AWS Application Discovery Service

User Guide

AWS Application Discovery Service: User Guide

Table of Contents

What Is AWS Application Discovery Service?

AWS Application Discovery Service collects and presents data to enable enterprise customers to understand the configuration, usage, and behavior of servers in their on-premises environment. Server data is stored securely by the service, where it can be tagged and grouped into applications to help with AWS migration planning. The data can be exported for analysis in Excel or other cloud migration analysis tools. For more information, see AWS Application Discovery Service FAQ.

Application Discovery Service offers two modes of operation:

- **Agentless discovery** mode is recommended for environments that use VMware vCenter Server. This mode doesn't require you to install an agent on each host. Agentless discovery gathers server information regardless of the operating systems, which minimizes the time required for initial on-premises infrastructure assessment. It collects static configuration data including server hostnames, IP addresses, MAC addresses, CPU allocation, network throughput, memory allocation, disk resource allocations, and DNS servers. It also captures resource utilization metrics such as CPU usage and memory usage.
- **Agent-based discovery** mode collects a richer set of data than agentless discovery by using Amazon software, the AWS Application Discovery agent, which you install on one or more hosts in your data center. The agent captures infrastructure and server information including system configuration, system performance, running processes, and details of the network connections between systems. The information collected by agents is secured at rest and in transit to the Application Discovery Service data store in the cloud. Agent-based discovery works in both VMware and non-virtualized environments.

For VMware environments, we recommend that you run the Agentless Discovery Connector first to perform the initial infrastructure assessment, and then to install agents selectively on individual VMs to gather additional details. For non-VMware environments, including physical servers, you can use agent-based discovery.

You can run agent-based and agentless discovery simultaneously. Use agentless discovery to quickly complete the initial infrastructure assessment and then install agents on select hosts.

> **Important**
> Application Discovery Service doesn't gather sensitive information. All data is handled according to the AWS Privacy Policy. For additional security, you can operate Application Discovery Service offline to inspect collected data before it is shared with the service.

For more information about the data that Application Discovery Service collects, see AWS Application Discovery Service Components (p. 2).

Contents

Prerequisites

Before using using the Application Discovery Service, confirm that your on-premises servers and your network environment meet the following requirements.

Supported Operating Systems

The Application Discovery agent supports the following operating systems and system versions.

Linux

- Amazon Linux 2012.03, 2015.03
- 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
- Windows Server 2008 R2 SP1
- Windows Server 2012 R1
- Windows Server 2012 R2
- Windows Server 2016

Firewall Configuration

The AWS Application Discovery Agent requires outbound access to arsenal.us-west-2.amazonaws.com:443. It does not require any inbound ports to be open. Agents also work with transparent web proxies.

AWS Application Discovery Service Components

AWS Application Discovery Service uses a combination of AWS software agents and cloud-based services to identify, map, and store an inventory of the assets in your computing environment.

Contents

Arsenal

Arsenal is an agent service managed and hosted by AWS that sends data from AWS Application Discovery Agents and the AWS Agentless Discovery Connector to Application Discovery Service in the cloud. The word *arsenal* is included in some URLs and IAM policies.

AWS Agentless Discovery Connector

Agentless discovery uses the AWS Agentless Discovery Connector to communicate with Arsenal. Install the connector as a virtual machine (VM) in your VMware vCenter Server environment using an Open Virtualization Archive (OVA) file. When you start the connector, it registers with Arsenal, and queries Arsenal for configuration information. When you send a command for the connector to start collecting data, it connects to VMWare vCenter Server and collects information about all the VMs and hosts managed by this specific vCenter. The collected data is sent to Arsenal using Secure Sockets Layer (SSL) encryption. The connector is configured to automatically upgrade when new versions of the connector become available. You can change this configuration setting at any time.

Agentless discovery relies on VMware metadata, and can therefore collect information about any server running in your VMware environment, regardless of operating system.

Data Collected by Agentless Discovery

Agentless discovery does not collect information about your applications. It only collects information about your VMware vCenter Server hosts and VMs, including performance data about those hosts and VMs. The information collected by agentless discovery is shown in the following tables. Items in **bold** are only captured if VMware vCenter Server tools are installed. Agentless discovery attempts to collect all the following data; however, in some situations, vCenter might not have the data, as noted in the following tables.

