AWS Directory Service

Administration Guide

Version 1.0

amazon
web services™

AWS Directory Service: Administration Guide

Table of Contents

What Is AWS Directory Service?

AWS Directory Service provides multiple ways to use Microsoft Active Directory with other AWS services. You can choose the directory service with the features you need at a cost that fits your budget.

Use Simple AD if you need an inexpensive Active Directory–compatible service with the common directory features.

Select AWS Directory Service for Microsoft Active Directory (Enterprise Edition) for a feature-rich managed Microsoft Active Directory hosted on the AWS cloud.

Our third option, AD Connector, lets you simply connect your existing on-premises Active Directory to AWS.

Which to Choose?

The following information will help you decide which AWS Directory Service option is right for you:

AWS Directory Service for Microsoft Active Directory (Enterprise Edition) is a managed Microsoft Active Directory hosted on the AWS cloud. It provides much of the functionality offered by Microsoft Active Directory plus integration with AWS applications. With the additional Active Directory functionality, you can, for example, easily set up trust relationships with your existing Active Directory domains to extend those directories to AWS services.

If you are in the Frankfurt region, you can use Microsoft AD to enable multi-factor authentication by integrating with your existing RADIUS-based MFA infrastructure to provide an additional layer of security when users access AWS applications. For more information, see Multi-Factor Authentication (p. 58).

AWS Directory Service for Microsoft Active Directory (Enterprise Edition) does not support fine-grained password polices.

When to use

Microsoft AD is your best choice if you have more than 5,000 users and need a trust relationship set up between an AWS hosted directory and your on-premises directories.

AD Connector is a proxy service for connecting your on-premises Microsoft Active Directory to the AWS cloud without requiring complex directory synchronization or the cost and complexity of hosting a federation infrastructure.

AD Connector forwards sign-in requests to your Active Directory domain controllers for authentication and provides the ability for applications to query the directory for data. After setup, your users can use their existing corporate credentials to log on to AWS applications, such as Amazon WorkSpaces, Amazon WorkDocs, or Amazon WorkMail. With the proper IAM permissions, they can also access the AWS Management Console and manage AWS resources such as Amazon EC2 instances or Amazon S3 buckets. You can also use AD Connector to enable multi-factor authentication by integrating with your existing RADIUS-based MFA infrastructure to provide an additional layer of security when users access AWS applications.

With AD Connector, you continue to manage your Active Directory as usual. For example, adding new users, adding new groups or updating passwords is all accomplished using standard directory administration tools with your on-premises directory. Thus, in addition to providing a streamlined experience for your users, AD Connector enables consistent enforcement of your existing security policies, such as password expiration, password history, and account lockouts, whether users are accessing resources on premises or in the AWS cloud.

When to use

AD Connector is your best choice when you want to use your existing on-premises directory with AWS services.

Simple AD is a Microsoft Active Directory–compatible directory from AWS Directory Service that is powered by Samba 4. Simple AD supports commonly used Active Directory features such as user accounts, group memberships, domain-joining Amazon Elastic Compute Cloud (Amazon EC2) instances running Linux and Microsoft Windows, Kerberos-based single sign-on (SSO), and group policies. This makes it even easier to manage Amazon EC2 instances running Linux and Windows, and deploy Windows applications in the AWS cloud.

Many of the applications and tools you use today that require Microsoft Active Directory support can be used with Simple AD. User accounts in Simple AD can also access AWS applications, such as Amazon WorkSpaces, Amazon WorkDocs, or Amazon WorkMail. They can also use AWS Identity and Access Management roles to access the AWS Management Console and manage AWS resources. Finally, Simple AD provides daily automated snapshots to enable point-in-time recovery.

Note that you cannot set up trust relationships between Simple AD and other Active Directory domains. Other common features not supported today by Simple AD include DNS dynamic update, schema extensions, multi-factor authentication, communication over LDAPS, PowerShell AD cmdlets, and the transfer of FSMO roles.

When to use

In most cases, Simple AD is the least expensive option and your best choice if you have 5,000 or less users and don't need the more advanced Microsoft Active Directory features.

For information about the AWS Directory Service API, see the *AWS Directory Service API Reference*.

Working with Amazon EC2

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

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

Setting Up

To work with AWS Directory Service, you need to meet the prerequisites for AWS Directory Service for Microsoft Active Directory (Enterprise Edition), AD Connector, or Simple AD. For more information, see Microsoft AD Prerequisites (p. 10), AD Connector Prerequisites (p. 47), or Simple AD Prerequisites (p. 61).

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

Topics

Sign Up for AWS

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

If you do not have an AWS account, use the following procedure to create one.

To sign up for AWS

1. Open https://aws.amazon.com/ and choose **Create an AWS Account**.
2. Follow the online instructions.

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

Create an IAM User

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

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

To create an IAM user for yourself and add the user to an Administrators group

1. Sign in to the Identity and Access Management (IAM) console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Users**, and then choose **Add user**.
3. For **User name**, type a user name, such as **`Administrator`**. The name can consist of letters, digits, and the following characters: plus (+), equal (=), comma (,), period (.), at (@), underscore (_), and hyphen (-). The name is not case sensitive and can be a maximum of 64 characters in length.
4. Select the check box next to **AWS Management Console access**, select **Custom password**, and then type the new user's password in the text box. You can optionally select **Require password reset** to force the user to select a new password the next time the user signs in.
5. Choose **Next: Permissions**.
6. On the **Set permissions for user** page, choose **Add user to group**.
7. Choose **Create group**.
8. In the **Create group** dialog box, type the name for the new group. The name can consist of letters, digits, and the following characters: plus (+), equal (=), comma (,), period (.), at (@), underscore (_), and hyphen (-). The name is not case sensitive and can be a maximum of 128 characters in length.
9. For **Filter**, choose **Job function**.
10. In the policy list, select the check box for **AdministratorAccess**. Then choose **Create group**.
11. 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.
12. Choose **Next: Review** to see the list of group memberships to be added to the new user. When you are ready to proceed, choose Add permissions.

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 to restrict users' permissions to specific AWS resources, go to Access Management and Example Policies for Administering AWS Resources.

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

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

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

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

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

For more information about using IAM policies to control access to your AWS Directory Service resources, see Identity-Based Policies (IAM Policies) (p. 121).

AWS Directory Service Best Practices

Here are some suggestions and guidelines you should consider to avoid problems and get the most out of AWS Directory Service.

Setting Up: Prerequisites

Consider these guidelines before creating your directory.

Choose the Right Directory Type

AWS Directory Service provides multiple ways to use Microsoft Active Directory with other AWS services. You can choose the directory service with the features you need at a cost that fits your budget:

- **AWS Directory Service for Microsoft Active Directory (Enterprise Edition)** is a feature-rich managed Microsoft Active Directory hosted on the AWS cloud. Microsoft AD is your best choice if you have more than 5,000 users and need a trust relationship set up between an AWS hosted directory and your on-premises directories.
- **AD Connector** simply connects your existing on-premises Active Directory to AWS. AD Connector is your best choice when you want to use your existing on-premises directory with AWS services.
- **Simple AD** is an inexpensive Active Directory–compatible service with the common directory features. In most cases, Simple AD is the least expensive option and your best choice if you have 5,000 or fewer users and don’t need the more advanced Microsoft Active Directory features.

For a more detailed comparison of AWS Directory Service options, see Which to Choose? (p. 1).

Ensure Your VPCs and Instances are Configured Correctly

In order to connect to, manage, and use your directories, you must properly configure the VPCs that the directories are associated with. See either Microsoft AD Microsoft AD Prerequisites (p. 10), AD

Connector AD Connector Prerequisites (p. 47), or Simple AD Simple AD Prerequisites (p. 61) for information about the VPC security and networking requirements.

If you are adding an instance to your domain, ensure that you have connectivity and remote access to your instance as described in Add an Instance to Your Directory (Simple AD and Microsoft AD) (p. 105).

Configure On-premises Sites and Subnets Correctly When Using AD Connector

If your on-premises network has Active Directory sites defined, you must make sure the subnets in the VPC where your AD Connector resides are defined in an Active Directory site, and that no conflicts exist between the subnets in your VPC and the subnets in your other sites.

To discover domain controllers, AD Connector uses the Active Directory site whose subnet IP address ranges are close to those in the VPC that contain the AD Connector. If you have a site whose subnets have the same IP address ranges as those in your VPC, AD Connector will discover the domain controllers in that site, which may not be physically close to your region.

Be Aware of Your Limits

By default, you are limited to 10 directories and 5 snapshots per each directory. You can increase those limits following the steps listed in AWS Directory Service Limits (p. 137).

Another limit you should pay attention to is number of users in a directory. Generally, you should not add more than 5,000 users to a Simple AD directory. If you have more than 5,000 users, consider AWS Directory Service for Microsoft Active Directory (Enterprise Edition) instead.

Use Microsoft AD If Trusts Are Required

Simple AD does not support trust relationships. If you need to establish a trust between your AWS Directory Service directory and another directory, you should use AWS Directory Service for Microsoft Active Directory (Enterprise Edition).

Setting Up: Creating Your Directory

Here are some suggestions to consider as you create your directory.

Remember Your Administrator ID and Password

When you set up your directory, you provide a password for the administrator account. That account ID is *Administrator* for Simple AD and *Admin* for Microsoft AD. Remember the password that you create for this account; otherwise you will not be able to add objects to your directory.

Create a DHCP Options Set

We recommend that you create a DHCP options set for your AWS Directory Service directory and assign the DHCP options set to the VPC that your directory is in. That way any instances in that VPC can point to the specified domain, and DNS servers can resolve their domain names.

For more information about DHCP options sets, see DHCP Options Set (p. 117).

Using Your Directory

Here are some suggestions to keep in mind when using your directory.

Do Not Alter Predefined Users, Groups and Organization Units

When you use AWS Directory Service to launch a directory, AWS creates an organizational unit (OU) that contains all your directory's objects. This OU, which has the NetBIOS name that you typed when you created your directory, is located in the domain root. The domain root is owned and managed by AWS. Several groups and an administrative user are also created.

Do not move, delete or in any other way alter these predefined objects. Doing so can make your directory inaccessible by both yourself and AWS.

Add Users to Your Simple AD Directory from a Windows Server Instance, Version 2008 R2 and Above

In Windows Server 2012 R2, an incompatibility between the Simple AD directory and the Active Directory Users and Computers tool causes user creation to fail. You can still use the tools on Windows Server 2012 R2 for other tasks, such as managing group policy. However, you should create your users from a Windows Server 2008 R2 instance.

Automatically Join Domains

When launching a Windows instance that is to be part of an AWS Directory Service domain, it is often easiest to join the domain as part of the instance creation process rather than manually adding the instance later. To automatically join a domain, simply select the correct directory for **Domain join directory** when launching a new instance. You can find details in Launching an Instance (Simple AD and Microsoft AD) (p. 106).

Don't Launch SQL Server Management Studio (SSMS) Using a Simple AD Domain Account

You might receive an error if you attempt to use SQL Server Management Studio (SSMS) with a SQL Server account to log into SQL Server running on a Windows 2012 R2 EC2 instance or in Amazon RDS. The issue occurs when SSMS is run as a domain user and can result in the error "Login failed for user," even when valid credentials are provided. To work around the issue, you can log into SQL Server with Windows Authentication instead of SQL Authentication. Or launch SSMS as a local user instead of a Simple AD domain user.

Set Up Trusts Correctly

When setting up trust relationship between your Microsoft AD directory and another directory, keep in mind these guidelines:

- Both trusts must be forest trusts.
- Both fully qualified domain names (FQDNs) must be unique.
- If adding a NetBIOS name, that should also be unique.

For more details and specific instructions on setting up a trust relationship, see When to Create a Trust Relationship (p. 18).

Use Unique AD Connectors for Each Domain

AD Connectors and your on-premises domains have a 1-to-1 relationship. That is, for each on-premises domain you want to authenticate against, you must create a unique AD Connector.

Managing Your Directory

Consider these suggestions for managing your directory.

Make a Backup of Your Instance

If you decide to manually add an instance to an existing AWS Directory Service domain, make a backup or take a snapshot of that instance first. This is particularly important when joining a Linux instance. Some of the procedures used to add an instance, if not performed correctly, can render your instance unreachable or unusable.

Set Up SNS Messaging

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

Also remember that if you have an SNS topic that receives messages from AWS Directory Service, before deleting that topic from the Amazon SNS console, you should associate your directory with a different SNS topic. Otherwise you risk missing important directory status messages.

Remove Amazon RDS Databases before Deleting a Directory

Before deleting a directory that is associated with an Amazon Relational Database Service (Amazon RDS), you must first remove that database from the directory.

Microsoft Active Directory

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

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

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

- Manage users and groups
- Provide single sign-on to applications and services
- Create and apply group policy
- Securely connect to Amazon EC2 Linux and Windows instances
- Simplify the deployment and management of cloud-based Linux and Microsoft Windows workloads

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

Topics

Create a Microsoft AD Directory

Microsoft AD creates a fully managed, Microsoft Active Directory in the AWS cloud. When you create a directory with Microsoft AD, AWS Directory Service creates two directory servers and DNS servers on your behalf. The directory servers are created in different subnets in a VPC; this redundancy helps ensure that your directory remains accessible even if a failure occurs.

Topics

Microsoft AD Prerequisites

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

- At least two subnets. Each of the subnets must be in a different Availability Zone.
- The following ports must be open between the two subnets that you deploy your directory into. This is necessary to allow the domain controllers that AWS Directory Service creates for you to communicate with each other.
  - TCP/UDP 53 - DNS
  - TCP/UDP 88 - Kerberos authentication
  - UDP 123 - NTP
  - TCP 135 - RPC
  - UDP 137-138 - Netlogon
  - TCP 139 - Netlogon
  - TCP/UDP 389 - LDAP; note that AWS Directory Service does not support LDAP with SSL (LDAPS) or LDAP signing
  - TCP/UDP 445 - SMB
  - TCP 873 - FRS
  - TCP 3268 - Global Catalog
  - TCP/UDP 1024-65535 - Ephemeral ports for RPC
- The VPC must have default hardware tenancy.
- You cannot create a Microsoft AD in a VPC using addresses in the 198.19.0.0/16 address space.
- AWS Directory Service does not support using Network Address Translation (NAT) with Active Directory. Using NAT can result in replication errors.

How to Create a Microsoft AD directory

To create a new directory, perform the following steps. Before starting this procedure, make sure you have completed the prerequisites identified in Microsoft AD Prerequisites (p. 10).

To create a Microsoft AD directory

1. In the AWS Directory Service console navigation pane, select **Directories** and choose **Set up Directory**.
2. Choose **Create Microsoft AD**.
3. Provide the following information:

Directory DNS

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

NetBIOS name

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

Administrator password

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

The password cannot include the word "admin."

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

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

Confirm password

Retype the administrator password.

Description

An optional description for the directory.

4. Provide the following information in the **VPC Details** section and choose **Next Step**.

   VPC

   The VPC for the directory.

   Subnets

   Select the subnets for the directory servers. The two subnets must be in different Availability Zones.
5. Review the directory information and make any necessary changes. When the information is correct, choose **Create Microsoft AD**.

It takes several minutes for the directory to be created. When it has been successfully created, the **Status** value changes to `Active`.

What Gets Created

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

- Sets up a Microsoft Active Directory within the VPC.
- Creates a directory administrator account with the user name `Admin` and the specified password. You use this account to manage your directory.

  Important

  Be sure to save this password. AWS Directory Service does not store this password and it cannot be retrieved or reset.
- Creates a security group for the directory controllers.

Admin Account Permissions

When you create an AWS Directory Service for Microsoft Active Directory (Enterprise Edition) directory, AWS creates an organizational unit (OU) that contains all your directory's objects. This OU, which has the NetBIOS name that you typed when you created your directory, is located in the domain root. The domain root is owned and managed by AWS.

The *admin* account that was created with your Microsoft AD has permissions for the most common administrative activities for your OU:

- Create update, or delete users, groups, and computers
- Add resources to your domain such as file or print servers, and then assign permissions for those resources to users and groups in your OU
- Create additional OUs and containers
- Delegate authority
- Create and link group policies
- Restore deleted objects from the Active Directory Recycle Bin
- Run AD and DNS Windows PowerShell modules on the Active Directory Web Service

The admin account also has rights to perform the following domain-wide activities:

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

Actions not listed here are not allowed for the admin account. The admin account also lacks permissions for any directory-related actions outside of your specific OU, such as on the parent OU.

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

Schema Extensions

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

Microsoft AD uses schemas to organize and enforce how directory data is stored. The process of adding definitions to the schema is referred to as “extending the schema.” Schema extensions make it possible for you to modify the schema of your Microsoft AD directory using a valid LDAP Data Interchange Format (LDIF) file. For more information about AD schemas and how to extend your schema, see the topics listed below.

Topics

Schema Elements

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

Attributes

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

Classes

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

Object identifier (OID)

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

Schema linked attributes

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

Related Topics

- When to Extend Your Microsoft AD Schema (p. 13)
- Tutorial: Extending Your Microsoft AD Schema (p. 14)

When to Extend Your Microsoft AD Schema

You can extend your Microsoft AD schema by adding new object classes and attributes. For example, you might do this if you have an application that requires changes to your schema in order to support single sign-on capabilities.

You can also use schema extensions to enable support for applications that rely on specific Active Directory object classes and attributes. This can be especially useful in the case where you need to migrate corporate applications that are dependent on Microsoft AD, to the AWS cloud.

Each attribute or class that is added to an existing Active Directory schema must be defined with a unique ID. That way when companies add extensions to the schema, they can be guaranteed to be unique and not to conflict with each other. These IDs are referred to as AD Object Identifiers (OIDs) and are stored in Microsoft AD.

To get started, see Tutorial: Extending Your Microsoft AD Schema (p. 14).

Related Topics

- Schema Extensions (p. 12)
- Schema Elements (p. 12)

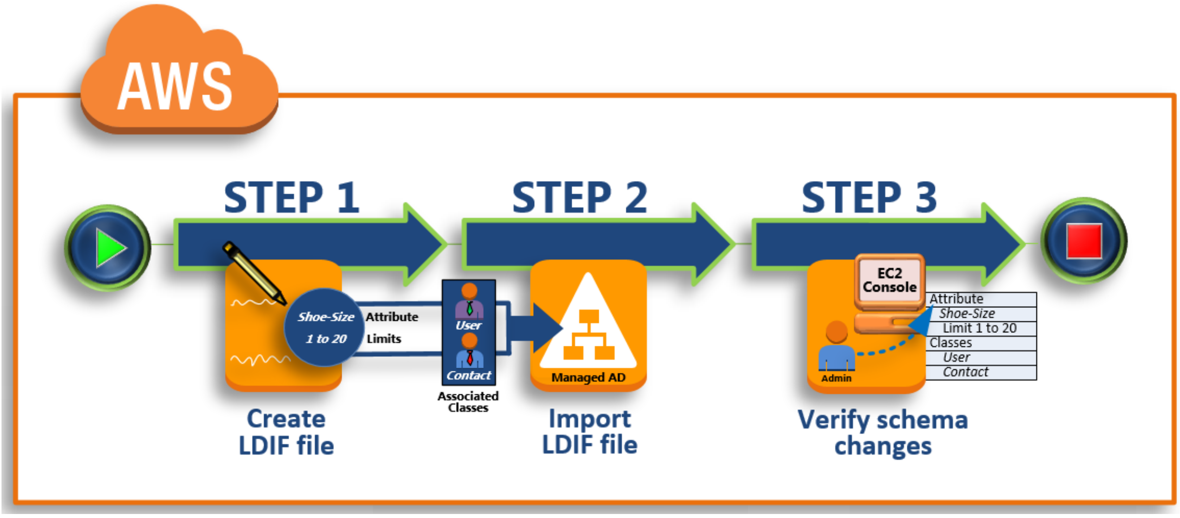
Tutorial: Extending Your Microsoft AD Schema

In this tutorial, you will learn how to extend the schema for your AWS Directory Service for Microsoft Active Directory (Enterprise Edition) directory, also known as Microsoft AD, by adding unique *attributes* and *classes* that meet your specific requirements. Microsoft AD schema extensions can only be uploaded and applied using a valid LDIF (Lightweight Directory Interchange Format) script file.

Attributes (attributeSchema) define the fields in the database while classes (classSchema) define the tables in the database. For example, all of the user objects in Active Directory are defined by the schema class *User* while the individual properties of a user, such as email address or phone number, are each defined by an attribute.

If you wanted to add a new property, such as Shoe-Size, you would define a new attribute, which would be of type *integer*. You could also define lower and upper limits like 1 to 20. Once the Shoe-Size attributeSchema object has been created, you would then alter the *User* classSchema object to contain that attribute. Attributes can be linked to multiple classes. Shoe-Size could also be added to the *Contact* class for example. For more information about Active Directory schemas, see When to Extend Your Microsoft AD Schema (p. 13).

This workflow has three basic steps.



Step 1: Create Your LDIF File (p. 14)
: First, you create an LDIF file and define the new attributes and any classes that the attributes should be added to. You use this file for the next phase of the workflow.

Step 2: Import Your LDIF File (p. 16)
: In this step, you use the AWS Directory Service console to import the LDIF file to your Microsoft AD environment.

Step 3: Verify If The Schema Extension Was Successful (p. 17)
: Finally, as an administrator, you use an EC2 instance to verify that the new extensions appear in the Active Directory Schema Snap-in.

Step 1: Create Your LDIF File

An LDIF file is a standard plain text data interchange format for representing LDAP (Lightweight Directory Access Protocol) directory content and update requests. LDIF conveys directory content as a set of records, one record for each object (or entry). It also represents update requests, such as Add, Modify, Delete, and Rename, as a set of records, one record for each update request.

The AWS Directory Service imports your LDIF file with the schema changes by running the `ldifde.exe` application on your Microsoft AD directory. Therefore, you'll find it helpful to understand the LDIF script syntax. For more information, see LDIF Scripts.

Several third-party LDIF tools can extract, clean-up, and update your schema updates. Regardless of which tool you use, it is important to understand that all identifiers used in your LDIF file must be unique.

We highly recommend that you review the following concepts and tips prior to creating your LDIF file.

- **Schema elements** – Learn about schema elements such as attributes, classes, object IDs, and linked attributes. For more information, see Schema Elements (p. 12).
- **Sequence of items** – Make sure that the order in which the items in your LDIF file are laid out follow the Directory Information Tree (DIT) from the top down. The general rules for sequencing in an LDIF file include the following:
  - Separate items with a blank line.
  - List child items after their parent items.
  - Ensure that items such as attributes or object classes exist in the schema. If they are not present, you must add them to the schema before they can be used. For example, before you can assign an attribute to a class, the attribute must be created.
- **Format of the DN** – For each new instruction in the LDIF file, define the distinguished name (DN) as the first line of the instruction. The DN identifies an Active Directory object within the Active Directory object's tree and must contain the domain components for your directory. For example, the domain components for the directory in this tutorial are `DC=example,DC=com`.

  The DN also must contain the common name (CN) of the Active Directory object. The first CN entry is the attribute or class name. Next, you must use `CN=Schema,CN=Configuration`. This CN ensures that you are able to extend the Active Directory schema. As mentioned before, you cannot add or modify Active Directory objects' content. The general format for a DN follows.

```
dn: CN=[attribute or class name],CN=Schema,CN=Configuration,DC=[domain_name]
```

  For this tutorial, the DN for the new Shoe-Size attribute would look like:

```
dn: CN=Shoe-Size,CN=Schema,CN=Configuration,DC=example,DC=com
```

- **Warnings** – Review the warnings below before you extend your schema.
  - Before you extend your Active Directory schema, it is important to review Microsoft's warnings on the impact of this operation. For more information, see What You Must Know Before Extending the Schema.
  - You cannot delete a schema attribute or class. Therefore, if you make a mistake and don't want to restore from backup, you can only disable the object. For more information, see Disabling Existing Classes and Attributes.

To learn more about how LDIF files are constructed and see a sample LDIF file that can be used for testing Microsoft AD schema extensions, see the article How to Extend your Microsoft AD directory Schema on the AWS Security Blog.

Next Step

Step 2: Import Your LDIF File (p. 16)

Step 2: Import Your LDIF File

You can extend your schema by importing an LDIF file from either the AWS Directory Service console or by using the API. For more information about how to do this with the schema extension APIs, see the *AWS Directory Service API Reference*. At this time, AWS does not support external applications, such as Microsoft Exchange, to perform schema updates directly.

> **Important**
> When you make an update to your Microsoft AD directory schema, the operation is not reversible. In other words, once you create a new class or attribute, Active Directory doesn’t allow you to remove it. However, you can disable it.
> If you must delete the schema changes, one option is to restore the directory from a previous snapshot. Restoring a snapshot rolls both the schema and the directory data back to a previous point, not just the schema.

Before the update process begins, Microsoft AD takes a snapshot to preserve the current state of your directory.

To import your LDIF file

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. In the **Directory ID** column, choose the link for your directory.
3. Under the **Schema extensions** tab, choose **Upload and update schema**.
4. In the dialog box, click **Browse**, select a valid LDIF file, type a description, and then choose **Update Schema**.

   > **Important**
   > Extending the schema is a critical operation. Don’t apply any schema update in production environment without first testing it with your application in a development or test environment.

How is the LDIF File Applied

After your LDIF file has been uploaded, Microsoft AD takes steps to protect your directory against errors as it applies the changes in the following order.

1. **Validates the LDIF file.** Since LDIF scripts can manipulate any object in the domain, Microsoft AD runs checks right after you upload to help ensure that the import operation will not fail. These include checks to ensure the following:
   - The objects to be updated are only held in the schema container
   - The DC (domain controllers) part matches the name of the domain where the LDIF script is running
2. **Takes a snapshot of your directory.** You can use the snapshot to restore your directory in case you encounter any problems with your application after updating the schema.
3. **Applies the changes to a single DC.** Microsoft AD isolates one of your DCs and applies the updates in the LDIF file to the isolated DC. It then selects one of your DCs to be the schema master, removes that DC from directory replication, and applies your LDIF file using `Ldifde.exe`.
4. **Replication occurs to all DCs.** Microsoft AD adds the isolated DC back in to replication to complete the update. While this is all happening, your directory continues to provide the Active Directory service to your applications without disruption.

Next Step

Step 3: Verify If The Schema Extension Was Successful (p. 17)

Step 3: Verify If The Schema Extension Was Successful

After you have finished the import process, it is important to verify that schema updates were applied to your directory. This is especially critical before you migrate or update any application that relies on the schema update. You can do this using a variety of different LDAP tools or by writing a test tool that issues the appropriate LDAP commands.

This procedure uses the Active Directory Schema Snap-in and/or PowerShell to verify that the schema updates were applied. You must run these tools from a computer that is domain joined to your Microsoft AD. This can be a Windows server running in your on-premises network with access to your virtual private cloud (VPC) or through a virtual private network (VPN) connection. You can also run these tools on an Amazon EC2 Windows instance (see How to launch a new EC2 instance with Seamless Domain Join).

To verify using the Active Directory Schema Snap-in

1. Install the Active Directory Schema Snap-In using the instructions on the TechNet website.
2. Open the Microsoft Management Console (MMC) and expand the **AD Schema** tree for your directory.
3. Navigate through the **Classes** and **Attributes** folders until you find the schema changes that you made earlier.

To verify using PowerShell

1. Open a PowerShell window.
2. Use the `Get-ADObject` cmdlet as shown below to verify the schema change. For example:

   ```
   get-adobject -Identity 'CN=Shoe-
   Size,CN=Schema,CN=Configuration,DC=example,DC=com' -Properties *
   ```

Optional Step

(Optional) Add a value to the new attribute (p. 17)

(Optional) Add a value to the new attribute

Use this optional step when you have created a new attribute and want to add a new value to the attribute in your Microsoft AD directory.

To add a value to an attribute

1. Open the Windows PowerShell command line utility and set the new attribute with the following command. In this example, we will add a new EC2InstanceID value to the attribute for a specific computer.

   ```
   PS C:\> set-adcomputer -Identity computer name -add @{example-
   EC2InstanceID = 'EC2 instance ID'}
   ```
2. You can validate if the EC2InstanceID value was added to the computer object by running the following command:

   ```
   PS C:\> get-adcomputer -Identity computer name –Property example-
   EC2InstanceID
   ```

Related Resources

The following resource links are located on the Microsoft website and provide related information.

- Extending the Schema (Windows)
- Active Directory Schema (Windows)
- Active Directory Schema
- Windows Administration: Extending the Active Directory Schema
- Restrictions on Schema Extension (Windows)
- Ldifde

When to Create a Trust Relationship

You can configure one and two-way forest trust relationships between your AWS Directory Service for Microsoft Active Directory (Enterprise Edition) and on-premises directories, as well as between multiple Microsoft AD directories in the AWS cloud. Microsoft AD supports all three trust relationship directions: Incoming, Outgoing and Two-way (Bi-directional).

> **Note**
> When setting up trust relationships, you must ensure that your on-premises directory is and remains compatible with AWS Directory Services. For more information on your responsibilities, please see our shared responsibility model.

Microsoft AD supports forest trusts only. External trusts are not supported.

Prerequisites

Creating the trust requires only a few steps, but you must first complete several prerequisite steps prior to setting up the trust.

Connect to VPC

If you are creating a trust relationship with your on-premises directory, you must first connect your on-premises network to the VPC containing your Microsoft AD. The firewall for your on-premises network must have the following ports open to the CIDRs for both subnets in the VPC.

- TCP/UDP 53 - DNS
- TCP/UDP 88 - Kerberos authentication
- TCP/UDP 389 - LDAP
- TCP 445 - SMB

These are the minimum ports that are needed to be able to connect to your directory. Your specific configuration may require additional ports be open.

Configure your VPC

The VPC that contains your Microsoft AD must have the appropriate outbound and inbound rules.

To configure your VPC outbound rules

1. In the AWS Directory Service console, on the **Directory Details** page, note your Microsoft AD directory ID.
2. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
3. Choose **Security Groups**.
4. Search for your Microsoft AD directory ID. In the search results, select the item with the description "AWS created security group for *directory ID* directory controllers".

> **Note**
> The selected security group is a security group that is automatically created when you initially create your directory.

5. Go to the **Outbound Rules** tab of that security group. Select **Edit**, then **Add another rule**. For the new rule, enter the following values:

   - **Type**: All Traffic
   - **Protocol**: All
   - **Destination**: *0.0.0.0/0*

6. Select **Save**.

To configure your VPC inbound rules

1. In the AWS Directory Service console, on the **Directory Details** page, note your Microsoft AD directory ID.
2. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
3. Choose **Security Groups**.
4. Search for your Microsoft AD directory ID. In the search results, select the item with the description "AWS created security group for *directory ID* directory controllers".

   > **Note**
   > The selected security group is a security group that is automatically created when you initially create your directory.

5. Go to the **Inbound Rules** tab of that security group. Select **Edit**, then **Add another rule**. For the new rule, enter the following values:

   - **Type**: Custom UDP Rule
   - **Protocol**: UDP
   - **Port Range**: 445
   - **Source**: *0.0.0.0/0*

6. Select **Save**.
7. Repeat these steps, adding each of the following rules:

Type	Protocol	Port Range	Source
Custom UDP Rule	UDP	88	*0.0.0.0/0*
Custom UDP Rule	UDP	123	*0.0.0.0/0*
Custom UDP Rule	UDP	138	*0.0.0.0/0*
Custom UDP Rule	UDP	389	*0.0.0.0/0*
Custom UDP Rule	UDP	464	*0.0.0.0/0*
Custom TCP Rule	TCP	88	*0.0.0.0/0*
Custom TCP Rule	TCP	135	*0.0.0.0/0*
Custom TCP Rule	TCP	445	*0.0.0.0/0*
Custom TCP Rule	TCP	464	*0.0.0.0/0*
Custom TCP Rule	TCP	636	*0.0.0.0/0*

Type	Protocol	Port Range	Source
Custom TCP Rule	TCP	1024 - 65535	*0.0.0.0/0*
Custom TCP Rule	TCP	3268 - 3269	*0.0.0.0/0*
DNS (UDP)	UDP	53	*0.0.0.0/0*
DNS (TCP)	TCP	53	*0.0.0.0/0*
LDAP	TCP	389	*0.0.0.0/0*
All ICMP	All	N/A	*0.0.0.0/0*
All traffic	All	All	*The current security group (The security group for your directory)*

These security rules impact an internal network interface that is not exposed publicly.

Enable Kerberos Pre-authentication

Your user accounts must have Kerberos pre-authentication enabled. For more information about this setting, review Preauthentication on Microsoft TechNet.

Configure DNS Conditional Forwarders On Your On-premises domain

You must set up DNS conditional forwarders on your on-premises domain. Refer to Assign a Conditional Forwarder for a Domain Name on Microsoft TechNet for details on conditional forwarders.

To perform the following steps, you must have access to following Windows Server tools for your on-premises domain:

- AD DS and AD LDS Tools
- DNS

To configure conditional forwarders on your on-premises domain

1. First you must get some information about your AWS Microsoft AD. Sign into the AWS Management Console and open the AWS Directory Service console at https://console.aws.amazon.com/directoryservice/.
2. In the navigation pane, select **Directories**.
3. Choose the directory ID of your Microsoft AD.
4. Take note of the fully qualified domain name (FQDN) and the DNS addresses of your directory.
5. Now, return to your on-premises domain controller. Open Server Manager.
6. On the **Tools** menu, choose **DNS**.

7. In the console tree, expand the DNS server of the domain for which you are setting up the trust.
8. In the console tree, choose **Conditional Forwarders**.
9. On the **Action** menu, choose **New conditional forwarder**.
10. In **DNS domain**, type the fully qualified domain name (FQDN) of your Microsoft AD, which you noted earlier.
11. Choose **IP addresses of the master servers** and type the DNS addresses of your Microsoft AD directory, which you noted earlier.

    After entering the DNS addresses, you might get a "timeout" or "unable to resolve" error. You can generally ignore these errors.
12. Select **Store this conditional forwarder in Active Directory and replicate as follows: All DNS servers in this domain**. Choose **OK**.

Trust Relationship Password

If you are creating a trust relationship with an existing domain, set up the trust relationship on that domain using Windows Server Administration tools. As you do so, note the trust password that you use. You will need to use this same password when setting up the trust relationship on the Microsoft AD. For more information, see Managing Trusts on Microsoft TechNet.

You are now ready to create the trust relationship on your Microsoft AD.

Create, Verify, or Delete a Trust Relationship

To create a trust relationship with your Microsoft AD

1. Open the AWS Directory Service console.
2. Choose the directory you want to configure.
3. On the **Details** page, choose the **Trusts** tab.
4. Choose **Add Trust Relationship**.
5. Provide the required information, including the fully qualified domain name (FQDN) of your trusted domain, the trust password and the trust direction.
6. For **Conditional forwarder**, type the IP address of your on-premises DNS server. If you have previously created conditional forwarders, you can type the fully qualified domain name (FQDN) of your on-premises domain instead of a DNS IP address.
7. (Optional) Choose **Add IP address** and type the IP address of an additional on-premises DNS server. You can repeat this step for each applicable DNS server address for a total of four addresses.
8. Choose **Create**.
9. If the DNS server for your on-premises domain uses a publicly addressable IP address, choose the **IP routing** tab and choose **Add route**. Type the IP address block of your DNS server using CIDR format, for example 10.0.0.0/24. This step is not necessary if your DNS server does not use a public IP address.
10. (Optional) We recommend that you also select **Add routes to the security group for this directory's VPC**. This will configure the security groups as detailed above in the "Configure your VPC." These security rules impact an internal network interface that is not exposed publicly.
    If this option is not available, you will instead see a message indicating that you have already customized your security groups.

