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Start scheduled: Data collection is scheduled to start. Data will be sent to Application Discovery Service on next ping, and status will change to <strong>Started</strong>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stopped: Data is not being collected or sent to Application Discovery Service.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stop scheduled: Data collection is scheduled to stop. Data will stop being sent to Application Discovery Service on next ping, and status will change to <strong>Stopped</strong>.</td>
</tr>
<tr>
<td>Health</td>
<td>==</td>
<td>Healthy: Data collection isn't turned on. The tool is functioning normally.</td>
</tr>
<tr>
<td></td>
<td>!=</td>
<td>Unhealthy: The tool is in an error state. Data isn't being collected or reported.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unknown: No connection established in over an hour.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Shutdown: The tool last communicated &quot;shutting down&quot; due to a system, service, or daemon shutdown. If a reboot or tool upgrade occurred, status will change to another state at the first reporting cycle.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Running: Data collection is turned on. The tool is functioning normally.</td>
</tr>
<tr>
<td>Hostname</td>
<td></td>
<td>For agents, any host name selected from the pre-populated list of hosts where an agent is installed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For connectors, not applicable.</td>
</tr>
<tr>
<td>IP address</td>
<td>==</td>
<td>Any IP address selected from the pre-populated list where a collection tool is installed.</td>
</tr>
<tr>
<td>Connector/Agent ID</td>
<td></td>
<td>Any connector or agent ID selected from the pre-populated list from which a collection tool is installed.</td>
</tr>
</tbody>
</table>

**To sort data collectors by applying search filters**

1. In the navigation pane, 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. 63)
- Tagging Servers (p. 63)
- Exporting Server Data (p. 64)
- Data Exploration in Athena (p. 65)
- Applications (p. 65)

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 navigation pane, 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. 64).

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 navigation pane, 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 navigation pane, 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 tag.
4. In the dialog box, type a value in the **Key** field, and optionally a value in the **Value** field.
   - Add more tags by choosing Additional tag and adding more information.
5. Choose Add Tags. A green confirmation message will be displayed at the top of the screen.
6. Optionally, tags can be added for an individual server from its detail page by choosing Actions, and then Add tag and repeating the above steps.

**To remove tags from one or more servers**

1. In the navigation pane, choose Servers.
2. In the Server info column, choose the server link for the server that you want to remove tags from. Click inside the check boxes of multiple servers to remove tags from more than one server at a time.
3. For Actions, choose Remove tag.
4. Select each tag you want to remove, or choose select all.
5. Choose Remove. A green confirmation message appears at the top of the screen.
6. Optionally, tags can be removed for an individual server from its detail page by choosing Actions, and then Remove tag and repeating the above steps.

## 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 System Performance Data for All Servers (p. 49). Or, if you want to bulk export data for all servers where agents have been installed, see Data Exploration in Amazon Athena (p. 50).

**To export detailed server data**

1. In the navigation pane, 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-progess**; 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 StartContinuousExport 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. 50).

### 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 navigation pane, 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**

<table>
<thead>
<tr>
<th>Field</th>
<th>Mandatory</th>
</tr>
</thead>
<tbody>
<tr>
<td>server.agentId</td>
<td></td>
</tr>
<tr>
<td>server.applications</td>
<td></td>
</tr>
<tr>
<td>server.applications.hasMoreValues</td>
<td></td>
</tr>
<tr>
<td>server.configurationId</td>
<td>x</td>
</tr>
<tr>
<td>server.cpuType</td>
<td></td>
</tr>
<tr>
<td>server.hostName</td>
<td></td>
</tr>
<tr>
<td>server.hypervisor</td>
<td></td>
</tr>
<tr>
<td>server.networkInterfaceInfo</td>
<td></td>
</tr>
<tr>
<td>server.networkInterfaceInfo.hasMoreValues</td>
<td></td>
</tr>
</tbody>
</table>
Using the DescribeConfigurations Action