Inventory Data about VMs in vCenter

Data	Data availability
Timestamp	Guaranteed
OSType	If available
SystemRelease	If available
MoRefID (Unique vCenter Managed Object Reference ID)	Guaranteed
instanceUuid (Unique ID for a virtual machine, not for the host system)	If available
FolderPath (VM folder path in vCenter)	Guaranteed
Name (Name of the vCenter VM or host)	Guaranteed
Hostname	If available
Hypervisor	Guaranteed
Manufacturer	Guaranteed
ToolsStatus (VMware tools status)	If available

Data	Data availability
HostSystem (MoRefId of the VM or host system)	Guaranteed for VM
Datacenter (MoRefID of the data center where the system is located)	Guaranteed
Type (Host or VM)	Guaranteed
vCenterId (Unique vCenter ID)	Guaranteed
smBiosId	If available
MacAddress	Guaranteed
IpAddress	If available
Network - List (A VMware object representation of a network)	If available
macAddress (For the network)	Guaranteed if the network exists
portGroupName (For the network)	If available
portGroupId (For the network)	If available
virtualSwitchName	If available
Name (Network name specified by the user)	If available
CPUType (vCPU for a VM, actual model for a host)	If available

Performance Data for VMs in vCenter

Data	Guaranteed or If Available
Timestamp	Guaranteed
MoRefID (Managed Object Reference ID of the system producing the metrics)	Guaranteed

Data	Guaranteed or If Available
Type (Host or VM)	Guaranteed
vCenterId (Unique ID of the vCenter)	Guaranteed
smBiosId	If available
PowerState	Guaranteed
MemorySize (Memory size of the VM/host)	If available
MemoryReservation (Reservation set for a VM)	If available
ActiveRAM (Average RAM over the polling period)	If available
MaxActiveRam (Max RAM over polling period)	If available
NetworkCards	If available
Name (Name associated with the metrics collected)	If available
BytesReadPerSecond (Average over the polling period)	If available
BytesWrittenPerSecond (Average over the polling period)	If available
TotalUsage (Average transmitted/received over the polling period)	If available
MaxTotalUsage (Max transmitted/received over the polling period)	If available
Disks	If available
DeviceID (Name associated with metrics collected; for a virtual device, it is the SCSI ID)	If available
Name	If available

Data	Guaranteed or If Available
Capacity	If available
scsi (For mapping performance metrics to a virtual disk)	If available
BytesReadPerSecond (Average over the polling period)	If available
BytesWrittenPerSecond (Average over the polling period)	If available
ReadOpsPerSecond (Average over the polling period)	If available
WriteOpsPerSecond (Average over the polling period)	If available
Cpus	If available
Name (Name associated with the metrics collected)	If available
UsagePct	If available
UsageMHz (Average over the polling period)	If available
MaxUsageMHz (Max over the polling period)	If available
numCores	If available
speedMHz	If available
reservationMHz (Reservation set for a VM)	If available

AWS Application Discovery Agent

The AWS Application Discovery Agent is AWS software that you install on on-premises servers and VMs targeted for discovery and migration. Agents run on Linux and Windows and collect server configuration and activity information about your applications and infrastructure. You can also install the agent on Amazon EC2 instances. When you start an agent, it registers with Arsenal and frequently pings the service for configuration information. When you send a command that tells an agent to start collecting data, it collects an extensive amount of data for the host or VM where it resides, including TCP connections to other hosts or VMs, which can help you map your IT assets without having to install an agent on every host or VM. Agents are configured to upgrade automatically when new versions become available. You can change this configuration setting at any time.

The agent does not require device drivers or kernel modules and operates only in user space. This follows industry best practices and reduces the risk of system instability.

After you install agents, you manage them using the Application Discovery Service API. The API includes actions to start and stop data collection. You can also retrieve information about agents, including the host names where agents reside, their health, and the version number of each agent. For more information, see the Application Discovery Service API Reference.

Data Collected by the Discovery Agent

The Application Discovery agent collects the following information about each system where it is installed:

- System identification information (hostname, IP addresses, MAC addresses, operating system name and version)
- System resource specifications (CPU, RAM, storage, etc.)
- System-level resource utilization
- Running processes
- TCP/IP (v4 and v6) connections

Related Services and Partner Tools

You have the flexibility to choose the discovery and migration tools that you need by integrating with AWS Partner tools using the Application Discovery Service API. For more information, see the Application Discovery Service API Reference.

After completing discovery of your virtualized inventory, use AWS Server Migration Service to perform an incremental, automated migration of your VMs to the Amazon EC2 cloud. For information, see the AWS SMS User Guide.

You can use the AWS VM Import/Export tools to import VM images manually from your local virtualization environment into AWS and convert them into ready-to-use Amazon EC2 Amazon Machine Images (AMIs) or instances. For more information, see Importing a VM as an Instance Using VM Import/Export.

Processing and Database Components

AWS Application Discovery Agents and the AWS Agentless Discovery Connector send data to Application Discovery Service, which includes a processing component that ingests the data and identifies (maps) IT assets. The service also includes the AWS Discovery database, a repository for discovered and mapped IT assets called *configuration items*.

Data in the Discovery database is encrypted at rest. Encryption keys for the data are managed using the AWS KMS.

> **Note**
> The AWS Discovery database is not a general purpose, enterprise configuration management database (CMDB). You can't save snapshots of discovered resources or track resource changes. The service does not alert you when resource configurations change. Similarly, though the service does collect performance data, it is not a general purpose, health monitoring solution.

When you use Application Discovery Service, you can specify filters and query specific configuration items in the AWS Discovery database. The service supports server, process, and connection configuration items. This means you can specify a value for the following keys and query your IT assets.

