Contents

What is Azure Active Directory Domain Services?

7/20/2020 • 8 minutes to read • Edit Online

Azure Active Directory Domain Services (Azure AD DS) provides managed domain services such as domain join, group policy, lightweight directory access protocol (LDAP), and Kerberos / NTLM authentication. You use these domain services without the need to deploy, manage, and patch domain controllers (DCs) in the cloud.

A managed domain is a DNS namespace and matching directory. The managed domain integrates with your existing Azure AD tenant, which makes it possible for users to sign in using their existing credentials. You can also use existing groups and user accounts to secure access to resources, which provides a smoother lift-and-shift of on-premises resources to Azure.

Azure AD DS integrates with your existing Azure AD tenant. This integration lets users sign in to service and applications connected to the managed domain using their existing credentials. You can also use existing groups and user accounts to secure access to resources. These features provide a smoother lift-and-shift of on-premises resources to Azure.

To get started, create an Azure AD DS managed domain using the Azure portal

Azure AD DS replicates identity information from Azure AD, so it works with Azure AD tenants that are cloud-only, or synchronized with an on-premises Active Directory Domain Services (AD DS) environment. The same set of Azure AD DS features exists for both environments.

- If you have an existing on-premises AD DS environment, you can synchronize user account information to provide a consistent identity for users. To learn more, see How objects and credentials are synchronized in a managed domain.
- For cloud-only environments, you don't need a traditional on-premises AD DS environment to use the centralized identity services of Azure AD DS.

To learn how to administrator a managed domain, see management concepts for user accounts, passwords, and administration in Azure AD DS.

The following video provides an overview of how Azure AD DS integrates with your applications and workloads to provide identity services in the cloud:

Common ways to provide identity solutions in the cloud

When you migrate existing workloads to the cloud, directory-aware applications may use LDAP for read or write access to an on-premises AD DS directory. Applications that run on Windows Server are typically deployed on domain-joined virtual machines (VMs) so they can be managed securely using Group Policy. To authenticate end users, the applications may also rely on Windows-integrated authentication, such as Kerberos or NTLM authentication.

IT administrators often use one of the following solutions to provide an identity service to applications that run in Azure:

- Configure a site-to-site VPN connection between workloads that run in Azure and an on-premises AD DS environment.
  - The on-premises domain controllers then provide authentication via the VPN connection.
- Create replica domain controllers using Azure virtual machines (VMs) to extend the AD DS domain / forest from

on-premises.
    - The domain controllers that run on Azure VMs provide authentication, and replicate directory information between the on-premises AD DS environment.
- Deploy a standalone AD DS environment in Azure using domain controllers that run on Azure VMs.
    - The domain controllers that run on Azure VMs provide authentication, but there's no directory information replicated from an on-premises AD DS environment.

With these approaches, VPN connections to the on-premises directory make applications vulnerable to transient network glitches or outages. If you deploy domain controllers using VMs in Azure, the IT team must manage the VMs, then secure, patch, monitor, backup, and troubleshoot them.

Azure AD DS offers alternatives to the need to create VPN connections back to an on-premises AD DS environment or run and manage VMs in Azure to provide identity services. As a managed service, Azure AD DS reduces the complexity to create an integrated identity solution for both hybrid and cloud-only environments.

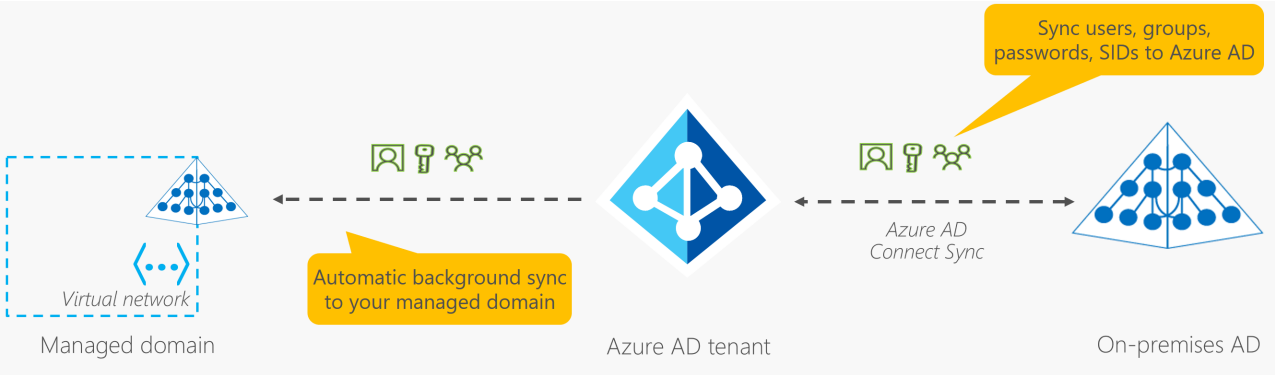
Compare Azure AD DS with Azure AD and self-managed AD DS on Azure VMs or on-premises

How does Azure AD DS work?

To provide identity services, Azure creates an AD DS managed domain on a virtual network of your choice. Behind the scenes, a pair of Windows Server domain controllers is created that run on Azure VMs. You don't need to manage, configure, or update these domain controllers. The Azure platform manages the domain controllers as part of the Azure AD DS service.

The managed domain is configured to perform a one-way synchronization from Azure AD to provide access to a central set of users, groups, and credentials. You can create resources directly in the managed domain, but they aren't synchronized back to Azure AD. Applications, services, and VMs in Azure that connect to this virtual network can then use common AD DS features such as domain join, group policy, LDAP, and Kerberos / NTLM authentication.

In a hybrid environment with an on-premises AD DS environment, Azure AD Connect synchronizes identity information with Azure AD, which is then synchronized to Azure AD DS.



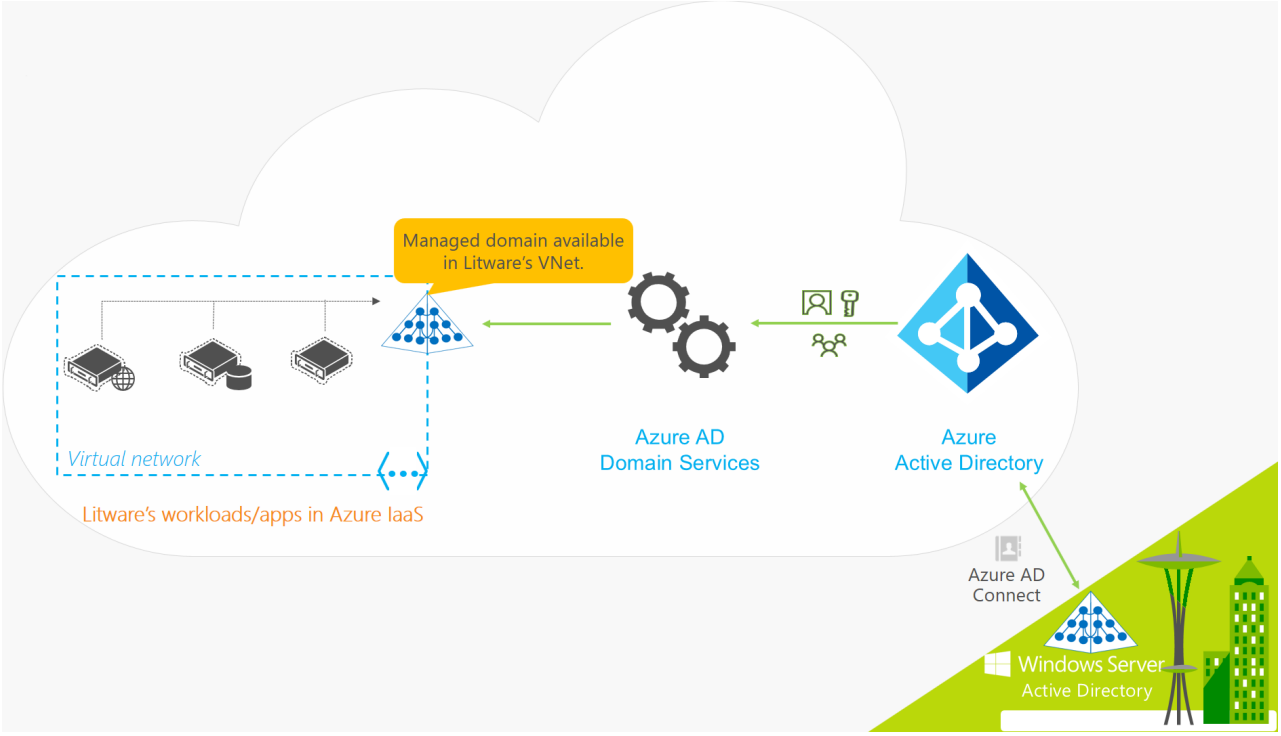
To see Azure AD DS in action, let's look at a couple of examples:

- Azure AD DS for hybrid organizations
- Azure AD DS for cloud-only organizations

Azure AD DS for hybrid organizations

Many organizations run a hybrid infrastructure that includes both cloud and on-premises application workloads. Legacy applications migrated to Azure as part of a lift and shift strategy may use traditional LDAP connections to provide identity information. To support this hybrid infrastructure, identity information from an on-premises AD DS environment can be synchronized to an Azure AD tenant. Azure AD DS then provides these legacy applications in Azure with an identity source, without the need to configure and manage application connectivity back to on-premises directory services.

Let's look at an example for Litware Corporation, a hybrid organization that runs both on-premises and Azure resources:



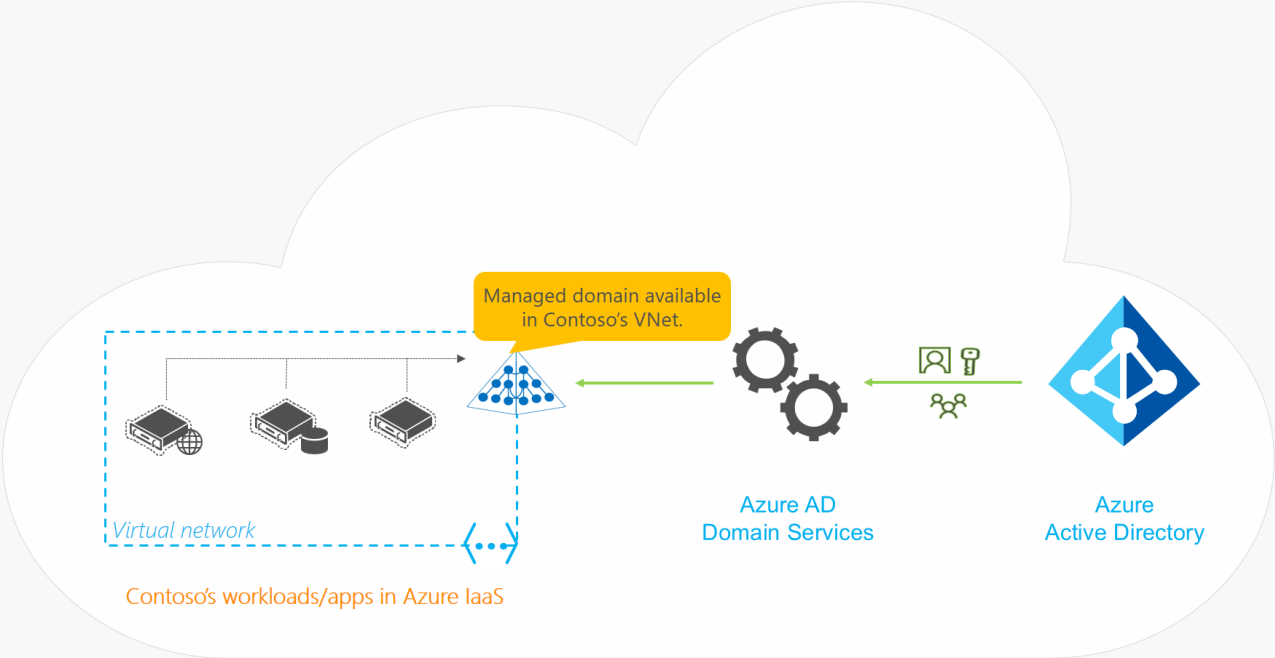
- Applications and server workloads that require domain services are deployed in a virtual network in Azure.
  - This may include legacy applications migrated to Azure as part of a lift and shift strategy.
- To synchronize identity information from their on-premises directory to their Azure AD tenant, Litware Corporation deploys Azure AD Connect.
  - Identity information that is synchronized includes user accounts and group memberships.
- Litware's IT team enables Azure AD DS for their Azure AD tenant in this, or a peered, virtual network.
- Applications and VMs deployed in the Azure virtual network can then use Azure AD DS features like domain join, LDAP read, LDAP bind, NTLM and Kerberos authentication, and Group Policy.

> **IMPORTANT**
> Azure AD Connect should only be installed and configured for synchronization with on-premises AD DS environments. It's not supported to install Azure AD Connect in a managed domain to synchronize objects back to Azure AD.

Azure AD DS for cloud-only organizations

A cloud-only Azure AD tenant doesn't have an on-premises identity source. User accounts and group memberships, for example, are created and managed directly in Azure AD.

Now let's look at an example for Contoso, a cloud-only organization that uses Azure AD for identity. All user identities, their credentials, and group memberships are created and managed in Azure AD. There is no additional configuration of Azure AD Connect to synchronize any identity information from an on-premises directory.



- Applications and server workloads that require domain services are deployed in a virtual network in Azure.
- Contoso's IT team enables Azure AD DS for their Azure AD tenant in this, or a peered, virtual network.
- Applications and VMs deployed in the Azure virtual network can then use Azure AD DS features like domain join, LDAP read, LDAP bind, NTLM and Kerberos authentication, and Group Policy.

Azure AD DS features and benefits

To provide identity services to applications and VMs in the cloud, Azure AD DS is fully compatible with a traditional AD DS environment for operations such as domain-join, secure LDAP (LDAPS), Group Policy, DNS management, and LDAP bind and read support. LDAP write support is available for objects created in the Azure AD DS managed domain, but not resources synchronized from Azure AD.

To learn more about your identity options, compare Azure AD DS with Azure AD, Active Directory Domain Services on Azure VMs, and Active Directory Domain Services on-premises.

The following features of Azure AD DS simplify deployment and management operations:

- **Simplified deployment experience:** Azure AD DS is enabled for your Azure AD tenant using a single wizard in the Azure portal.
- **Integrated with Azure AD:** User accounts, group memberships, and credentials are automatically available from your Azure AD tenant. New users, groups, or changes to attributes from your Azure AD tenant or your on-premises AD DS environment are automatically synchronized to Azure AD DS.
  - Accounts in external directories linked to your Azure AD aren't available in Azure AD DS. Credentials aren't available for those external directories, so can't be synchronized into an Azure AD DS managed domain.
- **Use your corporate credentials/passwords:** Passwords for users in Azure AD DS are the same as in your Azure AD tenant. Users can use their corporate credentials to domain-join machines, sign in interactively or over remote desktop, and authenticate against the Azure AD DS managed domain.
- **NTLM and Kerberos authentication:** With support for NTLM and Kerberos authentication, you can deploy applications that rely on Windows-integrated authentication.
- **High availability:** Azure AD DS includes multiple domain controllers, which provide high availability for your managed domain. This high availability guarantees service uptime and resilience to failures.
  - In regions that support Azure Availability Zones, these domain controllers are also distributed across zones for additional resiliency.