### Field

<table>
<thead>
<tr>
<th>Field</th>
<th>Mandatory</th>
</tr>
</thead>
<tbody>
<tr>
<td>server.osName</td>
<td></td>
</tr>
<tr>
<td>server.osVersion</td>
<td></td>
</tr>
<tr>
<td>server.tags</td>
<td></td>
</tr>
<tr>
<td>server.tags.hasMoreValues</td>
<td></td>
</tr>
<tr>
<td>server.timeOfCreation</td>
<td>X</td>
</tr>
<tr>
<td>server.type</td>
<td></td>
</tr>
<tr>
<td>server.performance.avgCpuUsagePct</td>
<td></td>
</tr>
<tr>
<td>server.performance.avgDiskReadIOPS</td>
<td></td>
</tr>
<tr>
<td>server.performance.avgDiskReadsPerSecondInKB</td>
<td></td>
</tr>
<tr>
<td>server.performance.avgDiskWriteIOPS</td>
<td></td>
</tr>
<tr>
<td>server.performance.avgDiskWritesPerSecondInKB</td>
<td></td>
</tr>
<tr>
<td>server.performance.avgFreeRAMInKB</td>
<td></td>
</tr>
<tr>
<td>server.performance.avgNetworkReadsPerSecondInKB</td>
<td></td>
</tr>
<tr>
<td>server.performance.avgNetworkWritesPerSecondInKB</td>
<td></td>
</tr>
<tr>
<td>server.performance.avgNetworkReadsPerSecondInKB</td>
<td></td>
</tr>
<tr>
<td>server.performance.avgNetworkWritesPerSecondInKB</td>
<td></td>
</tr>
<tr>
<td>server.performance.maxCpuUsagePct</td>
<td></td>
</tr>
<tr>
<td>server.performance.maxDiskReadIOPS</td>
<td></td>
</tr>
<tr>
<td>server.performance.maxDiskReadsPerSecondInKB</td>
<td></td>
</tr>
<tr>
<td>server.performance.maxDiskWriteIOPS</td>
<td></td>
</tr>
<tr>
<td>server.performance.maxNetworkReadsPerSecondInKB</td>
<td></td>
</tr>
<tr>
<td>server.performance.maxNetworkWritesPerSecondInKB</td>
<td></td>
</tr>
<tr>
<td>server.performance.minFreeRAMInKB</td>
<td></td>
</tr>
<tr>
<td>server.performance.numCores</td>
<td></td>
</tr>
<tr>
<td>server.performance.numCpus</td>
<td></td>
</tr>
<tr>
<td>server.performance.numDisks</td>
<td></td>
</tr>
<tr>
<td>server.performance.numNetworkCards</td>
<td></td>
</tr>
<tr>
<td>server.performance.totalRAMInKB</td>
<td></td>
</tr>
</tbody>
</table>

### Process assets

<table>
<thead>
<tr>
<th>Field</th>
<th>Mandatory</th>
</tr>
</thead>
<tbody>
<tr>
<td>process.commandLine</td>
<td></td>
</tr>
</tbody>
</table>
Using the ListConfigurationsAction

The ListConfigurations action 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 ListConfigurations action. The ones marked as mandatory are always present in the output.

### Server assets

<table>
<thead>
<tr>
<th>Field</th>
<th>Mandatory</th>
</tr>
</thead>
<tbody>
<tr>
<td>server.configurationId</td>
<td>x</td>
</tr>
<tr>
<td>server.agentId</td>
<td></td>
</tr>
<tr>
<td>server.hostName</td>
<td></td>
</tr>
<tr>
<td>server.osName</td>
<td></td>
</tr>
<tr>
<td>server.osVersion</td>
<td></td>
</tr>
<tr>
<td>server.timeOfCreation</td>
<td>x</td>
</tr>
<tr>
<td>server.type</td>
<td></td>
</tr>
</tbody>
</table>

### Process assets

<table>
<thead>
<tr>
<th>Field</th>
<th>Mandatory</th>
</tr>
</thead>
<tbody>
<tr>
<td>process.configurationId</td>
<td>x</td>
</tr>
<tr>
<td>process.name</td>
<td></td>
</tr>
<tr>
<td>process.path</td>
<td></td>
</tr>
<tr>
<td>process.timeOfCreation</td>
<td>x</td>
</tr>
</tbody>
</table>
Using the ListConfigurationsAction