> **Note**
> Server Performance, shown below, is an attribute of Server.

Server	Process	Connection	ServerPerformance
cpuType	name	sourceIp	numCores
hostName	commandLine	sourceProcess	numCpus
hypervisor	path	destinationIp	numDisks
osName	avgFreeRAMInKB	destinationPort	numNetworkCards
osVersion	minFreeRAMInKB	dstProcess	totalDiskSizeInKB
transportProtocol	avgDiskReadsPerSecondInKB	count	totalDiskFreeSizeInKB
lastReportedTime		ipVersion	totalRAMInKB
avgDiskWritesPerSecondInKB			
avgDiskReadIOPS			
avgDiskWriteIOPS			
maxDiskReadsPerSecondInKB			
maxDiskWritesPerSecondInKB			
maxDiskReadIOPS			
maxDiskWriteIOPS			
avgNetworkReadsPerSecondInKB			
avgNetworkWritesPerSecondInKB			
maxNetworkReadsPerSecondInKB			
maxNetworkWritesPerSecondInKB			

Service Limitations

Application Discovery Service has the following limitations for agentless and agent-based discovery.

Agentless discovery

The service limits you to 10 GB of data per day. If you reach this limit, the service won't process any more data for that day. If you frequently reach this limit, contact AWS Support about extending the limit.

Agent-based discovery

Agent-based discovery currently has the following limitations.

- The service enforces the following maximum limits:
  - 1000 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).

Setting up AWS Application Discovery Service

This section describes the steps required to set up Application Discovery Service.

Contents

Before You Install

Complete the following tasks before you install the AWS Application Discovery Service components.

> **Note**
> You must have an active AWS account to perform these procedures. To create one, open https://aws.amazon.com/, and choose **Create an AWS Account**.

Create an IAM User

Services in AWS, such as Application Discovery Service, require that you provide credentials when you access them. This allows the service to determine whether you have permissions to access its resources. We recommend that you avoid accessing AWS with the credentials for your AWS account; instead, use AWS Identity and Access Management (IAM). Create an IAM user, and then add the user to an IAM group with administrative permissions and grant administrative permissions to this user. You can then access AWS using a special URL and the credentials for the IAM user.

For more information about setting up an IAM administrator, see Creating Your First IAM Admin User and Group. For information about IAM, see What Is IAM?.

Attach Required IAM User Policies

Application Discovery Service uses the following IAM managed policies to control access to the service or components of the service. For information about how to attach managed policies to an IAM user account, see Working with Managed Policies.

> **Note**
> The policies described here only provide access to the Application Discovery Service. Application Discovery Service is now integrated with the AWS Migration Hub. For access to features of the Migration Hub other than Application Discovery, you must apply additional policies. For more information, see Authentication and Access Control for AWS Migration Hub.

AWSApplicationDiscoveryServiceFullAccess

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

AWSApplicationDiscoveryAgentAccess

Grants Application Discovery Agent access to register and communicate with Application Discovery Service. This policy should be attached to any user whose credentials are to be used by Application Discovery Agent.

AWSAgentlessDiscoveryService

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

Note
The `AWSAgentlessDiscoveryService` policy uses the following API actions: `awsconnector:RegisterConnector` and `awsconnector:GetConnectorHealth`. For more information, see API Actions of the AWSAgentlessDiscoveryService IAM Policy (p. 12).