You must set up the trust relationship on both domains. The relationships must be complementary. For example, if you create an outgoing trust on one domain, you must create an incoming trust on the other.

If you are creating a trust relationship with an existing domain, set up the trust relationship on that domain using Windows Server Administration tools.

You can create multiple trusts between your Microsoft AD and various Active Directory domains. However, only one trust relationship per pair can exist at a time. For example, if you have an existing, one-way trust in the “Incoming direction” and you then want to set up another trust relationship in the “Outgoing direction,” you will need to delete the existing trust relationship, and create a new “Two-way” trust.

To verify an outgoing trust relationship

1. Open the AWS Directory Service console.
2. Choose the directory you wish to configure.
3. On the **Details** page, choose the **Trusts** tab.
4. Choose the trust relationship to verify.
5. For **Actions**, choose **Verify**.

This process verifies only the outgoing direction of a two-way trust. AWS does not support verification of an incoming trusts. For more information on how to verify a trust to or from your on-premises Active Directory, refer to Verify a Trust on Microsoft Technet.

To delete an existing trust relationship

1. Open the AWS Directory Service console.
2. Choose the directory you wish to configure.
3. On the **Details** page, choose the **Trusts** tab.
4. Choose the trust relationship to delete.
5. For **Actions**, choose **Delete**.

Topics

Adding IP Routes When Using Public IP Addresses

You can use AWS Directory Service for Microsoft Active Directory (Enterprise Edition) to take advantage of many powerful Active Directory features, including establishing trusts with other directories. However, if the DNS servers for the other directories use public IP addresses, you must specify those IP addresses as part of configuring the trust. Instructions for doing this can be found in When to Create a Trust Relationship (p. 18).

Similarly, you must also enter the IP address information when routing traffic from your Microsoft AD on AWS to a peer AWS VPC, if the VPC uses public IP ranges.

When you add the IP addresses as described in When to Create a Trust Relationship (p. 18), you have the option of selecting **Add routes to the security group for this directory's VPC**. This option should be selected unless you have previously customized your security group to allow the necessary traffic as shown below. This option configures the security groups for your directory’s VPC as follows:

Inbound rules

Type	Protocol	Port Range	Source
Custom UDP Rule	UDP	88	*0.0.0.0/0*

Type	Protocol	Port Range	Source
Custom UDP Rule	UDP	123	*0.0.0.0/0*
Custom UDP Rule	UDP	138	*0.0.0.0/0*
Custom UDP Rule	UDP	389	*0.0.0.0/0*
Custom UDP Rule	UDP	445	*0.0.0.0/0*
Custom UDP Rule	UDP	464	*0.0.0.0/0*
Custom TCP Rule	TCP	88	*0.0.0.0/0*
Custom TCP Rule	TCP	135	*0.0.0.0/0*
Custom TCP Rule	TCP	445	*0.0.0.0/0*
Custom TCP Rule	TCP	464	*0.0.0.0/0*
Custom TCP Rule	TCP	636	*0.0.0.0/0*
Custom TCP Rule	TCP	1024 - 65535	*0.0.0.0/0*
Custom TCP Rule	TCP	3268 - 3269	*0.0.0.0/0*
DNS (UDP)	UDP	53	*0.0.0.0/0*
DNS (TCP)	TCP	53	*0.0.0.0/0*
LDAP	TCP	389	*0.0.0.0/0*
All ICMP	All	N/A	*0.0.0.0/0*

Outbound rules

Type	Protocol	Port Range	Destination
All traffic	All	All	*0.0.0.0/0*

These security rules affect an internal network interface that is not exposed publicly.

Tutorial: Create a Trust Relationship Between Your Microsoft AD and Your On-Premises Domain

This tutorial walks you through all the steps necessary to set up a trust relationship between AWS Directory Service for Microsoft Active Directory (Enterprise Edition) and your on-premises Microsoft Active Directory. Although creating the trust requires only a few steps, you must first complete the following prerequisite steps.

Topics

- Step 3: Create the Trust Relationship (p. 36)

Prerequisites

This tutorial assumes you already have the following:

- A Microsoft AD created on AWS. If you need help doing this, see Create a Microsoft AD Directory (p. 10).
- An EC2 instance running Windows added to that Microsoft AD. If you need help doing this, see Manually Add a Windows Instance (Simple AD and Microsoft AD) (p. 106).

  > **Important**
  > The admin account for your Microsoft AD must have administrative access to this instance.
- The following Windows Server tools installed on that instance:
  - AD DS and AD LDS Tools
  - DNS

  If you need help doing this, see Installing the Active Directory Administration Tools (p. 102).
- An on-premises Microsoft Active Directory

  You must have administrative access to this directory. The same Windows Server tools as listed above must also be available for this directory.
- An active connection between your on-premises network and the VPC containing your Microsoft AD. If you need help doing this, see Amazon Virtual Private Cloud Connectivity Options.

Tutorial Configuration

For this tutorial, we've already created a Microsoft AD and an on-premises domain. The on-premises network is connected to the Microsoft AD's VPC. Following are the properties of the two directories:

Microsoft AD running on AWS

- Domain name (FQDN): MyManagedAD.example.com
- NetBIOS name: MyManagedAD
- DNS Addresses: 10.0.10.246, 10.0.20.121
- VPC CIDR: 10.0.0.0/16

The Microsoft AD resides in VPC ID: vpc-12345678.

On-premises domain

- Domain name (FQDN): corp.example.com
- NetBIOS name: CORP
- DNS Addresses: 172.16.10.153
- On-premises CIDR: 172.16.0.0/16

You are now ready for Step 1: Prepare Your On-Premises Domain (p. 24).

Step 1: Prepare Your On-Premises Domain

First you need to complete several prerequisite steps on your on-premises domain.

Configure Your On-Premises Firewall

You must configure your on-premises firewall so that the following ports are open to the CIDRs for all subnets used by the VPC that contains your Microsoft AD. In this tutorial, we allow both incoming and outgoing traffic from 10.0.0.0/16 (the CIDR block of our Microsoft AD's VPC) on the following ports:

- TCP/UDP 53 - DNS
- TCP/UDP 88 - Kerberos authentication
- TCP/UDP 389 - LDAP
- TCP 445 - SMB

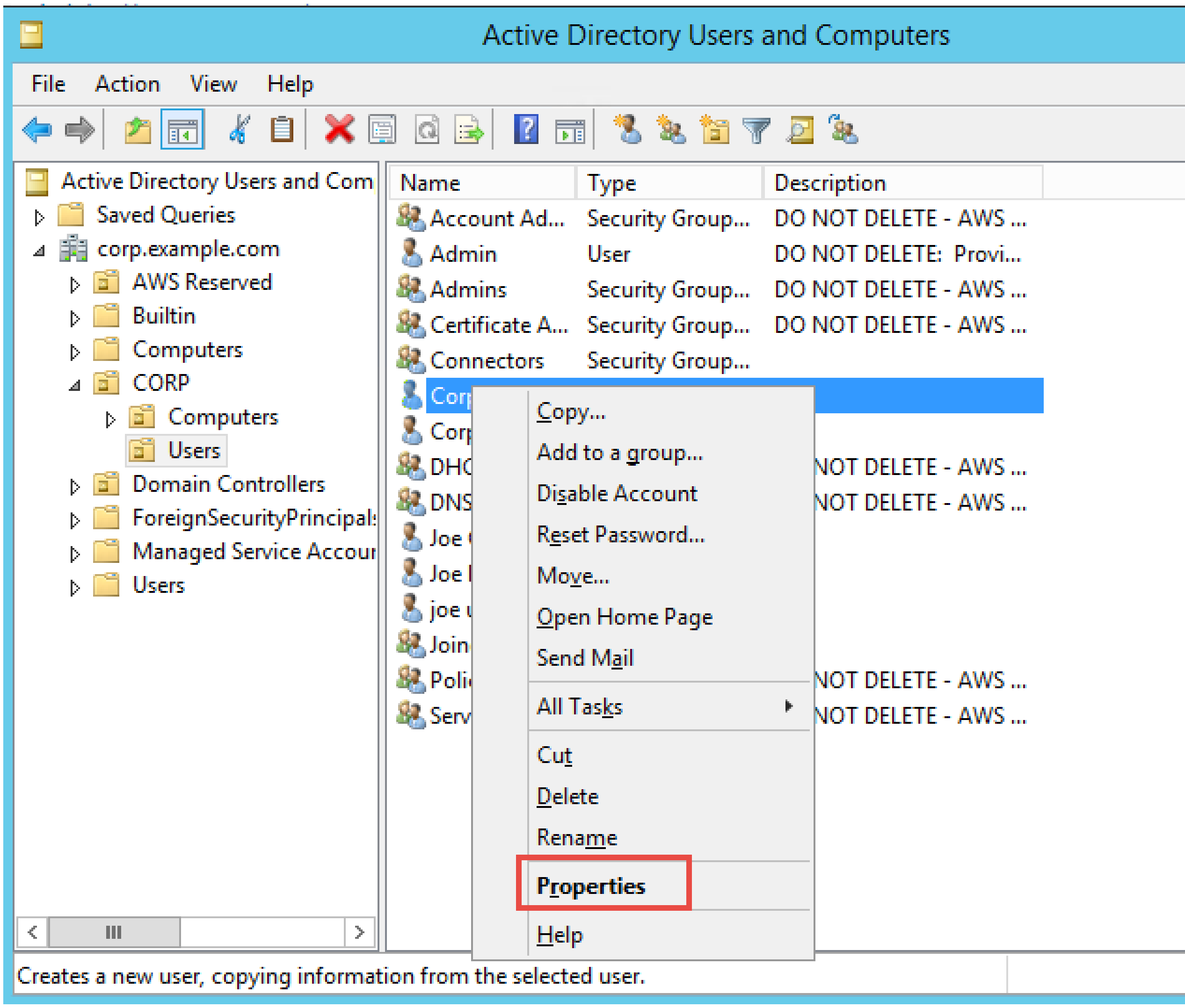
> **Note**
> These are the minimum ports that are needed to connect the VPC to the on-premises directory. Your specific configuration may require additional ports be open.

Ensure That Kerberos Pre-authentication Is Enabled

User accounts in both directories must have Kerberos preauthentication enabled. This is the default, but let's check to make sure nothing has changed.

To view user Kerberos settings

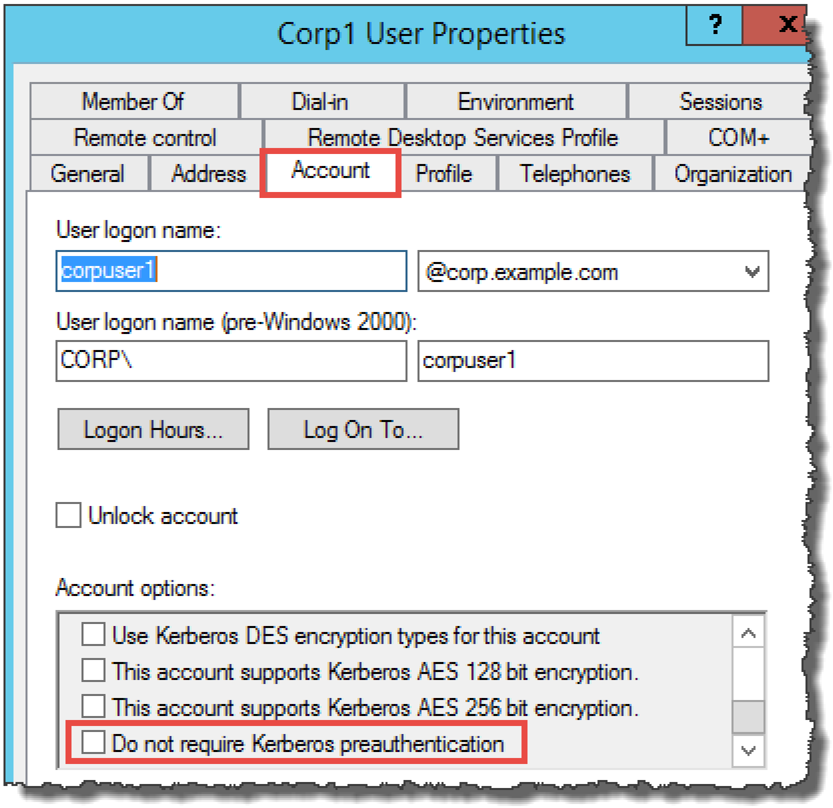
1. On your on-premises domain controller, open Server Manager.
2. On the **Tools** menu, choose **Active Directory Users and Computers**.
3. Choose the **Users** folder and open the context (right-click) menu for a user account listed in the right pane. Choose **Properties**.



Name	Type	Description
Account Ad...	Security Group...	DO NOT DELETE - AWS ...
Admin	User	DO NOT DELETE: Provi...
Admins	Security Group...	DO NOT DELETE - AWS ...
Certificate A...	Security Group...	DO NOT DELETE - AWS ...
Connectors	Security Group...	
Corp		
Corp		
DHC		NOT DELETE - AWS ...
DNS		NOT DELETE - AWS ...
Joe		
Joe		
joe		
Join		
Poli		NOT DELETE - AWS ...
Serv		NOT DELETE - AWS ...

Creates a new user, copying information from the selected user.

4. Choose the **Account** tab. In the **Account options** list, scroll down and ensure that **Do not require Kerberos preauthentication** is *not* checked.



User logon name:
corpuser1 @corp.example.com

User logon name (pre-Windows 2000):
CORP\ corpuser1

☐ Unlock account

Account options:

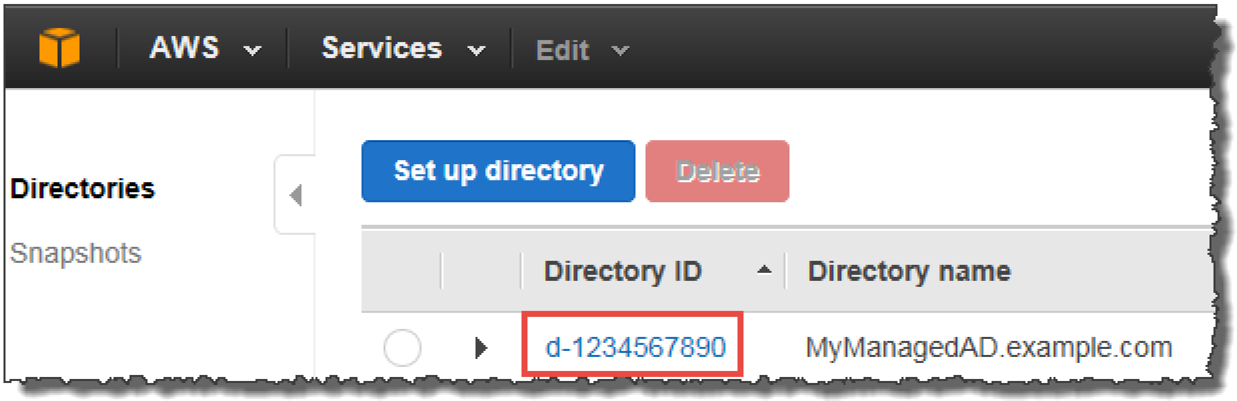
☐ Use Kerberos DES encryption types for this account
☐ This account supports Kerberos AES 128 bit encryption.
☐ This account supports Kerberos AES 256 bit encryption.
☐ Do not require Kerberos preauthentication

Configure DNS Conditional Forwarders for Your On-premises Domain

You must set up DNS conditional forwarders on each domain. Before doing this on your on-premises domain, you will first get some information about your AWS Microsoft AD.

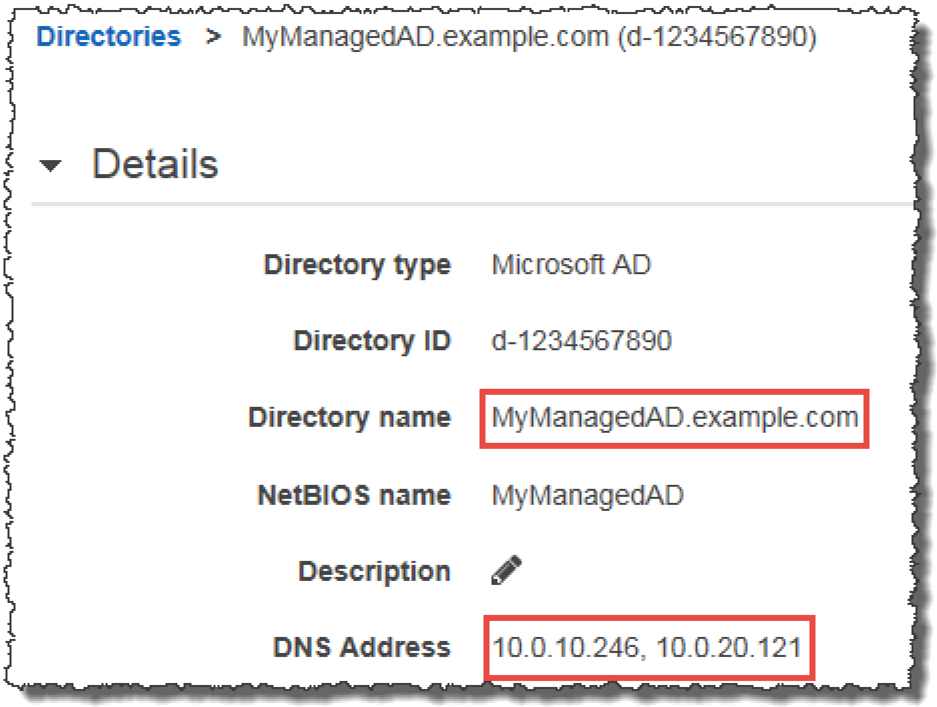
To configure conditional forwarders on your on-premises domain

1. Sign into the AWS Management Console and open the AWS Directory Service console at https://console.aws.amazon.com/directoryservice/.
2. In the navigation pane, select **Directories**.
3. Choose the directory ID of your Microsoft AD.



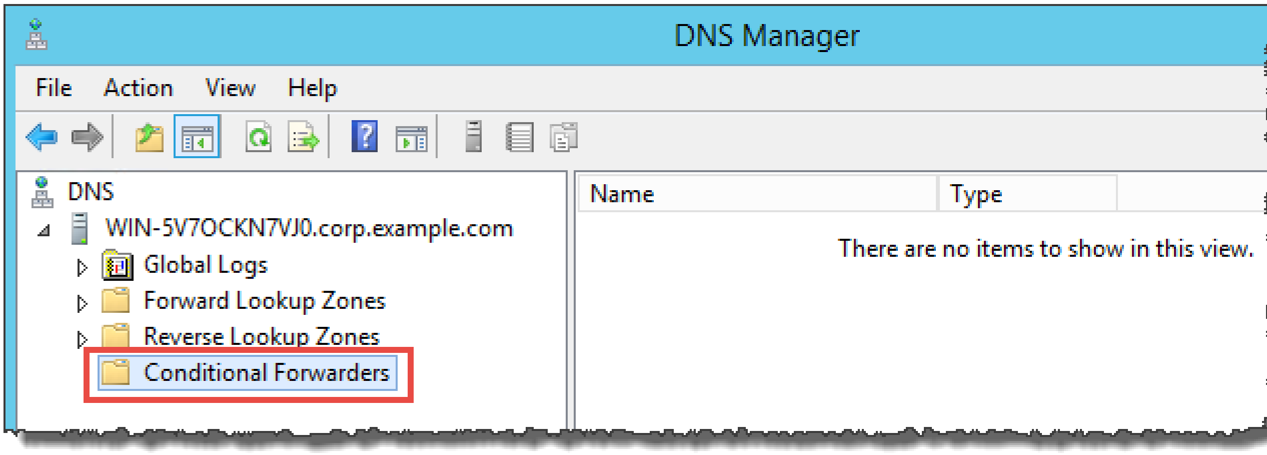
Directory ID	Directory name
d-1234567890	MyManagedAD.example.com

4. Take note of the fully qualified domain name (FQDN) and the DNS addresses of your directory.

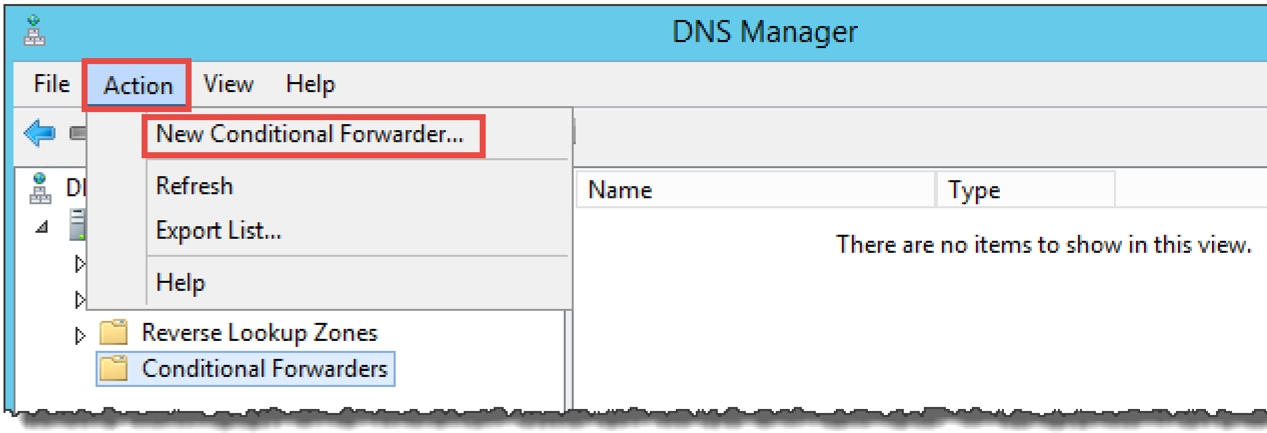


Directory type	Microsoft AD
Directory ID	d-1234567890
Directory name	MyManagedAD.example.com
NetBIOS name	MyManagedAD
Description	
DNS Address	10.0.10.246, 10.0.20.121

5. Now, return to your on-premises domain controller. Open Server Manager.
6. On the **Tools** menu, choose **DNS**.
7. In the console tree, expand the DNS server of the domain for which you are setting up the trust. Our server is WIN-5V70CN7VJ0.corp.example.com.
8. In the console tree, choose **Conditional Forwarders**.



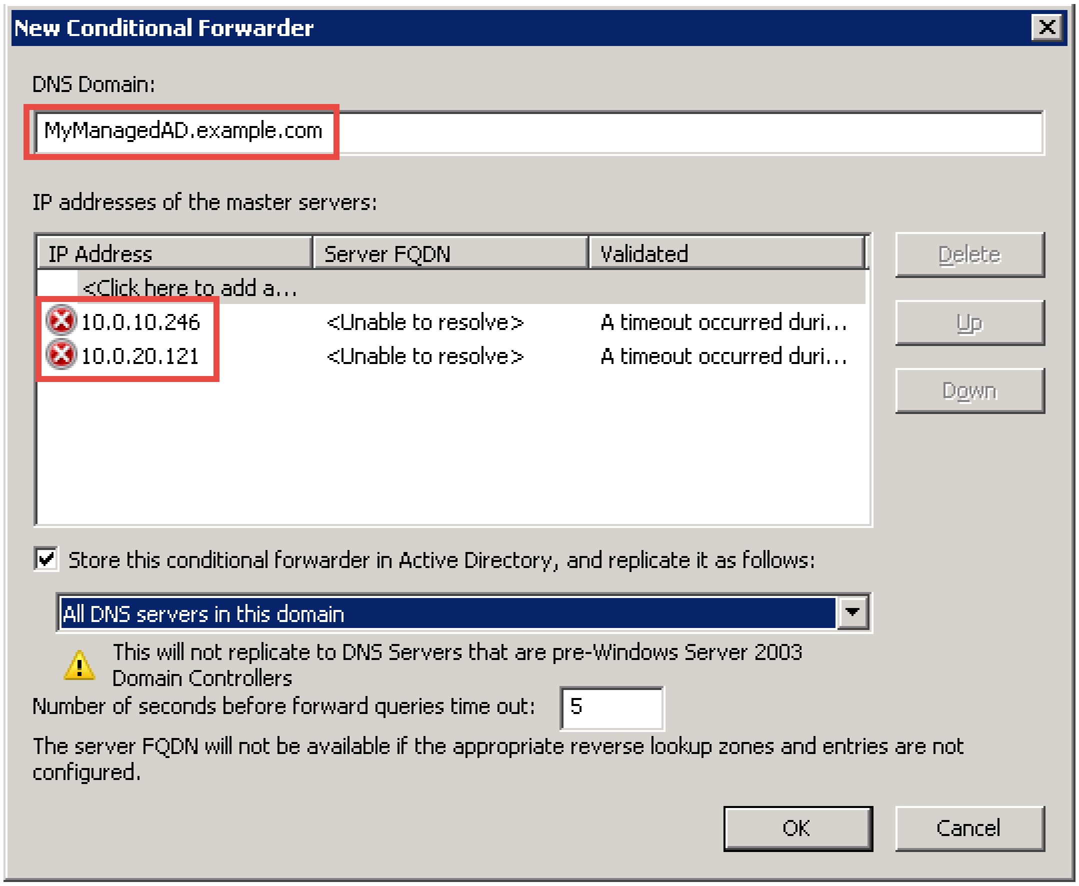
9. On the **Action** menu, choose **New conditional forwarder**.



10. In **DNS domain**, type the fully qualified domain name (FQDN) of your Microsoft AD, which you noted earlier. In this example, the FQDN is MyManagedAD.example.com.

11. Choose **IP addresses of the master servers** and type the DNS addresses of your Microsoft AD directory, which you noted earlier. In this example those are: 10.0.10.246, 10.0.20.121

    After entering the DNS addresses, you might get a "timeout" or "unable to resolve" error. You can generally ignore these errors.



DNS Domain: MyManagedAD.example.com

IP addresses of the master servers:

IP Address	Server FQDN	Validated
<Click here to add a...		
☒ 10.0.10.246	<Unable to resolve>	A timeout occurred duri...
☒ 10.0.20.121	<Unable to resolve>	A timeout occurred duri...

☑ Store this conditional forwarder in Active Directory, and replicate it as follows:

All DNS servers in this domain

⚠ This will not replicate to DNS Servers that are pre-Windows Server 2003 Domain Controllers

Number of seconds before forward queries time out: 5

The server FQDN will not be available if the appropriate reverse lookup zones and entries are not configured.

12. Select **Store this conditional forwarder in Active Directory and replicate as follows: All DNS servers in this forest**. Choose **OK**.

You're now done preparing your on-premises directory and ready for Step 2: Prepare Your Microsoft AD (p. 30).

Step 2: Prepare Your Microsoft AD

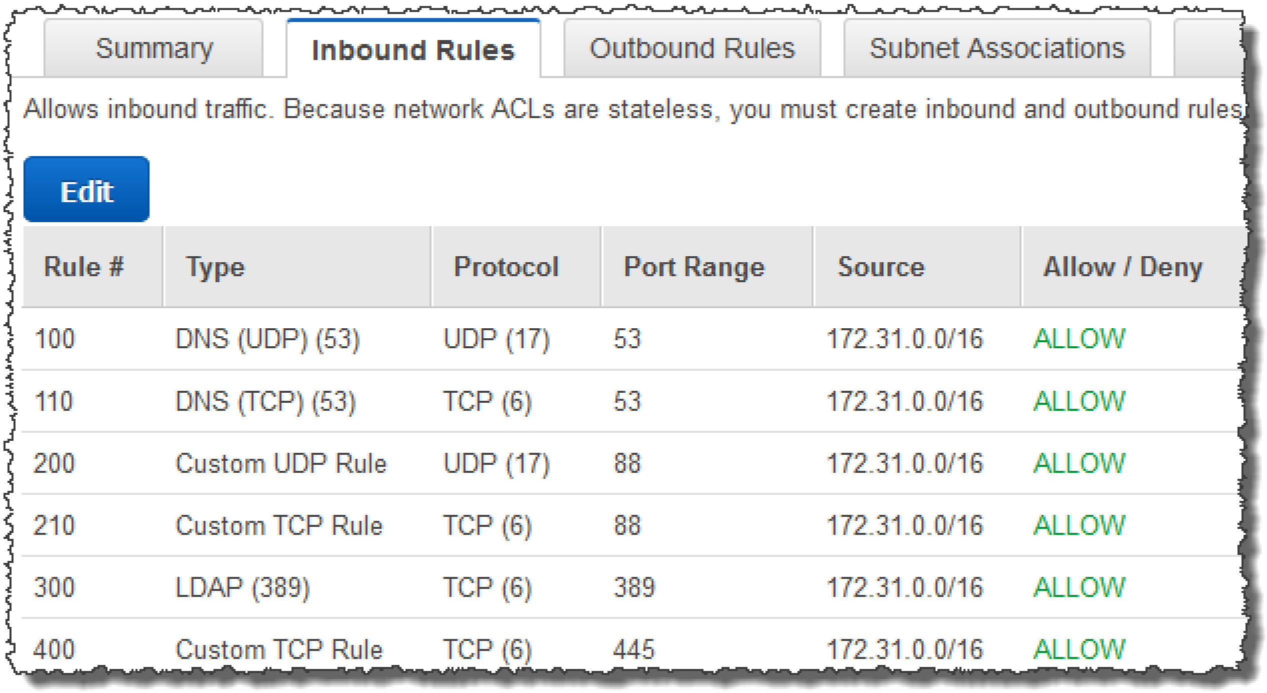
Now let's get your Microsoft AD ready for the trust relationship. Many of the following steps are almost identical to what you just completed for your on-premises domain. This time, however, you are working with your Microsoft AD.

Configure Your VPN Subnets and Security Groups

You must allow traffic from your on-premises network to the VPC containing your Microsoft AD. To do this, configure the VPC access control list (ACL) to allow both incoming and outgoing traffic from your on-premises directory for the following ports:

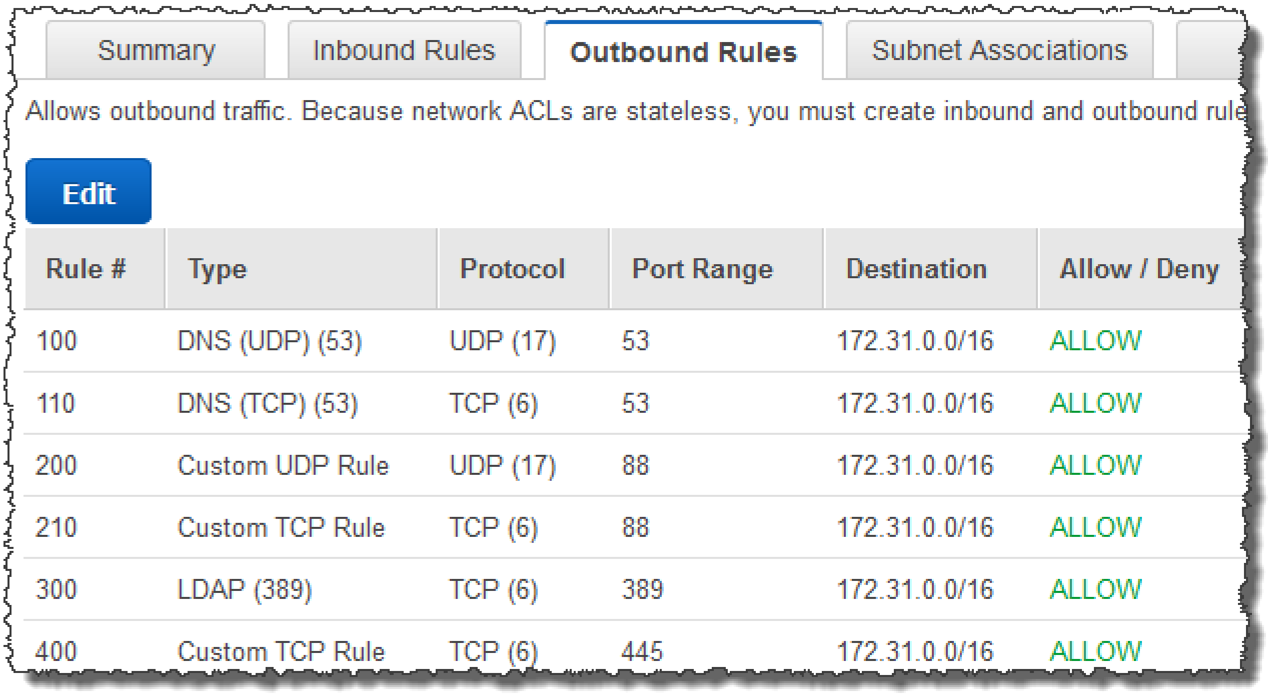
- TCP/UDP 53 - DNS
- TCP/UDP 88 - Kerberos authentication
- TCP/UDP 389 - LDAP
- TCP 445 - SMB

Note
These are the minimum ports that are needed to be able to connect the VPC and on-premises directory. Your specific configuration may require additional ports be open. For this tutorial, we have opened up all ports to our on-premises domain:



Allows inbound traffic. Because network ACLs are stateless, you must create inbound and outbound rules

Rule #	Type	Protocol	Port Range	Source	Allow / Deny
100	DNS (UDP) (53)	UDP (17)	53	172.31.0.0/16	ALLOW
110	DNS (TCP) (53)	TCP (6)	53	172.31.0.0/16	ALLOW
200	Custom UDP Rule	UDP (17)	88	172.31.0.0/16	ALLOW
210	Custom TCP Rule	TCP (6)	88	172.31.0.0/16	ALLOW
300	LDAP (389)	TCP (6)	389	172.31.0.0/16	ALLOW
400	Custom TCP Rule	TCP (6)	445	172.31.0.0/16	ALLOW



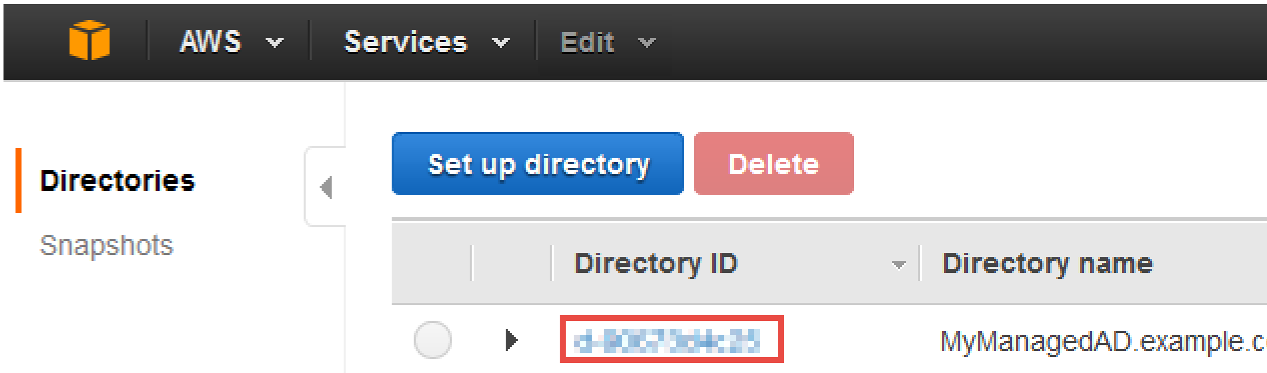
Allows outbound traffic. Because network ACLs are stateless, you must create inbound and outbound rule

Rule #	Type	Protocol	Port Range	Destination	Allow / Deny
100	DNS (UDP) (53)	UDP (17)	53	172.31.0.0/16	ALLOW
110	DNS (TCP) (53)	TCP (6)	53	172.31.0.0/16	ALLOW
200	Custom UDP Rule	UDP (17)	88	172.31.0.0/16	ALLOW
210	Custom TCP Rule	TCP (6)	88	172.31.0.0/16	ALLOW
300	LDAP (389)	TCP (6)	389	172.31.0.0/16	ALLOW
400	Custom TCP Rule	TCP (6)	445	172.31.0.0/16	ALLOW

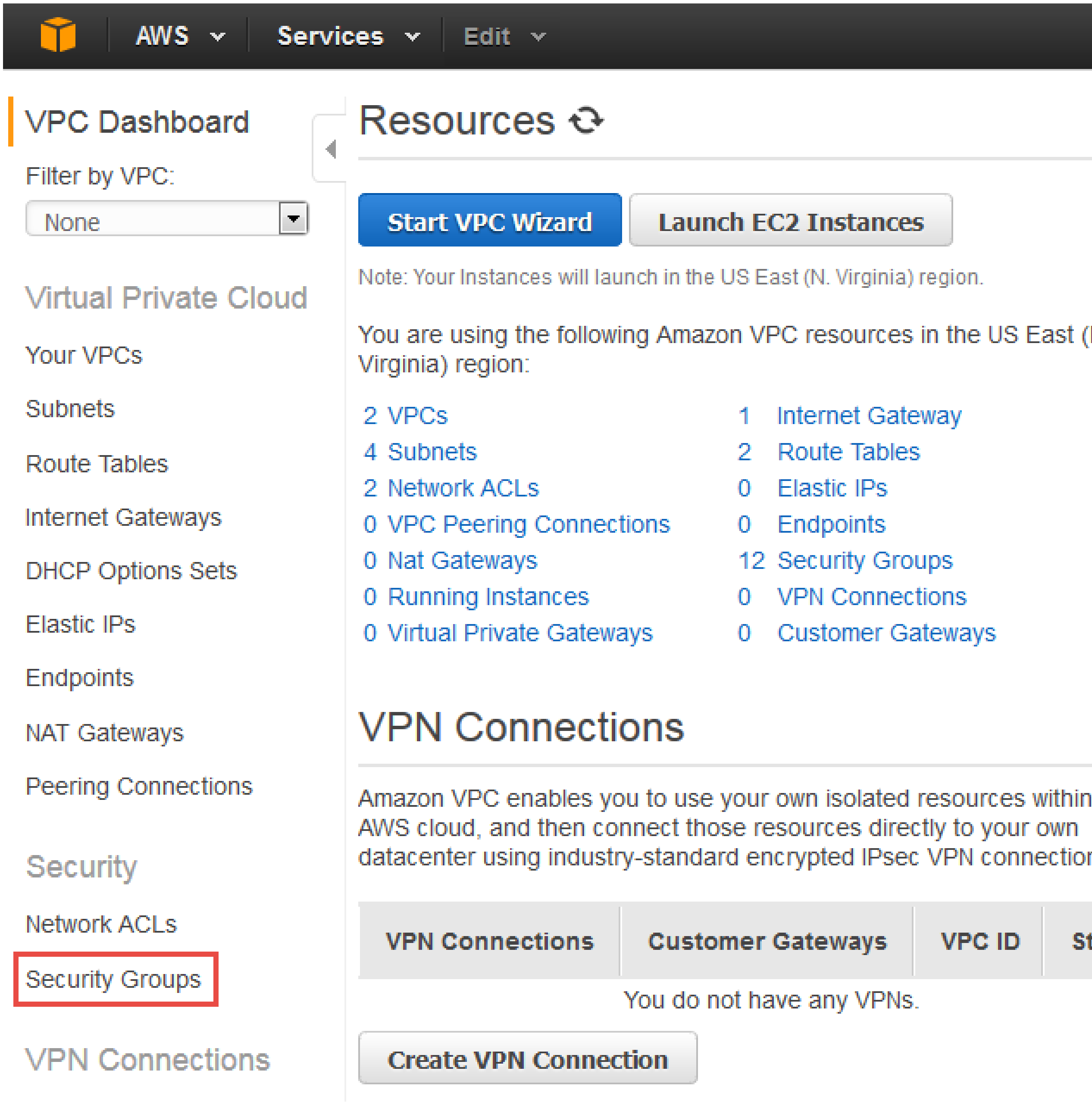
Similarly, your Microsoft AD domain controller must have the appropriate outbound and inbound rules.