<table>
<thead>
<tr>
<th>Field</th>
<th>Mandatory</th>
</tr>
</thead>
<tbody>
<tr>
<td>process.commandLine</td>
<td></td>
</tr>
<tr>
<td>process.configurationId</td>
<td>x</td>
</tr>
<tr>
<td>process.name</td>
<td></td>
</tr>
<tr>
<td>process.path</td>
<td></td>
</tr>
<tr>
<td>process.timeOfCreation</td>
<td>x</td>
</tr>
<tr>
<td>server.agentId</td>
<td></td>
</tr>
<tr>
<td>server.configurationId</td>
<td>x</td>
</tr>
</tbody>
</table>

**Application assets**

<table>
<thead>
<tr>
<th>Field</th>
<th>Mandatory</th>
</tr>
</thead>
<tbody>
<tr>
<td>application.configurationId</td>
<td>x</td>
</tr>
<tr>
<td>application.description</td>
<td></td>
</tr>
<tr>
<td>application.name</td>
<td>x</td>
</tr>
<tr>
<td>application.serverCount</td>
<td>x</td>
</tr>
<tr>
<td>application.timeOfCreation</td>
<td>x</td>
</tr>
<tr>
<td>application.lastModifiedTime</td>
<td>x</td>
</tr>
</tbody>
</table>

**Connection assets**

<table>
<thead>
<tr>
<th>Field</th>
<th>Mandatory</th>
</tr>
</thead>
<tbody>
<tr>
<td>connection.destinationIp</td>
<td>x</td>
</tr>
<tr>
<td>connection.destinationPort</td>
<td>x</td>
</tr>
<tr>
<td>connection.ipVersion</td>
<td>x</td>
</tr>
<tr>
<td>connection.latestTimestamp</td>
<td>x</td>
</tr>
<tr>
<td>connection.occurrence</td>
<td>x</td>
</tr>
<tr>
<td>connection.sourceIp</td>
<td>x</td>
</tr>
<tr>
<td>connection.transportProtocol</td>
<td></td>
</tr>
<tr>
<td>destinationProcess.configurationId</td>
<td></td>
</tr>
<tr>
<td>destinationProcess.name</td>
<td></td>
</tr>
<tr>
<td>destinationServer.configurationId</td>
<td></td>
</tr>
<tr>
<td>destinationServer.hostName</td>
<td></td>
</tr>
</tbody>
</table>
Using the `ListConfigurations` Action

<table>
<thead>
<tr>
<th>Field</th>
<th>Mandatory</th>
</tr>
</thead>
<tbody>
<tr>
<td>sourceProcess.configurationId</td>
<td></td>
</tr>
<tr>
<td>sourceProcess.name</td>
<td></td>
</tr>
<tr>
<td>sourceServer.configurationId</td>
<td></td>
</tr>
<tr>
<td>sourceServer.hostName</td>
<td></td>
</tr>
</tbody>
</table>

**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

<table>
<thead>
<tr>
<th>Filter</th>
<th>Supported conditions</th>
<th>Supported values</th>
<th>Supported sorting</th>
</tr>
</thead>
<tbody>
<tr>
<td>server.configurationId</td>
<td><strong>EQUALS</strong></td>
<td>• Any valid server</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>• NOT_EQUALS</td>
<td>configuration ID</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• EQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>server.hostName</td>
<td><strong>EQUALS</strong></td>
<td>• String</td>
<td>• ASC</td>
</tr>
<tr>
<td></td>
<td>• NOT_EQUALS</td>
<td></td>
<td>• DESC</td>
</tr>
<tr>
<td></td>
<td>• EQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NOT_CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>server.osName</td>
<td><strong>EQUALS</strong></td>
<td>• String</td>
<td>• ASC</td>
</tr>
<tr>
<td></td>
<td>• NOT_EQUALS</td>
<td></td>
<td>• DESC</td>
</tr>
<tr>
<td></td>
<td>• EQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NOT_CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>server.osVersion</td>
<td><strong>EQUALS</strong></td>
<td>• String</td>
<td>• ASC</td>
</tr>
<tr>
<td></td>
<td>• NOT_EQUALS</td>
<td></td>
<td>• DESC</td>
</tr>
<tr>
<td></td>
<td>• EQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NOT_CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>server.agentId</td>
<td><strong>EQUALS</strong></td>
<td>• String</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>• NOT_EQUALS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• EQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>server.connectorId</td>
<td><strong>EQUALS</strong></td>
<td>• String</td>
<td>None</td>
</tr>
</tbody>
</table>