Some key aspects of an Azure AD DS managed domain include the following:

- The Azure AD DS managed domain is a stand-alone domain. It isn't an extension of an on-premises domain.
  - If needed, you can create one-way outbound forest trusts from Azure AD DS to an on-premises AD DS environment. For more information, see Resource forest concepts and features for Azure AD DS.
- Your IT team doesn't need to manage, patch, or monitor domain controllers for this Azure AD DS managed domain.

For hybrid environments that run AD DS on-premises, you don't need to manage AD replication to the Azure AD DS managed domain. User accounts, group memberships, and credentials from your on-premises directory are synchronized to Azure AD via Azure AD Connect. These user accounts, group memberships, and credentials are automatically available within the Azure AD DS managed domain.

Next steps

To learn more about Azure AD DS compares with other identity solutions and how synchronization works, see the following articles:

- Compare Azure AD DS with Azure AD, Active Directory Domain Services on Azure VMs, and Active Directory Domain Services on-premises
- Learn how Azure AD Domain Services synchronizes with your Azure AD directory

To get started, create a managed domain using the Azure portal.

Compare self-managed Active Directory Domain Services, Azure Active Directory, and managed Azure Active Directory Domain Services

7/20/2020 • 7 minutes to read • Edit Online

To provide applications, services, or devices access to a central identity, there are three common ways to use Active Directory-based services in Azure. This choice in identity solutions gives you the flexibility to use the most appropriate directory for your organization's needs. For example, if you mostly manage cloud-only users that run mobile devices, it may not make sense to build and run your own Active Directory Domain Services (AD DS) identity solution. Instead, you could just use Azure Active Directory.

Although the three Active Directory-based identity solutions share a common name and technology, they're designed to provide services that meet different customer demands. At high level, these identity solutions and feature sets are:

- **Active Directory Domain Services (AD DS)** - Enterprise-ready lightweight directory access protocol (LDAP) server that provides key features such as identity and authentication, computer object management, group policy, and trusts.
  - AD DS is a central component in many organizations with an on-premises IT environment, and provides core user account authentication and computer management features.
  - For more information, see Active Directory Domain Services overview in the Windows Server documentation.
- **Azure Active Directory (Azure AD)** - Cloud-based identity and mobile device management that provides user account and authentication services for resources such as Office 365, the Azure portal, or SaaS applications.
  - Azure AD can be synchronized with an on-premises AD DS environment to provide a single identity to users that works natively in the cloud.
  - For more information about Azure AD, see What is Azure Active Directory?
- **Azure Active Directory Domain Services (Azure AD DS)** - Provides managed domain services with a subset of fully-compatible traditional AD DS features such as domain join, group policy, LDAP, and Kerberos / NTLM authentication.
  - Azure AD DS integrates with Azure AD, which itself can synchronize with an on-premises AD DS environment. This ability extends central identity use cases to traditional web applications that run in Azure as part of a lift-and-shift strategy.
  - To learn more about synchronization with Azure AD and on-premises, see How objects and credentials are synchronized in a managed domain.

This overview article compares and contrasts how these identity solutions can work together, or would be used independently, depending on the needs of your organization.

To get started, create an Azure AD DS managed domain using the Azure portal

Azure AD DS and self-managed AD DS

If you have applications and services that need access to traditional authentication mechanisms such as Kerberos or NTLM, there are two ways to provide Active Directory Domain Services in the cloud:

- A *managed domain* that you create using Azure Active Directory Domain Services (Azure AD DS). Microsoft

creates and manages the required resources.

- A *self-managed* domain that you create and configure using traditional resources such as virtual machines (VMs), Windows Server guest OS, and Active Directory Domain Services (AD DS). You then continue to administer these resources.

With Azure AD DS, the core service components are deployed and maintained for you by Microsoft as a *managed* domain experience. You don't deploy, manage, patch, and secure the AD DS infrastructure for components like the VMs, Windows Server OS, or domain controllers (DCs).

Azure AD DS provides a smaller subset of features to traditional self-managed AD DS environment, which reduces some of the design and management complexity. For example, there are no AD forests, domain, sites, and replication links to design and maintain. You can still create forest trusts between Azure AD DS and on-premises environments (currently in preview).

For applications and services that run in the cloud and need access to traditional authentication mechanisms such as Kerberos or NTLM, Azure AD DS provides a managed domain experience with the minimal amount of administrative overhead. For more information, see Management concepts for user accounts, passwords, and administration in Azure AD DS.

When you deploy and run a self-managed AD DS environment, you have to maintain all of the associated infrastructure and directory components. There's additional maintenance overhead with a self-managed AD DS environment, but you're then able to do additional tasks such as extend the schema or create forest trusts.

Common deployment models for a self-managed AD DS environment that provides identity to applications and services in the cloud include the following:

- **Standalone cloud-only AD DS** - Azure VMs are configured as domain controllers and a separate, cloud-only AD DS environment is created. This AD DS environment doesn't integrate with an on-premises AD DS environment. A different set of credentials is used to sign in and administer VMs in the cloud.
- **Resource forest deployment** - Azure VMs are configured as domain controllers and an AD DS domain that's part of an existing forest is created. A trust relationship is then configured to an on-premises AD DS environment. Other Azure VMs can domain-join to this resource forest in the cloud. User authentication runs over a VPN / ExpressRoute connection to the on-premises AD DS environment.
- **Extend on-premises domain to Azure** - An Azure virtual network connects to an on-premises network using a VPN / ExpressRoute connection. Azure VMs connect to this Azure virtual network, which lets them domain-join to the on-premises AD DS environment.
  - An alternative is to create Azure VMs and promote them as replica domain controllers from the on-premises AD DS domain. These domain controllers replicate over a VPN / ExpressRoute connection to the on-premises AD DS environment. The on-premises AD DS domain is effectively extended into Azure.

The following table outlines some of the features you may need for your organization, and the differences between a managed Azure AD DS domain or a self-managed AD DS domain:

FEATURE	AZURE AD DS	SELF-MANAGED AD DS
Managed service	✓	✕
Secure deployments	✓	Administrator secures the deployment
DNS server	✓ (managed service)	✓
Domain or Enterprise administrator privileges	✕	✓

FEATURE	AZURE AD DS	SELF-MANAGED AD DS
Domain join	✓	✓
Domain authentication using NTLM and Kerberos	✓	✓
Kerberos constrained delegation	Resource-based	Resource-based & account-based
Custom OU structure	✓	✓
Group Policy	✓	✓
Schema extensions	✕	✓
AD domain / forest trusts	✓ (one-way outbound forest trusts only)	✓
Secure LDAP (LDAPS)	✓	✓
LDAP read	✓	✓
LDAP write	✓ (within the managed domain)	✓
Geo-distributed deployments	✕	✓

Azure AD DS and Azure AD

Azure AD lets you manage the identity of devices used by the organization and control access to corporate resources from those devices. Users can also register their personal device (a bring-your-own (BYO) model) with Azure AD, which provides the device with an identity. Azure AD then authenticates the device when a user signs in to Azure AD and uses the device to access secured resources. The device can be managed using Mobile Device Management (MDM) software like Microsoft Intune. This management ability lets you restrict access to sensitive resources to managed and policy-compliant devices.

Traditional computers and laptops can also join to Azure AD. This mechanism offers the same benefits of registering a personal device with Azure AD, such as to allow users to sign in to the device using their corporate credentials.