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

AWSApplicationDiscoveryServiceFullAccess

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

AWSApplicationDiscoveryAgentAccess

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

AWSAgentlessDiscoveryService

```
{
```

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

```
}
```

API Actions of the AWSAgentlessDiscoveryService IAM Policy

The `AWSAgentlessDiscoveryService` IAM policy uses `awsconnector:RegisterConnector` and `awsconnector:GetConnectorHealth` to grant the AWS Agentless Discovery Connector access to register, communicate with, and share connector health metrics with Application Discovery Service. More specifically, these API actions perform the following operations and return the following errors:

```
<operation name="RegisterConnector">
    <input target="RegisterConnectorRequest" /> Request type for RegisterConnector API.
    <output target="RegisterConnectorResponse" /> Response type for RegisterConnector API.
    <member name="snsTopicArn" target="String" /> Metrics SNS topic arn that is created/
whitelisted for caller.
    <error target="AuthenticationFailureException" /> This exception is thrown if the
 credentials passed in the request could not be validated or user is not authorized to
 perform the operation.
    <error target="ServerInternalErrorException" /> This exception is thrown if there is
 erroneous logic in the service. It also includes all service dependency exceptions.
    <error target="ServiceUnavailableException" /> The request has failed due to a
 temporary failure of the server.
    <error target="ServerThrottleException" /> This exception is thrown if maximum number
 of request(for a given API) from an IAM user has been reached.
    <error target="InvalidParameterException" /> The request is missing required
 parameter(s) or has invalid parameter(s).
</operation>

<operation name="GetConnectorHealth">
    <input target="GetConnectorHealthRequest" /> Request type for GetConnectorHealth API.
    <member name="connectorId" target="String" /> Connector Id that will be used to verify
 identity of caller.
    <output target="GetConnectorHealthResponse" /> Response type for GetConnectorHealth
 API.
    <member name="serviceHealthList" target="ServiceHealthList" /> Contains all services'
 health information.
    <member target="ServiceHealth" /> The object that contains all health information for a
 given service.
    <member name="serviceName" target="String" /> The name of the service which was using
 connector health metrics publisher.
    <member name="healthList" target="HealthList" /> The list of health for the given
 service.
    <member target="Health" /> The object that represent a unique health metric that was
 published from the connector.
    <error target="AuthenticationFailureException" /> This exception is thrown if there is
 erroneous logic in the service. It also includes all service dependency exceptions.
    <error target="ServiceUnavailableException" /> The request has failed due to a
 temporary failure of the server.
    <error target="ServerThrottleException" /> This exception is thrown if maximum number
 of request(for a given API) from an IAM user has been reached.
    <error target="InvalidParameterException" /> The request is missing required
 parameter(s) or has invalid parameter(s).
</operation>

<structure name="Health"> The object that represent a unique health metric that was
 published from the connector.
    <member name="name" target="String" /> The name of the health that corresponds to
 "metric" field in Connector Metrics Publisher. The value of name is not user visible label
 that will show in Connector dashboard.
    <member name="value" target="String" /> The value for the health metric. It's a json
 that contains more information about how a health will display in connector dashboard.
    <member name="lastChecked" target="TimeStamp" /> The publish time for the last received
 metric.
</structure>
```

Setting up Agentless Discovery

To set up agentless discovery, you must deploy the AWS Agentless Discovery Connector virtual appliance on a VMware vCenter Server host in your on-premises environment. Download the Agentless Discovery Appliance OVA (and MD5 and SHA256 checksums for verification). The connector appliance is an Open Virtualization Archive (OVA) file that you must install in your on-premises VMware environment. Deploy and configure the connector as described in the following sections.

Contents

Deploying the AWS Agentless Discovery Connector Virtual Appliance

Deploy the downloaded OVA file in your VMware environment.

To deploy the connector virtual appliance

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

Configuring the AWS Agentless Discovery Connector

To finish the setup process, open a web browser and complete the following procedure.

To configure the connector using the console

1. In a web browser, type the following URL in the address bar: https://*ip_address*/, where *ip_address* is the IP address of the connector console that you saved earlier.
2. Choose **Get started now** and follow the wizard steps.
3. In **Step 5: Discovery Connector Set Up**, choose **Configure vCenter credentials**.

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

    b. For **vCenter Username**, type 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*.

c. For **vCenter Password**, type the local or domain user password.

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

4. Choose **Configure AWS credentials** and type the credentials for the IAM user who is assigned the `AWSAgentlessDiscoveryService` IAM policy that you created in Attach Required IAM User Policies (p. 9). Choose **Next**.
5. Choose **Configure where to publish data** and select suitable publishing options. Choose **Next**. You should see the AWS Agentless Discovery Connector console.

Note
After you complete this initial setup, you can access connector settings by using SSH and the connector IP address: root@*Connector_IP_address*. The default user name is ec2-user and the default password is ec2pass. We strongly encourage you to change the value of the default user name and password.

Enabling Auto-Upgrades on AWS Agentless Discovery Connector

To ensure that you are running the latest version of AWS Agentless Discovery Connector, we recommend that you enable auto-upgrades.

To enable 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 Agentless Discovery Connector.
2. In the Application Discovery Service console, under **Actions**, choose **Enable Auto-Upgrade**.

Troubleshooting the Agentless Discovery Connector

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 instance. Agentless discovery does not support a stand-alone ESX host that is not part of the vCenter Server instance.

Controlling the Scope of Data Collection

The vCenter user requires read-only permissions on each ESX host or virtual machine (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.

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.

Collecting and Exporting Data

After agentless-discovery setup is complete, you can use the console or API to start collecting data; managing service, tag, and query configuration items; and exporting data. You can export data as a CSV file to an Amazon S3 bucket or an application that enables you to view and evaluate the data. For more information, see Tutorial: Using the AWS Application Discovery Service Console or the Application Discovery Service API Reference.

Setting up AWS Application Discovery Agent

AWS Application Discovery Agent is installed on on-premises servers and virtual machines (VMs) that you target for discovery and migration. Agents can run on Linux and Windows servers and collect information including system configuration, system performance, running processes, and details of the network connections between systems. Agents send this data to Application Discovery Service using Secure Sockets Layer (SSL) encryption.

We recommend installing Application Discovery Agent on a small number of servers to confirm installation procedures and connectivity to Application Discovery Service. After it is running and connected, you can use the Application Discovery Service console to enable agent data collection. This instructs the agent to collect capacity and utilization information, which should start appearing in the console several minutes later.