70
Using the `ListConfigurations` Action

<table>
<thead>
<tr>
<th>Filter</th>
<th>Supported conditions</th>
<th>Supported values</th>
<th>Supported sorting</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>server.type</strong></td>
<td>• EQUALS</td>
<td>String with one of the following values:</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>• NOT_EQUALS</td>
<td>• EC2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• EQ</td>
<td>• OTHER</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NE</td>
<td>• VMWARE_VM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• VMWARE_HOST</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• VMWARE_VM_TEMPLATE</td>
<td></td>
</tr>
<tr>
<td><strong>server.vmWareInfo.morefId</strong></td>
<td>• EQUALS</td>
<td>• String</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>• NOT_EQUALS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• EQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NOT_CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>server.vmWareInfo.vcenterId</strong></td>
<td>• EQUALS</td>
<td>• String</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>• NOT_EQUALS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• EQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NOT_CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>server.vmWareInfo.hostId</strong></td>
<td>• EQUALS</td>
<td>• String</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>• NOT_EQUALS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• EQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NOT_CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>server.networkInterfaceInfo.portGroupId</strong></td>
<td>• EQUALS</td>
<td>• String</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>• NOT_EQUALS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• EQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NOT_CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>server.networkInterfaceInfo.portGroupName</strong></td>
<td>• EQUALS</td>
<td>• String</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>• NOT_EQUALS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• EQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NOT_CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Filter</td>
<td>Supported conditions</td>
<td>Supported values</td>
<td>Supported sorting</td>
</tr>
<tr>
<td>-------------------------</td>
<td>----------------------</td>
<td>------------------</td>
<td>-------------------</td>
</tr>
</tbody>
</table>
| server.networkInterfaceInfo.virtualSwitchName | • NOT_EQUALS  
• EQ  
• NE  
• CONTAINS  
• NOT_CONTAINS | String | None |
| server.networkInterfaceInfo.ipAddress | • NOT_EQUALS  
• EQ  
• NE  
• CONTAINS  
• NOT_CONTAINS | String | None |
| server.networkInterfaceInfo.macAddress | • NOT_EQUALS  
• EQ  
• NE  
• CONTAINS  
• NOT_CONTAINS | String | None |
| server.performance.avgCpuUsagePct | • 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 |
### Using the `ListConfigurations` Action

#### Filters

<table>
<thead>
<tr>
<th>Filter</th>
<th>Supported conditions</th>
<th>Supported values</th>
<th>Supported sorting</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>server.tag.key</code></td>
<td>EQUALS</td>
<td>String</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>NOT_EQUALS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>EQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NOT_CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>server.application.name</code></td>
<td>EQUALS</td>
<td>String</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>NOT_EQUALS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>EQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NOT_CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>server.application.description</code></td>
<td>EQUALS</td>
<td>String</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>NOT_EQUALS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>EQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NOT_CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>server.application.configurationId</code></td>
<td>EQUALS</td>
<td>Any valid application configuration ID</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>NOT_EQUALS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>EQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NE</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>server.process.configurationId</code></td>
<td>EQ</td>
<td></td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>NOT_EQUALS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NOT_CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>server.process.name</code></td>
<td>EQUALS</td>
<td>String</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>NOT_EQUALS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>EQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NOT_CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>server.process.commandLine</code></td>
<td>EQUALS</td>
<td>String</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>NOT_EQUALS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>EQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NOT_CONTAINS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Application assets