Azure AD joined devices give you the following benefits:

- Single-sign-on (SSO) to applications secured by Azure AD.
- Enterprise policy-compliant roaming of user settings across devices.
- Access to the Windows Store for Business using corporate credentials.
- Windows Hello for Business.
- Restricted access to apps and resources from devices compliant with corporate policy.

Devices can be joined to Azure AD with or without a hybrid deployment that includes an on-premises AD DS environment. The following table outlines common device ownership models and how they would typically be joined to a domain:

TYPE OF DEVICE	DEVICE PLATFORMS	MECHANISM
Personal devices	Windows 10, iOS, Android, macOS	Azure AD registered
Organization-owned device not joined to on-premises AD DS	Windows 10	Azure AD joined
Organization-owned device joined to an on-premises AD DS	Windows 10	Hybrid Azure AD joined

On an Azure AD-joined or registered device, user authentication happens using modern OAuth / OpenID Connect based protocols. These protocols are designed to work over the internet, so are great for mobile scenarios where users access corporate resources from anywhere.

With Azure AD DS-joined devices, applications can use the Kerberos and NTLM protocols for authentication, so can support legacy applications migrated to run on Azure VMs as part of a lift-and-shift strategy. The following table outlines differences in how the devices are represented and can authenticate themselves against the directory:

ASPECT	AZURE AD-JOINED	AZURE AD DS-JOINED
Device controlled by	Azure AD	Azure AD DS managed domain
Representation in the directory	Device objects in the Azure AD directory	Computer objects in the Azure AD DS managed domain
Authentication	OAuth / OpenID Connect based protocols	Kerberos and NTLM protocols
Management	Mobile Device Management (MDM) software like Intune	Group Policy
Networking	Works over the internet	Must be connected to, or peered with, the virtual network where the managed domain is deployed
Great for...	End-user mobile or desktop devices	Server VMs deployed in Azure

Next steps

To get started with using Azure AD DS, create an Azure AD DS managed domain using the Azure portal.

You can also learn more about management concepts for user accounts, passwords, and administration in Azure AD DS and how objects and credentials are synchronized in a managed domain.

Tutorial: Create and configure an Azure Active Directory Domain Services managed domain

7/20/2020 • 11 minutes to read • Edit Online

Azure Active Directory Domain Services (Azure AD DS) provides managed domain services such as domain join, group policy, LDAP, Kerberos/NTLM authentication that is fully compatible with Windows Server Active Directory. You consume these domain services without deploying, managing, and patching domain controllers yourself. Azure AD DS integrates with your existing Azure AD tenant. This integration lets users sign in using their corporate credentials, and you can use existing groups and user accounts to secure access to resources.

You can create a managed domain using default configuration options for networking and synchronization, or manually define these settings. This tutorial shows you how to use default options to create and configure an Azure AD DS managed domain using the Azure portal.

In this tutorial, you learn how to:

- Understand DNS requirements for a managed domain
- Create a managed domain
- Enable password hash synchronization

If you don't have an Azure subscription, create an account before you begin.

Prerequisites

To complete this tutorial, you need the following resources and privileges:

- An active Azure subscription.
  - If you don't have an Azure subscription, create an account.
- An Azure Active Directory tenant associated with your subscription, either synchronized with an on-premises directory or a cloud-only directory.
  - If needed, create an Azure Active Directory tenant or associate an Azure subscription with your account.
- You need *global administrator* privileges in your Azure AD tenant to enable Azure AD DS.
- You need *Contributor* privileges in your Azure subscription to create the required Azure AD DS resources.

Although not required for Azure AD DS, it's recommended to configure self-service password reset (SSPR) for the Azure AD tenant. Users can change their password without SSPR, but SSPR helps if they forget their password and need to reset it.

> **IMPORTANT**
>
> After you create a managed domain, you can't then move the managed domain to a different resource group, virtual network, subscription, etc. Take care to select the most appropriate subscription, resource group, region, and virtual network when you deploy the managed domain.

Sign in to the Azure portal

In this tutorial, you create and configure the managed domain using the Azure portal. To get started, first sign

in to the Azure portal.

Create a managed domain

To launch the **Enable Azure AD Domain Services** wizard, complete the following steps:

1. On the Azure portal menu or from the **Home** page, select **Create a resource**.
2. Enter *Domain Services* into the search bar, then choose *Azure AD Domain Services* from the search suggestions.
3. On the Azure AD Domain Services page, select **Create**. The **Enable Azure AD Domain Services** wizard is launched.
4. Select the Azure **Subscription** in which you would like to create the managed domain.
5. Select the **Resource group** to which the managed domain should belong. Choose to **Create new** or select an existing resource group.

When you create a managed domain, you specify a DNS name. There are some considerations when you choose this DNS name:

- **Built-in domain name:** By default, the built-in domain name of the directory is used (a *.onmicrosoft.com* suffix). If you wish to enable secure LDAP access to the managed domain over the internet, you can't create a digital certificate to secure the connection with this default domain. Microsoft owns the *.onmicrosoft.com* domain, so a Certificate Authority (CA) won't issue a certificate.
- **Custom domain names:** The most common approach is to specify a custom domain name, typically one that you already own and is routable. When you use a routable, custom domain, traffic can correctly flow as needed to support your applications.
- **Non-routable domain suffixes:** We generally recommend that you avoid a non-routable domain name suffix, such as *contoso.local*. The *.local* suffix isn't routable and can cause issues with DNS resolution.

> **TIP**
>
> If you create a custom domain name, take care with existing DNS namespaces. It's recommended to use a domain name separate from any existing Azure or on-premises DNS name space.
>
> For example, if you have an existing DNS name space of *contoso.com*, create a managed domain with the custom domain name of *aaddscontoso.com*. If you need to use secure LDAP, you must register and own this custom domain name to generate the required certificates.
>
> You may need to create some additional DNS records for other services in your environment, or conditional DNS forwarders between existing DNS name spaces in your environment. For example, if you run a webserver that hosts a site using the root DNS name, there can be naming conflicts that require additional DNS entries.
>
> In these tutorials and how-to articles, the custom domain of *aaddscontoso.com* is used as a short example. In all commands, specify your own domain name.

The following DNS name restrictions also apply:

- **Domain prefix restrictions:** You can't create a managed domain with a prefix longer than 15 characters. The prefix of your specified domain name (such as *aaddscontoso* in the *aaddscontoso.com* domain name) must contain 15 or fewer characters.
- **Network name conflicts:** The DNS domain name for your managed domain shouldn't already exist in the virtual network. Specifically, check for the following scenarios that would lead to a name conflict:
  - If you already have an Active Directory domain with the same DNS domain name on the Azure virtual network.
  - If the virtual network where you plan to enable the managed domain has a VPN connection with your on-premises network. In this scenario, ensure you don't have a domain with the same DNS

domain name on your on-premises network.
- If you have an existing Azure cloud service with that name on the Azure virtual network.

Complete the fields in the *Basics* window of the Azure portal to create a managed domain:

1. Enter a **DNS domain name** for your managed domain, taking into consideration the previous points.
2. Choose the Azure **Location** in which the managed domain should be created. If you choose a region that supports Azure Availability Zones, the Azure AD DS resources are distributed across zones for additional redundancy.

   > **TIP**
   >
   > Availability Zones are unique physical locations within an Azure region. Each zone is made up of one or more datacenters equipped with independent power, cooling, and networking. To ensure resiliency, there's a minimum of three separate zones in all enabled regions.
   >
   > There's nothing for you to configure for Azure AD DS to be distributed across zones. The Azure platform automatically handles the zone distribution of resources. For more information and to see region availability, see What are Availability Zones in Azure?