> **Note**
> Application Discovery Service has released a new 2.0 version of the Application Discovery agent offering better OS support. The 1.0 version of the agent has been deprecated and is no longer

recommended for new installations. For additional questions on 1.0 version, please reach out to ADS Support.

Contents

Preparing for Agent Installation

Before starting the installation, complete the following tasks.

- **Create an IAM user with a policy providing agent access to Application Discovery Service.** For information, see Create an IAM User (p. 9).
- Check the time skew from your NTP servers and correct if necessary. Incorrect time skew causes the agent registration call to fail.
- **Remove 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.

Commands to uninstall a previous-generation Application Discovery Agent 1.0

Operating System	Command
Amazon Linux, Red Hat Enterprise Linux, CentOS	**yum remove AwsAgent**
Ubuntu Server	**apt-get remove awsagent**
Windows Server	Use **Add/Remove Programs** to uninstall **AWS Agent**.

Agent Installation on Linux

Complete the following procedure on Linux.

Note
If you are using a noncurrent Linux version, see Requirements on Older Linux Platforms (p. 17).

To install AWS Application Discovery Agent in your data center

1. Log in to your Linux-based server or VM and download the installation script.

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

2. Verify the cryptographic signature of the installation package, and extract from the tarball:

```
curl -o ./agent.sig https://s3-us-west-2.amazonaws.com/aws-discovery-agent.us-west-2/
linux/latest/aws-discovery-agent.tar.gz.sig
curl -o ./discovery.gpg https://s3-us-west-2.amazonaws.com/aws-discovery-agent.us-
west-2/linux/latest/discovery.gpg
gpg --no-default-keyring --keyring ./discovery.gpg --verify agent.sig agent.tar.gz
tar -xzf agent.tar.gz
```

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

3. Run the following command to install the agent in the `us-west-2` region:

```
sudo bash install -r us-west-2 -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.

4. Optionally, use the following command to remove the agent installation script after it completes:

```
rm install
```

5. If outbound connections from your network are restricted, update your firewall settings. Agents require access to `arsenal.<region>.amazonaws.com:443`. They do not require any inbound ports to be open. Agents also work with transparent web proxies.

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, underpaid, extended-support agreements. These platforms may suffer from out-of-date cipher suites that prevent the Application Discovery agent installation script from downloading installation packages, and they may have a limited ability to find and download 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 may 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 us-west-2 -k <aws key id> -s <aws key secret> -p false -c true -b true
```

Using Application Discovery Agent on Linux

After you install, you can use the Application Discovery Service API to programmatically manage agents, tag and query configuration items, and export data. You can export data as a CSV file to an Amazon S3 bucket or an application that enables you to view and evaluate the data. For more information, see the Application Discovery Service API Reference.

You can manage the behavior of Application Discovery Agent at the system level using the following commands.

Linux Commands for Application Discovery Agent

Task	Command (Depending on Linux distribution)
Verify that an agent is running	**sudo systemctl status aws-discovery-daemon.service** **sudo initctl status aws-discovery-daemon** **sudo /etc/init.d/aws-discovery-daemon status**
Start an agent	**sudo systemctl start aws-discovery-daemon.service** **sudo initctl start aws-discovery-daemon** **sudo /etc/init.d/aws-discovery-daemon start**
Stop an agent	**sudo systemctl stop aws-discovery-daemon.service** **sudo initctl stop aws-discovery-daemon** **sudo /etc/init.d/aws-discovery-daemon stop**
Restart an agent	**sudo systemctl restart aws-discovery-daemon.service** **sudo initctl restart aws-discovery-daemon** **sudo /etc/init.d/aws-discovery-daemon restart**
Uninstall an agent from Amazon Linux, Red Hat Enterprise Linux, or CentOS	**yum remove aws-discovery-agent**
Uninstall an agent from Ubuntu Server	**apt-get remove aws-discovery-agent**
Uninstall an agent from SUSE Server	**zypper remove aws-discovery-agent**

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:

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

Logs 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:

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

Agent Installation on Windows

Complete the following procedure on Windows.

To install AWS Application Discovery Agent in your data center

1. Download and install Microsoft Visual C++ Runtime for x86 .

   Install the x86 version (`vc_redist.x86.exe`, not `vc_redist.x64.exe`) of the C++ runtime regardless of the architecture of the machine you are installing on.
2. Download the Windows agent installer.
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 command:

```
msiexec.exe /i AWSDiscoveryAgentInstaller.msi REGION="us-west-2" 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`
5. If outbound connections from your network are restricted, update your firewall settings. Agents require access to `arsenal.<region>.amazonaws.com:443`. They do not require any inbound ports to be open. Agents also work with transparent web proxies.

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 that *may* allow your Windows 2003 systems to read an SHA256 certificate. If you require SHA256 in your Windows 2003 environment, contact AWS Support for assistance.

Using Application Discovery Agent on Windows

To start or stop the Application Discovery Agent, use Windows Services Manager to start or stop the **AWS Discovery Agent** and **AWS Discovery Updater** services. To uninstall, use **Add/Remove Programs** to remove these services.

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**

  If the **msiexec** command described above appears to fail—for example, with the Windows Services Manager showing that the discovery services are not being created—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:

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

  On Windows Server 2003, agent log files can be found under:

  ```
  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:

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

  On Windows Server 2003, the agent configuration file can be found at:

  ```
  C:\Documents and Settings\All Users\Application Data\AWS\AWS Discovery\config
  ```

Frequently Asked Questions

This topic addresses common questions about the Application Discovery agent.