73
### Using the `ListConfigurations` Action

#### Filters

<table>
<thead>
<tr>
<th>Filter</th>
<th>Supported conditions</th>
<th>Supported values</th>
<th>Supported sorting</th>
</tr>
</thead>
</table>
| `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 | • String | • ASC  
  • DESC |
| `application.timeOfCreation` | Filtering not supported  
  Filtering not supported | • String | • ASC  
  • DESC |
| `application.lastModifiedTime` | Filtering not supported  
  Filtering not supported | • String | • ASC  
  • DESC |
| `server.configurationId` | • EQUALS  
  • NOT_EQUALS  
  • EQ  
  • NE | • ServerId | None |
| `process.configurationId` | • EQUALS  
  • NOT_EQUALS  
  • EQ  
  • NE | • ProcessId | None |
| `process.name` | • EQUALS  
  • NOT_EQUALS  
  • EQ  
  • NE  
  • CONTAINS  
  • NOT_CONTAINS | • String | • ASC  
  • DESC |
### Using the ListConfigurationsAction

<table>
<thead>
<tr>
<th>Filter</th>
<th>Supported conditions</th>
<th>Supported values</th>
<th>Supported sorting</th>
</tr>
</thead>
<tbody>
<tr>
<td>process.commandLine</td>
<td>• EQUALS</td>
<td>• String</td>
<td>• ASC</td>
</tr>
<tr>
<td></td>
<td>• NOT_EQUALS</td>
<td></td>
<td>• DESC</td>
</tr>
<tr>
<td></td>
<td>• EQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NOT_CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>server.configurationId</td>
<td>• EQUALS</td>
<td>• ServerId</td>
<td>• ASC</td>
</tr>
<tr>
<td></td>
<td>• NOT_EQUALS</td>
<td></td>
<td>• DESC</td>
</tr>
<tr>
<td></td>
<td>• EQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NOT_CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>server.hostName</td>
<td>• EQUALS</td>
<td>• String</td>
<td>• ASC</td>
</tr>
<tr>
<td></td>
<td>• NOT_EQUALS</td>
<td></td>
<td>• DESC</td>
</tr>
<tr>
<td></td>
<td>• EQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NOT_CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>server.osName</td>
<td>• EQUALS</td>
<td>• String</td>
<td>• ASC</td>
</tr>
<tr>
<td></td>
<td>• NOT_EQUALS</td>
<td></td>
<td>• DESC</td>
</tr>
<tr>
<td></td>
<td>• EQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NOT_CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>server.osVersion</td>
<td>• EQUALS</td>
<td>• String</td>
<td>• ASC</td>
</tr>
<tr>
<td></td>
<td>• NOT_EQUALS</td>
<td></td>
<td>• DESC</td>
</tr>
<tr>
<td></td>
<td>• EQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NOT_CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>server.agentId</td>
<td>• EQUALS</td>
<td>• String</td>
<td>• ASC</td>
</tr>
<tr>
<td></td>
<td>• NOT_EQUALS</td>
<td></td>
<td>• DESC</td>
</tr>
<tr>
<td></td>
<td>• EQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• CONTAINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NOT_CONTAINS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Connection assets

<table>
<thead>
<tr>
<th>Filter</th>
<th>Supported conditions</th>
<th>Supported values</th>
<th>Supported sorting</th>
</tr>
</thead>
<tbody>
<tr>
<td>connection.sourceIp</td>
<td>• EQUALS</td>
<td>• IP</td>
<td>• ASC</td>
</tr>
<tr>
<td></td>
<td>• NOT_EQUALS</td>
<td></td>
<td>• DESC</td>
</tr>
<tr>
<td></td>
<td>• EQ</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