To configure your Microsoft AD domain controller outbound and inbound rules

1. Return to the AWS Directory Service console at https://console.aws.amazon.com/directoryservice/. On the **Directory Details** page, note your Microsoft AD directory ID.



2. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
3. In the navigation pane, choose **Security Groups**.



Resources

Note: Your Instances will launch in the US East (N. Virginia) region.

You are using the following Amazon VPC resources in the US East (N. Virginia) region:

2	VPCs	1	Internet Gateway
4	Subnets	2	Route Tables
2	Network ACLs	0	Elastic IPs
0	VPC Peering Connections	0	Endpoints
0	Nat Gateways	12	Security Groups
0	Running Instances	0	VPN Connections
0	Virtual Private Gateways	0	Customer Gateways

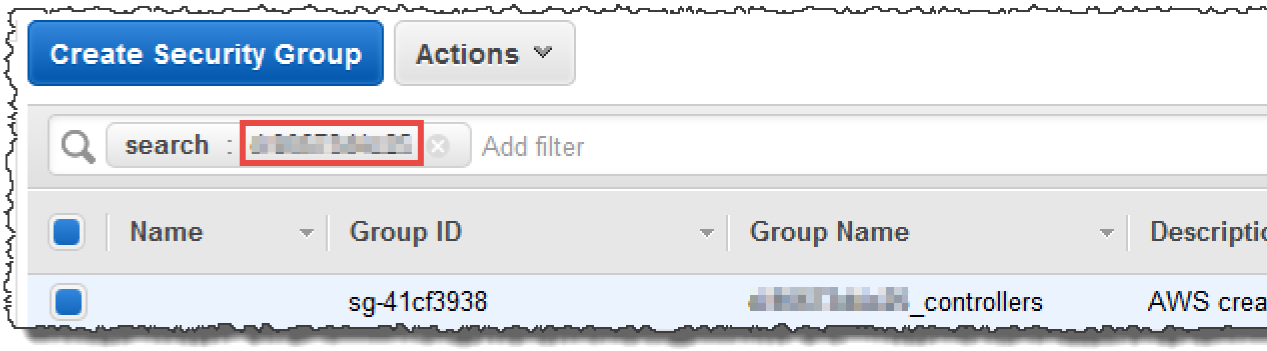
VPN Connections

Amazon VPC enables you to use your own isolated resources within AWS cloud, and then connect those resources directly to your own datacenter using industry-standard encrypted IPsec VPN connection

VPN Connections	Customer Gateways	VPC ID	St

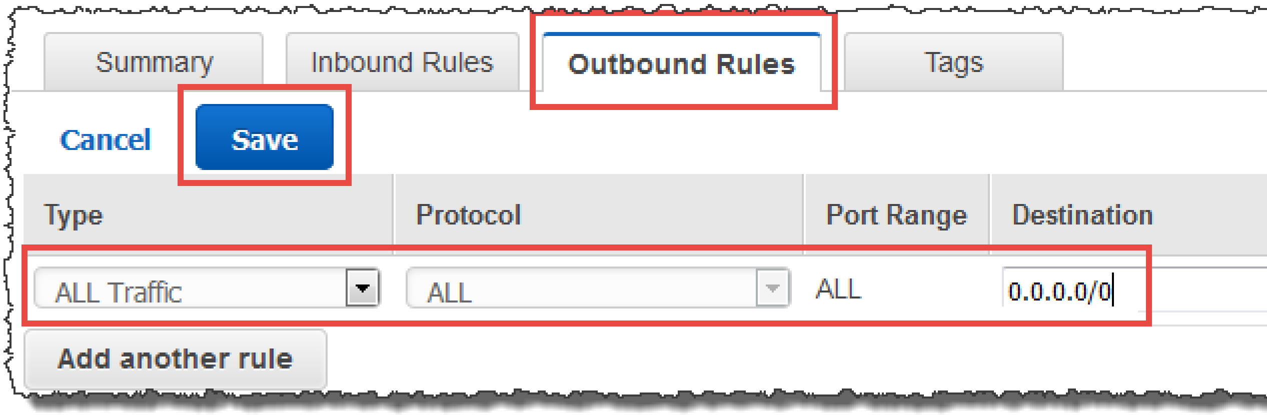
You do not have any VPNs.

4. Use the search box to search for your Microsoft AD directory ID. In the search results, select the item with the description **AWS created security group for *directory ID* directory controllers**.



Name	Group ID	Group Name	Descriptio
	sg-41cf3938	[redacted]_controllers	AWS crea

5. Go to the **Outbound Rules** tab for that security group. Choose **Edit**, and then **Add another rule**. For the new rule, enter the following values:
   - **Type**: ALL Traffic
   - **Protocol**: ALL
   - **Destination**: 0.0.0.0/0 (As a best practice, you can further restrict this by entering the address block of your on-premises domain, rather than 0.0.0.0/0.)
6. Select **Save**.



Type	Protocol	Port Range	Destination
ALL Traffic	ALL	ALL	0.0.0.0/0

7. Go to the **Inbound Rules** tab for that same security group. Choose **Edit**, and then **Add another rule**. For the new rule, enter the following values:
   - **Type**: Custom UDP Rule
   - **Protocol**: UDP
   - **Port Range**: 445
   - **Source**: *0.0.0.0/0*
8. Select **Save**.
9. Repeat these steps, adding each of the following rules:

Type	Protocol	Port Range	Destination/ Source
Custom UDP Rule	UDP	88	*0.0.0.0/0*
Custom UDP Rule	UDP	123	*0.0.0.0/0*

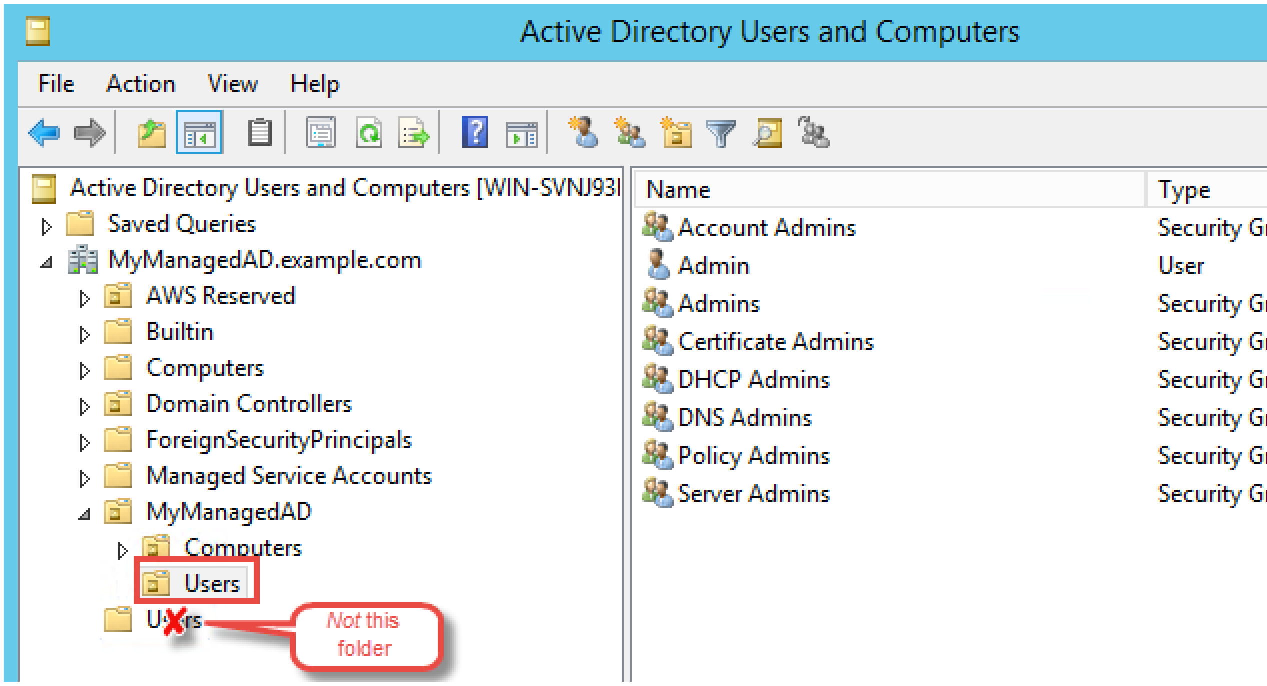
Type	**Protocol**	**Port Range**	**Destination/ Source**
Custom UDP Rule	UDP	138	*0.0.0.0/0*
Custom UDP Rule	UDP	389	*0.0.0.0/0*
Custom UDP Rule	UDP	464	*0.0.0.0/0*
Custom TCP Rule	TCP	88	*0.0.0.0/0*
Custom TCP Rule	TCP	135	*0.0.0.0/0*
Custom TCP Rule	TCP	445	*0.0.0.0/0*
Custom TCP Rule	TCP	464	*0.0.0.0/0*
Custom TCP Rule	TCP	636	*0.0.0.0/0*
Custom TCP Rule	TCP	1024 - 65535	*0.0.0.0/0*
Custom TCP Rule	TCP	3268 - 3269	*0.0.0.0/0*
DNS (UDP)	UDP	53	*0.0.0.0/0*
DNS (TCP)	TCP	53	*0.0.0.0/0*
LDAP	TCP	389	*0.0.0.0/0*
All ICMP	All	N/A	*0.0.0.0/0*
All traffic	All	All	*The current security group (The security group for your directory)*

Ensure That Kerberos Pre-authentication Is Enabled

Now you want to confirm that users in your Microsoft AD also have Kerberos pre-authentication enabled. This is the same process you completed for your on-premises directory. This is the default, but let's check to make sure nothing has changed.

To view user Kerberos settings

1. Log in to an instance that is a member of your Microsoft AD using an account that has domain administrative privileges.
2. If they are not already installed, install the Active Directory Users and Computers tool and the DNS tool. Learn how to install these tools in Installing the Active Directory Administration Tools (p. 102).
3. Open Server Manager. On the **Tools** menu, choose **Active Directory Users and Computers**.
4. Choose the **Users** folder in your domain. Note that this is the **Users** folder under your NetBIOS name, not the **Users** folder under the fully qualified domain name (FQDN).Open the context (right-click) menu for a user account and choose **Properties**.



Name	Type
Account Admins	Security Gr
Admin	User
Admins	Security Gr
Certificate Admins	Security Gr
DHCP Admins	Security Gr
DNS Admins	Security Gr
Policy Admins	Security Gr
Server Admins	Security Gr

5. Choose the **Account** tab. In the **Account options** list, ensure that **Do not require Kerberos preauthentication** is *not* checked.

You are now ready for Step 3: Create the Trust Relationship (p. 36).

Step 3: Create the Trust Relationship

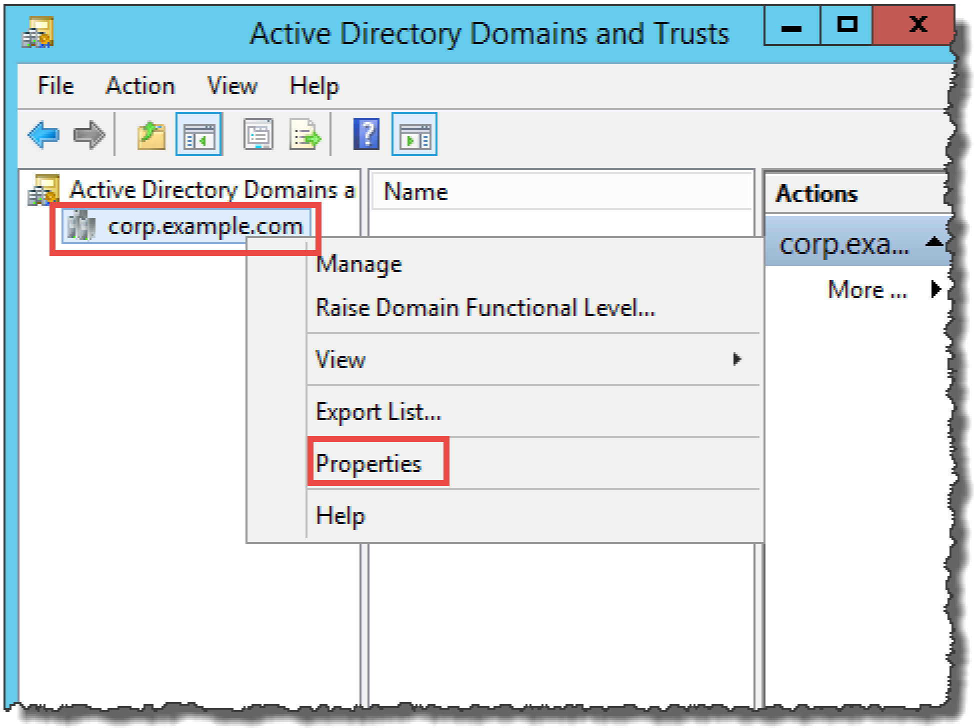
Now that the preparation work is complete, the final steps are to create the trusts. First you create the trust on your on-premises domain, and then finally on your Microsoft AD.

Configure Your On-Premises Trust

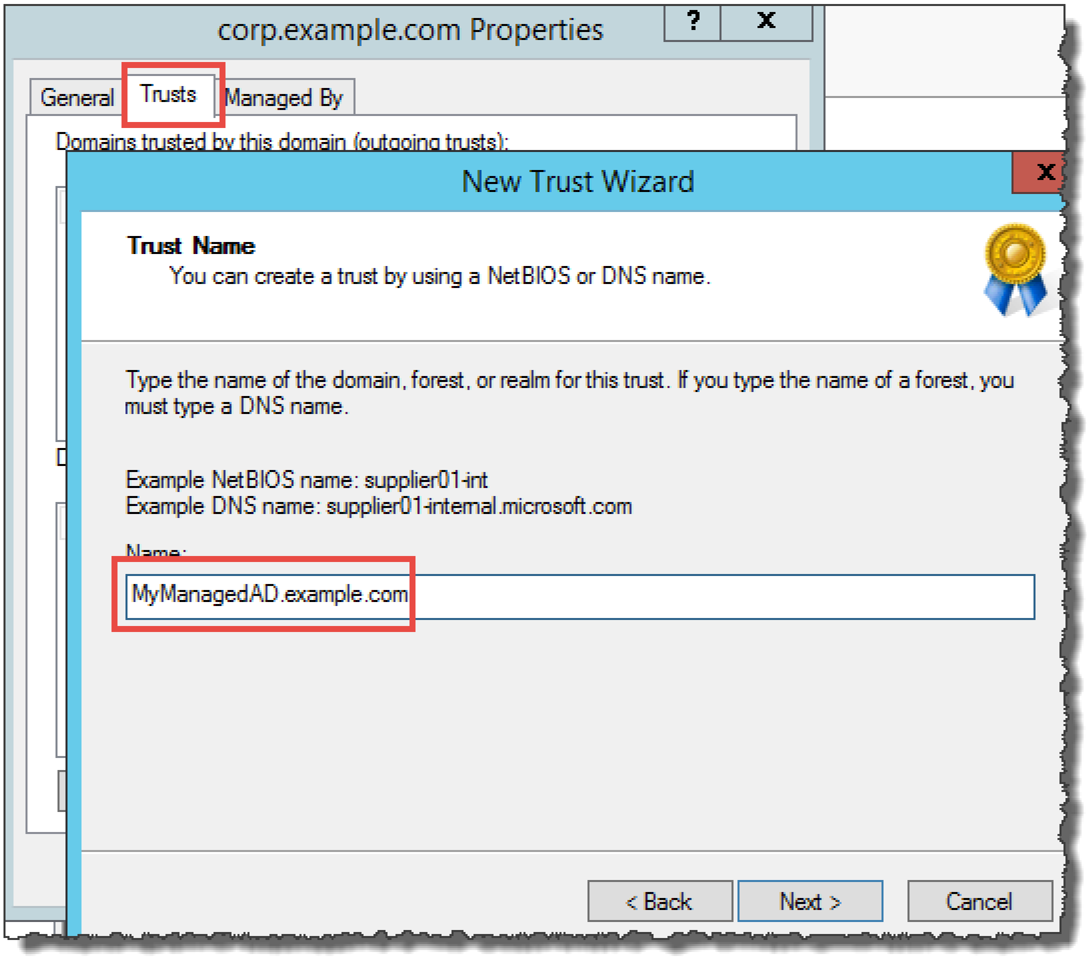
In this tutorial, you configure a two-way trust. However, if you create a one-way trust, be aware that the trust directions on each of your domains must be complementary. For example, if you create a one-way, outgoing trust on your on-premises domain, you need to create a one-way, incoming trust on your Microsoft AD.

To configure your on-premises trust relationship

1. Open Server Manager and on the **Tools** menu, choose **Active Directory Domains and Trusts**.
2. Open the context (right-click) menu of your domain and choose **Properties**.



3. Choose the **Trusts** tab and choose **New trust**. Type the name of your AWS Microsoft AD and choose **Next**.


corp.example.com Properties
?
X
General
Trusts
Managed By
Domains trusted by this domain (outgoing trusts):
New Trust Wizard
X

Trust Name

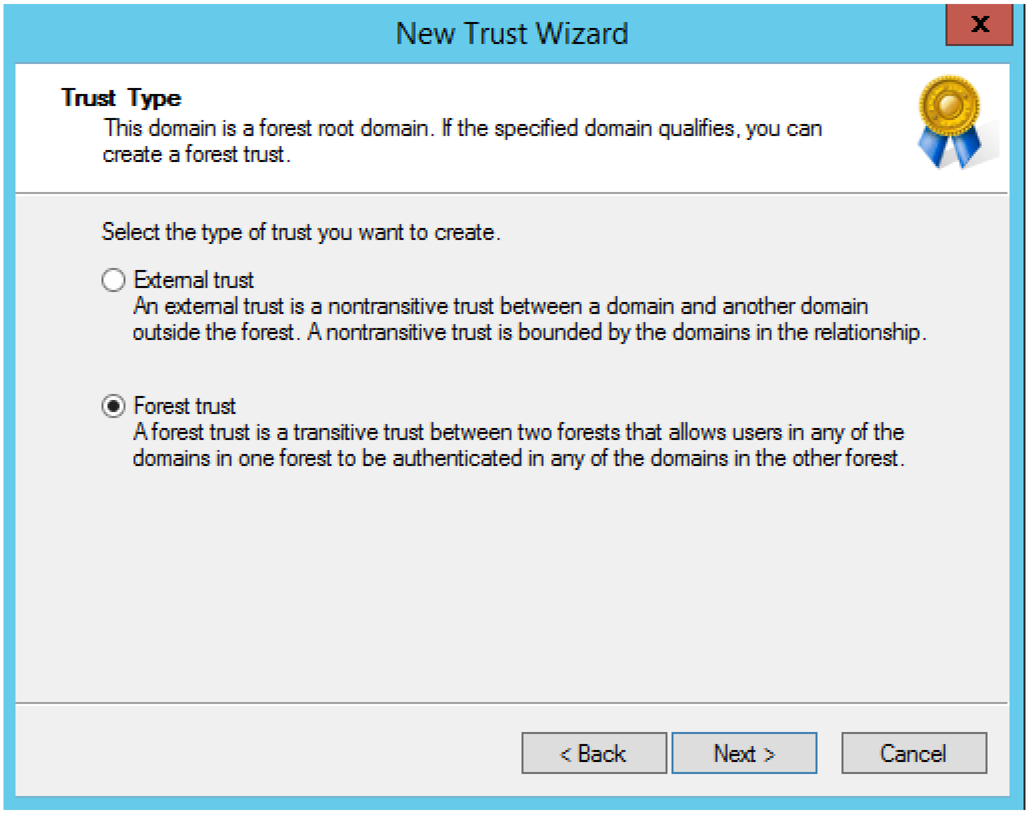
You can create a trust by using a NetBIOS or DNS name.

Type the name of the domain, forest, or realm for this trust. If you type the name of a forest, you must type a DNS name.

Example NetBIOS name: supplier01-int
Example DNS name: supplier01-internal.microsoft.com

Name:
MyManagedAD.example.com

4. Choose **Forest trust**. Choose **Next**.



Trust Type

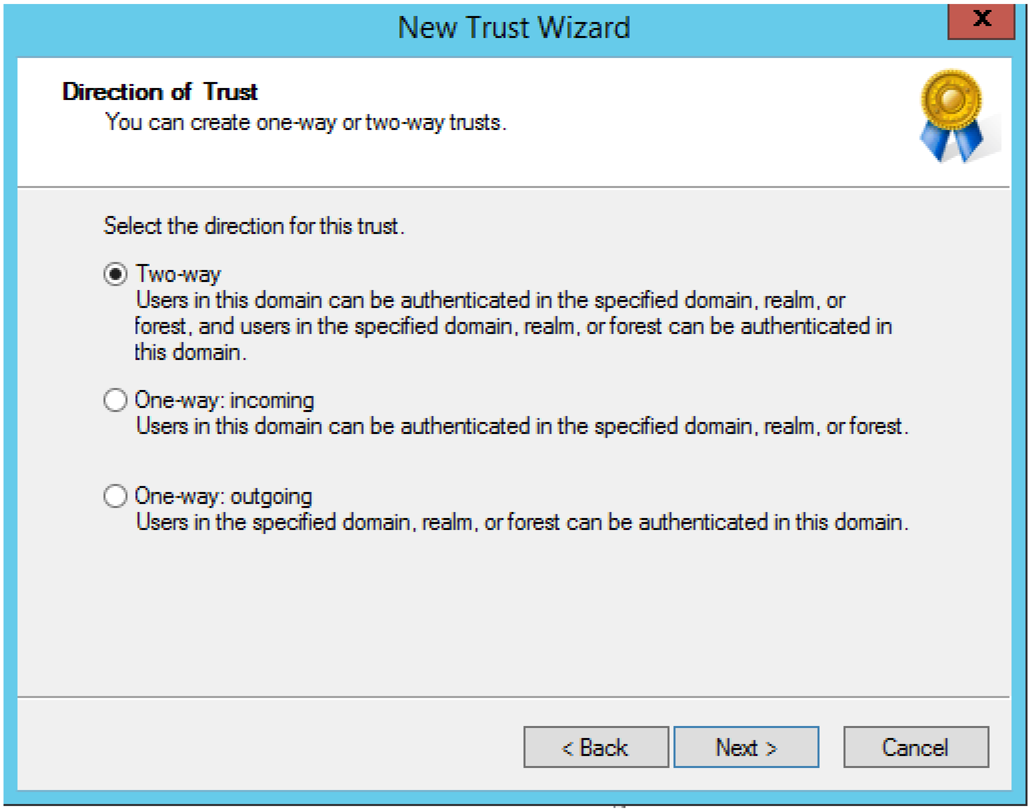
This domain is a forest root domain. If the specified domain qualifies, you can create a forest trust.

Select the type of trust you want to create.

○ External trust
An external trust is a nontransitive trust between a domain and another domain outside the forest. A nontransitive trust is bounded by the domains in the relationship.

● Forest trust
A forest trust is a transitive trust between two forests that allows users in any of the domains in one forest to be authenticated in any of the domains in the other forest.

5. Choose **Two-way**. Choose **Next**.



Direction of Trust

You can create one-way or two-way trusts.

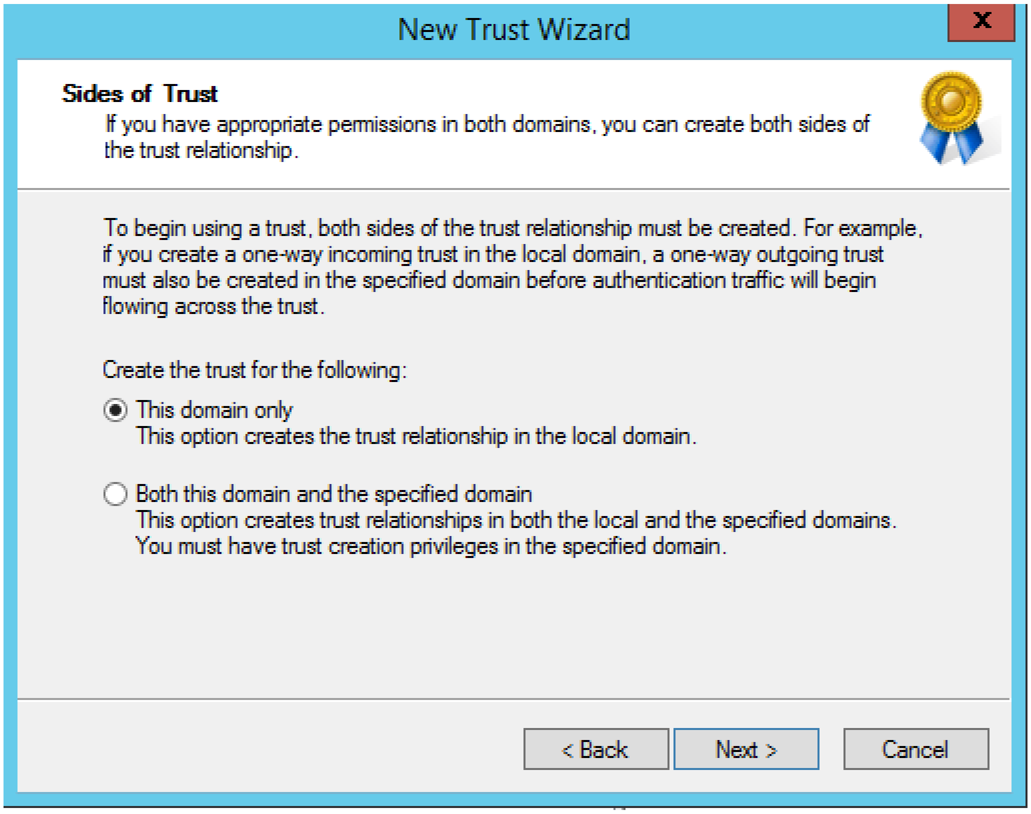
Select the direction for this trust.

● Two-way
Users in this domain can be authenticated in the specified domain, realm, or forest, and users in the specified domain, realm, or forest can be authenticated in this domain.

○ One-way: incoming
Users in this domain can be authenticated in the specified domain, realm, or forest.

○ One-way: outgoing
Users in the specified domain, realm, or forest can be authenticated in this domain.

6. Choose **This domain only**. Choose **Next**.



Sides of Trust

If you have appropriate permissions in both domains, you can create both sides of the trust relationship.

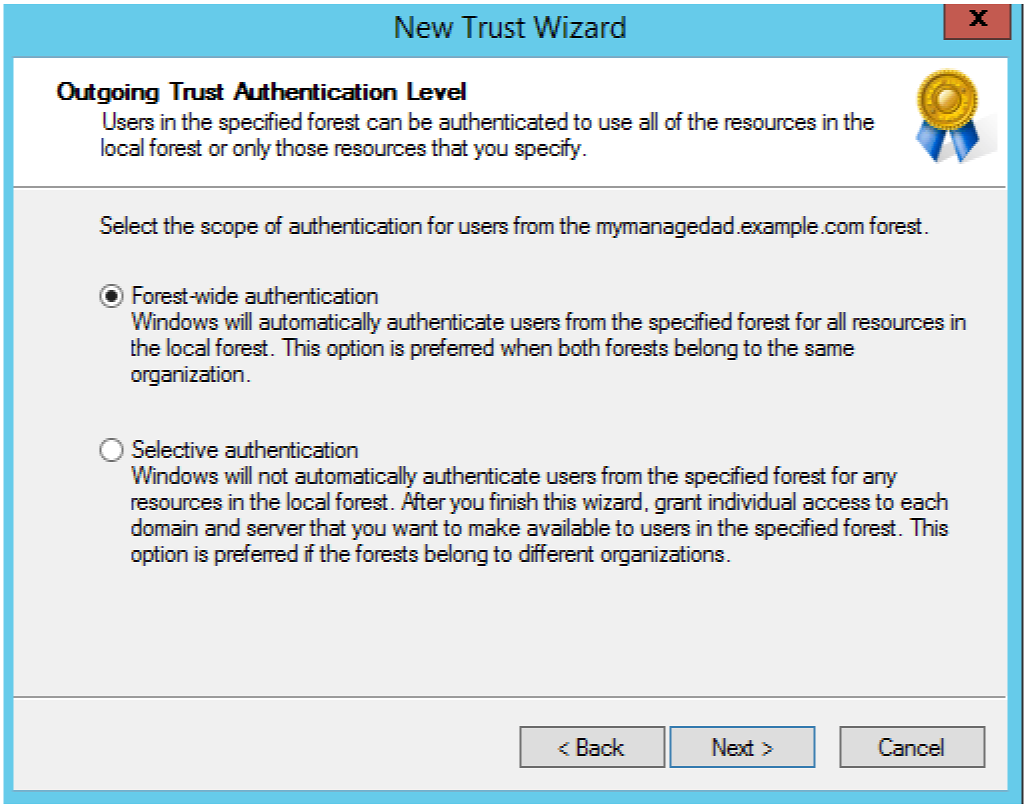
To begin using a trust, both sides of the trust relationship must be created. For example, if you create a one-way incoming trust in the local domain, a one-way outgoing trust must also be created in the specified domain before authentication traffic will begin flowing across the trust.

Create the trust for the following:

● This domain only
This option creates the trust relationship in the local domain.

○ Both this domain and the specified domain
This option creates trust relationships in both the local and the specified domains. You must have trust creation privileges in the specified domain.