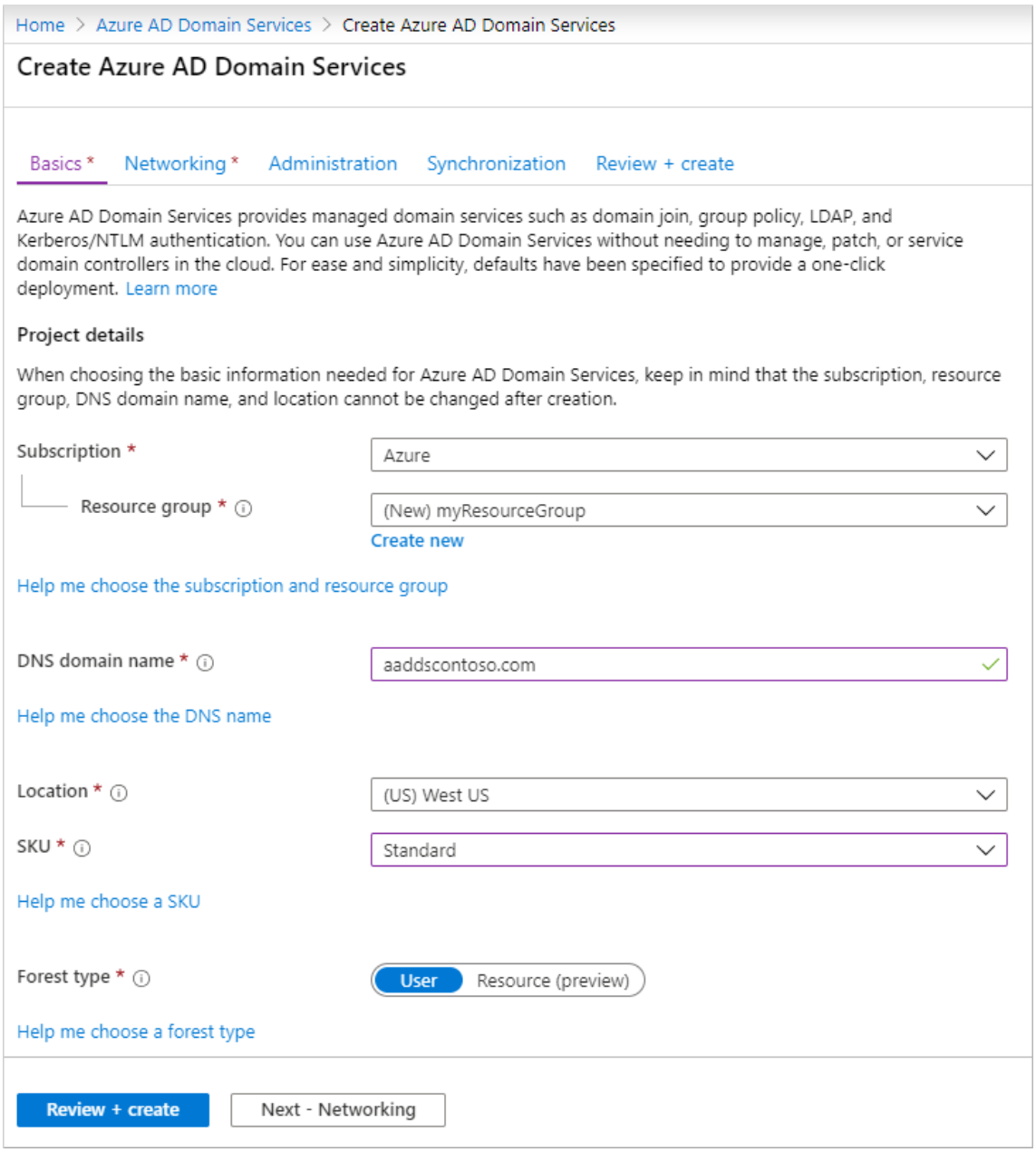
3. The **SKU** determines the performance, backup frequency, and maximum number of forest trusts you can create. You can change the SKU after the managed domain has been created if your business demands or requirements change. For more information, see Azure AD DS SKU concepts.

   For this tutorial, select the *Standard* SKU.

4. A *forest* is a logical construct used by Active Directory Domain Services to group one or more domains. By default, a managed domain is created as a *User* forest. This type of forest synchronizes all objects from Azure AD, including any user accounts created in an on-premises AD DS environment.

   A *Resource* forest only synchronizes users and groups created directly in Azure AD. Resource forests are currently in preview. For more information on *Resource* forests, including why you may use one and how to create forest trusts with on-premises AD DS domains, see Azure AD DS resource forests overview.

   For this tutorial, choose to create a *User* forest.



To quickly create a managed domain, you can select **Review + create** to accept additional default configuration options. The following defaults are configured when you choose this create option:

- Creates a virtual network named *aadds-vnet* that uses the IP address range of *10.0.2.0/24*.
- Creates a subnet named *aadds-subnet* using the IP address range of *10.0.2.0/24*.
- Synchronizes *All* users from Azure AD into the managed domain.

Select **Review + create** to accept these default configuration options.

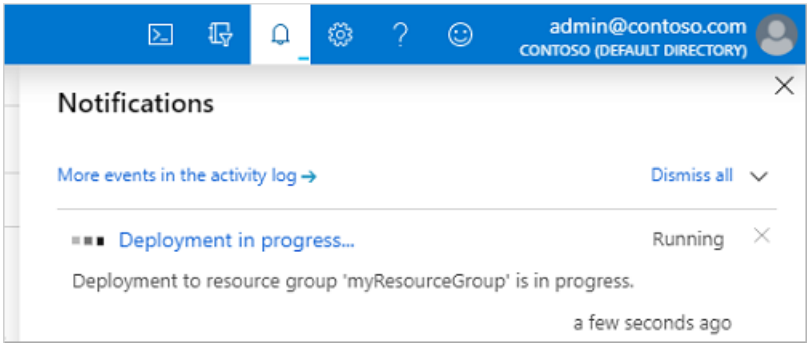
Deploy the managed domain

On the **Summary** page of the wizard, review the configuration settings for your managed domain. You can go back to any step of the wizard to make changes. To redeploy a managed domain to a different Azure AD tenant in a consistent way using these configuration options, you can also **Download a template for automation**.

1. To create the managed domain, select **Create**. A note is displayed that certain configuration options such as DNS name or virtual network can't be changed once the Azure AD DS managed has been created. To continue, select **OK**.

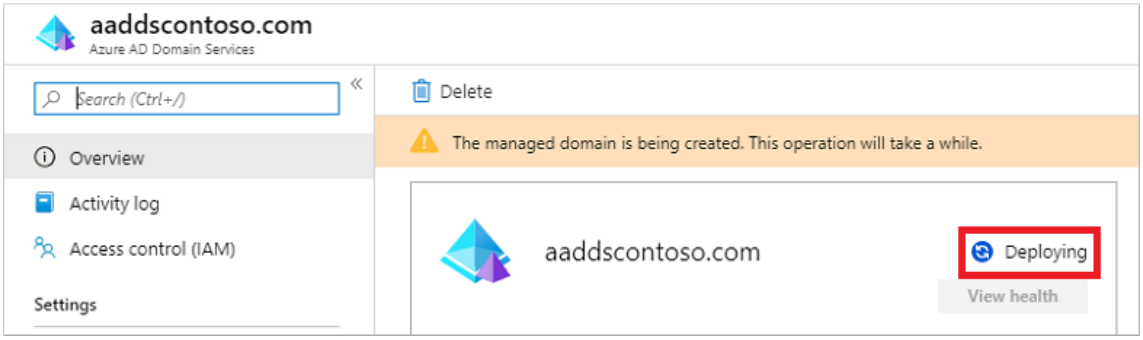
2. The process of provisioning your managed domain can take up to an hour. A notification is displayed

in the portal that shows the progress of your Azure AD DS deployment. Select the notification to see detailed progress for the deployment.