75
<table>
<thead>
<tr>
<th>Filter</th>
<th>Supported conditions</th>
<th>Supported values</th>
<th>Supported sorting</th>
</tr>
</thead>
<tbody>
<tr>
<td>connection.destinationIp</td>
<td>EQUALS, NE, CONTAINS, NOT_CONTAINS</td>
<td>IP</td>
<td>ASC, DESC</td>
</tr>
<tr>
<td>connection.destinationPort</td>
<td>EQUALS, NE, CONTAINS, NOT_CONTAINS</td>
<td>Integer</td>
<td>ASC, DESC</td>
</tr>
<tr>
<td>sourceServer.configurationId</td>
<td>EQUALS, NE, CONTAINS, NOT_CONTAINS</td>
<td>ServerId</td>
<td>ASC, DESC</td>
</tr>
<tr>
<td>sourceServer.hostName</td>
<td>EQUALS, NE, CONTAINS, NOT_CONTAINS</td>
<td>String</td>
<td>ASC, DESC</td>
</tr>
<tr>
<td>destinationServer.osName</td>
<td>EQUALS, NE, CONTAINS, NOT_CONTAINS</td>
<td>String</td>
<td>ASC, DESC</td>
</tr>
<tr>
<td>destinationServer.osVersion</td>
<td>EQUALS, NE, CONTAINS, NOT_CONTAINS</td>
<td>String</td>
<td>ASC, DESC</td>
</tr>
<tr>
<td>destinationServer.agentId</td>
<td>EQUALS, NE, CONTAINS, NOT_CONTAINS</td>
<td>String</td>
<td>ASC, DESC</td>
</tr>
</tbody>
</table>
### Using the ListConfigurationsAction

#### Supported conditions

<table>
<thead>
<tr>
<th>Filter</th>
<th>Supported conditions</th>
<th>Supported values</th>
<th>Supported sorting</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>sourceProcess.configurationId</code></td>
<td><code>EQUALS</code>&lt;br&gt;<code>NOT_EQUALS</code>&lt;br&gt;<code>EQ</code>&lt;br&gt;<code>NE</code>&lt;br&gt;<code>CONTAINS</code>&lt;br&gt;<code>NOT_CONTAINS</code></td>
<td><code>ProcessId</code></td>
<td></td>
</tr>
<tr>
<td><code>sourceProcess.name</code></td>
<td><code>EQUALS</code>&lt;br&gt;<code>NOT_EQUALS</code>&lt;br&gt;<code>EQ</code>&lt;br&gt;<code>NE</code>&lt;br&gt;<code>CONTAINS</code>&lt;br&gt;<code>NOT_CONTAINS</code></td>
<td><code>String</code></td>
<td>ASC&lt;br&gt;DESC</td>
</tr>
<tr>
<td><code>sourceProcess.commandLine</code></td>
<td><code>EQUALS</code>&lt;br&gt;<code>NOT_EQUALS</code>&lt;br&gt;<code>EQ</code>&lt;br&gt;<code>NE</code>&lt;br&gt;<code>CONTAINS</code>&lt;br&gt;<code>NOT_CONTAINS</code></td>
<td><code>String</code></td>
<td>ASC&lt;br&gt;DESC</td>
</tr>
<tr>
<td><code>destinationProcess.configurationId</code></td>
<td><code>EQUALS</code>&lt;br&gt;<code>NOT_EQUALS</code>&lt;br&gt;<code>EQ</code>&lt;br&gt;<code>NE</code>&lt;br&gt;<code>CONTAINS</code>&lt;br&gt;<code>NOT_CONTAINS</code></td>
<td><code>ProcessId</code></td>
<td></td>
</tr>
<tr>
<td><code>destinationProcess.name</code></td>
<td><code>EQUALS</code>&lt;br&gt;<code>NOT_EQUALS</code>&lt;br&gt;<code>EQ</code>&lt;br&gt;<code>NE</code>&lt;br&gt;<code>CONTAINS</code>&lt;br&gt;<code>NOT_CONTAINS</code></td>
<td><code>String</code></td>
<td>ASC&lt;br&gt;DESC</td>
</tr>
<tr>
<td><code>destinationprocess.commandLine</code></td>
<td><code>EQUALS</code>&lt;br&gt;<code>NOT_EQUALS</code>&lt;br&gt;<code>EQ</code>&lt;br&gt;<code>NE</code>&lt;br&gt;<code>CONTAINS</code>&lt;br&gt;<code>NOT_CONTAINS</code></td>
<td><code>String</code></td>
<td>ASC&lt;br&gt;DESC</td>
</tr>
</tbody>
</table>
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
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`

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.
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 Step 3: Provide Application Discovery Service Access to Non-Administrator Users by Attaching Policies (p. 6).

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

• 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 Step 3: Provide Application Discovery Service Access to Non-Administrator Users by Attaching Policies (p. 6).