7. Choose **Forest-wide authentication**. Choose **Next**.



Outgoing Trust Authentication Level

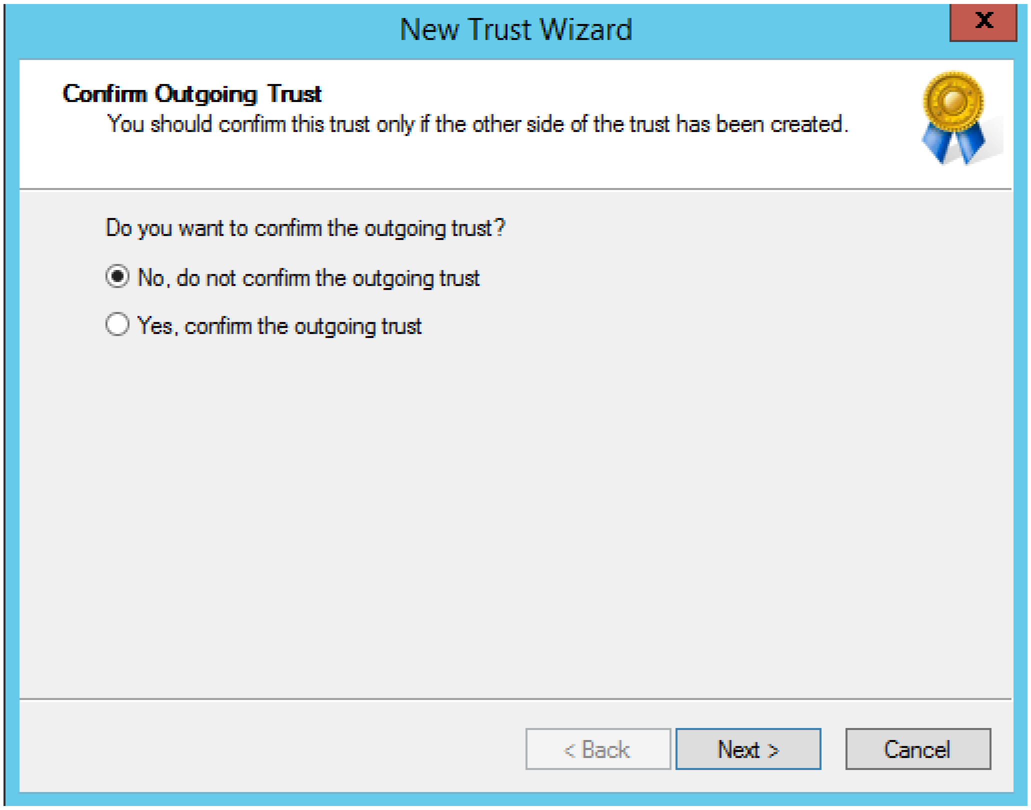
Users in the specified forest can be authenticated to use all of the resources in the local forest or only those resources that you specify.

Select the scope of authentication for users from the mymanagedad.example.com forest.

● Forest-wide authentication
Windows will automatically authenticate users from the specified forest for all resources in the local forest. This option is preferred when both forests belong to the same organization.

○ Selective authentication
Windows will not automatically authenticate users from the specified forest for any resources in the local forest. After you finish this wizard, grant individual access to each domain and server that you want to make available to users in the specified forest. This option is preferred if the forests belong to different organizations.

8. Type a **Trust password**. Make sure to remember this password as you will need it when setting up the trust for your AWS Microsoft AD.
9. In the next dialog box, confirm your settings and choose **Next**. Confirm that the trust was created successfully and again choose **Next**.
10. Choose **No, do not confirm the outgoing trust**. Choose **Next**.



Confirm Outgoing Trust

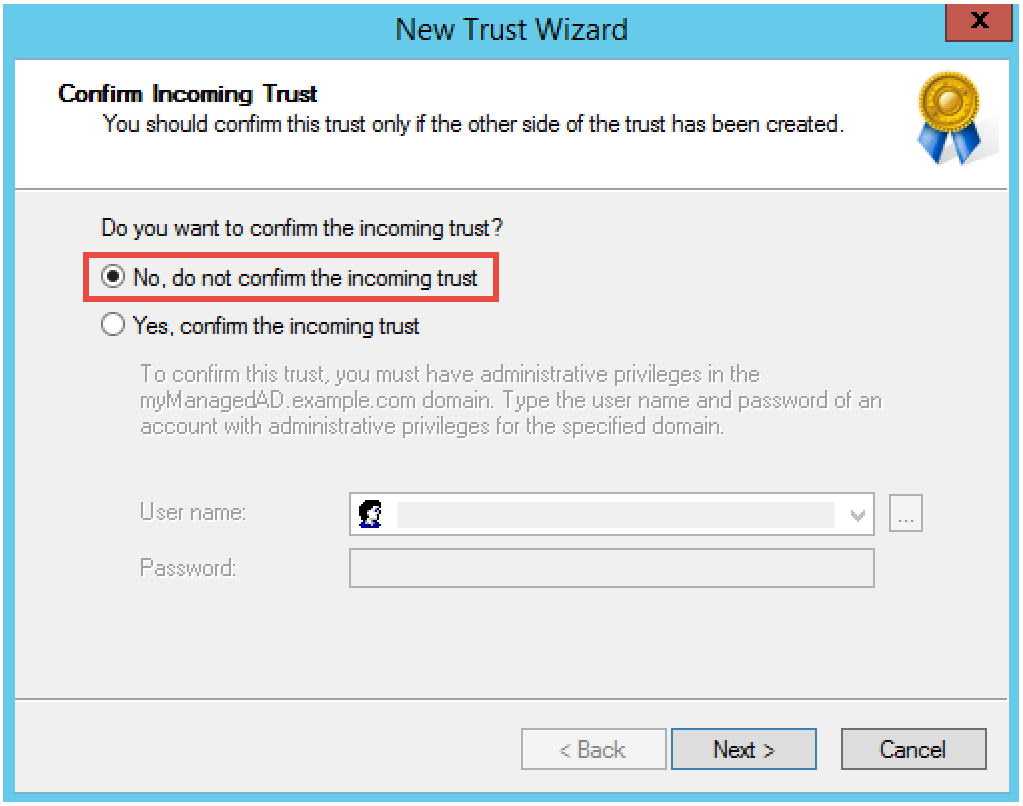
You should confirm this trust only if the other side of the trust has been created.

Do you want to confirm the outgoing trust?

● No, do not confirm the outgoing trust

○ Yes, confirm the outgoing trust

11. Choose **No, do not confirm the incoming trust**. Choose **Next**.



New Trust Wizard

Confirm Incoming Trust
You should confirm this trust only if the other side of the trust has been created.

Do you want to confirm the incoming trust?

● No, do not confirm the incoming trust

○ Yes, confirm the incoming trust

To confirm this trust, you must have administrative privileges in the myManagedAD.example.com domain. Type the user name and password of an account with administrative privileges for the specified domain.

User name:

Password:

Configure Your Microsoft AD Trust

Finally, you configure the trust relationship for your AWS Microsoft AD. Because you created a two-way trust on the on-premises domain, you also create a two-way trust on our Microsoft AD.

To configure your Microsoft AD trust relationship

1. Return to the AWS Directory Service console. On the **Directory Details** page, choose your Microsoft AD ID.
2. Choose the **Trust relationships** tab.
3. Choose **Add trust relationship**.
4. Type the FQDN of your on-premises domain (in this tutorial `corp.example.com`). Type the same trust password that you used when creating the trust on your on-premises domain. Specify the direction. In this case we choose **Two-way**.
5. In the **Conditional forwarder** field, enter the IP address of your on-premises DNS server. In this example, enter 172.16.10.153.
6. (Optional) Choose **Add IP address** and enter a second IP address your on-premises DNS server. You can specify up to a total of four DNS servers.
7. Choose **Create**.

Congratulations. You now have a trust relationship between your on-premises domain (corp.example.com) and your AWS Microsoft AD (MyManagedAD.example.com). Only one relationship can be set up between these two domains. If for example, you want to change the trust direction to one-way, you would first need to delete this existing trust relationship and create a new one.

For more information, including instructions about verifying or deleting trusts, see When to Create a Trust Relationship (p. 18).

Active Directory Connector

AD Connector is a directory gateway with which you can redirect directory requests to your on-premises Microsoft Active Directory without caching any information in the cloud. AD Connector comes in two sizes, small and large. A small AD Connector is designed for smaller organizations of up to 500 users. A large AD Connector can support larger organizations of up to 5,000 users.

Once set up, AD Connector offers the following benefits:

- Your end users and IT administrators can use their existing corporate credentials to log on to AWS applications such as Amazon WorkSpaces, Amazon WorkDocs, or Amazon WorkMail.
- You can manage AWS resources like Amazon EC2 instances or Amazon S3 buckets through IAM role-based access to the AWS Management Console.
- You can consistently enforce existing security policies (such as password expiration, password history, and account lockouts) whether users or IT administrators are accessing resources in your on-premises infrastructure or in the AWS Cloud.
- You can use AD Connector to enable multi-factor authentication by integrating with your existing RADIUS-based MFA infrastructure to provide an additional layer of security when users access AWS applications.

Continue reading the topics in this section to learn how to connect to a directory and make the most of AD Connector features.

Topics

Connect to a Directory

With AD Connector you can connect AWS Directory Service to your existing enterprise directory. When connected to your on-premises directory, all of your directory data remains on your directory servers. AWS Directory Service does not replicate any of your directory data.

Topics

Best Practices for AD Connector

We recommend that you follow these best practices for creating your AD Connector:

- Each AD Connector that you create must use a different service account, even if they are connected to the same directory.
- If your on-premises network has Active Directory sites defined, you must make sure the subnets in the VPC where your AD Connector resides are defined in an Active Directory site, and that there are no conflicts between the subnets in your VPC and the subnets in your other sites. To discover domain controllers AD Connector uses the Active Directory site whose subnet IP address ranges are close to those in the VPC containing the AD Connector. If you have a site that has subnets with the same IP address ranges as those in your VPC, the AD Connector will discover the domain controllers in that site, which may not be physically close to your region.
- When using AD Connector, you must ensure that your on-premises directory is and remains compatible with AWS Directory Services. For more information on your responsibilities, please see our shared responsibility model.

AD Connector Prerequisites

To connect to your on-premises directory with AD Connector, you need the following:

VPC

Set up a VPC with the following:

- At least two subnets. Each of the subnets must be in a different Availability Zone.
- The VPC must be connected to your on-premises network through a virtual private network (VPN) connection or AWS Direct Connect.
- The VPC must have default hardware tenancy.

For more information, see the following topics in the *Amazon VPC User Guide*:

- What is Amazon VPC?
- Subnets in your VPC
- Adding a Hardware Virtual Private Gateway to Your VPC

For more information about AWS Direct Connect, see the AWS Direct Connect User Guide.

On-premises network

You'll need an on-premises network with an Active Directory domain. The functional level of this domain must be `Windows Server 2003` or higher. AD Connector also supports connecting to a domain hosted on an AWS EC2 instance.

Credentials

You must have credentials for an account in the on-premises directory with the following privileges. For more information, see Delegating Connect Privileges (p. 48).

- Read users and groups
- Create computer objects
- Join computers to the domain

IP addresses

Get the IP addresses of two DNS servers or domain controllers in your on-premises directory.

AD Connector obtains the `_ldap._tcp.<DnsDomainName>` and `_kerberos._tcp.<DnsDomainName>` SRV records from these servers when connecting to your directory, so these servers must contain these SRV records. The AD Connector attempts to find a common domain controller that will provide both LDAP and Kerberos services, so these SRV records must include at least one common domain controller. For more information about SRV records, go to SRV Resource Records on Microsoft TechNet.

Ports for subnets

For AWS Directory Service to communicate with your on-premises directory, the firewall for your on-premises network must have the following ports open to the CIDRs for both subnets in the VPC.

- TCP/UDP 53 - DNS
- TCP/UDP 88 - Kerberos authentication
- TCP/UDP 389 - LDAP; note that AWS Directory Service does not support LDAP with SSL (LDAPS) or LDAP signing

These are the minimum ports that are needed to be able to connect to your directory. Your specific configuration may require additional ports be open.

Kerberos preauthentication

Your user accounts must have Kerberos preauthentication enabled. For more information about this setting, go to Preauthentication on Microsoft TechNet.

Encryption type

Your on-premises domain controller must have RC4-HMAC encryption enabled.

Multi-factor Authentication Prerequisites

To support multi-factor authentication with your AD Connector directory, you need the following:

- A Remote Authentication Dial In User Service (RADIUS) server in your on-premises network that has two client endpoints. The RADIUS client endpoints have the following requirements:
  - To create the endpoints, you need the IP addresses of the AWS Directory Service servers. These IP addresses can be obtained from the **Directory IP Address** field of your directory details.
  - Both RADIUS endpoints must use the same shared secret code.
- Your on-premises network must allow inbound traffic over the default RADIUS server port (1812) from the AWS Directory Service servers.
- The usernames between your RADIUS server and your on-premises directory must be identical.

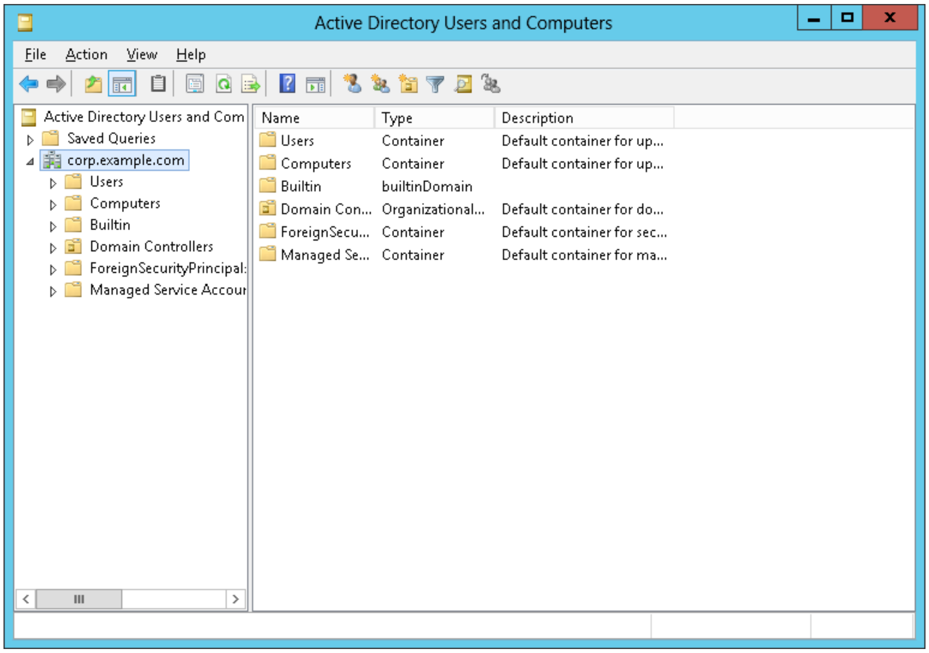
Delegating Connect Privileges

To connect to your on-premises directory, you must have the credentials for an account in the on-premises directory that has certain privileges. While members of the **Domain Admins** group have sufficient privileges to connect to the directory, as a best practice, you should use an account that only has the minimum privileges necessary to connect to the directory. The following procedure demonstrates how to create a new group called `Connectors`, and delegate the privileges to this group that are needed to connect AWS Directory Service to the directory.

This procedure must be performed on a machine that is joined to your directory and has the **Active Directory User and Computers** MMC snap-in installed. You must also be logged in as a domain administrator.

To delegate connect privileges

1. Open **Active Directory User and Computers** and select your domain root in the navigation tree.

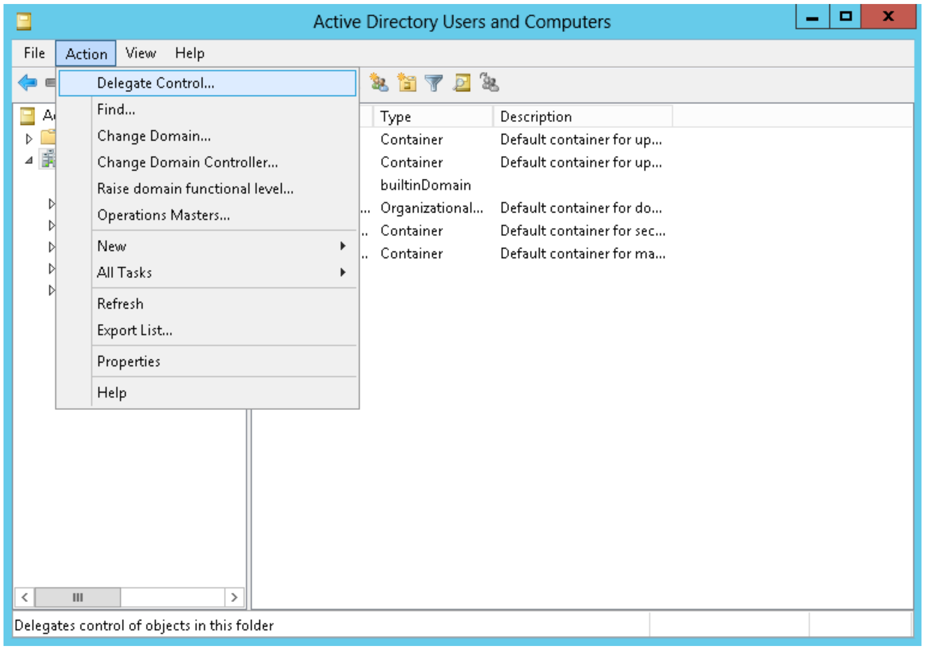


Name	Type	Description
Users	Container	Default container for up...
Computers	Container	Default container for up...
Builtin	builtinDomain	
Domain Con...	Organizational...	Default container for do...
ForeignSecu...	Container	Default container for sec...
Managed Se...	Container	Default container for ma...

2. In the list in the left-hand pane, right-click **Users**, select **New**, and then select **Group**.
3. In the **New Object - Group** dialog box, enter the following and click **OK**.

Field	Value/Selection
Group name	`Connectors`
Group scope	**Global**
Group type	**Security**

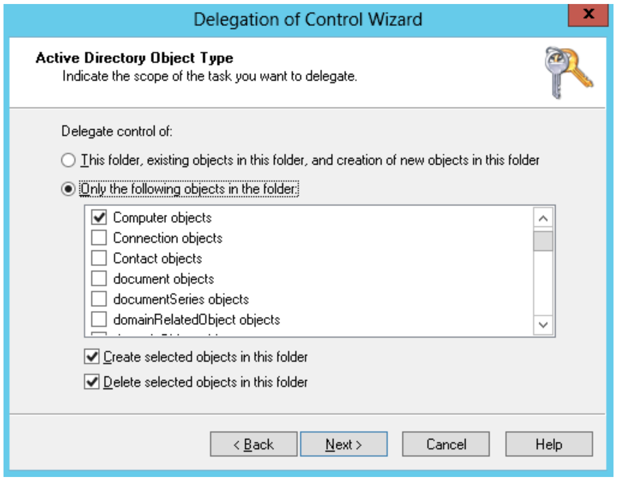
4. In the **Active Directory User and Computers** navigation tree, select your domain root. In the menu, select **Action**, and then **Delegate Control**.



Type	Description
Container	Default container for up...
Container	Default container for up...
builtinDomain	
Organizational...	Default container for do...
Container	Default container for sec...
Container	Default container for ma...

Delegates control of objects in this folder

5. On the **Delegation of Control Wizard** page, click **Next**, then click **Add**.
6. In the **Select Users, Computers, or Groups** dialog box, enter `Connectors` and click **OK**. If more than one object is found, select the `Connectors` group created above. Click **Next**.
7. On the **Tasks to Delegate** page, select **Create a custom task to delegate**, and then choose **Next**.
8. Select **Only the following objects in the folder**, and then select **Computer objects** and **User objects**.
9. Select **Create selected objects in this folder** and **Delete selected objects in this folder**. Then choose **Next**.



Active Directory Object Type
Indicate the scope of the task you want to delegate.

Delegate control of:

○ This folder, existing objects in this folder, and creation of new objects in this folder

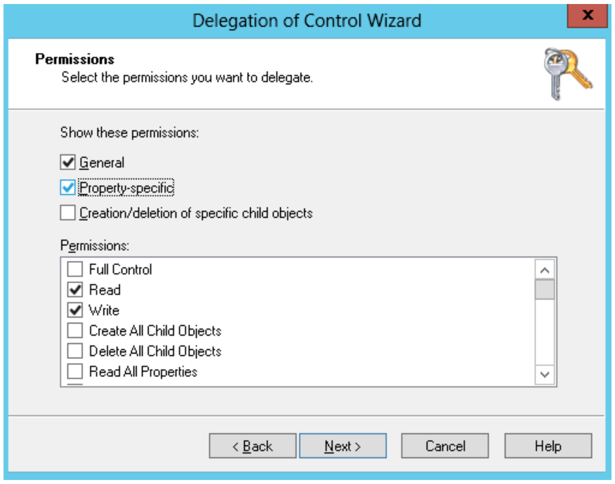
● Only the following objects in the folder:

- ☑ Computer objects
- □ Connection objects
- □ Contact objects
- □ document objects
- □ documentSeries objects
- □ domainRelatedObject objects

☑ Create selected objects in this folder

☑ Delete selected objects in this folder

10. Select **Read** and **Write**, and then choose **Next**.



Permissions
Select the permissions you want to delegate.

Show these permissions:

☑ General
☑ Property-specific
☐ Creation/deletion of specific child objects

Permissions:

☐ Full Control
☑ Read
☑ Write
☐ Create All Child Objects
☐ Delete All Child Objects
☐ Read All Properties

11. Verify the information on the **Completing the Delegation of Control Wizard** page, and click **Finish**.
12. Create a user with a strong password and add that user to the `Connectors` group. The user has sufficient privileges to connect AWS Directory Service to the directory.

Connect Verification

For AD Connector to connect to your on-premises directory, the firewall for your on-premises network must have certain ports open to the CIDRs for both subnets in the VPC. To test if these conditions are met, perform the following steps:

To verify the connection

1. Launch a Windows instance in the VPC and connect to it over RDP. The instance must be a member of your on-premises domain. The remaining steps are performed on this VPC instance.
2. Download and unzip the DirectoryServicePortTest test application. The source code and Visual Studio project files are included so you can modify the test application if desired.

   Note
   This script is not supported on Windows Server 2003 or older operating systems.
3. From a Windows command prompt, run the **DirectoryServicePortTest** test application with the following options:

```
DirectoryServicePortTest.exe -d <domain_name> \
-ip <server_IP_address> \
-tcp "53,88,389" \
-udp "53,88,389"
```

<domain_name>
The fully qualified domain name. This is used to test the forest and domain functional levels. If you exclude the domain name, the functional levels won't be tested.

<server_IP_address>
The IP address of a domain controller in your on-premises domain. The ports will be tested against this IP address. If you exclude the IP address, the ports won't be tested.

This test app determines if the necessary ports are open from the VPC to your domain, and also verifies the minimum forest and domain functional levels.

The output will be similar to the following:

```
Testing forest functional level.
Forest Functional Level = Windows2008R2Forest : PASSED

Testing domain functional level.
Domain Functional Level = Windows2008R2Domain : PASSED

Testing required TCP ports to <server_IP_address>:
Checking TCP port 53: PASSED
Checking TCP port 88: PASSED
Checking TCP port 389: PASSED

Testing required UDP ports to <server_IP_address>:
Checking UDP port 53: PASSED
Checking UDP port 88: PASSED
Checking UDP port 389: PASSED
```

The following is the source code for the **DirectoryServicePortTest** application.

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Net;
using System.Net.Sockets;
using System.Text;
using System.Threading.Tasks;
using System.DirectoryServices.ActiveDirectory;
using System.Threading;
using System.DirectoryServices.AccountManagement;
using System.DirectoryServices;
using System.Security.Authentication;
using System.Security.AccessControl;
using System.Security.Principal;

namespace DirectoryServicePortTest
{
    class Program
    {
        private static List<int> _tcpPorts;
        private static List<int> _udpPorts;

        private static string _domain = "";
        private static IPAddress _ipAddr = null;

        static void Main(string[] args)
        {
            if (ParseArgs(args))
            {
                try
                {
                    if (_domain.Length > 0)
                    {
                        try
                        {
                            TestForestFunctionalLevel();

                            TestDomainFunctionalLevel();
                        }
                        catch (ActiveDirectoryObjectNotFoundException)
                        {
                            Console.WriteLine("The domain {0} could not be
found.\n", _domain);
                        }
                    }

                    if (null != _ipAddr)
                    {
                        if (_tcpPorts.Count > 0)
                        {
                            TestTcpPorts(_tcpPorts);
                        }

                        if (_udpPorts.Count > 0)
                        {
                            TestUdpPorts(_udpPorts);
                        }
                    }
                }
                catch (AuthenticationException ex)
                {
                    Console.WriteLine(ex.Message);
                }
            }
            else
            {
                PrintUsage();
            }

            Console.Write("Press <enter> to continue.");
            Console.ReadLine();
        }

        static void PrintUsage()
        {
            string currentApp =
Path.GetFileName(System.Reflection.Assembly.GetExecutingAssembly().Location);
            Console.WriteLine("Usage: {0} \n-d <domain> \n-ip \"<server
IP address>\" \n[-tcp \"<tcp_port1>,<tcp_port2>,etc\"] \n[-udp
\"<udp_port1>,<udp_port2>,etc\"]", currentApp);
        }

        static bool ParseArgs(string[] args)
        {
            bool fReturn = false;
            string ipAddress = "";

            try
            {
                _tcpPorts = new List<int>();
                _udpPorts = new List<int>();

                for (int i = 0; i < args.Length; i++)
                {
                    string arg = args[i];

                    if ("-tcp" == arg | "/tcp" == arg)
                    {
                        i++;
                        string portList = args[i];
                        _tcpPorts = ParsePortList(portList);
                    }

                    if ("-udp" == arg | "/udp" == arg)
                    {
                        i++;
                        string portList = args[i];
                        _udpPorts = ParsePortList(portList);
                    }

                    if ("-d" == arg | "/d" == arg)
                    {
                        i++;
                        _domain = args[i];
                    }

                    if ("-ip" == arg | "/ip" == arg)
                    {
                        i++;
                        ipAddress = args[i];
                    }
                }
            }
            catch (ArgumentOutOfRangeException)
            {
                return false;
            }

            if (_domain.Length > 0 || ipAddress.Length > 0)
            {
                fReturn = true;
            }

            if (ipAddress.Length > 0)
            {
                _ipAddr = IPAddress.Parse(ipAddress);
            }

            return fReturn;
        }

        static List<int> ParsePortList(string portList)
        {
            List<int> ports = new List<int>();

            char[] separators = {',', ';', ':'};

            string[] portStrings = portList.Split(separators);
            foreach (string portString in portStrings)
            {
                try
                {
                    ports.Add(Convert.ToInt32(portString));
                }
                catch (FormatException)
                {
                }
            }

            return ports;
        }

        static void TestForestFunctionalLevel()
        {
            Console.WriteLine("Testing forest functional level.");

            DirectoryContext dirContext = new
DirectoryContext(DirectoryContextType.Forest, _domain, null, null);
            Forest forestContext = Forest.GetForest(dirContext);

            Console.Write("Forest Functional Level = {0} : ",
forestContext.ForestMode);

            if (forestContext.ForestMode >= ForestMode.Windows2003Forest)
            {
                Console.WriteLine("PASSED");
            }
            else
            {
                Console.WriteLine("FAILED");
            }

            Console.WriteLine();
        }

        static void TestDomainFunctionalLevel()
        {
            Console.WriteLine("Testing domain functional level.");

            DirectoryContext dirContext = new
DirectoryContext(DirectoryContextType.Domain, _domain, null, null);
            Domain domainObject = Domain.GetDomain(dirContext);

            Console.Write("Domain Functional Level = {0} : ",
domainObject.DomainMode);

            if (domainObject.DomainMode >= DomainMode.Windows2003Domain)
            {
                Console.WriteLine("PASSED");
            }
            else
            {
                Console.WriteLine("FAILED");
            }

            Console.WriteLine();
        }

        static List<int> TestTcpPorts(List<int> portList)
        {
            Console.WriteLine("Testing TCP ports to {0}:",
_ipAddr.ToString());
```

```
           List<int> failedPorts = new List<int>();

           foreach (int port in portList)
           {
               Console.Write("Checking TCP port {0}: ", port);

               TcpClient tcpClient = new TcpClient();

               try
               {
                   tcpClient.Connect(_ipAddr, port);

                   tcpClient.Close();
                   Console.WriteLine("PASSED");
               }
               catch (SocketException)
               {
                   failedPorts.Add(port);
                   Console.WriteLine("FAILED");
               }
           }

           Console.WriteLine();

           return failedPorts;
       }

       static List<int> TestUdpPorts(List<int> portList)
       {
           Console.WriteLine("Testing UDP ports to {0}:",
_ipAddr.ToString());

           List<int> failedPorts = new List<int>();

           foreach (int port in portList)
           {
               Console.Write("Checking UDP port {0}: ", port);

               UdpClient udpClient = new UdpClient();

               try
               {
                   udpClient.Connect(_ipAddr, port);
                   udpClient.Close();
                   Console.WriteLine("PASSED");
               }
               catch (SocketException)
               {
                   failedPorts.Add(port);
                   Console.WriteLine("FAILED");
               }
           }

           Console.WriteLine();