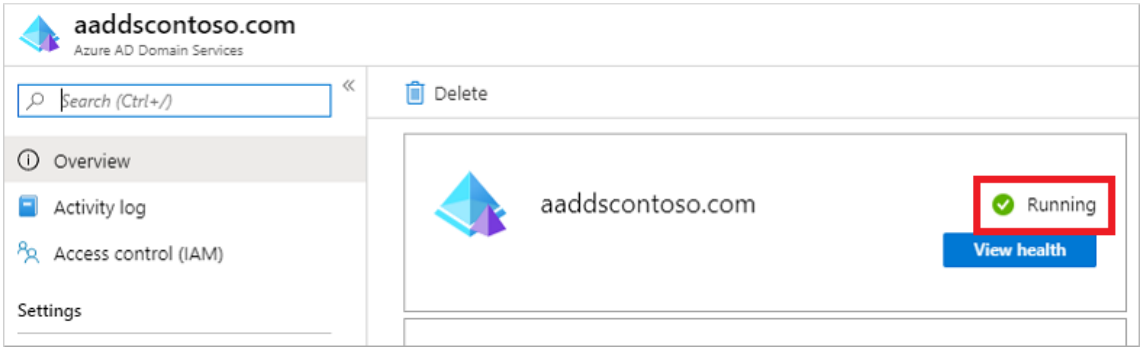


3. The page will load with updates on the deployment process, including the creation of new resources in your directory.

4. Select your resource group, such as *myResourceGroup*, then choose your managed domain from the list of Azure resources, such as *aaddscontoso.com*. The **Overview** tab shows that the managed domain is currently *Deploying*. You can't configure the managed domain until it's fully provisioned.



5. When the managed domain is fully provisioned, the **Overview** tab shows the domain status as *Running*.



IMPORTANT

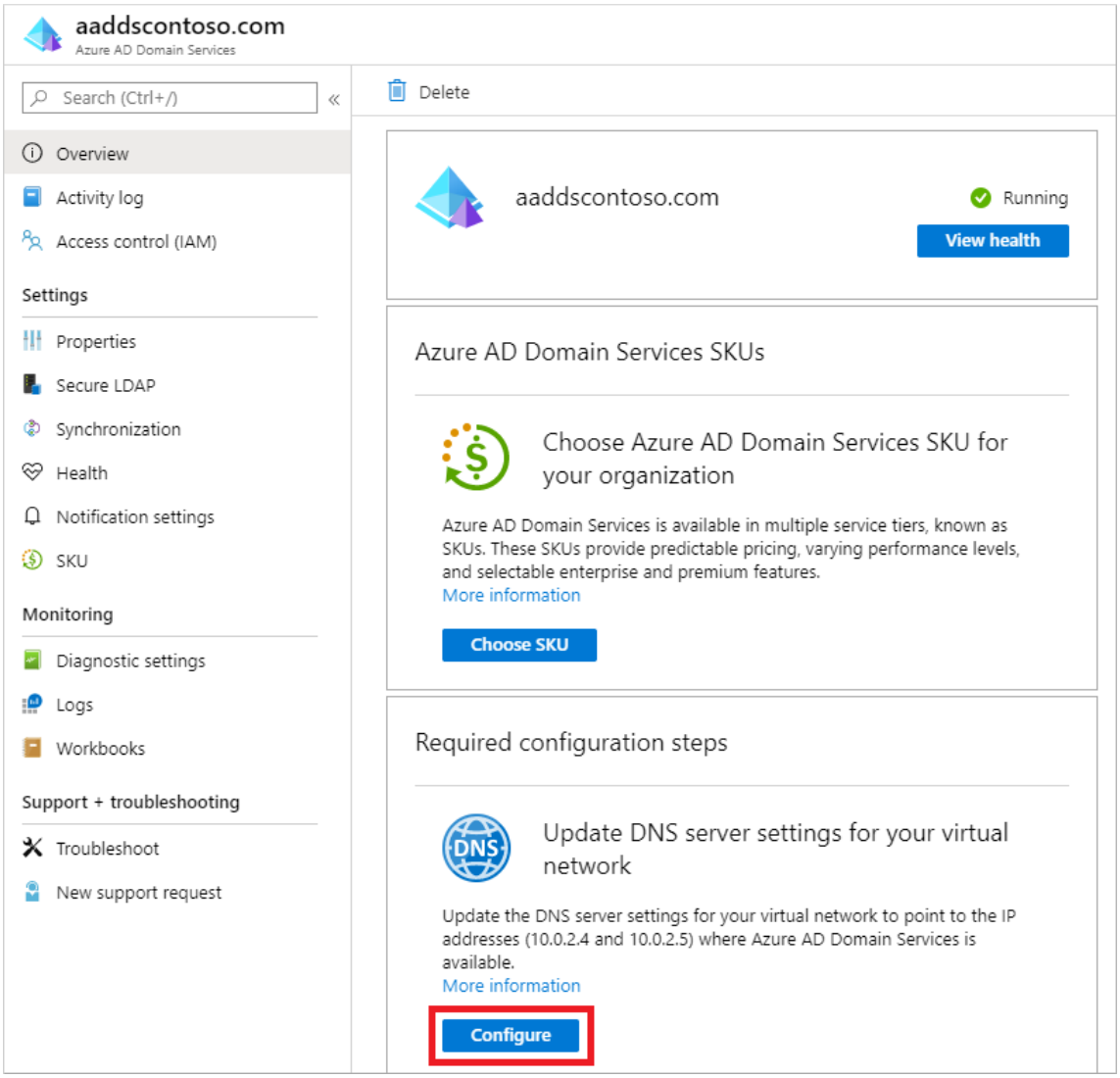
The managed domain is associated with your Azure AD tenant. During the provisioning process, Azure AD DS creates two Enterprise Applications named *Domain Controller Services* and *AzureActiveDirectoryDomainControllerServices* in the Azure AD tenant. These Enterprise Applications are needed to service your managed domain. Don't delete these applications.

Update DNS settings for the Azure virtual network

With Azure AD DS successfully deployed, now configure the virtual network to allow other connected VMs and applications to use the managed domain. To provide this connectivity, update the DNS server settings for your virtual network to point to the two IP addresses where the managed domain is deployed.

1. The **Overview** tab for your managed domain shows some **Required configuration steps**. The first configuration step is to update DNS server settings for your virtual network. Once the DNS settings are correctly configured, this step is no longer shown.

   The addresses listed are the domain controllers for use in the virtual network. In this example, those addresses are *10.0.2.4* and *10.0.2.5*. You can later find these IP addresses on the **Properties** tab.



2. To update the DNS server settings for the virtual network, select the **Configure** button. The DNS settings are automatically configured for your virtual network.

> **TIP**
> If you selected an existing virtual network in the previous steps, any VMs connected to the network only get the new DNS settings after a restart. You can restart VMs using the Azure portal, Azure PowerShell, or the Azure CLI.

Enable user accounts for Azure AD DS

To authenticate users on the managed domain, Azure AD DS needs password hashes in a format that's suitable for NT LAN Manager (NTLM) and Kerberos authentication. Azure AD doesn't generate or store password hashes in the format that's required for NTLM or Kerberos authentication until you enable Azure AD DS for your tenant. For security reasons, Azure AD also doesn't store any password credentials in clear-text form. Therefore, Azure AD can't automatically generate these NTLM or Kerberos password hashes based on users' existing credentials.

NOTE

Once appropriately configured, the usable password hashes are stored in the managed domain. If you delete the managed domain, any password hashes stored at that point are also deleted.

Synchronized credential information in Azure AD can't be re-used if you later create a managed domain - you must reconfigure the password hash synchronization to store the password hashes again. Previously domain-joined VMs or users won't be able to immediately authenticate - Azure AD needs to generate and store the password hashes in the new managed domain.