Why doesn't the Application Discovery agent have a 64-bit version?

Providing a 32-bit agent executable that works on both 32- and 64-bit operating systems reduces the number of installation packages that our customers need to qualify for deployment. Additionally, on 64-bit operating systems, memory use is reduced.

How does the Application Discovery agent ensure that communication with Application Discovery Service is secure?

All connections are outbound from the agent. On all host platforms, the agent enforces secure communications over TLS using up-to-date encryption libraries.

How does the agent's automatic update process work, and is it secure?

Updates are initiated by the agent using TLS. The agent retrieves information about the latest available agent release from the Application Discovery agent repository, which is securely hosted on Amazon S3.

Before the agent installs a new package, it verifies that the package has a valid digital signature from Amazon.

Accessing Application Discovery Service

Application Discovery Service supports the following console, command line, and programmatic access options:

Application Discovery Service console

The ADS console provides a simple and intuitive web-based user interface that allows you to manage your application discovery jobs. If you are whitelisted with the Application Discovery Service, you can access the console at https://console.aws.amazon.com/discovery/home. For information about using the console, see Using the AWS Application Discovery Service Console (p. 23).

AWS Command Line Interface

The AWS CLI provides commands for a broad set of AWS products. It is supported on Windows, Mac, and Linux. For more information, see AWS Command Line Interface User Guide.

Application Discovery Service API

You can use the Application Discovery Service API to manage software agents in your data center, query discovered assets, categorize discovered assets using tags, and export data. Application Discovery Service uses JavaScript Object Notation format (JSON) to send and receive formatted data. JSON presents data in a hierarchy so that both data values and data structure are conveyed simultaneously. For more information, see the Application Discovery Service API Reference.

AWS SDKs and tools

If you prefer to build applications using language-specific APIs instead of submitting a request over HTTP or HTTPS, AWS provides libraries, sample code, tutorials, and other resources for software developers. These libraries provide basic functions that automate tasks such as cryptographically signing your requests, retrying requests, and handling error responses, making it is easier for you to get started. For more information, see Tools for Amazon Web Services.

Using the AWS Application Discovery Service Console

This topic provides console-based examples of workflows involved in using the AWS Application Discovery Service. Using Application Discovery Service, you can efficiently plan the migration of applications in your virtualized on-premises environment to Amazon EC2.

Contents

Dashboard

The console dashboard offers high-level summaries of Application Discovery Service status in a single view, along with links to other console pages. You can view high-level statistics about **Servers & Applications**, a summary of **Discovery agents** and their status, and an overview of agentless **Discovery connectors**.

Data collection

The **Data collection** page displays a tab for each type of data collection tool supported, currently **Agents** (application discovery agents) and **Connectors** (agentless discovery connectors). On this page, you can:

- View and search data collection tools (p. 24)
- Start data collection for both agents and connectors (p. 24)
- Stop data collection (p. 25)

> **Note**
> You must explicitly start data collection for discovery to begin.

The table of available tools provides the following information about each:

- **Agent ID** or **Connector ID**
- **Hostname**
- **Collection status**

  The supported states can be understood as follows:
  - **STARTED**—The collection tool has started collecting and sending data to Discovery service.
  - **START_SCHEDULED**—The data collection has been scheduled to be started. The next time collection tool contacts AWS, it will start sending data to the Discovery Service and the collection status will change to **STARTED**.
  - **STOPPED**—The collection tool has stopped sending data to the Discovery service.

  - **STOP_SCHEDULED**—The data collection has been scheduled to be stopped. The next time collection tool contacts AWS, it will stop sending data to the Discovery service and the Collection status will change to **STOPPED**.
- **Health**
- **IP address**
- **Version** (of the collection tool)
- **Registered time** (when the collection tool was created)
- **Last health ping time**

The following procedures demonstrate how to carry out typical data-related management tasks. Though these examples focus on discovery agents, the steps for agentless discovery connectors are nearly identical.

To view and filter data collection tools

1. In the navigation menu, choose **Data collection**.
2. Choose **Agents** to view a table of installed application discovery agents. Each entry provides detailed information about an agent, such as its ID and host name.
3. To filter the display, choose the menu-driven filter bar, and select one of the available fields in the resulting menu:

   - **Collection status**
   - **Health**
   - **Host name**
   - **IP address**
   - **Agent ID**
4. Select one of the available operators:

   - **==**
   - **!=**
5. Select a field value. These vary based on the filter selected earlier. For **Health**, you see a menu with the following possible values:

   - **HEALTHY**
   - **RUNNING**
   - **UNHEALTHY**
   - **UNKNOWN**
   - **BLACKLISTED**
   - **SHUTDOWN**

The table now displays only the entries that match your filter criterion. You can also define multiple filters, delete filters, and bypass the filter menus by typing into the filter bar directly. For more information about agent health status and collection status, see Querying Discovered Configuration Items in the *Application Discovery Service API Reference*.