           return failedPorts;
       }
   }
```

```
}
```

How to Create an AD Connector

To connect to your on-premises directory with AD Connector, perform the following steps. Before starting this procedure, make sure you have completed the prerequisites identified in AD Connector Prerequisites (p. 47).

To connect with AD Connector

1. In the AWS Directory Service console navigation pane, select **Directories** and choose **Set up Directory**.
2. In the **Connect using AD Connector** area, choose **Create AD Connector**.
3. Provide the following information:

   Connected domain name
   The fully qualified name of your on-premises directory, such as `corp.example.com`.

   Connected NetBIOS name
   The short name of your on-premises directory, such as `CORP`.

   Connector account username
   The user name of a user in the on-premises directory. For more information about this account, see the AD Connector Prerequisites (p. 47).

   Connector account password
   The password for the on-premises user account.

   Confirm password
   Retype the password for the on-premises user account.

   DNS IP address
   The IP address of at least one DNS server in your on-premises directory. These servers must be accessible from each subnet specified in the next section.

   Description
   An optional description for the directory.

   Size
   Select the size of the directory.
4. Provide the following information in the **VPC Details** section and choose **Next Step**.

   VPC
   The VPC for the directory.

   Subnets
   Select the subnets for the directory servers. The two subnets must be in different Availability Zones.
5. Review the directory information and make any necessary changes. When the information is correct, choose **Create AD Connector**.

It takes several minutes for your directory to be connected. When it has been successfully extended, the **Status** value changes to `Active`.

Update Directory Credentials for Your AD Connector

The AD Connector directory credentials represent the account that is used to access your on-premises directory. You can modify these account credentials by performing the following steps.

Note
If single sign-on is enabled for the directory, AWS Directory Service must transfer the service principal name (SPN) from the current service account to the new service account. If the current service account does not have permission to delete the SPN or the new service account does not have permission to add the SPN, you are prompted for the credentials of a directory account that does have permission to perform both actions. These credentials are only used to transfer the SPN and are not stored by the service.

To modify your account credentials for AD Connector

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. Click the directory ID link for your directory.
3. Select the **Connector Account** tab.
4. Enter the new user name and password, and click **Update Directory**.

Update the DNS Address for Your AD Connector

Use the following steps to update the DNS addresses that your AD Connector is pointing to.

Note
If you have an update in progress, you must wait until it is complete before submitting another update.

To update your DNS settings for AD Connector

1. In the AWS Directory Service console navigation pane, choose **Directories**.
2. Choose the name of the directory to be updated.
3. Select the **DNS settings** tab.
4. Type the updated DNS IP addresses and choose **Update directory**.

Multi-Factor Authentication

You can enable multi-factor authentication for your AD Connector directory by performing the following procedure. For more information about using multi-factor authentication with AWS Directory Service, see AD Connector Prerequisites (p. 47).

Note
Multi-factor authentication is not available for Simple AD or Microsoft AD.

To enable multi-factor authentication for your AD Connector

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. Click the directory ID link for your directory.
3. Select the **Multi-Factor Authentication** tab.
4. Enter the following values and click **Update Directory**.

   Enable Multi-Factor Authentication
   Check to enable multi-factor authentication.

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

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

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

Confirm shared secret code
Confirm the shared secret code for your RADIUS endpoints.

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

Server timeout
The amount of time, in seconds, to wait for the RADIUS server to respond. This must be a value between 1 and 20.

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

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

Simple Active Directory

Simple AD is a standalone managed directory that is powered by Samba 4 Active Directory Compatible Server. Simple AD is available in two sizes, small and large. A small Simple AD supports up to 500 users (approximately 2,000 objects including users, groups, and computers). A large Simple AD supports up to 5,000 users (approximately 20,000 objects, including users, groups, and computers).

Simple AD provides a subset of the features offered by Microsoft AD, including the ability to manage user accounts and group memberships, create and apply group policies, securely connect to Amazon EC2 instances, and provide Kerberos-based single sign-on (SSO). However, note that Simple AD does not support features such as trust relationships with other domains, Active Directory Administrative Center, PowerShell support, Active Directory recycle bin, group managed service accounts, and schema extensions for POSIX and Microsoft applications.

Simple AD offers many advantages:

- Simple AD makes it easier to manage Amazon EC2 instances running Linux and Windows and deploy Windows applications in the AWS Cloud.
- Many of the applications and tools that you use today that require Microsoft Active Directory support can be used with Simple AD.
- User accounts in Simple AD allow access to AWS applications such as Amazon WorkSpaces, Amazon WorkDocs, or Amazon WorkMail.
- You can manage AWS resources through IAM role–based access to the AWS Management Console.
- Daily automated snapshots enable point-in-time recovery.

Continue reading the topics in this section to learn how to create your own Simple AD.

Topics

Create a Simple AD Directory

Simple AD creates a fully managed, Samba-based directory in the AWS cloud. When you create a directory with Simple AD, AWS Directory Service creates two directory servers and DNS servers on your behalf. The directory servers are created in different subnets in a VPC; this redundancy helps ensures that your directory remains accessible even if a failure occurs.

Topics

Supported Applications

The following applications have been tested for use with AWS Directory Service Simple AD directories:

- Microsoft Internet Information Services (IIS) on the following platforms:
  - Windows Server 2003 R2
  - Windows Server 2008 R1
  - Windows Server 2008 R2
  - Windows Server 2012
  - Windows Server 2012 R2
- Microsoft SQL Server:
  - SQL Server 2005 R2 (Express, Web, and Standard editions)
  - SQL Server 2008 R2 (Express, Web, and Standard editions)
  - SQL Server 2012 (Express, Web, and Standard editions)
  - SQL Server 2014 (Express, Web, and Standard editions)
- Microsoft SharePoint:
  - SharePoint 2010 Foundation
  - SharePoint 2010 Enterprise
  - SharePoint 2013 Enterprise

Simple AD Prerequisites

To create a Simple AD directory, you need a VPC with the following:

- At least two subnets. Each of the subnets must be in a different Availability Zone.
- The following ports must be open between the two subnets that you deploy your directory into. This is necessary to allow the domain controllers that AWS Directory Service creates for you to communicate with each other.
  - TCP/UDP 53 - DNS
  - TCP/UDP 88 - Kerberos authentication
  - UDP 123 - NTP
  - TCP 135 - RPC
  - UDP 137-138 - Netlogon
  - TCP 139 - Netlogon
  - TCP/UDP 389 - LDAP; note that AWS Directory Service does not support LDAP with SSL (LDAPS) or LDAP signing
  - TCP/UDP 445 - SMB
  - TCP 873 - FRS
  - TCP 3268 - Global Catalog
  - TCP/UDP 1024-65535 - Ephemeral ports for RPC
- The VPC must have default hardware tenancy.

How to Create a Simple AD Directory

To create a new directory, perform the following steps. Before starting this procedure, make sure you have completed the prerequisites identified in Simple AD Prerequisites (p. 61).

To create a Simple AD directory

1. In the AWS Directory Service console navigation pane, select **Directories** and choose **Set up Directory**.
2. Choose **Create Simple AD**
3. Provide the following information:

   Organization name
   A unique organization name for your directory that will be used to register client devices.

   This field is only available if you are creating your directory as part of launching Amazon WorkSpaces.

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

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

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

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

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

   Confirm password
   Retype the administrator password.

   Description
   An optional description for the directory.

   Directory Size
   Select the size of the directory.
4. Provide the following information in the **VPC Details** section and choose **Next Step**.

   VPC
   The VPC for the directory.

   Subnets
   Select the subnets for the directory servers. The two subnets must be in different Availability Zones.
5. Review the directory information and make any necessary changes. When the information is correct, choose **Create Simple AD**.

It takes several minutes for the directory to be created. When it has been successfully created, the **Status** value changes to `Active`.

What Gets Created

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

- Sets up a Samba-based directory within the VPC.
- Creates a directory administrator account with the user name `Administrator` and the specified password. You use this account to manage your directory.

  Important
  Be sure to save this password. AWS Directory Service does not store this password and it cannot be retrieved or reset.
- Creates a security group for the directory controllers.
- Creates an account with the name `AWSAdminD-`*`xxxxxxxx`* that has domain admin privileges. This account is used by AWS Directory Service to perform automated operations for directory maintenance operations, such as taking directory snapshots and FSMO role transfers. The credentials for this account are securely stored by AWS Directory Service.

Tutorial: Create a Simple AD Directory

The following tutorial walks you through all of the steps necessary to set up an AWS Directory Service Simple AD directory. It is intended to get you started with AWS Directory Service quickly and easily, but is not intended to be used in a large-scale production environment.

Topics

Prerequisites

This tutorial assumes the following:

- You have an active AWS account. For more information about managing accounts and your directory, see Managing Your Directory (p. 66).
- Your account has not reached its limit of VPCs for the region in which you want to use AWS Directory Service. For more information about Amazon VPC, see What is Amazon VPC? and Subnets in Your VPC in the *Amazon VPC User Guide*.
- You do not have an existing VPC in the region with a CIDR of 10.0.0.0/16.

Step 1: Create and Configure Your VPC

The following sections demonstrate how to create and configure a VPC for use with AWS Directory Service.

Topics

Create a New VPC

This tutorial uses one of the VPC creation wizards to create the following:

- The VPC
- One of the subnets
- An Internet gateway

To create your VPC using the VPC wizard

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, click **VPC Dashboard**. If you do not already have any VPC resources, locate the **Your Virtual Private Cloud** area of the dashboard and click **Get started creating a VPC**. Otherwise, click **Start VPC Wizard**.
3. Select the second option, **VPC with a Single Public Subnet**, and then click **Select**.
4. Enter the following information into the wizard and click **Create VPC**.

   IP CIDR block
   `10.0.0.0/16`

   VPC name
   `ADS VPC`

   Public subnet
   `10.0.0.0/24`

   Availability Zone
   No Preference

   Subnet name
   `ADS Subnet 1`

   Enable DNS hostnames
   Leave default selection

   Hardware tenancy
   Default
5. It takes several minutes for the VPC to be created. After the VPC is created, proceed to the following section to add a second subnet.

Add a Second Subnet

AWS Directory Service requires two subnets in your VPC, and each subnet must be in a different Availability Zone. The VPC wizard only creates one subnet, so you must manually create the second subnet, and specify a different Availability Zone than the first subnet. Create the second subnet by performing the following steps.

To create a subnet

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, select **Subnets**, select the subnet with the name `ADS Subnet 1`, and select the **Summary** tab at the bottom of the page. Make a note of the Availability Zone of this subnet.
3. Click **Create Subnet** and enter the following information in the **Create Subnet** dialog box and click **Yes, Create**.

   Name tag
   `ADS Subnet 2`

VPC
Select your VPC. This is the VPC with the name `ADS VPC`.

Availability Zone
Select any Availability Zone other than the one noted in step 2. The two subnets used by AWS Directory Service must reside in different Availability Zones.

CIDR Block
`10.0.1.0/24`

Step 2: Create Your Simple AD Directory

To create your AWS Directory Service Simple AD directory, perform the following steps. For more information about this process, see Create a Simple AD Directory (p. 60).

To create your Simple AD directory

1. Open the AWS Directory Service console for your desired region.
2. In the navigation pane, select **Directories**, click **Set up Directory**, then select **Create Simple AD**.
3. Enter the following fields.

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

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

   Administrator password
   The password for the directory administrator. The directory creation process creates an administrator account with the username `Administrator` and this password.

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

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

   Confirm password
   Re-enter the administrator password.

   Description
   An optional description for the directory.

   Directory Size
   Select the size of the directory.
4. Enter the following fields in the **VPC Details** section and click **Next Step**.

   VPC
   The VPC for the directory.

   Subnets
   Select the subnets for the directory servers. The two subnets must be in different Availability Zones.
5. Review the directory information and make any necessary changes. When the information is correct, click **Create Simple AD**.

   It takes several minutes for the directory to be created. When it has been successfully created, the **Status** value changes to `Active`.

Managing Your Directory

You use the AWS Directory Service management console to perform certain directory-related actions, such as changing directory information or deleting an existing directory. After a directory is created, most administrative functions are performed with directory management tools, such as the Active Directory Administration Tools.

> **Note**
> Simple AD directories do not support Active Directory Web Services. Because of this, tools that rely on Active Directory Web Services, such as the Active Directory Administrative Center, do not work with your Simple AD directory.

Topics

View Directory Information

You can view basic information about a directory within the directories page, or more detailed information in the directory details page.

Basic Information

To view basic information about a directory, perform the following steps:

To view basic directory information

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. Click the arrow button next to your directory. Basic information about the directory is displayed below the directory entry in the list.

For more information about the **Status** field, see Directory Status (p. 131).

Detailed Information

To view more detailed information about a directory, perform the following steps:

To view detailed directory information

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. Click the directory ID link for your directory. Information about the directory is displayed in the **Directory Details** section.

For more information about the **Status** field, see Directory Status (p. 131).

Get Notified of Directory Status Updates Using Amazon SNS

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

How It Works

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

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

To enable SNS messaging for your directory

1. Sign in to the AWS Management Console and open the AWS Directory Service console at https://console.aws.amazon.com/directoryservice/.
2. On the **Directories** page, choose your directory ID.
3. Choose the **Monitoring** tab and then **Create Notification**.
4. Choose **Create a new notification**. Alternatively, if you already have an existing SNS topic, you can choose **Associate with existing SNS topic** to send status messages from this directory to that topic.

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

   Note
   If you are logged in using an IAM user or role that has only the DirectoryServiceFullAccess managed policy, your topic name must start with

"DirectoryMonitoring". If you'd like to further customize your topic name you'll need additional privileges for SNS.

7. Choose **Add**.

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

To remove directory status messages from a topic

1. Sign in to the AWS Management Console and open the AWS Directory Service console at https://console.aws.amazon.com/directoryservice/.
2. On the **Directories** page, choose your directory ID.
3. Choose the **Monitoring** tab and then choose an **SNS topic name**.
4. Choose **Remove**.

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

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

Delete Your Directory

When a Simple AD or AWS Directory Service for Microsoft Active Directory (Enterprise Edition) directory is deleted, all of the directory data and snapshots are deleted and cannot be recovered. After the directory is deleted, all instances that are joined to the directory remain intact. You cannot, however, use your directory credentials to log in to these instances. You need to log in to these instances with a user account that is local to the instance.

When an AD Connector directory is deleted, your on-premises directory remains intact. All instances that are joined to the directory also remain intact and remain joined to your on-premises directory. You can still use your directory credentials to log in to these instances.

To delete a directory

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. Ensure that no AWS applications are enabled for the directory.

    1. Click the directory ID link for your directory.
    2. Select the **Apps & Services** tab. In the Apps & Services section, you see which AWS applications are enabled for your directory.

        - To disable Amazon WorkSpaces, you must deregister the service from the directory in the Amazon WorkSpaces console. For more information, see Deregistering From a Directory in the *Amazon WorkSpaces Administration Guide*.

- To disable Amazon WAM, you must remove all application assignments in the Amazon WAM console. For more information, see Removing All Application Assignments in the *Amazon WAM Administration Guide*.
- To disable Amazon WorkDocs, you must delete the Amazon WorkDocs site in the Amazon WorkDocs console. For more information, see Delete a Site in the *Amazon WorkDocs Administration Guide*.
- To disable Amazon WorkMail, you must remove the Amazon WorkMail organization in the Amazon WorkMail console. For more information, see Remove an Organization in the *Amazon WorkMail Administrator Guide*.
- To disable AWS console access, see Disabling AWS Management Console Access (p. 79).

3. Go back to the AWS Directory Service console navigation pane and again select **Directories**.
4. Select only the directory to be deleted and click **Delete**. It takes several minutes for the directory to be deleted. When the directory has been deleted, it is removed from your directory list.

Note
Before deleting a directory that is associated with an Amazon RDS database, you must first remove that database from the directory.

Snapshots (Simple AD and Microsoft AD)

AWS Directory Service provides the ability to take manual snapshots of data for a Simple AD or AWS Directory Service for Microsoft Active Directory (Enterprise Edition) directory. These snapshots can be used to perform a point-in-time restore for your directory.

Note
You cannot take snapshots of AD Connector directories.

Topics

Creating a Snapshot of Your Directory

A snapshot can be used to restore your directory to what it was at the point in time that the snapshot was taken. To create a manual snapshot of your directory, perform the following steps.

Note
You are limited to 5 manual snapshots for each directory. If you have already reached this limit, you must delete one of your existing manual snapshots before you can create another.

To create a manual snapshot

1. In the AWS Directory Service console navigation pane, select **Snapshots**.
2. Choose **Create Snapshot**.
3. In the **Create directory snapshot** dialog box, select the directory to take a snapshot of. You can also apply a description to the snapshot, if desired. When ready, choose **Create Snapshot**.

Depending on the size of your directory, it may take several minutes to create the snapshot. When the snapshot is ready, the **Status** value changes to `Completed`.

Restoring Your Directory from a Snapshot

Restoring a directory from a snapshot is equivalent to moving the directory back in time.

> **Warning**
> Before you restore a directory from a snapshot, it is important you understand that all of the DCs and DNS servers associated with the directory will be offline until the restore operation has been completed.

To restore your directory from a snapshot, perform the following steps.

To restore a directory from a snapshot

1. In the AWS Directory Service console navigation pane, select **Snapshots**.
2. Select the snapshot to restore from.
3. Choose **Restore**, review the information in the dialog box, and choose **Restore**.

It takes several minutes for the directory to be restored. When it has been successfully restored, the **Status** value of the directory changes to `Active`. Any changes made to the directory after the snapshot date are overwritten.

Deleting a Snapshot

To delete a snapshot

1. In the AWS Directory Service console navigation pane, select **Snapshots**.
2. Select the snapshot to delete.
3. Choose **Delete**, verify that you want to delete the snapshot, and choose **Delete**.

Use Your Directory

Topics

Single Sign-On

AWS Directory Service provides the ability to allow users in your directory to access certain AWS apps and services from a computer joined to the directory without having to enter their credentials separately.

Users may need to modify their web browser settings to enable single sign-on. For more information, see Single Sign-On for IE and Chrome (p. 72) and Single Sign-On for Firefox (p. 77).

> **Note**
> Single sign-on only works when used on a computer that is joined to the AWS Directory Service directory. It cannot be used on computers that are not joined to the directory.

To enable or disable single sign-on for a directory

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. Choose the directory ID link for your directory.
3. Select the **Apps & Services** tab.
4. If single sign-on is currently disabled, perform the following steps to enable it.

   a. Click **Enable**.

   b. In the **Enable Single Sign-On for this Directory** dialog box, choose **Enable**. Single sign-on is enabled for the directory.

      If the directory is an AD Connector directory and the AD Connector service account does not have permission to add a service principle name, you are prompted for the username and password for a directory user that has this permission. These credentials are only used to enable single sign-on and are not stored by the service. The AD Connector service account is not changed.

5. If single sign-on is currently enabled, perform the following steps to disable it.

   a. Click **Disable**.

   b. In the **Disable Single Sign-On for this Directory** dialog box, choose **Disable**. Single sign-on is disabled for the directory.

      If the directory is an AD Connector directory and the AD Connector service account does not have permission to remove a service principle name, you are prompted for the username and password for a directory user that has this permission. These credentials are only used to disable single sign-on and are not stored by the service. The AD Connector service account is not changed.

Before you enable single sign-on for your directory, you need to take additional steps to enable your users' web browsers to support single sign-on.

> **Note**
> Single sign-on only works when used on a computer that is joined to the AWS Directory Service directory. It cannot be used on computers that are not joined to the directory.

Topics

- Single Sign-On for IE and Chrome (p. 72)
- Single Sign-On for Firefox (p. 77)

Single Sign-On for IE and Chrome

To allow Microsoft's Internet Explorer (IE) and Google's Chrome browsers to support single sign-on, the following tasks must be performed on the client computer:

- Add your access URL (e.g., https://*<alias>*.awsapps.com) to the list of approved sites for single sign-on.
- Enable active scripting (JavaScript).
- Allow automatic logon.
- Enable integrated authentication.

You or your users can perform these tasks manually, or you can change these settings using Group Policy settings.

Topics

- Manual Update for Single Sign-On on Windows (p. 72)
- Manual Update for Single Sign-On on OS X (p. 75)
- Group Policy Settings for Single Sign-On (p. 75)

Manual Update for Single Sign-On on Windows

To manually enable single sign-on on a Windows computer, perform the following steps on the client computer. Some of these settings may already be set correctly.

To manually enable single sign-on for Internet Explorer and Chrome on Windows

1. To open the **Internet Properties** dialog box, choose the **Start** menu, type `Internet Options` in the search box, and choose **Internet Options**.
2. Add your access URL to the list of approved sites for single sign-on by performing the following steps:

a. In the **Internet Properties** dialog box, select the **Security** tab.

b. Select **Local intranet** and choose **Sites**.

c. In the **Local intranet** dialog box, choose **Advanced**.

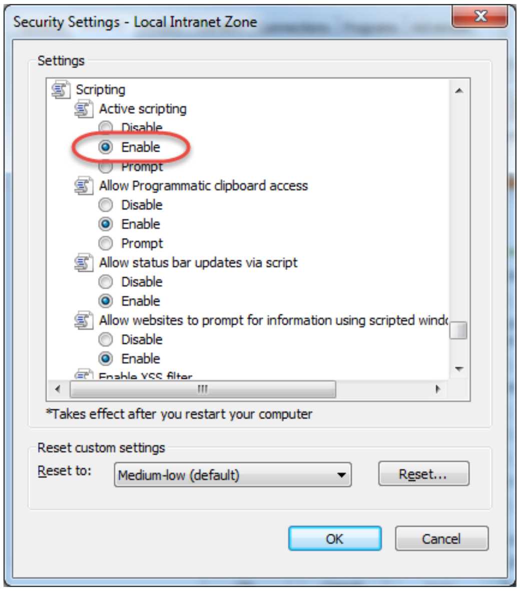
d. Add your access URL to the list of websites and choose **Close**.

e. In the **Local intranet** dialog box, choose **OK**.

3. To enable active scripting, perform the following steps:

   a. In the **Security** tab of the **Internet Properties** dialog box, choose **Custom level**.

   b. In the **Security Settings - Local Intranet Zone** dialog box, scroll down to **Scripting** and select **Enable** under **Active scripting**.



Security Settings - Local Intranet Zone

Settings

Scripting
Active scripting
○ Disable
● Enable
○ Prompt
Allow Programmatic clipboard access
○ Disable
● Enable
○ Prompt
Allow status bar updates via script
○ Disable
● Enable
Allow websites to prompt for information using scripted windo
○ Disable
● Enable
Enable XSS filter

*Takes effect after you restart your computer

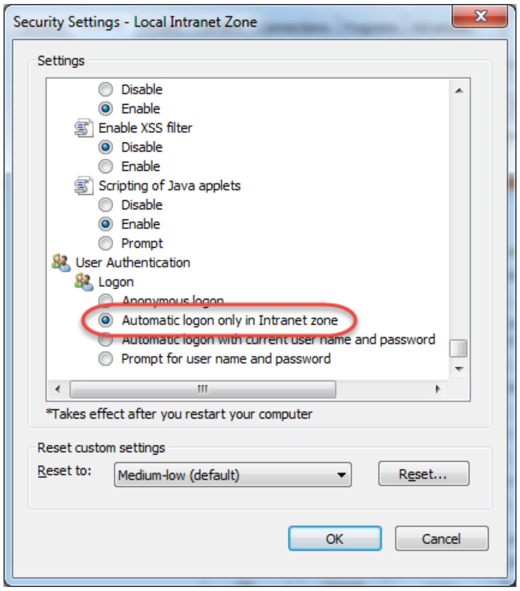
Reset custom settings
Reset to: Medium-low (default)

   c. In the **Security Settings - Local Intranet Zone** dialog box, choose **OK**.

4. To enable automatic logon, perform the following steps:

   a. In the **Security** tab of the **Internet Properties** dialog box, choose **Custom level**.

   b. In the **Security Settings - Local Intranet Zone** dialog box, scroll down to **User Authentication** and select **Automatic logon only in Intranet zone** under **Logon**.



Security Settings - Local Intranet Zone

Settings

○ Disable
● Enable
Enable XSS filter
● Disable
○ Enable
Scripting of Java applets
○ Disable
● Enable
○ Prompt
User Authentication
Logon
○ Anonymous logon
● Automatic logon only in Intranet zone
○ Automatic logon with current user name and password
○ Prompt for user name and password

*Takes effect after you restart your computer

Reset custom settings
Reset to: Medium-low (default)

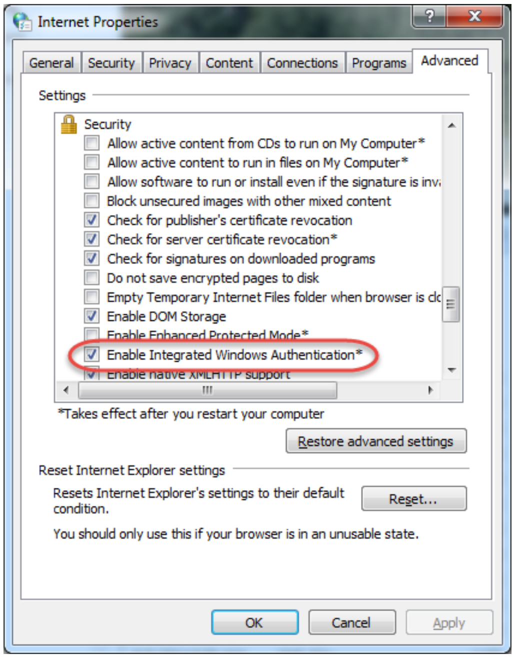
c. In the **Security Settings - Local Intranet Zone** dialog box, choose **OK**.

d. In the **Security Settings - Local Intranet Zone** dialog box, choose **OK**.

5. To enable integrated authentication, perform the following steps:

   a. In the **Internet Properties** dialog box, select the **Advanced** tab.

   b. Scroll down to **Security** and select **Enable Integrated Windows Authentication**.



Internet Properties

Settings

Security
□ Allow active content from CDs to run on My Computer*
□ Allow active content to run in files on My Computer*
□ Allow software to run or install even if the signature is inv
□ Block unsecured images with other mixed content
☑ Check for publisher's certificate revocation
☑ Check for server certificate revocation*
☑ Check for signatures on downloaded programs
□ Do not save encrypted pages to disk
□ Empty Temporary Internet Files folder when browser is clo
☑ Enable DOM Storage
□ Enable Enhanced Protected Mode*
☑ Enable Integrated Windows Authentication*
☑ Enable native XMLHTTP support

*Takes effect after you restart your computer

Reset Internet Explorer settings

Resets Internet Explorer's settings to their default condition.

You should only use this if your browser is in an unusable state.

   c. In the **Internet Properties** dialog box, choose **OK**.

6. Close and re-open your browser to have these changes take effect.

Manual Update for Single Sign-On on OS X

To manually enable single sign-on for Chrome on OS X, perform the following steps on the client computer. You will need administrator rights on your computer to complete these steps.

To manually enable single sign-on for Chrome on OS X

1. Add your access URL to the AuthServerWhitelist policy by running the following command:

   ```
   defaults write com.google.Chrome AuthServerWhitelist
    "https://<alias>.awsapps.com"
   ```

2. Open **System Preferences**, go to the **Profiles** panel, and delete the `Chrome Kerberos Configuration` profile.
3. Restart Chrome and open `chrome://policy` in Chrome to confirm that the new settings are in place.

Group Policy Settings for Single Sign-On

The domain administrator can implement Group Policy settings to make the single sign-on changes on client computers that are joined to the domain.

> **Note**
> If you manage the Chrome web browsers on the computers in your domain with Chrome policies, you must add your access URL to the AuthServerWhitelist policy. For more information about setting Chrome policies, go to Policy Settings in Chrome.

To enable single sign-on for Internet Explorer and Chrome using Group Policy settings

1. Create a new Group Policy object by performing the following steps:

   a. Open the Group Policy Management tool, navigate to your domain and select **Group Policy Objects**.
   b. From the main menu, choose **Action** and select **New**.
   c. In the **New GPO** dialog box, enter a descriptive name for the Group Policy object, such as `SSO Policy`, and leave **Source Starter GPO** set to **(none)**. Click **OK**.

2. Add the access URL to the list of approved sites for single sign-on by performing the following steps:

   a. In the Group Policy Management tool, navigate to your domain, select **Group Policy Objects**, open the context (right-click) menu for your SSO policy, and choose **Edit**.
   b. In the policy tree, navigate to **User Configuration** > **Preferences** > **Windows Settings**.
   c. In the **Windows Settings** list, open the context (right-click) menu for **Registry** and choose **New registry item**.
   d. In the **New Registry Properties** dialog box, enter the following settings and choose **OK**:

      Action
      `Update`

      Hive
      `HKEY_CURRENT_USER`

      Path
      `Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\Domains\awsapps.com\<alias>`

      The value for *<alias>* is derived from your access URL. If your access URL is `https://examplecorp.awsapps.com`, the alias is `examplecorp`, and the registry key will be `Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\Domains\awsapps.com\examplecorp`.

Value name
`https`

Value type
`REG_DWORD`

Value data
`1`

3. To enable active scripting, perform the following steps:

   a. In the Group Policy Management tool, navigate to your domain, select **Group Policy Objects**, open the context (right-click) menu for your SSO policy, and choose **Edit**.

   b. In the policy tree, navigate to **Computer Configuration** > **Policies** > **Administrative Templates** > **Windows Components** > **Internet Explorer** > **Internet Control Panel** > **Security Page** > **Intranet Zone**.

   c. In the **Intranet Zone** list, open the context (right-click) menu for **Allow active scripting** and choose **Edit**.

   d. In the **Allow active scripting** dialog box, enter the following settings and choose **OK**:

      - Select the **Enabled** radio button.
      - Under **Options** set **Allow active scripting** to **Enable**.

4. To enable automatic logon, perform the following steps:

   a. In the Group Policy Management tool, navigate to your domain, select Group Policy Objects, open the context (right-click) menu for your SSO policy, and choose **Edit**.

   b. In the policy tree, navigate to **Computer Configuration** > **Policies** > **Administrative Templates** > **Windows Components** > **Internet Explorer** > **Internet Control Panel** > **Security Page** > **Intranet Zone**.

   c. In the **Intranet Zone** list, open the context (right-click) menu for **Logon options** and choose **Edit**.

   d. In the **Logon options** dialog box, enter the following settings and choose **OK**:

      - Select the **Enabled** radio button.
      - Under **Options** set **Logon options** to **Automatic logon only in Intranet zone**.

5. To enable integrated authentication, perform the following steps:

   a. In the Group Policy Management tool, navigate to your domain, select **Group Policy Objects**, open the context (right-click) menu for your SSO policy, and choose **Edit**.

   b. In the policy tree, navigate to **User Configuration** > **Preferences** > **Windows Settings**.

   c. In the **Windows Settings** list, open the context (right-click) menu for **Registry** and choose **New registry item**.

   d. In the **New Registry Properties** dialog box, enter the following settings and choose **OK**:

      Action
      `Update`

      Hive
      `HKEY_CURRENT_USER`

      Path
      `Software\Microsoft\Windows\CurrentVersion\Internet Settings`

      Value name
      `EnableNegotiate`

      Value type
      `REG_DWORD`

Value data

`1`

6. Close the **Group Policy Management Editor** window if it is still open.
7. Assign the new policy to your domain by following these steps:

   a. In the Group Policy Management tree, open the context (right-click) menu for your domain and choose **Link an Existing GPO**.

   b. In the **Group Policy Objects** list, select your SSO policy and choose **OK**.

These changes will take effect after the next Group Policy update on the client, or the next time the user logs in.

Single Sign-On for Firefox

To allow Mozilla's Firefox browser to support single sign-on, add your access URL (e.g., https://*<alias>*.awsapps.com) to the list of approved sites for single sign-on. This can be done manually, or automated with a script.

Topics

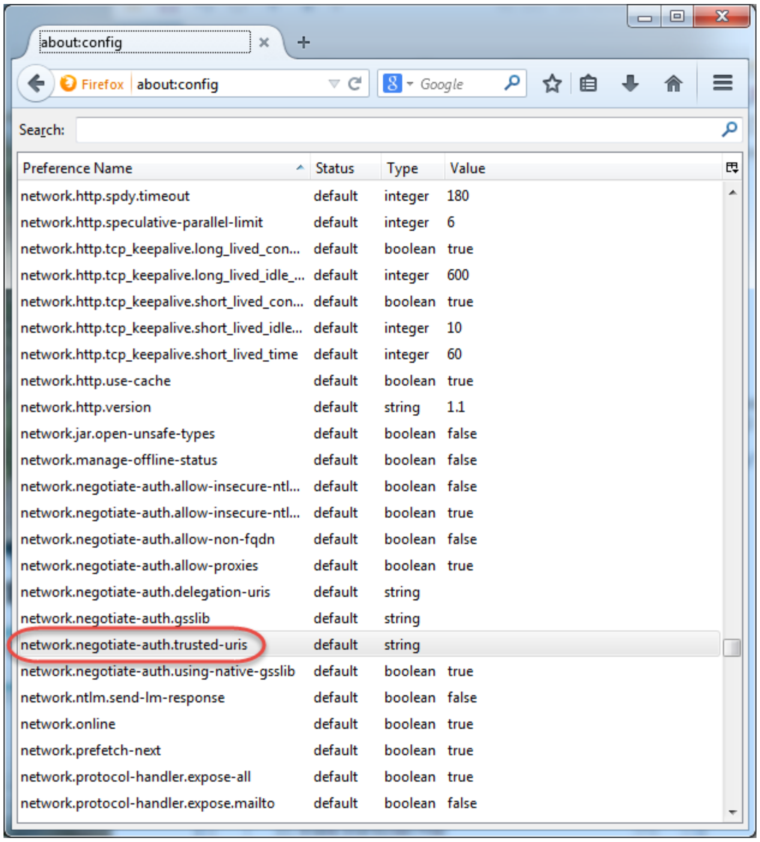
- Manual Update for Single Sign-On (p. 77)
- Automatic Update for Single Sign-On (p. 78)

Manual Update for Single Sign-On

To manually add your access URL to the list of approved sites in Firefox, perform the following steps on the client computer.

To manually add your access URL to the list of approved sites in Firefox

1. Open Firefox and open the `about:config` page.
2. Open the `network.negotiate-auth.trusted-uris` preference and add your access URL to the list of sites. Use a comma (,) to separate multiple entries.



Preference Name	Status	Type	Value
network.http.spdy.timeout	default	integer	180
network.http.speculative-parallel-limit	default	integer	6
network.http.tcp_keepalive.long_lived_con...	default	boolean	true
network.http.tcp_keepalive.long_lived_idle_...	default	integer	600
network.http.tcp_keepalive.short_lived_con...	default	boolean	true
network.http.tcp_keepalive.short_lived_idle...	default	integer	10
network.http.tcp_keepalive.short_lived_time	default	integer	60
network.http.use-cache	default	boolean	true
network.http.version	default	string	1.1
network.jar.open-unsafe-types	default	boolean	false
network.manage-offline-status	default	boolean	false
network.negotiate-auth.allow-insecure-ntl...	default	boolean	false
network.negotiate-auth.allow-insecure-ntl...	default	boolean	true
network.negotiate-auth.allow-non-fqdn	default	boolean	false
network.negotiate-auth.allow-proxies	default	boolean	true
network.negotiate-auth.delegation-uris	default	string	
network.negotiate-auth.gsslib	default	string	
network.negotiate-auth.trusted-uris	default	string	
network.negotiate-auth.using-native-gsslib	default	boolean	true
network.ntlm.send-lm-response	default	boolean	false
network.online	default	boolean	true
network.prefetch-next	default	boolean	true
network.protocol-handler.expose-all	default	boolean	true
network.protocol-handler.expose.mailto	default	boolean	false

Automatic Update for Single Sign-On

As a domain administrator, you can use a script to add your access URL to the Firefox `network.negotiate-auth.trusted-uris` user preference on all computers on your network. For more information, go to https://support.mozilla.org/en-US/questions/939037.

Managing Console Access for AWS Directory Service

AWS Directory Service allows you to grant members of your directory access to the AWS Management Console. By default, your directory members do not have access to any AWS resources. You assign IAM roles to your directory members to give them access to the various AWS services and resources. The IAM role defines the services, resources, and level of access that your directory members have.

Before you can grant console access to your directory members, your directory must have an access URL. For more information about how to view directory details and get your access URL, see View Directory Information (p. 66). For more information about how to create an access URL, see Creating an Access URL (p. 100)

For more information about how to create and assign IAM roles to your directory members, see Managing IAM Roles and Policies (p. 79).

Topics

Enabling AWS Management Console Access

By default, console access is not enabled for any directory. To enable console access for your directory users and groups, perform the following steps:

To enable console access

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. Click the directory ID link for your directory.
3. In the **Directory Details** page, select the **Apps & Services** tab.
4. In the **Services** area, click the **Manage Access** link for **AWS Management Console**.
5. In the **Enable AWS Management Console** dialog box, click **Enable Application**. Console access is now enabled for your directory.

After the IAM roles have been assigned to your directory members, they can access the console at the following URL:

```
https://<alias>.awsapps.com/console/
```

For example, if your directory's access URL is `example-corp.awsapps.com`, the URL to access the console is `https://example-corp.awsapps.com/console/`.

> **Note**
> Access for users in nested groups within your directory are not supported. Members of the parent group will have console access, but members of child groups will not.

Disabling AWS Management Console Access

To disable console access for your directory users and groups, perform the following steps:

To disable console access

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. Click the directory ID link for your directory.
3. In the **Directory Details** page, select the **Apps & Services** tab.
4. In the **Services** area, click the **Manage Access** link for **AWS Management Console**.
5. If any IAM roles have been assigned to users or groups in the directory, the **Disable Access** button in the **Manage access to AWS Resources** dialog box is disabled. In this case, you need to click **Continue** and remove all IAM role assignments for the directory before proceeding. For more information, see Removing a Role (p. 82).

   After all IAM role assignments have been removed, repeat the steps above. When the **Manage access to AWS Resources** dialog box is displayed, click **Disable Access**.

Managing IAM Roles and Policies

AWS Directory Service provides the ability to give your directory users and groups access to AWS services and resources, such as access to the Amazon EC2 console. Similar to granting IAM users access to manage directories as described in Identity-Based Policies (IAM Policies) (p. 121), in order for users in your directory to have access to other AWS resources, such as Amazon EC2 you must assign IAM roles and policies to those users and groups. For more information, see IAM Roles in the *IAM User Guide*.

Topics

Editing the Trust Relationship for an Existing Role

You can assign your existing IAM roles to your AWS Directory Service users and groups. To do this, however, the role must have a trust relationship with AWS Directory Service. When you use AWS Directory Service to create a role using the procedure in Creating a New Role (p. 80), this trust relationship is automatically set. You only need to establish this trust relationship for IAM roles that are not created by AWS Directory Service.

To establish a trust relationship for an existing role to AWS Directory Service

1. In the navigation pane of the IAM console, click **Roles**.

   The console displays the roles for your account.
2. Click the name of the role that you want to modify, and open the **Trust Relationships** section in the details page.
3. Click **Edit Trust Relationship**.
4. Enter the following for the **Policy Document** field and click **Update Trust Policy**.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ds.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Creating a New Role

In addition, AWS Directory Service provides a set of predefined templates for common access needs. You can use these templates to create roles for your AWS Directory Service users. During this process, AWS Directory Service creates an IAM role in your account on your behalf. Because AWS

Directory Service is using the IAM service to create the role on your behalf, you must provide explicit permission for this action in step 9.

> **Note**
> The user performing this task must have permission to perform the following IAM actions. For more information, see Identity-Based Policies (IAM Policies) (p. 121).
>
> - iam:PassRole
> - iam:GetRole
> - iam:CreateRole
> - iam:PutRolePolicy

To create a new role from an AWS Directory Service template

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. Click the directory ID link for your directory.
3. In the **Directory Details** page, select the **Apps & Services** tab.
4. In the **Apps & Services** area, click the **Manage Access** link for **AWS Management Console**.
5. In the **Manage Access to AWS Resource** dialog box, click **Continue**.
6. In the **AWS Management Console Access** page, click **New Role**.
7. In the **Select Role Type** page, click **Create New Role**.
8. In the **Select Role Template** page, select the template to create the role from and click **Next Step**. For more information about the templates provided by AWS Directory Service, see Using IAM Managed Policies with AWS Directory Service (p. 82).
9. Review the information and click **Allow** to allow AWS Directory Service to create the IAM role on your behalf. When the role has been created, you will be taken to step 9 of Assigning a Role (p. 81) to select the users to apply the role to.

Assigning a Role

You can assign an existing IAM role to an AWS Directory Service user or group. The role must have a trust relationship with AWS Directory Service. For more information, see Editing the Trust Relationship for an Existing Role (p. 80).

To assign a role to an AWS Directory Service user or group

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. Click the directory ID link for your directory.
3. In the **Directory Details** page, select the **Apps & Services** tab.
4. In the **Apps & Services** area, click the **Manage Access** link for **AWS Management Console**.
5. In the **Manage Access to AWS Resource** dialog box, click **Continue**.
6. In the **AWS Management Console Access** page, click **New Role**.
7. In the **Select Role Type** page, click **Use Existing Role**.
8. In the **Select Existing Role** page, select the role to add, and click **Next Step**.
9. In the **Select Users/Groups** page, select the users and groups to apply the role to. You can search for the user or group by typing all or part of the name in the text box. When enough context has been entered, a list of possible matches is displayed. Select the desired user or group and the user or group will is added to the list. Click **Next Step** when the list of users and groups to apply the role to is complete.
10. Review the information and click **Create Role Assignment** to apply the selected role to the selected AWS Directory Service users and groups.

Viewing Role Details

To view the details for a role, perform the following steps:

To view details for a role

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. Click the directory ID link for your directory.
3. In the **Directory Details** page, select the **Apps & Services** tab.
4. In the **Apps & Services** area, click the **Manage Access** link for **AWS Management Console**.
5. In the **Manage Access to AWS Resource** dialog box, click **Continue**.
6. In the **AWS Console Access** page, click the role. The **Role Detail** page is displayed.

Removing a Role

To remove a role from all AWS Directory Service users or groups, perform the following steps.

To remove a role

1. In the AWS Directory Service console navigation pane, select **Directories**.
2. Click the directory ID link for your directory.
3. In the **Directory Details** page, select the **Apps & Services** tab.
4. In the **Apps & Services** area, click the **Manage Access** link for **AWS Management Console**.
5. In the **Manage Access to AWS Resource** dialog box, click **Continue**.
6. In the **AWS Console Access** page, select the role to remove, and click **Remove Role**.
7. In the **Remove Role Access** dialog box, verify that you want to remove the role and click **Remove**. The role is removed from all users and groups that the role is assigned to, but the role is not removed from your account.

Removing a Role from a User or Group

To remove a role from specific AWS Directory Service users or groups, perform the following steps.

To remove a role from a user or group

1. View the details for the role to remove as shown in Viewing Role Details (p. 82).
2. In the **Role Detail** page, in the **Assigned Users and Groups** section, select the users or groups to remove the role from and click **Remove**.
3. In the **Remove Role Access** dialog box, verify that you want to remove the role from the selected users and/or groups, and click **Remove**. The role is removed from the specified users and groups, but the role is not removed from your account.

Using IAM Managed Policies with AWS Directory Service

AWS Directory Service provides the following managed policies that allow you to quickly and easily use predefined IAM policies for use with AWS Directory Service. For more information, see Managed Policies in the *IAM User Guide*.

Topics

Read Only Access

This IAM policy provides an AWS Directory Service user or group with read-only access to the following AWS services and resources.

- Auto Scaling
- Elastic Load Balancing
- AWS CloudFormation
- Amazon CloudFront
- AWS CloudTrail
- Amazon CloudWatch
- AWS Direct Connect
- Amazon DynamoDB
- Amazon Elastic Compute Cloud
- Amazon ElastiCache
- AWS Elastic Beanstalk
- Amazon Elastic Transcoder
- AWS Identity and Access Management
- Amazon Kinesis
- AWS OpsWorks
- Amazon Route 53
- Amazon Redshift
- Amazon Relational Database Service
- Amazon Simple Storage Service
- Amazon SimpleDB
- Amazon Simple Email Service

- Amazon Simple Notification Service
- Amazon Simple Queue Service
- AWS Storage Gateway
- AWS Trusted Advisor

The following lists the contents of the policy document.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "acm:DescribeCertificate",
        "acm:GetCertificate",
        "acm:ListCertificates",
        "acm:ListTagsForCertificate",
        "apigateway:GET",
        "application-autoscaling:Describe*",
        "appstream:Get*",
        "autoscaling:Describe*",
        "cloudformation:Describe*",
        "cloudformation:Get*",
        "cloudformation:List*",
        "cloudfront:Get*",
        "cloudfront:List*",
        "cloudsearch:Describe*",
        "cloudsearch:List*",
        "cloudtrail:DescribeTrails",
        "cloudtrail:GetTrailStatus",
        "cloudtrail:LookupEvents",
        "cloudtrail:ListTags",
        "cloudtrail:ListPublicKeys",
        "cloudwatch:Describe*",
        "cloudwatch:Get*",
        "cloudwatch:List*",
        "codecommit:BatchGetRepositories",
        "codecommit:Get*",
        "codecommit:GitPull",
        "codecommit:List*",
        "codedeploy:Batch*",
        "codedeploy:Get*",
        "codedeploy:List*",
        "config:Deliver*",
        "config:Describe*",
        "config:Get*",
        "config:List*",
        "datapipeline:DescribeObjects",
        "datapipeline:DescribePipelines",
        "datapipeline:EvaluateExpression",
        "datapipeline:GetAccountLimits",
        "datapipeline:GetPipelineDefinition",
        "datapipeline:ListPipelines",
        "datapipeline:QueryObjects",
        "datapipeline:ValidatePipelineDefinition",
        "directconnect:Describe*",
        "dms:Describe*",
        "dms:List*",
        "ds:Check*",
        "ds:Describe*",
        "ds:Get*",
        "ds:List*",
        "ds:Verify*",
        "dynamodb:BatchGetItem",
        "dynamodb:DescribeLimits",
        "dynamodb:DescribeTable",
        "dynamodb:GetItem",
        "dynamodb:ListTables",
        "dynamodb:Query",
        "dynamodb:Scan",
        "ec2:Describe*",
        "ec2:GetConsoleOutput",
        "ec2:GetConsoleScreenshot",
        "ecr:BatchCheckLayerAvailability",
        "ecr:BatchGetImage",
        "ecr:Describe*",
        "ecr:Get*",
        "ecr:List*",
        "ecs:Describe*",
        "ecs:List*",
        "elasticache:Describe*",
        "elasticache:List*",
        "elasticbeanstalk:Check*",
        "elasticbeanstalk:Describe*",
        "elasticbeanstalk:List*",
        "elasticbeanstalk:RequestEnvironmentInfo",
        "elasticbeanstalk:RetrieveEnvironmentInfo",
        "elasticfilesystem:Describe*",
        "elasticloadbalancing:Describe*",
        "elasticmapreduce:Describe*",
        "elasticmapreduce:List*",
        "elastictranscoder:List*",
        "elastictranscoder:Read*",
        "es:DescribeElasticsearchDomain",
        "es:DescribeElasticsearchDomains",
        "es:DescribeElasticsearchDomainConfig",
        "es:ListDomainNames",
        "es:ListTags",
        "es:ESHttpGet",
        "es:ESHttpHead",
        "events:DescribeRule",
        "events:ListRuleNamesByTarget",
        "events:ListRules",
        "events:ListTargetsByRule",
        "events:TestEventPattern",
        "firehose:Describe*",
        "firehose:List*",
        "glacier:ListVaults",
        "glacier:DescribeVault",
        "glacier:GetDataRetrievalPolicy",
        "glacier:GetVaultAccessPolicy",
        "glacier:GetVaultLock",
        "glacier:GetVaultNotifications",
        "glacier:ListJobs",
        "glacier:ListMultipartUploads",
        "glacier:ListParts",
        "glacier:ListTagsForVault",
        "glacier:DescribeJob",
        "glacier:GetJobOutput",
        "health:Describe*",
        "health:Get*",
        "health:List*",
        "iam:GenerateCredentialReport",
        "iam:GenerateServiceLastAccessedDetails",
        "iam:Get*",
        "iam:List*",
        "inspector:Describe*",
        "inspector:Get*",
        "inspector:List*",
        "inspector:LocalizeText",
        "inspector:PreviewAgentsForResourceGroup",
        "iot:Describe*",
        "iot:Get*",
        "iot:List*",
        "kinesisanalytics:DescribeApplication",
        "kinesisanalytics:DiscoverInputSchema",
        "kinesisanalytics:GetApplicationState",
        "kinesisanalytics:ListApplications",
        "kinesis:Describe*",
        "kinesis:Get*",
        "kinesis:List*",
        "kms:Describe*",
        "kms:Get*",
        "kms:List*",
        "lambda:List*",
        "lambda:Get*",
        "logs:Describe*",
        "logs:Get*",
        "logs:FilterLogEvents",
        "logs:TestMetricFilter",
        "machinelearning:Describe*",
        "machinelearning:Get*",
        "mobilehub:GetProject",
        "mobilehub:ListAvailableFeatures",
        "mobilehub:ListAvailableRegions",
        "mobilehub:ListProjects",
        "mobilehub:ValidateProject",
        "mobilehub:VerifyServiceRole",
        "opsworks:Describe*",
        "opsworks:Get*",
        "rds:Describe*",
        "rds:ListTagsForResource",
        "redshift:Describe*",
        "redshift:ViewQueriesInConsole",
        "route53:Get*",
        "route53:List*",
        "route53domains:CheckDomainAvailability",
        "route53domains:GetDomainDetail",
        "route53domains:GetOperationDetail",
        "route53domains:ListDomains",
        "route53domains:ListOperations",
        "route53domains:ListTagsForDomain",
        "s3:Get*",
        "s3:List*",
        "sdb:GetAttributes",
        "sdb:List*",
        "sdb:Select*",
        "ses:Get*",
        "ses:List*",
        "sns:Get*",
        "sns:List*",
        "sqs:GetQueueAttributes",
        "sqs:ListQueues",
        "sqs:ReceiveMessage",
        "ssm:Describe*",
        "ssm:Get*",
        "ssm:List*",
        "storagegateway:Describe*",
        "storagegateway:List*",
        "swf:Count*",
        "swf:Describe*",
        "swf:Get*",
        "swf:List*",
        "tag:Get*",
        "trustedadvisor:Describe*",
        "waf:Get*",
        "waf:List*",
        "workspaces:Describe*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Power User Access

This IAM policy provides an AWS Directory Service user or group with full access to AWS services and resources, but does not allow management of IAM users and groups. The following lists the contents of the policy document.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "NotAction": ["iam:*", "organizations:*"],
      "Resource": "*"
    },{
      "Effect": "Allow",
      "Action": "organizations:DescribeOrganization",
      "Resource": "*"
    }
  ]
}
```

Directory Service Full Access

This IAM policy provides an AWS Directory Service user or group with the following:

- Full access to AWS Directory Service
- Access to key Amazon EC2 services required to use AWS Directory Service
- Ability to list Amazon Simple Notification Service topics

- Ability to create, manage, and delete Amazon Simple Notification Service topics with a name beginning with “DirectoryMonitoring”

The following lists the contents of the policy document.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ds:*",
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CreateNetworkInterface",
                "ec2:CreateSecurityGroup",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteSecurityGroup",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:RevokeSecurityGroupEgress",
                "ec2:RevokeSecurityGroupIngress",
                "sns:GetTopicAttributes",
                "sns:ListSubscriptions",
                "sns:ListSubscriptionsByTopic",
                "sns:ListTopics"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "sns:CreateTopic",
                "sns:DeleteTopic",
                "sns:SetTopicAttributes",
                "sns:Subscribe",
                "sns:Unsubscribe"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:sns:*:*:DirectoryMonitoring*"
        }
    ]
}
```

Directory Service Read Only Access

This IAM policy provides an AWS Directory Service user or group with read only access to the following services and resources.

- AWS Directory Service
- Amazon Elastic Compute Cloud
- Amazon Simple Notification Service

The following lists the contents of the policy document.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ds:Check*",
        "ds:Describe*",
        "ds:Get*",
        "ds:List*",
        "ds:Verify*",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcs",
        "sns:ListTopics",
        "sns:GetTopicAttributes",
        "sns:ListSubscriptions",
        "sns:ListSubscriptionsByTopic"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Amazon EC2 Full Access

This IAM policy provides an AWS Directory Service user or group with full access to the following Amazon EC2 services and resources.

- Amazon Elastic Compute Cloud
- Elastic Load Balancing
- Amazon CloudWatch
- Auto Scaling

The following lists the contents of the policy document.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "ec2:*",
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "elasticloadbalancing:*",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "cloudwatch:*",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "autoscaling:*",
      "Resource": "*"
    }
  ]
}
```

Amazon EC2 Read Only Access

This IAM policy provides an AWS Directory Service user or group with read only access to the following Amazon EC2 services and resources.

- Amazon Elastic Compute Cloud
- Elastic Load Balancing
- Amazon CloudWatch
- Auto Scaling

The following lists the contents of the policy document.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:Describe*",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "elasticloadbalancing:Describe*",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:ListMetrics",
        "cloudwatch:GetMetricStatistics",
        "cloudwatch:Describe*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "autoscaling:Describe*",
      "Resource": "*"
    }
  ]
}
```

Amazon VPC Full Access

This IAM policy provides an AWS Directory Service user or group with full access to Amazon VPC services and resources. The following lists the contents of the policy document.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:AcceptVpcPeeringConnection",
        "ec2:AllocateAddress",
        "ec2:AssignPrivateIpAddresses",
        "ec2:AssociateAddress",
        "ec2:AssociateDhcpOptions",
        "ec2:AssociateRouteTable",
        "ec2:AttachClassicLinkVpc",
        "ec2:AttachInternetGateway",
        "ec2:AttachNetworkInterface",
        "ec2:AttachVpnGateway",
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:CreateCustomerGateway",
        "ec2:CreateDhcpOptions",
        "ec2:CreateFlowLogs",
        "ec2:CreateInternetGateway",
        "ec2:CreateNatGateway",
        "ec2:CreateNetworkAcl",
        "ec2:CreateNetworkAcl",
        "ec2:CreateNetworkAclEntry",
        "ec2:CreateNetworkInterface",
        "ec2:CreateRoute",
        "ec2:CreateRouteTable",
        "ec2:CreateSecurityGroup",
        "ec2:CreateSubnet",
        "ec2:CreateTags",
        "ec2:CreateVpc",
        "ec2:CreateVpcEndpoint",
        "ec2:CreateVpcPeeringConnection",
        "ec2:CreateVpnConnection",
        "ec2:CreateVpnConnectionRoute",
        "ec2:CreateVpnGateway",
        "ec2:DeleteCustomerGateway",
        "ec2:DeleteDhcpOptions",
        "ec2:DeleteFlowLogs",
        "ec2:DeleteInternetGateway",
        "ec2:DeleteNatGateway",
        "ec2:DeleteNetworkAcl",
        "ec2:DeleteNetworkAclEntry",
        "ec2:DeleteNetworkInterface",
        "ec2:DeleteRoute",
        "ec2:DeleteRouteTable",
        "ec2:DeleteSecurityGroup",
        "ec2:DeleteSubnet",
        "ec2:DeleteTags",
        "ec2:DeleteVpc",
        "ec2:DeleteVpcEndpoints",
        "ec2:DeleteVpcPeeringConnection",
        "ec2:DeleteVpnConnection",
        "ec2:DeleteVpnConnectionRoute",
        "ec2:DeleteVpnGateway",
        "ec2:DescribeAddresses",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeClassicLinkInstances",
        "ec2:DescribeCustomerGateways",
        "ec2:DescribeDhcpOptions",
        "ec2:DescribeFlowLogs",
        "ec2:DescribeInstances",
        "ec2:DescribeInternetGateways",
        "ec2:DescribeKeyPairs",
        "ec2:DescribeMovingAddresses",
        "ec2:DescribeNatGateways",
        "ec2:DescribeNetworkAcls",
        "ec2:DescribeNetworkInterfaceAttribute",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribePrefixLists",
        "ec2:DescribeRouteTables",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeTags",
        "ec2:DescribeVpcAttribute",
        "ec2:DescribeVpcClassicLink",
        "ec2:DescribeVpcEndpoints",
        "ec2:DescribeVpcEndpointServices",
        "ec2:DescribeVpcPeeringConnections",
        "ec2:DescribeVpcs",
        "ec2:DescribeVpnConnections",
        "ec2:DescribeVpnGateways",
        "ec2:DetachClassicLinkVpc",
        "ec2:DetachInternetGateway",
        "ec2:DetachNetworkInterface",
        "ec2:DetachVpnGateway",
        "ec2:DisableVgwRoutePropagation",
        "ec2:DisableVpcClassicLink",
        "ec2:DisassociateAddress",
        "ec2:DisassociateRouteTable",
        "ec2:EnableVgwRoutePropagation",
        "ec2:EnableVpcClassicLink",
        "ec2:ModifyNetworkInterfaceAttribute",
        "ec2:ModifySubnetAttribute",
        "ec2:ModifyVpcAttribute",
        "ec2:ModifyVpcEndpoint",
        "ec2:MoveAddressToVpc",
        "ec2:RejectVpcPeeringConnection",
        "ec2:ReleaseAddress",
        "ec2:ReplaceNetworkAclAssociation",
        "ec2:ReplaceNetworkAclEntry",
        "ec2:ReplaceRoute",
        "ec2:ReplaceRouteTableAssociation",
        "ec2:ResetNetworkInterfaceAttribute",
        "ec2:RestoreAddressToClassic",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:RevokeSecurityGroupIngress",
        "ec2:UnassignPrivateIpAddresses"
      ],
      "Resource": "*"
    }
  ]
}
```

Amazon VPC Read Only Access

This IAM policy provides an AWS Directory Service user or group with read only access to Amazon VPC services and resources. The following lists the contents of the policy document.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeAddresses",
        "ec2:DescribeClassicLinkInstances",
        "ec2:DescribeCustomerGateways",
        "ec2:DescribeDhcpOptions",
        "ec2:DescribeFlowLogs",
        "ec2:DescribeInternetGateways",
        "ec2:DescribeMovingAddresses",
        "ec2:DescribeNatGateways",
        "ec2:DescribeNetworkAcls",
        "ec2:DescribeNetworkInterfaceAttribute",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribePrefixLists",
        "ec2:DescribeRouteTables",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeTags",
        "ec2:DescribeVpcAttribute",
        "ec2:DescribeVpcClassicLink",
        "ec2:DescribeVpcEndpoints",
        "ec2:DescribeVpcEndpointServices",
        "ec2:DescribeVpcPeeringConnections",
        "ec2:DescribeVpcs",
        "ec2:DescribeVpnConnections",
        "ec2:DescribeVpnGateways"
      ],
      "Resource": "*"
    }
  ]
}
```

Amazon RDS Full Access

This IAM policy provides an AWS Directory Service user or group with full access to Amazon RDS services and resources. The following lists the contents of the policy document.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "rds:*",
        "cloudwatch:DescribeAlarms",
        "cloudwatch:GetMetricStatistics",
        "ec2:DescribeAccountAttributes",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcs",
        "sns:ListSubscriptions",
        "sns:ListTopics",
        "logs:DescribeLogStreams",
        "logs:GetLogEvents"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Amazon RDS Read Only Access

This IAM policy provides an AWS Directory Service user or group with read only access to Amazon RDS services and resources. The following lists the contents of the policy document.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "rds:Describe*",
        "rds:ListTagsForResource",
        "ec2:DescribeAccountAttributes",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeVpcs"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": [
        "cloudwatch:GetMetricStatistics",
        "logs:DescribeLogStreams",
        "logs:GetLogEvents"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Amazon DynamoDB Full Access

This IAM policy provides an AWS Directory Service user or group with full access to DynamoDB services and resources. The following lists the contents of the policy document.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "dynamodb:*",
        "cloudwatch:DeleteAlarms",
        "cloudwatch:DescribeAlarmHistory",
        "cloudwatch:DescribeAlarms",
        "cloudwatch:DescribeAlarmsForMetric",
        "cloudwatch:GetMetricStatistics",
        "cloudwatch:ListMetrics",
        "cloudwatch:PutMetricAlarm",
        "datapipeline:ActivatePipeline",
        "datapipeline:CreatePipeline",
        "datapipeline:DeletePipeline",
        "datapipeline:DescribeObjects",
        "datapipeline:DescribePipelines",
        "datapipeline:GetPipelineDefinition",
        "datapipeline:ListPipelines",
        "datapipeline:PutPipelineDefinition",
        "datapipeline:QueryObjects",
        "iam:ListRoles",
        "sns:CreateTopic",
        "sns:DeleteTopic",
        "sns:ListSubscriptions",
        "sns:ListSubscriptionsByTopic",
        "sns:ListTopics",
        "sns:Subscribe",
        "sns:Unsubscribe",
        "sns:SetTopicAttributes",
        "lambda:CreateFunction",
        "lambda:ListFunctions",
        "lambda:ListEventSourceMappings",
        "lambda:CreateEventSourceMapping",
        "lambda:DeleteEventSourceMapping",
        "lambda:GetFunctionConfiguration",
        "lambda:DeleteFunction"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Amazon DynamoDB Read Only Access

This IAM policy provides an AWS Directory Service user or group with read only access to DynamoDB services and resources. The following lists the contents of the policy document.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "cloudwatch:DescribeAlarmHistory",
        "cloudwatch:DescribeAlarms",
        "cloudwatch:DescribeAlarmsForMetric",
        "cloudwatch:GetMetricStatistics",
        "cloudwatch:ListMetrics",
        "datapipeline:DescribeObjects",
        "datapipeline:DescribePipelines",
        "datapipeline:GetPipelineDefinition",
        "datapipeline:ListPipelines",
        "datapipeline:QueryObjects",
        "dynamodb:BatchGetItem",
        "dynamodb:DescribeTable",
        "dynamodb:GetItem",
        "dynamodb:ListTables",
        "dynamodb:Query",
        "dynamodb:Scan",
        "dynamodb:DescribeReservedCapacity",
        "dynamodb:DescribeReservedCapacityOfferings",
        "sns:ListSubscriptionsByTopic",
        "sns:ListTopics",
        "lambda:ListFunctions",
        "lambda:ListEventSourceMappings",
        "lambda:GetFunctionConfiguration"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Amazon S3 Full Access

This IAM policy provides an AWS Directory Service user or group with full access to Amazon S3 services and resources. The following lists the contents of the policy document.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:*",
      "Resource": "*"
    }
  ]
}
```

Amazon S3 Read Only Access

This IAM policy provides an AWS Directory Service user or group with read only access to Amazon S3 services and resources. The following lists the contents of the policy document.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
      "Resource": "*"
    }
  ]
}
```

AWS CloudTrail Full Access

This IAM policy provides an AWS Directory Service user or group with full access to CloudTrail services and resources. The following lists the contents of the policy document.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sns:AddPermission",
        "sns:CreateTopic",
        "sns:DeleteTopic",
        "sns:ListTopics",
        "sns:SetTopicAttributes",
        "sns:GetTopicAttributes"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:CreateBucket",
        "s3:DeleteBucket",
        "s3:ListAllMyBuckets",
        "s3:PutBucketPolicy",
        "s3:ListBucket",
        "s3:GetObject",
        "s3:GetBucketLocation",
        "s3:GetBucketPolicy"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "cloudtrail:*",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole",
        "iam:ListRoles",
        "iam:GetRolePolicy",
        "iam:GetUser"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:ListKeys",
        "kms:ListAliases"
      ],
      "Resource": "*"
    }
  ]
```

```
}
```