For more information, see Password hash sync process for Azure AD DS and Azure AD Connect.

The steps to generate and store these password hashes are different for cloud-only user accounts created in Azure AD versus user accounts that are synchronized from your on-premises directory using Azure AD Connect.

A cloud-only user account is an account that was created in your Azure AD directory using either the Azure portal or Azure AD PowerShell cmdlets. These user accounts aren't synchronized from an on-premises directory.

> In this tutorial, let's work with a basic cloud-only user account. For more information on the additional steps required to use Azure AD Connect, see Synchronize password hashes for user accounts synced from your on-premises AD to your managed domain.

TIP

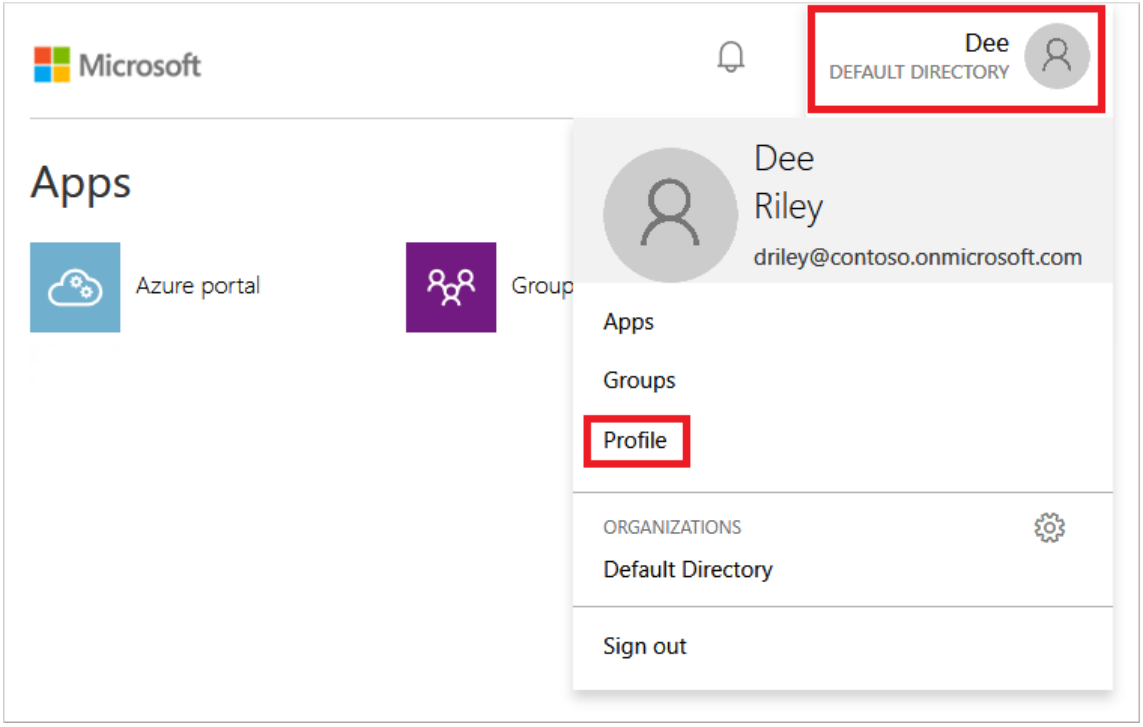
If your Azure AD tenant has a combination of cloud-only users and users from your on-premises AD, you need to complete both sets of steps.

For cloud-only user accounts, users must change their passwords before they can use Azure AD DS. This password change process causes the password hashes for Kerberos and NTLM authentication to be generated and stored in Azure AD. The account isn't synchronized from Azure AD to Azure AD DS until the password is changed. Either expire the passwords for all cloud users in the tenant who need to use Azure AD DS, which forces a password change on next sign-in, or instruct cloud users to manually change their passwords. For this tutorial, let's manually change a user password.

Before a user can reset their password, the Azure AD tenant must be configured for self-service password reset.

To change the password for a cloud-only user, the user must complete the following steps:

1. Go to the Azure AD Access Panel page at https://myapps.microsoft.com.
2. In the top-right corner, select your name, then choose **Profile** from the drop-down menu.



3. On the **Profile** page, select **Change password**.
4. On the **Change password** page, enter your existing (old) password, then enter and confirm a new password.
5. Select **Submit**.

It takes a few minutes after you've changed your password for the new password to be usable in Azure AD DS and to successfully sign in to computers joined to the managed domain.

Next steps

In this tutorial, you learned how to:

- Understand DNS requirements for a managed domain
- Create a managed domain
- Add administrative users to domain management
- Enable user accounts for Azure AD DS and generate password hashes

Before you domain-join VMs and deploy applications that use the managed domain, configure an Azure virtual network for application workloads.

Configure Azure virtual network for application workloads to use your managed domain

Tutorial: Configure virtual networking for an Azure Active Directory Domain Services managed domain

7/20/2020 • 8 minutes to read • Edit Online

To provide connectivity to users and applications, an Azure Active Directory Domain Services (Azure AD DS) managed domain is deployed into an Azure virtual network subnet. This virtual network subnet should only be used for the managed domain resources provided by the Azure platform.

When you create your own VMs and applications, they shouldn't be deployed into the same virtual network subnet. Instead, you should create and deploy your applications into a separate virtual network subnet, or in a separate virtual network that's peered to the Azure AD DS virtual network.

This tutorial shows you how to create and configure a dedicated virtual network subnet or how to peer a different network to the Azure AD DS managed domain's virtual network.

In this tutorial, you learn how to:

- Understand the virtual network connectivity options for domain-joined resources to Azure AD DS
- Create an IP address range and additional subnet in the Azure AD DS virtual network
- Configure virtual network peering to a network that's separate from Azure AD DS

If you don't have an Azure subscription, create an account before you begin.

Prerequisites

To complete this tutorial, you need the following resources and privileges:

- An active Azure subscription.
  - If you don't have an Azure subscription, create an account.
- An Azure Active Directory tenant associated with your subscription, either synchronized with an on-premises directory or a cloud-only directory.
  - If needed, create an Azure Active Directory tenant or associate an Azure subscription with your account.
- You need *global administrator* privileges in your Azure AD tenant to configure Azure AD DS.
- You need *Contributor* privileges in your Azure subscription to create the required Azure AD DS resources.
- An Azure Active Directory Domain Services managed domain enabled and configured in your Azure AD tenant.
  - If needed, the first tutorial creates and configures an Azure Active Directory Domain Services managed domain.

Sign in to the Azure portal

In this tutorial, you create and configure the managed domain using the Azure portal. To get started, first sign in to the Azure portal.

Application workload connectivity options

In the previous tutorial, a managed domain was created that used some default configuration options for the virtual network. These default options created an Azure virtual network and virtual network subnet. The Azure AD DS domain controllers that provide the managed domain services are connected to this virtual network subnet.

When you create and run VMs that need to use the managed domain, network connectivity needs to be provided.

This network connectivity can be provided in one of the following ways:

- Create an additional virtual network subnet in the managed domain's virtual network. This additional subnet is where you create and connect your VMs.
  - As the VMs are part of the same virtual network, they can automatically perform name resolution and communicate with the Azure AD DS domain controllers.
- Configure Azure virtual network peering from the managed domain's virtual network to one or more separate virtual networks. These separate virtual networks are where you create and connect your VMs.
  - When you configure virtual network peering, you must also configure DNS settings to use name resolution back to the Azure AD DS domain controllers.

Usually, you only use one of these network connectivity options. The choice is often down to how you wish to manage separate your Azure resources.