To start data collection for both agents and connectors

1. In the navigation menu, choose **Data collection**, **Agents**.
2. In the table, select the check box associated with each of the agents to start.
3. Choose **Start data collection**. In the **Collection status** field, note that the status of each of your selected collection tools changes to either **START_SCHEDULED** or **STARTED**. The next time each

of your selected collection tools contacts AWS, it collects and sends data to Application Discovery Service.

To stop data collection

1. In the navigation menu, choose **Data collection**, **Agents**.
2. In the table, select the check box associated with each of the agents to stop.
3. Choose **Stop data collection**. In the **Collection status** field, note that the status of each of your selected collection tools is now either **STOP_SCHEDULED** or **STOPPED**. The next time each of your selected collections tools contacts AWS, it stops sending discovery data to Application Discovery Service. The status of each selected collection tool changes to **STOPPED** after data collection has halted.

Servers

The **Servers** page provides system configuration and performance data about each server instance known to the data discovery tools. You can group servers into logical units called *applications*, which represent higher-level solutions that need to be migrated jointly to remain functional. You can also apply custom tags to servers to assist in your migration planning.

On this page, you can:

The following procedures describe how to carry out typical server-related management tasks.

To view and search server information

1. In the navigation menu, choose **Servers**.
2. To filter the display, choose the menu-driven filter bar, and select one of the available fields in the resulting menu:

   - **Server ID**—Server configuration item identifier generated by Application Discovery Service to uniquely identify a server.
   - **Host name**—Host name of the server.
   - **OS name**—Operating system name.
   - **OS version**—Operating system version.
   - **Agent ID**—Identifier for agent that collected server data.
   - **Connector ID**—Identifier of the connector that collected the server data.
   - **Type**—Server type.
   - **VMware more ID**—VMware VM Managed Object Reference (MoRef) ID.
   - **VMware vCenter ID**—Identifier for the vCenter.
   - **VMware host system ID**—Identifier for the host system.
   - **IP address**—IP address of the server.

- **MAC address**—Mac address of the server.
- **Avg CPU usage %**—Average CPU usage percentage.
- **Total disk free size (KB)**—Total disk free size in KB.
- **Avg free RAM (KB)**—Total disk free size in KB.
- **Tag value**—Configuration tag value.
- **Tag key**—Configuration tag key.
- **Application name**—Name of application to which the server belongs.
- **Application description**—Description of application to which the server belongs.
- **Application ID**—Unique identifier of Application configuration item to which the server belongs.

3. Select one of the available operators:

   - **==**
   - **!=**
   - **Contains**
   - **Not Contains**

4. Supply a value or choose from values offered by the menu. These vary based on the filter that you selected earlier. For **Type**, you see a menu with values such as the following:

   - **EC2**
   - **OTHER**
   - **VMWARE_VM**
   - **VMWARE_HOST**
   - **VMWARE_VM_TEMPLATE**

   The table now displays only the entries that match your filter criterion. You can also define multiple filters, delete filters, and bypass the filter menus by typing into the filter bar directly.

5. In the table, choose a **Server ID** link to display details about that server. This opens a details page with basic system information, system performance, and the status of recent export jobs. Under the server name at the top of the page, the console displays applications and tags associated with the server. Choose **Actions** to see the actions that can be performed on the server, including **Group as application**, **Add tag**, **Remove server from application**, **Remove tag**, and **Export server details**.

Tags are user-defined key/value pairs that can store meta-information about servers. Application discovery tags are similar to AWS tags, but the two types of tag cannot be used interchangeably. You can add or remove tags on up to 10 servers at a time on the **Servers** page. On server details pages, you can add or remove tags only for the selected server. Up to five tags can be added or removed from a server or servers in a single operation.

To tag multiple servers

1. In the navigation menu, choose **Servers**.
2. In the table, select the check boxes of the servers to tag and choose **Add tag**.
3. In the **Add Tags** window, provide a key (tag name) and a value.
4. To add additional tags to the selected servers, choose **Additional tag** and repeat the previous step.
5. Choose **Save**. Note that table entries for the tagged servers now display the new tag keys in the **Tags** field.
6. Open the **Server ID** link of a newly tagged server. On the details page, note that the **Tags** list near the top of the page displays the new tag keys.
7. Choose the tag key names. The **Tags** window opens with a table containing all of the tag key/value pairs for the selected server.