AWS CloudTrail Read Only Access

This IAM policy provides an AWS Directory Service user or group with read only access to CloudTrail services and resources. The following lists the contents of the policy document.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetBucketLocation"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudtrail:GetTrailStatus",
        "cloudtrail:DescribeTrails",
        "cloudtrail:LookupEvents",
        "cloudtrail:ListTags",
        "s3:ListAllMyBuckets",
        "kms:ListAliases"
      ],
      "Resource": "*"
    }
  ]
}
```

CloudWatch Full Access

This IAM policy provides an AWS Directory Service user or group with full access to CloudWatch services and resources. The following lists the contents of the policy document.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "autoscaling:Describe*",
        "cloudwatch:*",
        "logs:*",
        "sns:*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

CloudWatch Read Only Access

This IAM policy provides an AWS Directory Service user or group with read only access to CloudWatch services and resources. The following lists the contents of the policy document.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "autoscaling:Describe*",
        "cloudwatch:Describe*",
        "cloudwatch:Get*",
        "cloudwatch:List*",
        "logs:Get*",
        "logs:Describe*",
        "logs:TestMetricFilter",
        "sns:Get*",
        "sns:List*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

CloudWatch Logs Full Access

This IAM policy provides an AWS Directory Service user or group with full access to CloudWatch Logs services and resources. The following lists the contents of the policy document.

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

CloudWatch Logs Read Only Access

This IAM policy provides an AWS Directory Service user or group with read only access to CloudWatch Logs services and resources. The following lists the contents of the policy document.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:Describe*",
                "logs:Get*",
                "logs:TestMetricFilter",
                "logs:FilterLogEvents"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Managing Directory Applications and Services

AWS Directory Service can give other AWS applications and services, such as Amazon WorkSpaces, access to your directory users.

To display the applications and services that can work with your AWS Directory Service directory, perform the following steps.

To display the applications and services for a directory

1. In the AWS Directory Service console navigation pane, choose **Directories**.
2. Choose the directory ID link for your directory.
3. Choose the **Apps & Services** tab.

You manage access to your directories in the console of the application or service that you want to give access to your directory. To enable or disable access to your directories, choose the link for the application or service that you want to modify. The following applications and services can be used with AWS Directory Service:

Amazon WorkSpaces

> You can create a Simple AD, Microsoft AD, or AD Connector directly from Amazon WorkSpaces. Simply launch **Advanced Setup** when creating your Workspace.
>
> For more information, see the Amazon WorkSpaces Administration Guide.

Amazon WorkDocs

> For more information, see the Amazon WorkDocs Administration Guide.

Amazon WorkMail

> For more information, see the Amazon WorkMail Administrator Guide.

AWS Management Console

> For more information, see Managing Console Access for AWS Directory Service (p. 78).

Creating an Access URL

You can create an access URL for your directory by performing the following steps.

> **Warning**
> After an access URL has been created, it cannot be deleted or reused, so this procedure should only be used when absolutely necessary.

To create an access URL

1. In the AWS Directory Service console navigation pane, select **Directories**.

2. Choose the directory ID link for your directory.
3. In the **Access URL** section, if an access URL has not been assigned to the directory, the **Create Access URL** button is displayed. Enter a directory alias and choose **Create Access URL**. If an **Entity Already Exists** error is returned, the specified directory alias has already been allocated. Choose another alias and repeat this procedure.

   Your directory URL is changed to `<alias>.awsapps.com`.

Add Users and Groups (Simple AD and Microsoft AD)

You can create users and groups with the Active Directory Users and Computers tool, which is part of the Active Directory Domain Services and Active Directory Lightweight Directory Services tools. Users represent individual people or entities that have access to your directory. Groups are very useful for giving or denying privileges to groups of users, rather than having to apply those privileges to each individual user. If a user moves to a different organization, you move that user to a different group and they automatically receive the privileges needed for the new organization.

> **Note**
> Your user accounts must have Kerberos preauthentication enabled. This is the default setting for new user accounts, but it should not be modified. For more information about this setting, go to Preauthentication on Microsoft TechNet.

The following examples demonstrate how to create a user, create a group, and add the user to the group. To create users and groups in a AWS Directory Service directory, you must be connected to a Windows instance that is a member of the AWS Directory Service directory, and be logged in as a user that has privileges to create users and groups.

> **Note**
> When using Simple AD, if you create a user account on a Linux instance with the option "Force user to change password at first login," that user will not be able to initially change their password using **kpasswd**. In order to change the password the first time, a domain administrator must update the user password using the Active Directory Management Tools.

To create a user

1. Open the Active Directory Users and Computers tool. There is a shortcut to this tool in the **Administrative Tools** folder.

   > **Tip**
   > You can run the following from a command prompt on the instance to open the Active Directory Users and Computers tool box directly.

   ```
   %SystemRoot%\system32\dsa.msc
   ```
2. In the directory tree, open your directory and select the **Users** folder.
3. On the **Action** menu, click **New**, and then click **User** to open the new user wizard.
4. In the first page of the new user wizard, enter the following values and click **Next**.

   First name
   `Mary`

   Last name
   `Major`

   User logon name
   `marym`

5. In the second page of the new user wizard, enter a temporary password for **Password** and **Confirm Password**. Make sure the **User must change password at next logon** option is selected. None of the other options should be selected. Click **Next**.
6. In the third page of the new user wizard, verify that the new user information is correct and click **Finish**. The new user will appear in the **Users** folder.

To create a group

1. Open the Active Directory Users and Computers tool. There is a shortcut to this tool in the **Administrative Tools** folder.

   Tip
   You can run the following from a command prompt on the instance to open the Active Directory Users and Computers tool box directly.

   ```
   %SystemRoot%\system32\dsa.msc
   ```
2. In the directory tree, open your directory and select the **Users** folder.
3. On the **Action** menu, click **New**, and then click **Group** to open the new group wizard.
4. Enter **`Division Managers`** for the **Group name**, select **Global** for the **Group scope**, and select **Security** for the **Group type**. Click **OK**. The new group, **Division Managers**, appears in the **Users** folder.

To add a user to a group

1. Open the Active Directory Users and Computers tool. There is a shortcut to this tool in the **Administrative Tools** folder.

   Tip
   You can run the following from a command prompt on the instance to open the Active Directory Users and Computers tool box directly.

   ```
   %SystemRoot%\system32\dsa.msc
   ```
2. In the directory tree, open your directory, select the **Users** folder, and select the **Division Managers** group.
3. On the **Action** menu, click **Properties** to open the properties dialog box for the **Division Managers** group.
4. Select the **Members** tab and click **Add...**.
5. For **Enter the object names to select**, enter `marym` and click **OK**. **Mary Major** is displayed in the **Members** list. Click **OK** again to update the group membership.
6. Verify that Mary Major is now a member of the **Division Managers** group by selecting **Mary Major** in the **Users** folder, click **Properties** in the **Action** menu to open the properties dialog box for Mary Major. Select the **Member Of** tab. **Division Managers** is in the list of groups that Mary Major belongs to.

Installing the Active Directory Administration Tools

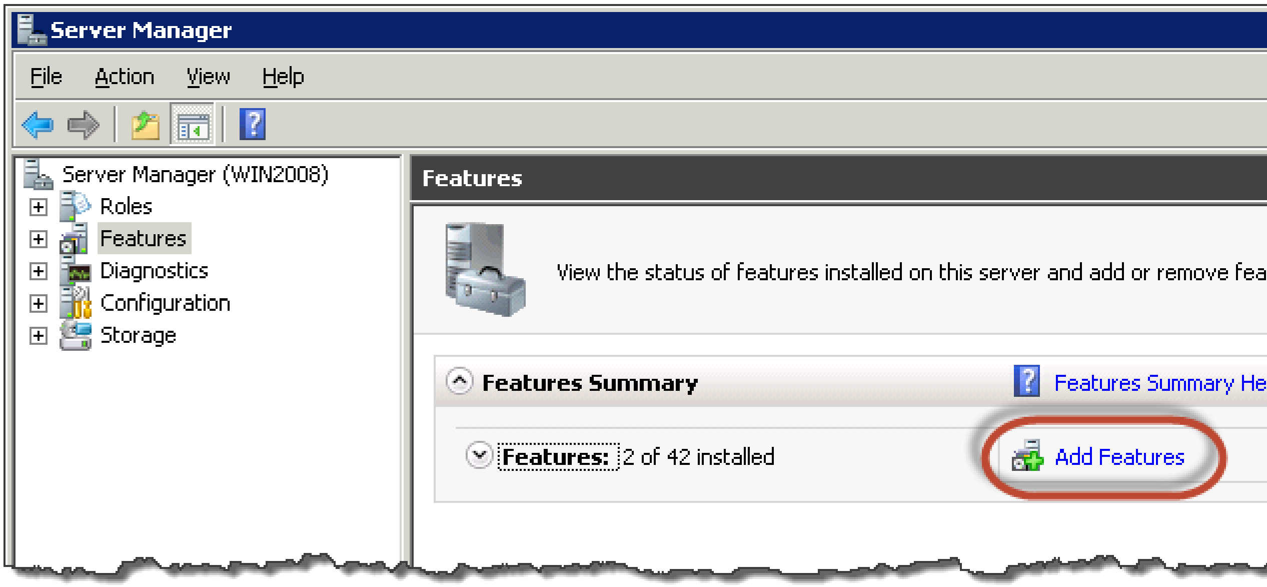
To manage your directory from an EC2 Windows instance, you need to install the Active Directory Domain Services and Active Directory Lightweight Directory Services Tools on the instance.

Topics

Install the Active Directory Administration Tools on Windows Server 2008

To install the Active Directory administration tools on Windows Server 2008

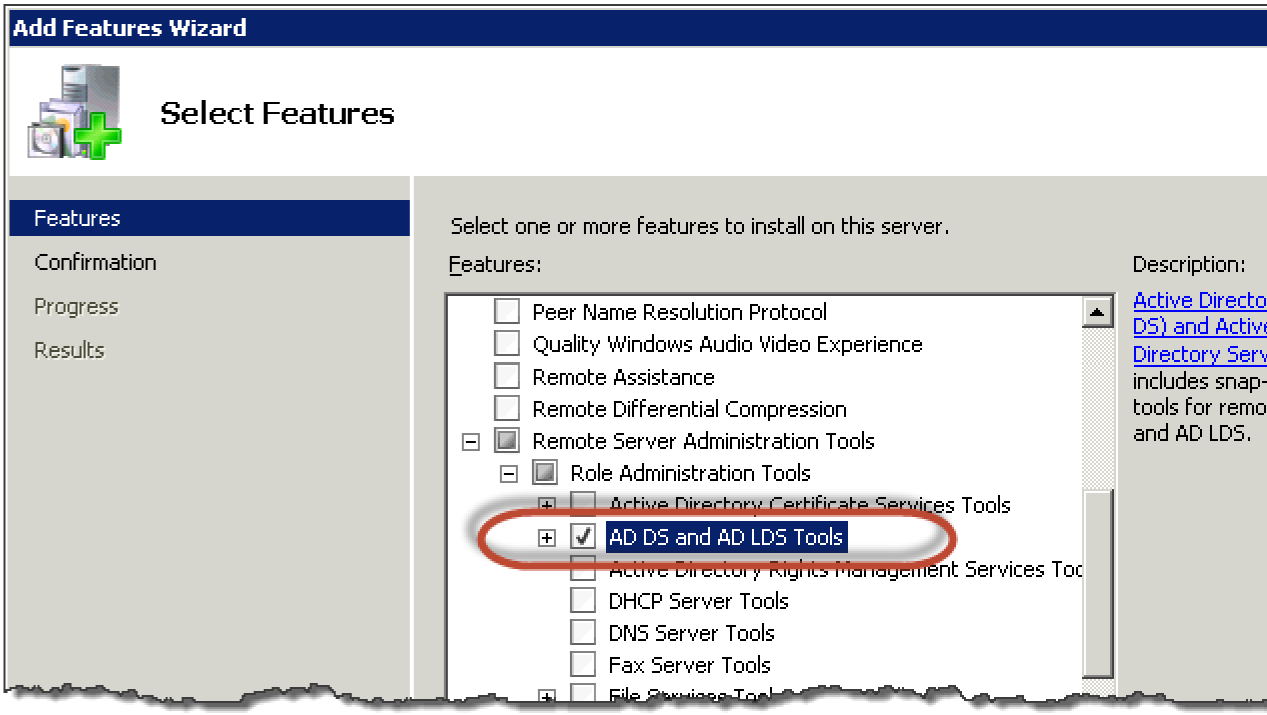
1. Open Server Manager by choosing **Start**, **Administrative Tools**, **Server Manager**.
2. In the **Server Manager** tree pane, select **Features**, and choose **Add Features**,



View the status of features installed on this server and add or remove fea

Features: 2 of 42 installed

3. In the **Add Features Wizard**, open **Remote Server Administration Tools**, **Role Administration Tools**, select **AD DS and AD LDS Tools**, scroll down and select **DNS**, then choose **Next**.



Select one or more features to install on this server.

- ☐ Peer Name Resolution Protocol
- ☐ Quality Windows Audio Video Experience
- ☐ Remote Assistance
- ☐ Remote Differential Compression
- ▣ Remote Server Administration Tools
  - ▣ Role Administration Tools
    - ☐ Active Directory Certificate Services Tools
    - ☑ AD DS and AD LDS Tools
    - ☐ Active Directory Rights Management Services Too
    - ☐ DHCP Server Tools
    - ☐ DNS Server Tools
    - ☐ Fax Server Tools
    - ☐ File Services Tool

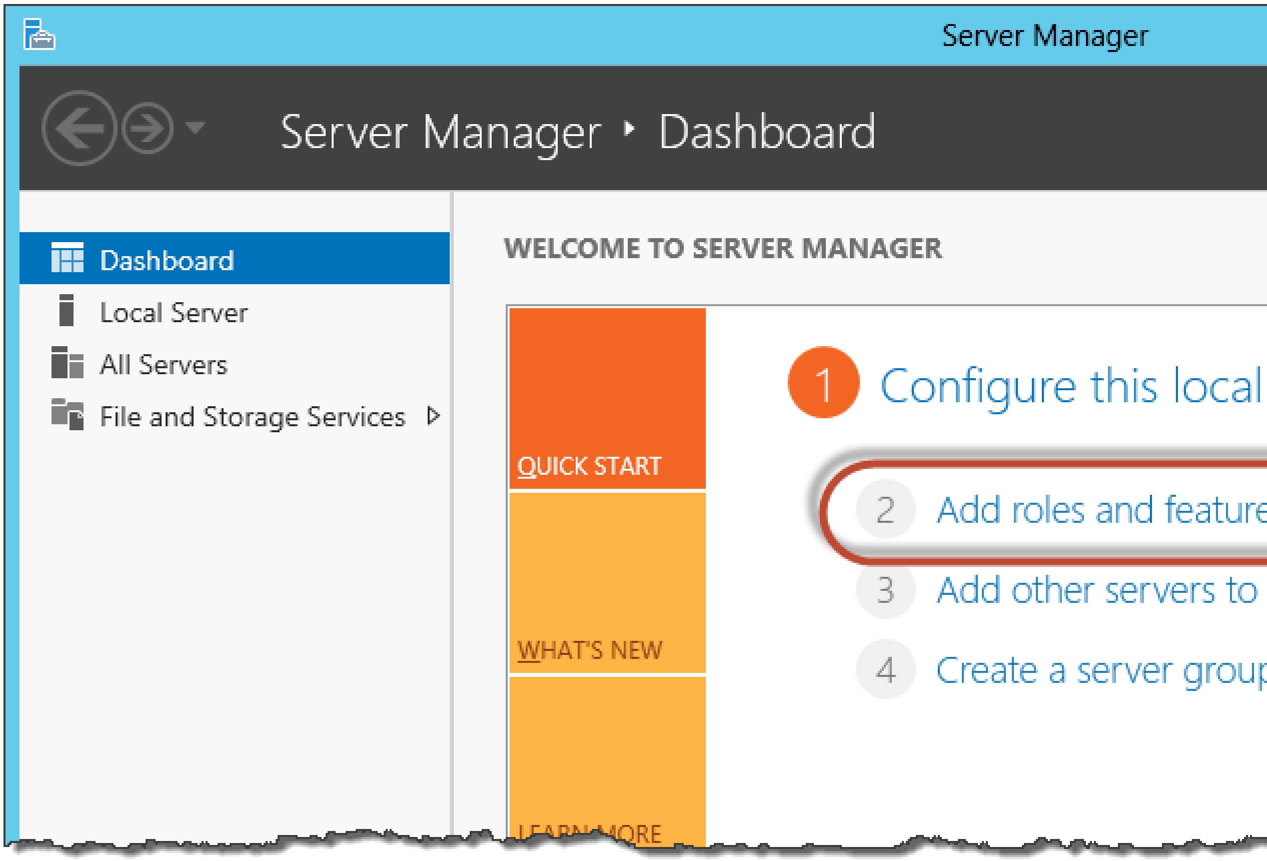
Active Directo DS) and Active Directory Serv includes snap- tools for remo and AD LDS.

4. Review the information and choose **Install**. The feature installation requires that the instance be restarted. When the instance has restarted, the Active Directory Domain Services and Active Directory Lightweight Directory Services Tools are available on the **Start** menu, under **All Programs** > **Administrative Tools**.

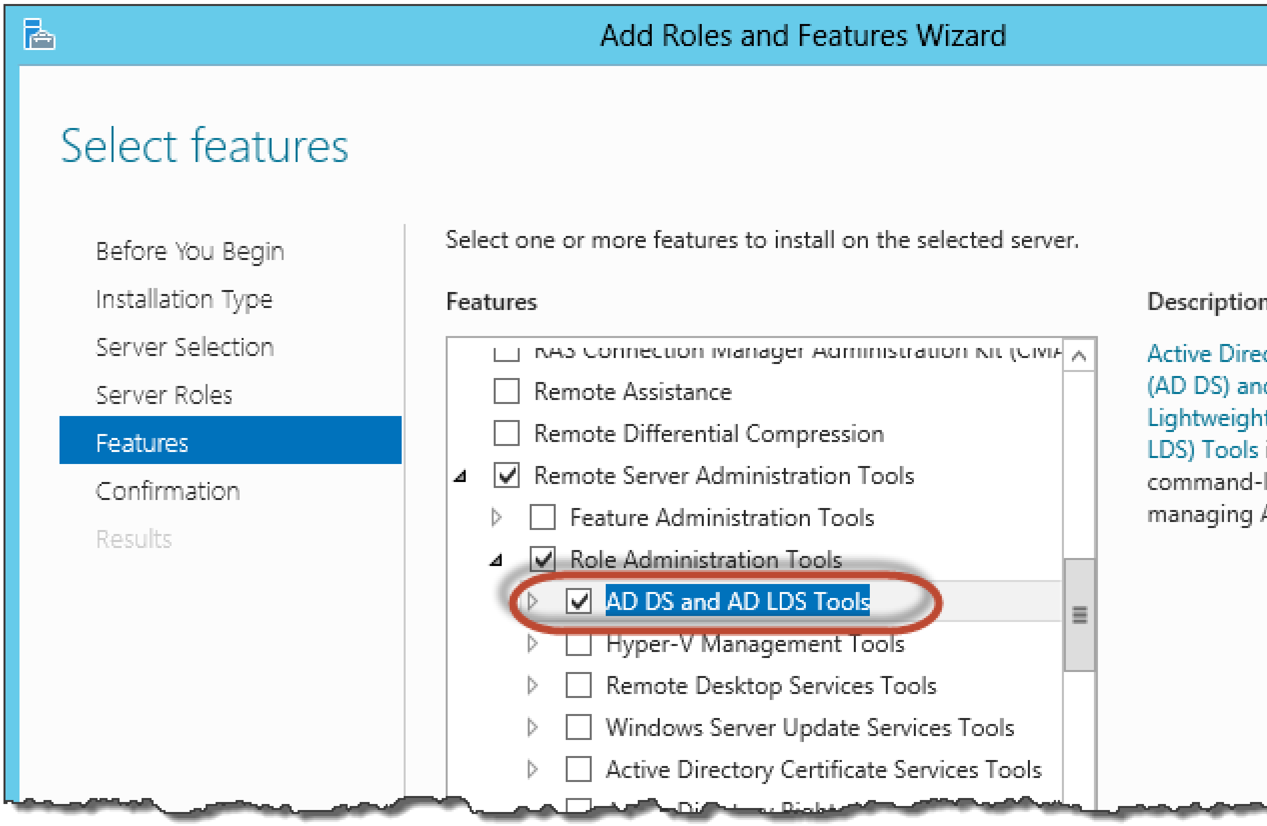
Install the Active Directory Administration Tools on Windows Server 2012

To install the Active Directory administration tools on Windows Server 2012

1. Open Server Manager by from the Start screen by choosing **Server Manager**.
2. In the **Server Manager Dashboard**, choose **Add roles and features**,



3. In the **Add Roles and Features Wizard** choose **Installation Type**, select **Role-based or feature-based installation**, and choose **Next**.
4. Under **Server Selection**, make sure the local server is selected, and choose **Features**.
5. In the **Features** tree, open **Remote Server Administration Tools**, **Role Administration Tools**, select **AD DS and AD LDS Tools**, scroll down and select **DNS**, then choose **Next**.



Add Roles and Features Wizard

Select features

- Before You Begin
- Installation Type
- Server Selection
- Server Roles
- Features
- Confirmation
- Results

Select one or more features to install on the selected server.

Features

- □ RAS Connection Manager Administration Kit (CMA
- □ Remote Assistance
- □ Remote Differential Compression
- ☑ Remote Server Administration Tools
  - □ Feature Administration Tools
  - ☑ Role Administration Tools
    - ☑ AD DS and AD LDS Tools
    - □ Hyper-V Management Tools
    - □ Remote Desktop Services Tools
    - □ Windows Server Update Services Tools
    - □ Active Directory Certificate Services Tools

Description

Active Direc
(AD DS) and
Lightweight
LDS) Tools i
command-l
managing A

6. Review the information and choose **Install**. When the feature installation is finished, the Active Directory Domain Services and Active Directory Lightweight Directory Services Tools are available on the Start screen in the **Administrative Tools** folder.

Add an Instance to Your Directory (Simple AD and Microsoft AD)

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

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

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

Topics

- DHCP Options Set (p. 117)

Launching an Instance (Simple AD and Microsoft AD)

To launch an instance to be manually joined to a directory in a VPC

1. Sign in to the AWS Management Console and open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the region selector in the navigation bar, select the same region as the existing directory.
3. From the Amazon EC2 console dashboard, choose **Launch Instance**.
4. Select the appropriate AMI.
5. In the **Configure Instance Details** page of the launch wizard, make the following selections:

   Network
   Select the VPC that your directory was created in.

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

   Auto-assign Public IP
   Although a public IP address is not required to join the domain, in order to connect to the instance, the instance must have a public IP address. Set this to **Enable** to assign a public IP address automatically, or assign an Elastic IP address to the instance after it is launched.

   Domain join directory
   To manually join the instance to your domain, leave this empty. If you want to have your instance seamlessly joined to your domain, select your domain from the **Domain join directory** list, and select the IAM role to associate with the instance from the **IAM role** list. If you choose this option, you will not have to manually join the instance to the domain as that will be done for you when the instance is launched. For more information, see Seamlessly Joining a Windows Instance to an AWS Directory Service Domain in the *Amazon EC2 User Guide for Windows Instances*.

   > **Note**
   > This option is only available for Windows instances. Linux instances must be manually joined to the directory as explained in Manually Add a Linux Instance (Simple AD and Microsoft AD) (p. 108).

6. The security group you select for the instance must allow remote access to the instance from your network.

Manually Add a Windows Instance (Simple AD and Microsoft AD)

To manually join an existing Amazon EC2 Windows instance to a Simple AD or AWS Directory Service for Microsoft Active Directory (Enterprise Edition) directory, the instance must be launched as specified in Launching an Instance (Simple AD and Microsoft AD) (p. 106).

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

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

a. Open **Network Connections**.

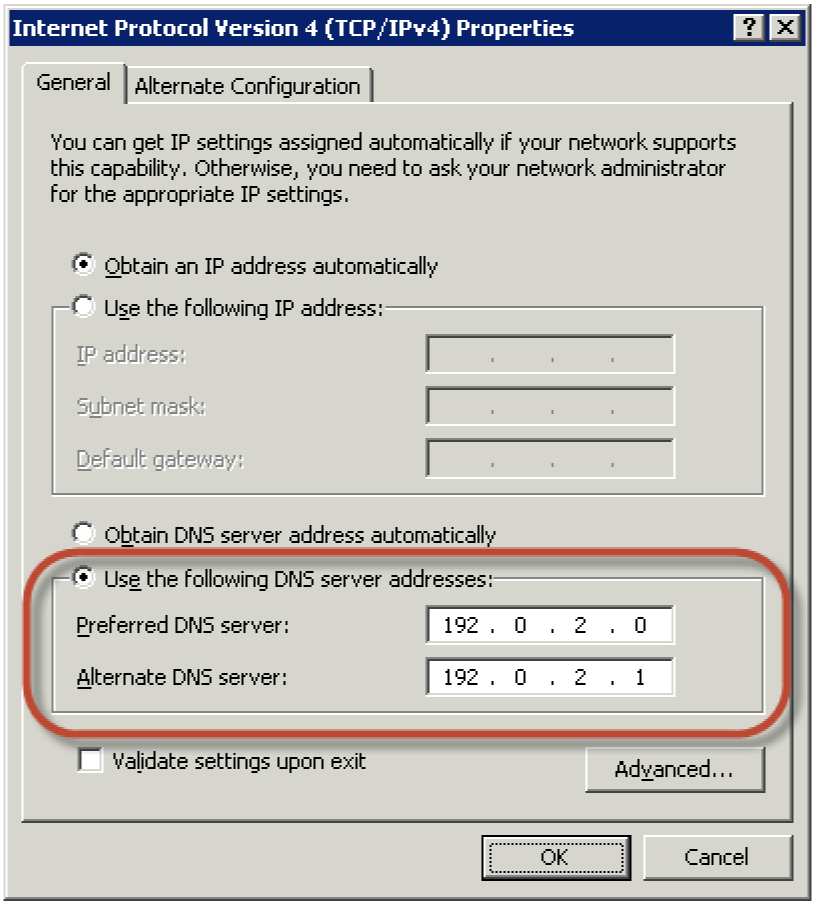
Tip
You can open **Network Connections** directly by running the following from a command prompt on the instance.

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

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

c. In the connection properties dialog box, open (double-click) **Internet Protocol Version 4**.

3. (Optional) Select **Use the following DNS server addresses**, change the **Preferred DNS server** and **Alternate DNS server** addresses to the IP addresses of the AWS Directory Service-provided DNS servers, and choose **OK**.



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

● Obtain an IP address automatically

○ Use the following IP address:

IP address: . . .

Subnet mask: . . .

Default gateway: . . .

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: 192 . 0 . 2 . 0

Alternate DNS server: 192 . 0 . 2 . 1

□ Validate settings upon exit

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

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

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

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

6. When prompted for the name and password for the domain administrator, enter the username and password of an account that has domain join privileges. For more information about delegating these privileges, see Delegating Directory Join Privileges (Simple AD and Microsoft AD) (p. 111).
7. After you receive the message welcoming you to the domain, restart the instance to have the changes take effect.

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

Manually Add a Linux Instance (Simple AD and Microsoft AD)

In addition to Amazon EC2 Windows instances, you can also join certain Amazon EC2 Linux instances to a Simple AD or AWS Directory Service for Microsoft Active Directory (Enterprise Edition) directory. The following Linux instance distributions and versions are supported:

- Amazon Linux AMI 2015.03
- Red Hat Enterprise Linux 7.2
- Ubuntu Server 14.04 LTS
- CentOS 7

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

Prerequisites for Joining an Instance to a Simple AD or Microsoft AD Directory

To join a Linux instance to your directory, the instance must be launched as specified in Launching an Instance (Simple AD and Microsoft AD) (p. 106).

> **Important**
> When using Simple AD, if you create a user account on a Linux instance with the option "Force user to change password at first login," that user will not be able to initially change their password using **kpasswd**. In order to change the password the first time, a domain administrator must update the user password using the Active Directory Management Tools.
> Some of the following procedures, if not performed correctly, can render your instance unreachable or unusable. Therefore, we strongly suggest you make a backup or take a snapshot of your instance before performing these procedures.

To join a Linux instance to a Simple AD or Microsoft AD directory

1. Connect to the instance using any SSH client.
2. Make sure the instance is up to date.

   Amazon Linux - 64bit/Red Hat - 64bit/CentOS 7

   ```
   $ sudo yum -y update
   ```

   Ubuntu - 64bit

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

3. Install the required packages on your Linux instance.

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

   Amazon Linux - 64bit/Red Hat - 64bit/CentOS 7

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

   Ubuntu - 64bit

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

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

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

   `join_account@example.com`
   An account in the `example.com` domain that has domain join privileges. Enter the password for the account when prompted. For more information about delegating these privileges, see Delegating Directory Join Privileges (Simple AD and Microsoft AD) (p. 111).

   `example.com`
   The fully-qualified DNS name of your directory.

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

   Note
   If you are using Ubuntu 14.04, and encounter the following error:
   realm: Couldn't join realm: Failed to enroll machine in realm. See diagnostics..
   Complete the following additional steps, and then attempt a domain join to resolve the issue:

   ```
   $ killall aptd
   $ apt-get install packagekit adcli
   ```

5. Set the SSH service to allow password authentication.

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

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

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

   ```
   PasswordAuthentication yes
   ```

6. Start the SSSD service.

   ```
   $ sudo systemctl start sssd.service
   ```

   Alternatively:

```
$ sudo service sssd start
```

7. Restart the instance.
8. After the instance has restarted, connect to it with any SSH client and add the domain administrators to the sudoers list by performing the following steps:

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

   ```
   $ sudo visudo -f /etc/sudoers
   ```

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

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

   (The above example uses "\<space>" in order to create the Linux space character.)
9. By default, all users in the directory can log in to the instance. You can allow only specific users to log in to the instance with **ad_access_filter**. For example:

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

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

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

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

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

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

You must manually add **ad_access_filter** to your **sssd.conf**. After you do this, your **sssd.conf** might look like this:

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

[domain/example.com]
ad_domain = example.com
krb5_realm = EXAMPLE.COM
realmd_tags = manages-system joined-with-samba
cache_credentials = True
id_provider = ad
krb5_store_password_if_offline = True
default_shell = /bin/bash
ldap_id_mapping = True
```

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

For more information about **ad_access_filter**, see Active Directory client access control.

Connecting to the Instance

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

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

Delegating Directory Join Privileges (Simple AD and Microsoft AD)

To join a computer to your directory, you need an account that has privileges to join computers to the directory.

With Simple AD, members of the **Domain Admins** group have sufficient privileges to join computers to the directory.

With AWS Directory Service for Microsoft Active Directory (Enterprise Edition), members of the **Admins** and **Server Admins** groups have these privileges.

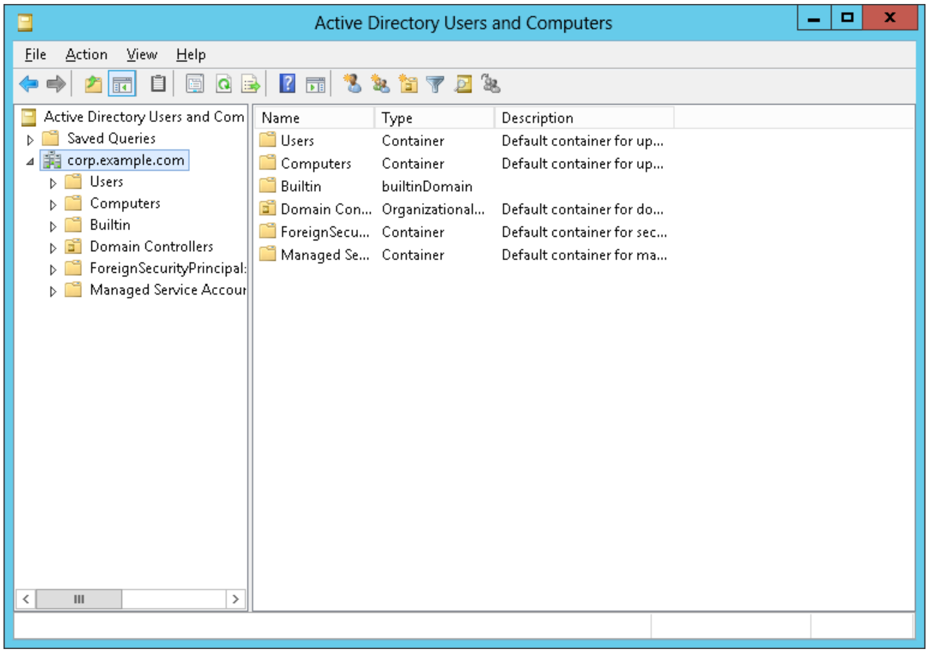
However, as a best practice, you should use an account that has only the minimum privileges necessary. The following procedure demonstrates how to create a new group called `Joiners` and delegate the privileges to this group that are needed to join computers to the directory.

You must perform this procedure on a machine that is joined to your directory and has the **Active Directory User and Computers** MMC snap-in installed. You must also be logged in as a domain administrator.

Simple AD

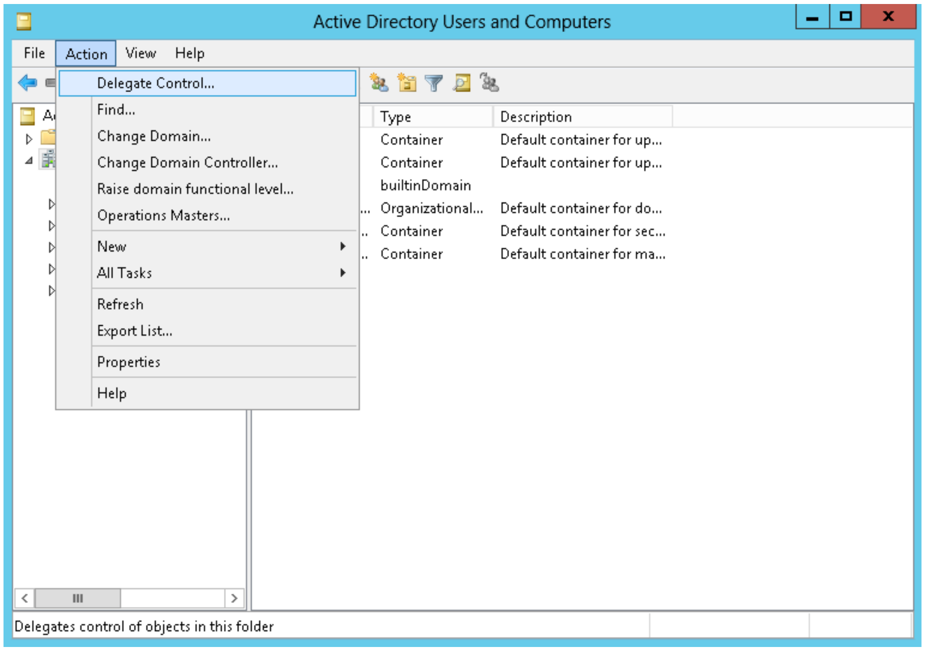
To delegate join privileges

1. Open **Active Directory User and Computers** and select your domain root in the navigation tree.



Name	Type	Description
Users	Container	Default container for up...
Computers	Container	Default container for up...
Builtin	builtinDomain	
Domain Con...	Organizational...	Default container for do...
ForeignSecu...	Container	Default container for sec...
Managed Se...	Container	Default container for ma...

2. In the navigation tree on the left, open the context menu (right-click) for **Users**, choose **New**, and then choose **Group**.
3. In the **New Object - Group** box, type the following and choose **OK**.
   - For **Group name**, type `Joiners`.
   - For **Group scope**, choose **Global**.
   - For **Group type**, choose **Security**.
4. In the navigation tree, select your domain root. From the **Action** menu, choose **Delegate Control**.

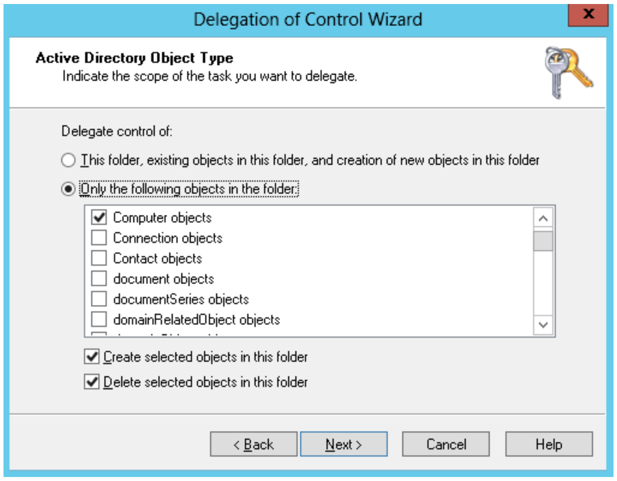


Type	Description
Container	Default container for up...
Container	Default container for up...
builtinDomain	
Organizational...	Default container for do...
Container	Default container for sec...
Container	Default container for ma...

Delegates control of objects in this folder

5. On the **Delegation of Control Wizard** page, choose **Next**, and then choose **Add**.

6. In the **Select Users, Computers, or Groups** box, type `Joiners` and choose **OK**. If more than one object is found, select the `Joiners` group created above. Choose **Next**.
7. On the **Tasks to Delegate** page, select **Create a custom task to delegate**, and then choose **Next**.
8. Select **Only the following objects in the folder**, and then select **Computer objects**.
9. Select **Create selected objects in this folder** and **Delete selected objects in this folder**. Then choose **Next**.



Delegation of Control Wizard

Active Directory Object Type
Indicate the scope of the task you want to delegate.

Delegate control of:

○ This folder, existing objects in this folder, and creation of new objects in this folder

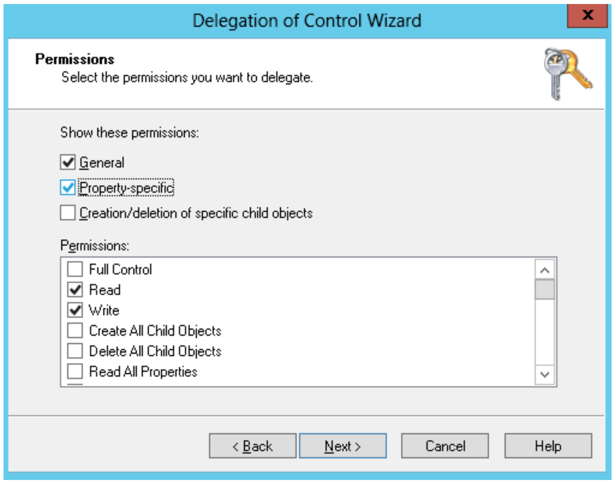
● Only the following objects in the folder:

- ☑ Computer objects
- ☐ Connection objects
- ☐ Contact objects
- ☐ document objects
- ☐ documentSeries objects
- ☐ domainRelatedObject objects

☑ Create selected objects in this folder

☑ Delete selected objects in this folder

10. Select **Read** and **Write**, and then choose **Next**.



Delegation of Control Wizard

Permissions
Select the permissions you want to delegate.

Show these permissions:

☑ General

☑ Property-specific

☐ Creation/deletion of specific child objects

Permissions:

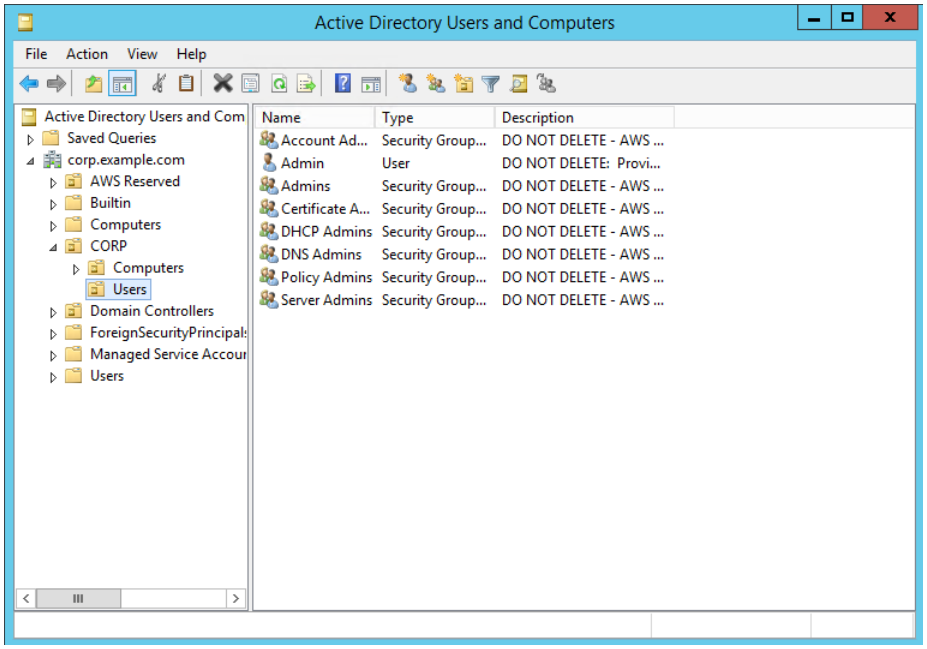
- ☐ Full Control
- ☑ Read
- ☑ Write
- ☐ Create All Child Objects
- ☐ Delete All Child Objects
- ☐ Read All Properties

11. Verify the information on the **Completing the Delegation of Control Wizard** page and choose **Finish**.
12. Create a user with a strong password and add that user to the `Joiners` group. The user will then have sufficient privileges to connect AWS Directory Service to the directory.

AWS Directory Service for Microsoft Active Directory (Enterprise Edition)

To delegate join privileges

1. Open **Active Directory User and Computers** and select the organizational unit (OU) that has your NetBIOS name in the navigation tree, then select the **Users** OU.

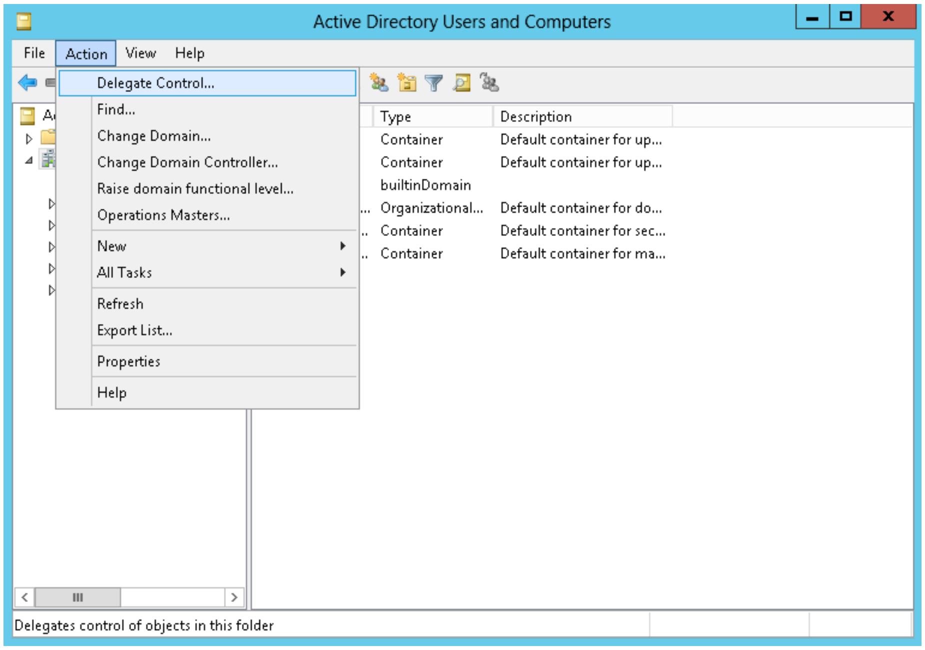


Name	Type	Description
Account Ad...	Security Group...	DO NOT DELETE - AWS ...
Admin	User	DO NOT DELETE: Provi...
Admins	Security Group...	DO NOT DELETE - AWS ...
Certificate A...	Security Group...	DO NOT DELETE - AWS ...
DHCP Admins	Security Group...	DO NOT DELETE - AWS ...
DNS Admins	Security Group...	DO NOT DELETE - AWS ...
Policy Admins	Security Group...	DO NOT DELETE - AWS ...
Server Admins	Security Group...	DO NOT DELETE - AWS ...

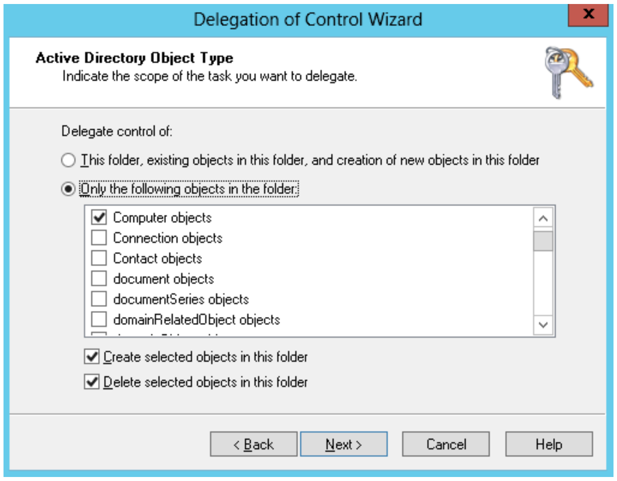
> **Important**
> When you launch a AWS Directory Service for Microsoft Active Directory (Enterprise Edition), AWS creates an organizational unit (OU) that contains all your directory’s objects. This OU, which has the NetBIOS name that you typed when you created your directory, is located in the domain root. The domain root is owned and managed by AWS. You cannot make changes to the domain root itself, therefore, you must create the **`Joiners`** group within the OU that has your NetBIOS name.

2. Open the context menu (right-click) for **Users**, choose **New**, and then choose **Group**.
3. In the **New Object - Group** box, type the following and choose **OK**.

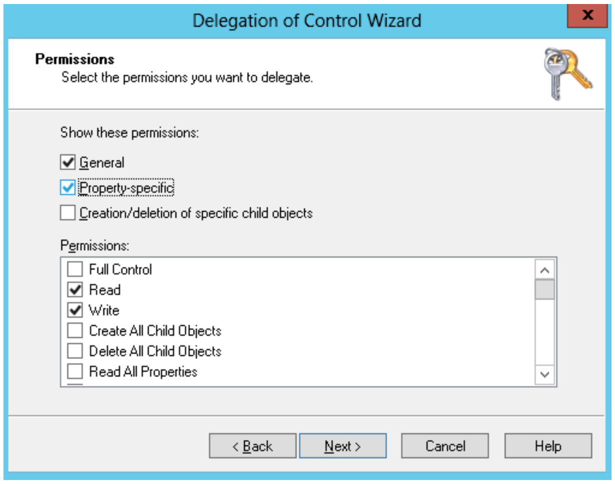
   - For **Group name**, type **`Joiners`**.
   - For **Group scope**, choose **Global**.
   - For **Group type**, choose **Security**.
4. In the navigation tree, select the **Computers** container under your NetBIOS name. From the **Action** menu, choose **Delegate Control**.


Active Directory Users and Computers
File
Action
View
Help
Delegate Control...
Find...
Change Domain...
Change Domain Controller...
Raise domain functional level...
Operations Masters...
New
All Tasks
Refresh
Export List...
Properties
Help
Type
Description
Container
Default container for up...
Container
Default container for up...
builtinDomain
Organizational...
Default container for do...
Container
Default container for sec...
Container
Default container for ma...
Delegates control of objects in this folder

5. On the **Delegation of Control Wizard** page, choose **Next**, and then choose **Add**.
6. In the **Select Users, Computers, or Groups** box, type `Joiners` and choose **OK**. If more than one object is found, select the `Joiners` group created above. Choose **Next**.
7. On the **Tasks to Delegate** page, select **Create a custom task to delegate**, and then choose **Next**.
8. Select **Only the following objects in the folder**, and then select **Computer objects**.
9. Select **Create selected objects in this folder** and **Delete selected objects in this folder**. Then choose **Next**.