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 limit was reached – There are limits associated with import tasks. If you make an import task request that would exceed one of the limits, then the request will fail and return an error. For example, if you import 5,000 servers from a single import file, and then try to import that file again, it will fail the second time. For more information, see
• 25,000 imported records per account.

• 5,000 imported servers per account.

• 25,000 deletions of import records per 24 hour period, starting every day at 00:00 UTC.

• 400 servers per application.

You can take the following steps to request an increase for these limits. These increases are not granted immediately, so it may take a couple of days for your increase to become effective.

**To request a limit increase**

1. Open the AWS Support Center page, sign in, if necessary, and then choose **Create Case**.
2. Under **Regarding**, choose **Service Limit Increase**.
3. Under **Limit Type**, choose the type of limit to increase, fill in the necessary fields in the form, and then choose your preferred method of contact.

(p. 84).

**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**

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.:

```bash
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. 45).
AWS Application Discovery Service

Limits

Application Discovery Service has the following limits per account:

- **Applications** – 1,000 applications per account. If you reach this limit, 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](https://docs.aws.amazon.com/ataug/latest/userguide/).

Application Discovery Service has the following limits for discovery:

- **Import** – Each import file can have a maximum file size of 10 MB.
- **Agentless discovery** – The service limits you to 10 GB of data per day. If you reach this limit, the service doesn't process any more data for that day. If you frequently reach this limit, contact AWS Support about extending the limit.
- **Agent-based discovery** – currently, the following limits are enforced:
  - 1,000 active agents — agents that are collecting and sending data to Application Discovery Service in the cloud.
  - 10,000 inactive agents — agents that are responsive but not collecting data.
  - 10 GB of data per day — collected by all agents associated with a given AWS account.
  - 90 days of data storage — after which the data is purged.

**Limits That Can Be Increased**

Following are the limits for Application Discovery Service that can be increased by contacting AWS Support.

**Import Limits**

- 25,000 imported records per account.
- 5,000 imported servers per account.
- 25,000 deletions of import records per 24 hour period, starting every day at 00:00 UTC.
- 400 servers per application.

You can take the following steps to request an increase for these limits. These increases are not granted immediately, so it may take a couple of days for your increase to become effective.

**To request a limit increase**

1. Open the [AWS Support Center](https://console.aws.amazon.com/support/home) page, sign in, if necessary, and then choose **Create Case**.
2. Under **Regarding**, choose **Service Limit Increase**.
3. Under **Limit Type**, choose the type of limit to increase, fill in the necessary fields in the form, and then choose your preferred method of contact.
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.

<table>
<thead>
<tr>
<th>update-history-change</th>
<th>update-history-description</th>
<th>update-history-date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introducing the Home Region (p. 85)</td>
<td>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.</td>
<td>November 20, 2019</td>
</tr>
<tr>
<td>Introducing the Migration Hub import feature (p. 85)</td>
<td>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.</td>
<td>January 18, 2019</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Feature</td>
<td>Updated docs to support Data Exploration in Amazon Athena and added Troubleshooting chapter.</td>
<td>August 09, 2018</td>
</tr>
<tr>
<td>Major revision</td>
<td>Rewrites to usage &amp; output details; entire document restructured.</td>
<td>May 25, 2018</td>
</tr>
<tr>
<td>Discovery Agent 2.0</td>
<td>A new and improved Application Discovery agent was released.</td>
<td>October 19, 2017</td>
</tr>
<tr>
<td>Console</td>
<td>The AWS Management Console was added.</td>
<td>December 19, 2016</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Agentless discovery</td>
<td>This release describes how to set up and configure agentless discovery.</td>
<td>July 28, 2016</td>
</tr>
<tr>
<td>New details for Microsoft Windows Server and command issue fixes</td>
<td>This update adds details about Microsoft Windows Server. It also documents fixes to various command issues.</td>
<td>May 20, 2016</td>
</tr>
<tr>
<td>Initial publication</td>
<td>This is the first release of the Application Discovery Service User Guide.</td>
<td>May 12, 2016</td>
</tr>
</tbody>
</table>
AWS Glossary

For the latest AWS terminology, see the AWS Glossary in the AWS General Reference.