To tag a single server

1. In the navigation menu, choose **Servers**.
2. In the table of servers, select a server ID to open the server details page.
3. Choose **Actions**, **Add tag**.
4. In the **Add Tags** window, provide a key (tag name) and a value.
5. To add additional tags to the selected server, choose **Additional tag** and repeat the previous step.
6. Choose **Save**. On the details page, note that the **Tags** list near the top of the page displays the new tag keys.

To remove tags from multiple servers

1. In the navigation menu, choose **Servers**.
2. In the table of servers, select the check boxes of the servers sharing a tag to remove.
3. Choose **Actions**, **Remove tag**.
4. In the **Remove Tags** window, enter the key/value pair of the tag to remove.
5. To remove additional tags from the selected servers, choose **Additional tag** and repeat the previous step.
6. Choose **Remove** to delete the tags from the selected servers. Note that table entries for the selected servers no longer display keys for removed tags.

To remove tags from a single server

1. In the navigation menu, choose **Servers**.
2. Choose **Actions**, **Remove tag**.
3. In the **Remove Tags** window, select the check boxes of key/value pairs to remove.
4. To remove additional tags from the selected server, choose **Additional tag** and repeat the previous step.
5. Choose **Save**. On the details page, note that the **Tags** list near the top of the page no longer displays keys for removed tags.

To export data about a server

Application Discovery Service supports up to five concurrent exports of server data. Start export jobs as described in the following procedure.

1. In the navigation menu, choose **Servers**.
2. Choose the **Server ID** of a server. This opens a details page that displays information about the server's hardware, operating system, and performance.
3. The **Exports** section of the details page displays a table of export jobs (if any) for the server. Information includes:

   - **Export ID**—A unique ID for the export job
   - **From date/time**—Starting date and time of the data sample period
   - **To date/time**—Ending date and time of the data sample period
   - **Export status**—Possible values are **In-progress** and **Completed**
   - **Export requested time**—Date and time of the export request
   - A link to the exported data (in zip format) stored in your AWS account's S3 bucket

   To request a new export, choose **Export server details**.

4. In the **Export server details** window, the "From date/time" field will be pre-populated with the current date and time and data will be generated for the next 72 hours.
5. Choose **Export** to start the job. The initial status will be **In-progress**; to update the status, click the refresh icon in the **Exports** section heading.
6. When the export job is complete, choose **Download** and save the zip file.
7. Unzip the saved file. The exported data is contained in a set of CSV 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 Excel and review the exported server data.

   Also present is a JSON file containing data about the export task and its results. The customer can write a script to read the JSON and manipulate the downloaded server data.

Applications

The **Applications** page lists your applications and allows you to view the servers that compose them. Each table entry represents an application and displays the application ID, application name, description, number of member servers, and creation time. You can search applications, create applications, and delete applications. On the details page for each application, you can add and remove servers.

On this page, you can:

- View and search applications (p. 28)
- Create an application (p. 28)
- Delete one or more applications (p. 29)
- Delete one or more servers from an application (p. 29)

To view and search applications

1. In the navigation menu, choose **Applications**.
2. In the table, for **Application ID**, select an application with a non-zero number of member servers. A details page with a list of all the member servers displays.
3. On the details page of the selected application, choose the **Server ID** of a member server. From the details page for that server, you can perform management operations for the selected server described in Servers (p. 25), including deleting it from the application. To delete servers in bulk from an application, see To delete one or more servers from an application (p. 29) below.

The table now displays only the entries that match your filter criterion. You can also define multiple filters, delete filters, and bypass the filter menus by typing into the filter bar directly.

To create an application

1. In the navigation menu, choose **Applications**.
2. Choose **Create new application**.

3. In the **Create new application** window, provide values for **Application name** and (optionally) **Application description**.
4. When done, choose **Create**. Note that the newly created application is listed in the table.

To delete one or more applications

1. In the navigation menu, choose **Applications**.
2. In the table of applications, select the check boxes of applications to delete.
3. Choose **Delete applications**.
4. In the **Delete application** window, choose **Delete**. Note that entries for deleted applications have been removed from the table.

To delete one or more servers from an application

1. In the navigation menu, choose **Applications**.
2. In the table, for **Application ID**, select an application with a non-zero number of member servers. A details page with a list of all the member servers displays.
3. On the details page of the selected application, select the check boxes of servers to delete from the application.
4. When done, choose **Remove server from application**.
5. In the **Remove server from applications** window, when you are sure you want to proceed, choose **Remove**. Note that entries for deleted servers have been removed from the table.

Document History for AWS Application Discovery Service

The following table describes the important changes to the documentation since the last release of Application Discovery Service.

Latest documentation update: October 19, 2017

Change	Description	Date
Discovery Agent 2.0	New and improved Application Discovery agent released.	October 19, 2017
Console	AWS management console added.	December 19, 2016
Launch of agentless discovery	Added content that describes how to set. up and configure agentless discovery.	July 28, 2016
Updates	Added Windows Server details to *AWS Application Discovery Service CLI Walkthrough*. Fixed various command issues.	May 20, 2016
Initial publication	Released *Application Discovery Service User Guide*.	May 12, 2016

AWS Glossary

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