Delegation of Control Wizard
Active Directory Object Type
Indicate the scope of the task you want to delegate.
Delegate control of:
○ This folder, existing objects in this folder, and creation of new objects in this folder
● Only the following objects in the folder:
☑ Computer objects
☐ Connection objects
☐ Contact objects
☐ document objects
☐ documentSeries objects
☐ domainRelatedObject objects
☑ Create selected objects in this folder
☑ Delete selected objects in this folder
< Back
Next >
Cancel
Help

10. Select **Read** and **Write**, and then choose **Next**.



Permissions

Select the permissions you want to delegate.

Show these permissions:

☑ General

☑ Property-specific

□ Creation/deletion of specific child objects

Permissions:

□ Full Control

☑ Read

☑ Write

□ Create All Child Objects

□ Delete All Child Objects

□ Read All Properties

11. Verify the information on the **Completing the Delegation of Control Wizard** page and choose **Finish**.

12. Create a user with a strong password and add that user to the `Joiners` group. This user must be in the **Users** container that is under your NetBIOS name. The user will then have sufficient privileges to connect instances to the directory.

Using DNS with Simple AD and Microsoft AD

DNS and Simple AD

Simple AD forwards DNS requests to the IP address of the Amazon-provided DNS servers for your VPC. These DNS servers will resolve names configured in your Amazon Route 53 private hosted zones. By pointing your on-premises computers to your Simple AD, you can now resolve DNS requests to the private hosted zone.

Note that to enable your Simple AD to respond to external DNS queries, the network access control list (ACL) for the VPC containing your Simple AD must be configured to allow traffic from outside the VPC.

If you are not using Amazon Route 53 private hosted zones, your DNS requests will be forwarded to public DNS servers.

If you're using custom DNS servers that are outside of your VPC and you want to use private DNS, you must reconfigure to use custom DNS servers on EC2 instances within your VPC. For more information, see Working with Private Hosted Zones.

If you want your Simple AD to resolve names using both DNS servers within your VPC and private DNS servers outside of your VPC, you can do this using a DHCP options set. For a detailed example, see this article.

> **Note**
> DNS dynamic updates are not supported in Simple AD domains. You can instead make the changes directly by connecting to your directory using DNS Manager on an instance that is joined to your domain.

For more information on Amazon Route 53, see What is Amazon Route 53.

DNS and AWS Directory Service for Microsoft Active Directory (Enterprise Edition)

The Microsoft AD directory service is tightly integrated with Microsoft Active Directory. Members of either the **Admins** group or the **DNS Admins** group can manage DNS zones, records, logs, forwarders and more using a variety of tools, such as the DNSMGMT console, PowerShell, or the DNSCMD features included with Microsoft's Remote Server Administration Tools (RSAT).

DHCP Options Set

AWS recommends that you create a DHCP options set for your AWS Directory Service directory and assign the DHCP options set to the VPC that your directory is in. This allows any instances in that VPC to point to the specified domain and DNS servers to resolve their domain names.

For more information about DHCP options sets, see DHCP Options Sets in the *Amazon VPC User Guide*.

To create a DHCP options set for your directory

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. Choose **DHCP Options Sets** in the navigation pane, and choose **Create DHCP options set**.
3. In the **Create DHCP options set** dialog box, enter the following values for your directory:

   Name tag
   An optional tag for the options set.

   Domain name
   The fully-qualified name of your directory, such as `corp.example.com`.

   Domain name servers
   The IP addresses of your directory's DNS servers. These are the IP addresses of your AWS-provided directory. You can find these addresses by going to the AWS Directory Service console navigation pane, selecting **Directories** and then choosing the correct directory ID.

   NTP servers
   Leave this field blank.

   NetBIOS name servers
   Leave this field blank.

   NetBIOS node type
   Leave this field blank.
4. Choose **Yes, Create**. The new set of DHCP options appears in your list of DHCP options.
5. Make a note of the ID of the new set of DHCP options (dopt-*xxxxxxxx*). You need it to associate the new options set with your VPC.

To change the DHCP options set associated with a VPC

After you create a set of DHCP options, you can't modify them. If you want your VPC to use a different set of DHCP options, you must create a new set and associate them with your VPC. You can also set up your VPC to use no DHCP options at all.

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. Choose **Your VPCs** in the navigation pane.
3. Select the VPC, and choose **Edit DHCP Options Set** from the **Actions** list.
4. In the **DHCP Options Set** list, select the desired options set from the list, and choose **Save**.

Authentication and Access Control for AWS Directory Service

Access to AWS Directory Service requires credentials that AWS can use to authenticate your requests. Those credentials must have permissions to access AWS resources, such as an AWS Directory Service directory. The following sections provide details on how you can use AWS Identity and Access Management (IAM) and AWS Directory Service to help secure your resources by controlling who can access them:

- Authentication (p. 118)
- Access Control (p. 119)

Authentication

You can access AWS as any of the following types of identities:

- **AWS account root user** – When you sign up for AWS, you provide an email address and password that is associated with your AWS account. These are your *root credentials* and they provide complete access to all of your AWS resources.

  Important
  For security reasons, we recommend that you use the root credentials only to create an *administrator user*, which is an *IAM user* with full permissions to your AWS account. Then, you can use this administrator user to create other IAM users and roles with limited permissions. For more information, see IAM Best Practices and Creating an Admin User and Group in the *IAM User Guide*.
- **IAM user** – An IAM user is simply an identity within your AWS account that has specific custom permissions (for example, permissions to create a directory in AWS Directory Service). You can use an IAM user name and password to sign in to secure AWS webpages like the AWS Management Console, AWS Discussion Forums, or the AWS Support Center.

In addition to a user name and password, you can also generate access keys for each user. You can use these keys when you access AWS services programmatically, either through one of the several SDKs or by using the AWS Command Line Interface (CLI). The SDK and CLI tools use the access keys to cryptographically sign your request. If you don’t use the AWS tools, you must sign the request yourself. AWS Directory Service supports *Signature Version 4*, a protocol for authenticating inbound API requests. For more information about authenticating requests, see Signature Version 4 Signing Process in the *AWS General Reference*.

- **IAM role** – An IAM role is another IAM identity you can create in your account that has specific permissions. It is similar to an *IAM user*, but it is not associated with a specific person. An IAM role enables you to obtain temporary access keys that can be used to access AWS services and resources. IAM roles with temporary credentials are useful in the following situations:

  - **Federated user access** – Instead of creating an IAM user, you can use preexisting user identities from AWS Directory Service, your enterprise user directory, or a web identity provider. These are known as *federated users*. AWS assigns a role to a federated user when access is requested through an identity provider. For more information about federated users, see Federated Users and Roles in the *IAM User Guide*.

  - **Cross-account access** – You can use an IAM role in your account to grant another AWS account permissions to access your account’s resources. For an example, see Tutorial: Delegate Access Across AWS Accounts Using IAM Roles in the *IAM User Guide*.

  - **AWS service access** – You can use an IAM role in your account to grant an AWS service permissions to access your account’s resources. For example, you can create a role that allows Amazon Redshift to access an Amazon S3 bucket on your behalf and then load data stored in the bucket into an Amazon Redshift cluster. For more information, see Creating a Role to Delegate Permissions to an AWS Service in the *IAM User Guide*.

  - **Applications running on Amazon EC2** – Instead of storing access keys within the EC2 instance for use by applications running on the instance and making AWS API requests, you can use an IAM role to manage temporary credentials for these applications. To assign an AWS role to an EC2 instance and make it available to all of its applications, you can create an instance profile that is attached to the instance. An instance profile contains the role and enables programs running on the EC2 instance to get temporary credentials. For more information, see Using Roles for Applications on Amazon EC2 in the *IAM User Guide*.

Access Control

You can have valid credentials to authenticate your requests, but unless you have permissions you cannot create or access AWS Directory Service resources. For example, you must have permissions to create an AWS Directory Service directory or create a directory snapshot.

The following sections describe how to manage permissions for AWS Directory Service. We recommend that you read the overview first.

- Overview of Managing Access Permissions to Your AWS Directory Service Resources (p. 120)

- Using Identity-Based Policies (IAM Policies) for AWS Directory Service (p. 123)

Overview of Managing Access Permissions to Your AWS Directory Service Resources

Every AWS resource is owned by an AWS account, and permissions to create or access the resources are governed by permissions policies. An account administrator can attach permissions policies to IAM identities (that is, users, groups, and roles), and some services (such as AWS Lambda) also support attaching permissions policies to resources.

Note
An *account administrator* (or administrator user) is a user with administrator privileges. For more information, see IAM Best Practices in the *IAM User Guide*.

When granting permissions, you decide who is getting the permissions, the resources they get permissions for, and the specific actions that you want to allow on those resources.

Topics

AWS Directory Service Resources and Operations

In AWS Directory Service, the primary resource is a *directory*. AWS Directory Service supports directory snapshot resources as well. However, you can create snapshots only in the context of an existing directory. Therefore, a snapshot is referred to as a *subresource*.

These resources have unique Amazon Resource Names (ARNs) associated with them as shown in the following table.

Resource Type	ARN Format
Directory	`arn:aws:ds:region:account-id:directory/external-directory-id`
Snapshot	`arn:aws:ds:region:account-id:snapshot/external-snapshot-id`

AWS Directory Service provides a set of operations to work with the appropriate resources. For a list of available operations, see Actions.

Understanding Resource Ownership

A *resource owner* is the AWS account that created a resource. That is, the resource owner is the AWS account of the *principal entity* (the root account, an IAM user, or an IAM role) that authenticates the request that creates the resource. The following examples illustrate how this works:

- If you use the root account credentials of your AWS account to create an AWS Directory Service resource, such as a directory, your AWS account is the owner of that resource.

- If you create an IAM user in your AWS account and grant permissions to create AWS Directory Service resources to that user, the user can also create AWS Directory Service resources. However, your AWS account, to which the user belongs, owns the resources.
- If you create an IAM role in your AWS account with permissions to create AWS Directory Service resources, anyone who can assume the role can create AWS Directory Service resources. Your AWS account, to which the role belongs, owns the AWS Directory Service resources.

Managing Access to Resources

A *permissions policy* describes who has access to what. The following section explains the available options for creating permissions policies.

> **Note**
> This section discusses using IAM in the context of AWS Directory Service. It doesn't provide detailed information about the IAM service. For complete IAM documentation, see What Is IAM? in the *IAM User Guide*. For information about IAM policy syntax and descriptions, see AWS IAM Policy Reference in the *IAM User Guide*.

Policies attached to an IAM identity are referred to as *identity-based* policies (IAM polices) and policies attached to a resource are referred to as *resource-based* policies. AWS Directory Service supports only identity-based policies (IAM policies).

Topics

Identity-Based Policies (IAM Policies)

You can attach policies to IAM identities. For example, you can do the following:

- **Attach a permissions policy to a user or a group in your account** – An account administrator can use a permissions policy that is associated with a particular user to grant permissions for that user to create an AWS Directory Service resource, such as a new directory.
- **Attach a permissions policy to a role (grant cross-account permissions)** – You can attach an identity-based permissions policy to an IAM role to grant cross-account permissions. For example, the administrator in Account A can create a role to grant cross-account permissions to another AWS account (for example, Account B) or an AWS service as follows:

  1. Account A administrator creates an IAM role and attaches a permissions policy to the role that grants permissions on resources in Account A.
  2. Account A administrator attaches a trust policy to the role identifying Account B as the principal who can assume the role.
  3. Account B administrator can then delegate permissions to assume the role to any users in Account B. Doing this allows users in Account B to create or access resources in Account A. The principal in the trust policy can also be an AWS service principal if you want to grant an AWS service permissions to assume the role.

  For more information about using IAM to delegate permissions, see Access Management in the *IAM User Guide*.

The following permissions policy grants permissions to a user to run all of the actions that begin with `Describe`. These actions show information about an AWS Directory Service resource, such as a directory or snapshot. Note that the wildcard character (*) in the `Resource` element indicates that the actions are allowed for all AWS Directory Service resources owned by the account.

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

For more information about using identity-based policies with AWS Directory Service, see Using Identity-Based Policies (IAM Policies) for AWS Directory Service (p. 123). For more information about users, groups, roles, and permissions, see Identities (Users, Groups, and Roles) in the *IAM User Guide*.

Resource-Based Policies

Other services, such as Amazon S3, also support resource-based permissions policies. For example, you can attach a policy to an S3 bucket to manage access permissions to that bucket. AWS Directory Service doesn't support resource-based policies.

Specifying Policy Elements: Actions, Effects, Resources, and Principals

For each AWS Directory Service resource (see AWS Directory Service Resources and Operations (p. 120)), the service defines a set of API operations (see Actions). To grant permissions for these API operations, AWS Directory Service defines a set of actions that you can specify in a policy. Note that, performing an API operation can require permissions for more than one action.

The following are the basic policy elements:

- **Resource** – In a policy, you use an Amazon Resource Name (ARN) to identify the resource to which the policy applies. For AWS Directory Service resources, you always use the wildcard character (*) in IAM policies. For more information, see AWS Directory Service Resources and Operations (p. 120).
- **Action** – You use action keywords to identify resource operations that you want to allow or deny. For example, the `ds:DescribeDirectories` permission allows the user permissions to perform the AWS Directory Service `DescribeDirectories` operation.
- **Effect** – You specify the effect when the user requests the specific action—this can be either allow or deny. If you don't explicitly grant access to (allow) a resource, access is implicitly denied. You can also explicitly deny access to a resource, which you might do to make sure that a user cannot access it, even if a different policy grants access.
- **Principal** – In identity-based policies (IAM policies), the user that the policy is attached to is the implicit principal. For resource-based policies, you specify the user, account, service, or other entity that you want to receive permissions (applies to resource-based policies only). AWS Directory Service doesn't support resource-based policies.

To learn more about IAM policy syntax and descriptions, see AWS IAM Policy Reference in the *IAM User Guide*.

For a table showing all of the AWS Directory Service API actions and the resources that they apply to, see AWS Directory Service API Permissions: Actions, Resources, and Conditions Reference (p. 126).

Specifying Conditions in a Policy

When you grant permissions, you can use the access policy language to specify the conditions when a policy should take effect. For example, you might want a policy to be applied only after a specific date. For more information about specifying conditions in a policy language, see Condition in the *IAM User Guide*.

To express conditions, you use predefined condition keys. There are no condition keys specific to AWS Directory Service. However, there are AWS-wide condition keys that you can use as appropriate. For a complete list of AWS-wide keys, see Available Keys for Conditions in the *IAM User Guide*.

Using Identity-Based Policies (IAM Policies) for AWS Directory Service

This topic provides examples of identity-based policies in which an account administrator can attach permissions policies to IAM identities (that is, users, groups, and roles).

> **Important**
> We recommend that you first review the introductory topics that explain the basic concepts and options available for you to manage access to your AWS Directory Service resources. For more information, see Overview of Managing Access Permissions to Your AWS Directory Service Resources (p. 120).

The sections in this topic cover the following:

- Permissions Required to Use the AWS Directory Service Console (p. 124)
- AWS Managed (Predefined) Policies for AWS Directory Service (p. 125)
- Customer Managed Policy Examples (p. 125)

The following shows an example of a permissions policy.

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

    {
      "Action" : [
        "iam:PassRole",
        "iam:GetRole",
        "iam:CreateRole",
        "iam:PutRolePolicy"
       ],
      "Effect" : "Allow",
      "Resource" : "*"
    },
    {
      "Action" : [
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcs",
        "ec2:CreateSecurityGroup",
        "ec2:DeleteSecurityGroup",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:RevokeSecurityGroupIngress",
        "ec2:RevokeSecurityGroupEgress"
      ],
      "Effect" : "Allow",
      "Resource" : "*"
    }
  ]
}
```

The policy includes the following:

- The first statement grants permission to create a AWS Directory Service directory. AWS Directory Service doesn't support permissions for this particular action at the resource-level. Therefore, the policy specifies a wildcard character (*) as the `Resource` value.
- The second statement grants permissions to certain IAM actions. The access to IAM actions is needed so that AWS Directory Service can read and create IAM roles on your behalf. The wildcard character (*) at the end of the `Resource` value means that the statement allows permission for the IAM actions on any IAM role. To limit this permission to a specific role, replace the wildcard character (*) in the resource ARN with the specific role name. For more information, see IAM Actions.
- The third statement grants permissions to a specific set of Amazon EC2 resources that are necessary to allow AWS Directory Service to create, configure, and destroy its directories. The wildcard character (*) at the end of the `Resource` value means that the statement allows permission for the EC2 actions on any EC2 resource or subresource. To limit this permission to a specific role, replace the wildcard character (*) in the resource ARN with the specific resource or subresource. For more information, see EC2 Actions

The policy doesn't specify the `Principal` element because in an identity-based policy you don't specify the principal who gets the permission. When you attach policy to a user, the user is the implicit principal. When you attach a permission policy to an IAM role, the principal identified in the role's trust policy gets the permissions.

For a table showing all of the AWS Directory Service API actions and the resources that they apply to, see AWS Directory Service API Permissions: Actions, Resources, and Conditions Reference (p. 126).

Permissions Required to Use the AWS Directory Service Console

For a user to work with the AWS Directory Service console, that user must have permissions listed in the policy above or the permissions granted by the Directory Service Full Access Role or Directory Service Read Only role, described in AWS Managed (Predefined) Policies for AWS Directory Service (p. 125).

If you create an IAM policy that is more restrictive than the minimum required permissions, the console won't function as intended for users with that IAM policy.

AWS Managed (Predefined) Policies for AWS Directory Service

AWS addresses many common use cases by providing standalone IAM policies that are created and administered by AWS. Managed policies grant necessary permissions for common use cases so you can avoid having to investigate what permissions are needed. For more information, see AWS Managed Policies in the *IAM User Guide*.

The following AWS managed policies, which you can attach to users in your account, are specific to AWS Directory Service:

- **AmazonDirectoryServiceReadOnlyAccess** – Grants a user or group read-only access to all AWS Directory Service resources, EC2 subnets, EC2 network interfaces, and AWS SNS topics and subscriptions for the root AWS account.
- **AWSDirectoryServiceFullAccess** – Grants a user or group the following:
  - Full access to AWS Directory Service
  - Access to key Amazon EC2 services required to use AWS Directory Service
  - Ability to list Amazon Simple Notification Service topics
  - Ability to create, manage, and delete Amazon Simple Notification Service topics with a name beginning with "DirectoryMonitoring"

In addition, there are other AWS-managed policies that are suitable for use with other IAM roles. These policies are assigned to the roles associated with users in your AWS Directory Service directory and are required in order for those users to have access to other AWS resources, such as Amazon EC2. For more information, see Managing IAM Roles and Policies (p. 79).

You can also create custom IAM policies that allow users to access the required Amazon AWS Directory Service API actions and resources. You can attach these custom policies to the IAM users or groups that require those permissions. For more information on these polices, see Directory Service Read Only Access (p. 88) and Directory Service Full Access (p. 87).

Customer Managed Policy Examples

In this section, you can find example user policies that grant permissions for various AWS Directory Service actions.

> **Note**
> All examples use the US West (Oregon) Region (`us-west-2`) and contain fictitious account IDs.

Examples

Example 1: Allow a User to Perform Any Describe Action on Any AWS Directory Service Resource

The following permissions policy grants permissions to a user to run all of the actions that begin with `Describe`. These actions show information about an AWS Directory Service resource, such as a directory or snapshot. Note that the wildcard character (*) in the `Resource` element indicates that the actions are allowed for all AWS Directory Service resources owned by the account.

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

Example 2: Allow a User to Create a Directory

The following permissions policy grants permissions to allow a user to create a directory and all other related resources, such as snapshots and trusts. In order to do so, permissions to certain Amazon EC2 services are also required.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action": [
                "ds:Create*",
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CreateNetworkInterface",
                "ec2:CreateSecurityGroup",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteSecurityGroup",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:RevokeSecurityGroupEgress",
                "ec2:RevokeSecurityGroupIngress"
                  ],
         "Resource":"*"
         ]
      }
   ]
}
```

AWS Directory Service API Permissions: Actions, Resources, and Conditions Reference

When you are setting up Access Control (p. 119) and writing permissions policies that you can attach to an IAM identity (identity-based policies), you can use the following table as a reference. The list includes each AWS Directory Service API operation, the corresponding actions for which you can grant permissions to perform the action, the AWS resource for which you can grant the permissions. You specify the actions in the policy's `Action` field and the resource value in the policy's `Resource` field.

You can use AWS-wide condition keys in your AWS Directory Service policies to express conditions. For a complete list of AWS-wide keys, see Available Keys in the *IAM User Guide*.

Note
To specify an action, use the `ds:` prefix followed by the API operation name (for example, `ds:CreateDirectory`).

Related Topics

- Access Control (p. 119)

Troubleshooting AWS Directory Service Administration Issues

The following can help you troubleshoot some common issues you might encounter when creating or using your directory.

Simple AD

Here are some common problems with Simple AD.

I am not able to update the DNS name or IP address of an instance joined to my domain (DNS dynamic update)

DNS dynamic updates are not supported in Simple AD domains. You can instead make the changes directly by connecting to your directory using DNS Manager on an instance that is joined to your domain.

I cannot log onto SQL Server using a SQL Server account

You might receive an error if you attempt to use SQL Server Management Studio (SSMS) with a SQL Server account to log into SQL Server running on a Windows 2012 R2 EC2 instance or in Amazon RDS. The issue occurs when SSMS is run as a domain user and can result in the error "Login failed for user," even when valid credentials are provided. This is a known issue and AWS is actively working to resolve it.

To work around the issue, you can log into SQL Server with Windows Authentication instead of SQL Authentication. Or launch SSMS as a local user instead of a Simple AD domain user.

My directory is stuck in the "Requested" state

If you have a directory that has been in the "Requested" state for more than five minutes, try deleting the directory and recreating it. If this problem persists, contact the AWS Support Center.

I receive an "AZ Constrained" error when I create a directory

Some AWS accounts created before 2012 might have access to Availability Zones in the US East (N. Virginia), US West (N. California), or Asia Pacific (Tokyo) region that do not support AWS Directory Service directories. If you receive an error such as this when creating a directory, choose a subnet in a different Availability Zone and try to create the directory again.

Some of my users cannot authenticate with my directory

Your user accounts must have Kerberos preauthentication enabled. This is the default setting for new user accounts, and it should not be modified. For more information about this setting, go to Preauthentication on Microsoft TechNet.

AD Connector

Here are some common problems with AD Connector.

I receive a "DNS unavailable" error when I try to connect to my on-premises directory

You receive an error message similar to the following when connecting to your on-premises directory:

```
DNS unavailable (TCP port 53) for IP: <DNS IP address>
```

AD Connector must be able to communicate with your on-premises DNS servers via TCP and UDP over port 53. Verify that your security groups and on-premises firewalls allow TCP and UDP communication over this port. For more information, see AD Connector Prerequisites (p. 47).

I receive a "Connectivity issues detected" error when I try to connect to my on-premises directory

You receive an error message similar to the following when connecting to your on-premises directory:

```
Connectivity issues detected: LDAP unavailable (TCP port 389) for IP: <IP
 address>
Kerberos/authentication unavailable (TCP port 88) for IP: <IP address>
Please ensure that the listed ports are available and retry the operation.
```

AD Connector must be able to communicate with your on-premises domain controllers via TCP and UDP over the following ports. Verify that your security groups and on-premises firewalls allow TCP and UDP communication over these ports. For more information, see AD Connector Prerequisites (p. 47).

- 88 (Kerberos)
- 389 (LDAP)

I receive an "SRV record" error when I try to connect to my on-premises directory

You receive an error message similar to one or more of the following when connecting to your on-premises directory:

```
SRV record for LDAP does not exist for IP: <DNS IP address>

SRV record for Kerberos does not exist for IP: <DNS IP address>
```

AD Connector needs to obtain the `_ldap._tcp.<DnsDomainName>` and `_kerberos._tcp.<DnsDomainName>` SRV records when connecting to your directory. You will get this error if the service cannot obtain these records from the DNS servers that you specified when connecting to your directory. For more information about these SRV records, see SRV record requirements (p. 48).

My directory is stuck in the "Requested" state

If you have a directory that has been in the "Requested" state for more than five minutes, try deleting the directory and recreating it. If this problem persists, contact the AWS Support Center.

I receive an "AZ Constrained" error when I create a directory

Some AWS accounts created before 2012 might have access to Availability Zones in the US East (N. Virginia), US West (N. California), or Asia Pacific (Tokyo) region that do not support AWS Directory Service directories. If you receive an error such as this when creating a directory, choose a subnet in a different Availability Zone and try to create the directory again.

Some of my users cannot authenticate with my directory

Your user accounts must have Kerberos preauthentication enabled. This is the default setting for new user accounts, but it should not be modified. For more information about this setting, go to Preauthentication on Microsoft TechNet.

I receive an "Invalid Credentials" error when the service account used by AD Connector attempts to authenticate

This can occur if the hard drive on your domain controller runs out of space. Ensure that your domain controller's hard drives are not full.

AWS Directory Service for Microsoft Active Directory (Enterprise Edition)

Here are some common problems with Microsoft AD.

I get an error when creating multiple password policies (fine-grained password policies) for my Microsoft AD directory

AWS Directory Service for Microsoft Active Directory (Enterprise Edition) does not support fine grained password policies.

Topics

Directory Status

The following are the various statuses for a directory. For more information, see Directory Status Reasons (p. 132).

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

Creating
The directory is currently being created. Directory creation typically take 5 to 10 minutes but may vary depending on the system load.

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

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

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

Impaired
The directory is running in a degraded state. One or more issues have been detected, and not all directory operations may be working at full operational capacity.

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

Requested
A request to create your directory is currently pending.

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

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

Directory Status Reasons

When a directory is impaired or inoperable, the directory status message contains additional information. The status message is displayed in the AWS Directory Service console, or returned in the `DirectoryDescription.StageReason` member by the `DescribeDirectories` API. For more information about the directory status, see Directory Status (p. 131).

The following are the status messages for a Simple AD directory:

Topics

The directory service's elastic network interface is not attached

Description

The critical elastic network interface (ENI) that was created on your behalf during directory creation to establish network connectivity with your VPC is not attached to the directory instance. AWS applications backed by this directory will not be functional. Your directory cannot connect to your on-premises network.

Troubleshooting

If the ENI is detached but still exists, contact AWS Support. If the ENI is deleted, there is no way to resolve the issue and your directory is permanently unusable. You must delete the directory and create a new one.

Issue(s) detected by instance

Description

An internal error was detected by the instance. This usually signifies that the monitoring service is actively attempting to recover the impaired instances.

Troubleshooting

In most cases, this is a transient issue, and the directory eventually returns to the Active state. If the problem persists, contact AWS Support for more assistance.

The critical AWS Directory Service reserved user is missing from the directory

Description

When a Simple AD is created, AWS Directory Service creates a service account in the directory with the name `AWSAdmin`*`D-xxxxxxxxx`*. This error is received when this service account cannot be found. Without this account, AWS Directory Service cannot perform administrative functions on the directory, rendering the directory unusable.

Troubleshooting

To correct this issue, restore the directory to a previous snapshot that was created before the service account was deleted. Automatic snapshots are taken of your Simple AD directory one time a day. If it has been more than five days after this account was deleted, you may not be able to restore the directory to a state where this account exists. If you are not able to restore the directory from a snapshot where this account exists, your directory may become permanently unusable. If this is the case, you must delete your directory and create a new one.

The critical AWS Directory Service reserved user needs to belong to the Domain Admins AD group

Description

When a Simple AD is created, AWS Directory Service creates a service account in the directory with the name `AWSAdmin`*`D-xxxxxxxxxx`*. This error is received when this service account is not a member of the `Domain Admins` group. Membership in this group is needed to give AWS Directory Service the privileges it needs to perform maintenance and recovery operations, such as transferring FSMO roles, domain joining new directory controllers, and restoring from snapshots.

Troubleshooting

Use the Active Directory Users and Computers tool to re-add the service account to the `Domain Admins` group.

The critical AWS Directory Service reserved user is disabled

Description

When a Simple AD is created, AWS Directory Service creates a service account in the directory with the name `AWSAdmin`*`D-xxxxxxxxxx`*. This error is received when this service account is disabled. This account must be enabled so that AWS Directory Service can perform maintenance and recovery operations on the directory.

Troubleshooting

Use the Active Directory Users and Computers tool to re-enable the service account.

The main domain controller does not have all FSMO roles

Description

All the FSMO roles are not owned by the Simple AD directory controller. AWS Directory Service cannot guarantee certain behavior and functionality if the FSMO roles do not belong to the correct Simple AD directory controller.

Troubleshooting

Use Active Directory tools to move the FSMO roles back to the original working directory controller. For more information about moving the FSMO roles, go to https://support.microsoft.com/en-us/kb/324801. If this does not correct the problem, please contact AWS Support for more assistance.

Domain controller replication failures

Description

The Simple AD directory controllers are failing to replicate with one another. This can be caused by one or more of the following issues:

- The security groups for the directory controllers does not have the correct ports open.
- The network ACLs are too restrictive.
- The VPC route table is not routing network traffic between the directory controllers correctly.
- Another instance has been promoted to a domain controller in the directory.

Troubleshooting

For more information about your VPC network requirements, see either Microsoft AD Microsoft AD Prerequisites (p. 10), AD Connector AD Connector Prerequisites (p. 47), or Simple AD Simple AD Prerequisites (p. 61). If there is an unknown domain controller in your directory, you must demote it. If your VPC network setup is correct, but the error persists, please contact AWS Support for more assistance.

Schema Extension Errors

The following can help you troubleshoot some error messages you might encounter when extending the schema for your Microsoft AD directory.

Referral

Error

Add error on entry starting on line 1: Referral The server side error is: 0x202b A referral was returned from the server. The extended server error is: 0000202B: RefErr: DSID-0310082F, data 0, 1 access points \tref 1: ‘example.com’ Number of Objects Modified: 0

Troubleshooting

Ensure that all of the distinguished name fields have the correct domain name. In the example above, `DC=example,dc=com` should be replaced with the `DistinguishedName` shown by the cmdlet `Get-ADDomain`.

Unable to Read Import File

Error

Unable to read the import file. Number of Objects Modified: 0

Troubleshooting

The imported LDIF file is empty (0 bytes). Ensure the correct file was uploaded.

Syntax Error

Error

There is a syntax error in the input file Failed on line 21. The last token starts with 'q'. Number of Objects Modified: 0

Troubleshooting

The text on line 21 is not formatted correctly. The first letter of the invalid text is `A`. Update line 21 with valid LDIF syntax. For more information about how to format the LDIF file, see Step 1: Create Your LDIF File (p. 14).

Attribute or Value Exists

Error

Add error on entry starting on line 1: Attribute Or Value Exists The server side error is: 0x2083 The specified value already exists. The extended server error is: 00002083: AtrErr: DSID-03151830, #1: \t0: 00002083: DSID-03151830, problem 1006 (ATT_OR_VALUE_EXISTS), data 0, Att 20019 (mayContain):len 4 Number of Objects Modified: 0

Troubleshooting

The schema change has already been applied.

No Such Attribute

Error

Add error on entry starting on line 1: No Such Attribute The server side error is: 0x2085 The attribute value cannot be removed because it is not present on the object. The extended server error is: 00002085: AtrErr: DSID-03152367, #1: \t0: 00002085: DSID-03152367, problem 1001 (NO_ATTRIBUTE_OR_VAL), data 0, Att 20019 (mayContain):len 4 Number of Objects Modified: 0

Troubleshooting

The LDIF file is trying to remove an attribute from a class, but that attribute is currently not attached to the class. Schema change was probably already applied.

Error

Add error on entry starting on line 41: No Such Attribute 0x57 The parameter is incorrect. The extended server error is: 0x208d Directory object not found. The extended server error is: "00000057: LdapErr: DSID-0C090D8A, comment: Error in attribute conversion operation, data 0, v2580" Number of Objects Modified: 0

Troubleshooting

The attribute listed on line 41 is incorrect. Double-check the spelling.

No Such Object

Error

Add error on entry starting on line 1: No Such Object The server side error is: 0x208d Directory object not found. The extended server error is: 0000208D: NameErr: DSID-03100238, problem 2001 (NO_OBJECT), data 0, best match of: 'CN=Schema,CN=Configuration,DC=example,DC=com' Number of Objects Modified: 0

Troubleshooting

The object referenced by the distinguished name (DN) does not exist.

Trust Creation Status Reasons

When trust creation fails, the status message contains additional information. Here's some help understanding what those messages mean.

Access is denied

Access was denied when trying to create the trust. Either the trust password is incorrect or the remote domain's security settings do not allow a trust to be configured. Ensure that you are using the same trust password that you used when creating the corresponding trust on the remote domain. Also confirm that your domain security settings allow for trust creation.

The specified domain name does not exist or could not be contacted

To solve this problem, ensure the security group settings for your domain and access control list (ACL) for your VPC are correct and you have accurately entered the information for your conditional forwarder. For more information about security requirements, see When to Create a Trust Relationship (p. 18).

If this does not solve the issue, it is possible that information for a previously created conditional forwarder has been cached, preventing the creation of a new trust. Please wait several minutes and then try creating the trust and conditional forwarder again.

AWS Directory Service Limits

The following are the default limits for AWS Directory Service. Each limit is per region unless otherwise noted.

Simple AD

AWS Directory Service Limits

Resource	Default Limit
Simple AD directories	10
Manual snapshots	5 per Simple AD

AWS Directory Service for Microsoft Active Directory (Enterprise Edition)

AWS Directory Service Limits

Resource	Default Limit
Microsoft AD directories	10
Manual snapshots	5 per Microsoft AD

AD Connector

AWS Directory Service Limits

Resource	Default Limit
AD Connector directories	10

Increase Your Limit

Perform the following steps to increase your limit for a region.

To request a limit increase for a region

1. Go to the AWS Support Center page, sign in, if necessary, and click **Open a new case**.
2. Under **Regarding**, select **Service Limit Increase**.
3. Under **Limit Type**, select **AWS Directory Service**.
4. Fill in all of the necessary fields in the form and click the button at the bottom of the page for your desired method of contact.

Document History

The following table describes the important changes since the last release of the *AWS Directory Service Administrator Guide*.

- **Latest documentation update:** November 14, 2016

Change	Description	Date Changed
Schema Extensions	Added documentation for schema extensions with AWS Directory Service for Microsoft Active Directory (Enterprise Edition). For more information, see Schema Extensions (p. 12).	November 14, 2016
Major reorganization of the Directory Service Admin Guide	Reorganized the content to more directly map to customer needs.	November 14, 2016
Authorization and Authentication	Added additional documentation for using IAM with Directory Services.	February 25, 2016
SNS notifications	Added documentation for SNS notifications. For more information, see Get Notified of Directory Status Updates Using Amazon SNS (p. 67).	February 25, 2016
Microsoft AD	Added documentation for Microsoft AD, and combined guides into a single guide.	November 17, 2015
Allow Linux instances to be joined to a Simple AD directory	Added documentation for joining a Linux instance to a Simple AD directory. For more information, see Manually Add a Linux Instance (Simple AD and Microsoft AD) (p. 108).	July 23, 2015
Guide separation	The *AWS Directory Service administration guide* is split into separate guides, the *Simple AD guide* and the *AD Connector guide*.	July 14, 2015
Single sign-on support	Added single sign-on documentation. For more information, see Single Sign-On (p. 71).	March 31, 2015

Change	Description	Date Changed
New guide	This is the first release of the *AWS Directory Service Administration Guide*.	October 21, 2014