- If you want to manage Azure AD DS and connected VMs as one group of resources, you can create an additional virtual network subnet for VMs.
- If you want to separate the management of Azure AD DS and then any connected VMs, you can use virtual network peering.
  - You may also choose to use virtual network peering to provide connectivity to existing VMs in your Azure environment that are connected to an existing virtual network.

In this tutorial, you only need to configure one these virtual network connectivity options.

For more information on how to plan and configure the virtual network, see networking considerations for Azure Active Directory Domain Services.

Create a virtual network subnet

By default, the Azure virtual network created with the managed domain contains a single virtual network subnet. This virtual network subnet should only be used by the Azure platform to provide managed domain services. To create and use your own VMs in this Azure virtual network, create an additional subnet.

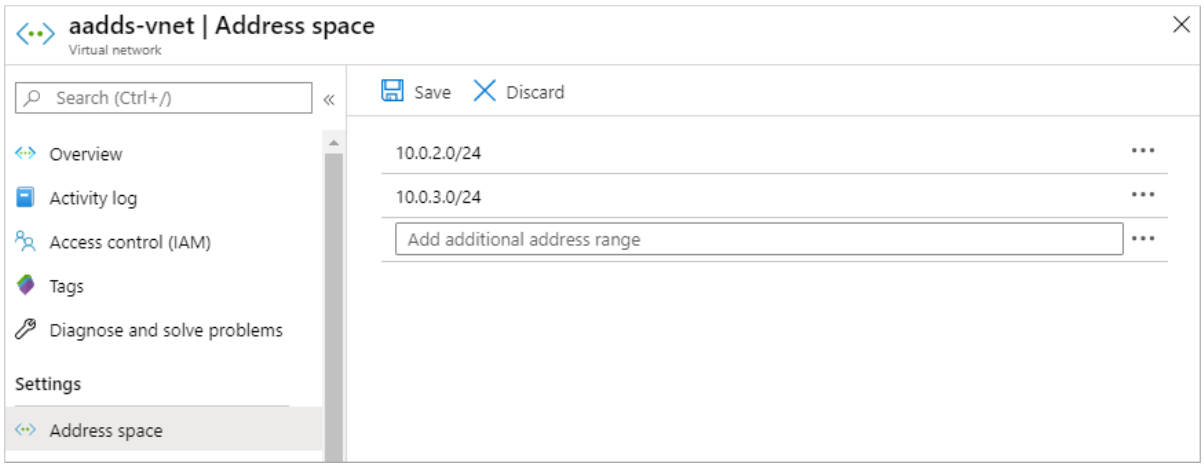
To create a virtual network subnet for VMs and application workloads, complete the following steps:

1. In the Azure portal, select the resource group of your managed domain, such as *myResourceGroup*. From the list of resources, choose the default virtual network, such as *aadds-vnet*.

2. In the left-hand menu of the virtual network window, select **Address space**. The virtual network is created with a single address space of *10.0.2.0/24*, which is used by the default subnet.

   Add an additional IP address range to the virtual network. The size of this address range and the actual IP address range to use depends on other network resources already deployed. The IP address range shouldn't overlap with any existing address ranges in your Azure or on-premises environment. Make sure that you size the IP address range large enough for the number of VMs you expect to deploy into the subnet.

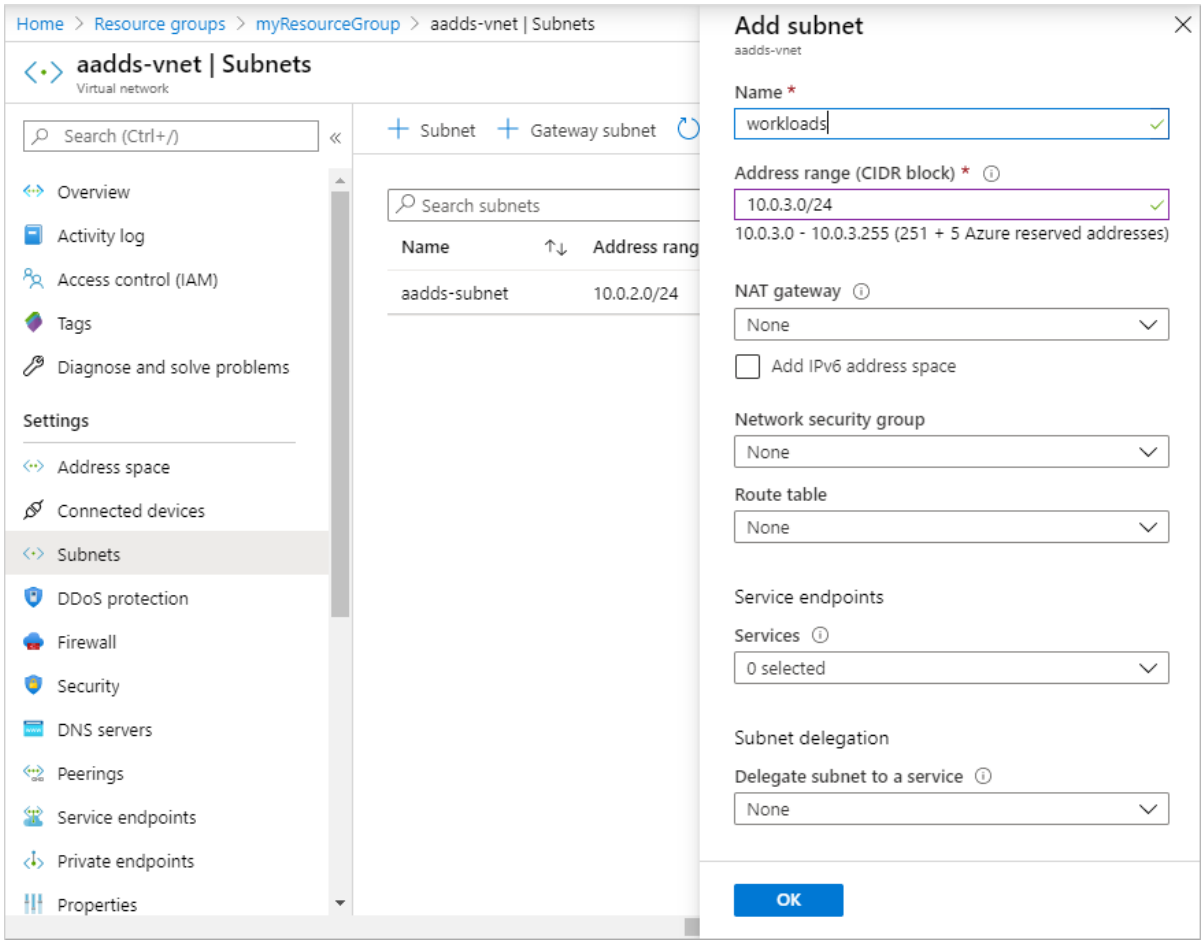
   In the following example, an additional IP address range of *10.0.3.0/24* is added. When ready, select **Save**.



3. Next, in the left-hand menu of the virtual network window, select **Subnets**, then choose **+ Subnet** to add a subnet.

4. Enter a name for the subnet, such as *workloads*. If needed, update the **Address range** if you want to use a subset of the IP address range configured for the virtual network in the previous steps. For now, leave the defaults for options like network security group, route table, service endpoints.

   In the following example, a subnet named *workloads* is created that uses the *10.0.3.0/24* IP address range:



5. When ready, select **OK**. It takes a few moments to create the virtual network subnet.

When you create a VM that needs to use the managed domain, make sure you select this virtual network subnet. Don't create VMs in the default *aadds-subnet*. If you select a different virtual network, there's no network connectivity and DNS resolution to reach the managed domain unless you configure virtual network peering.

Configure virtual network peering

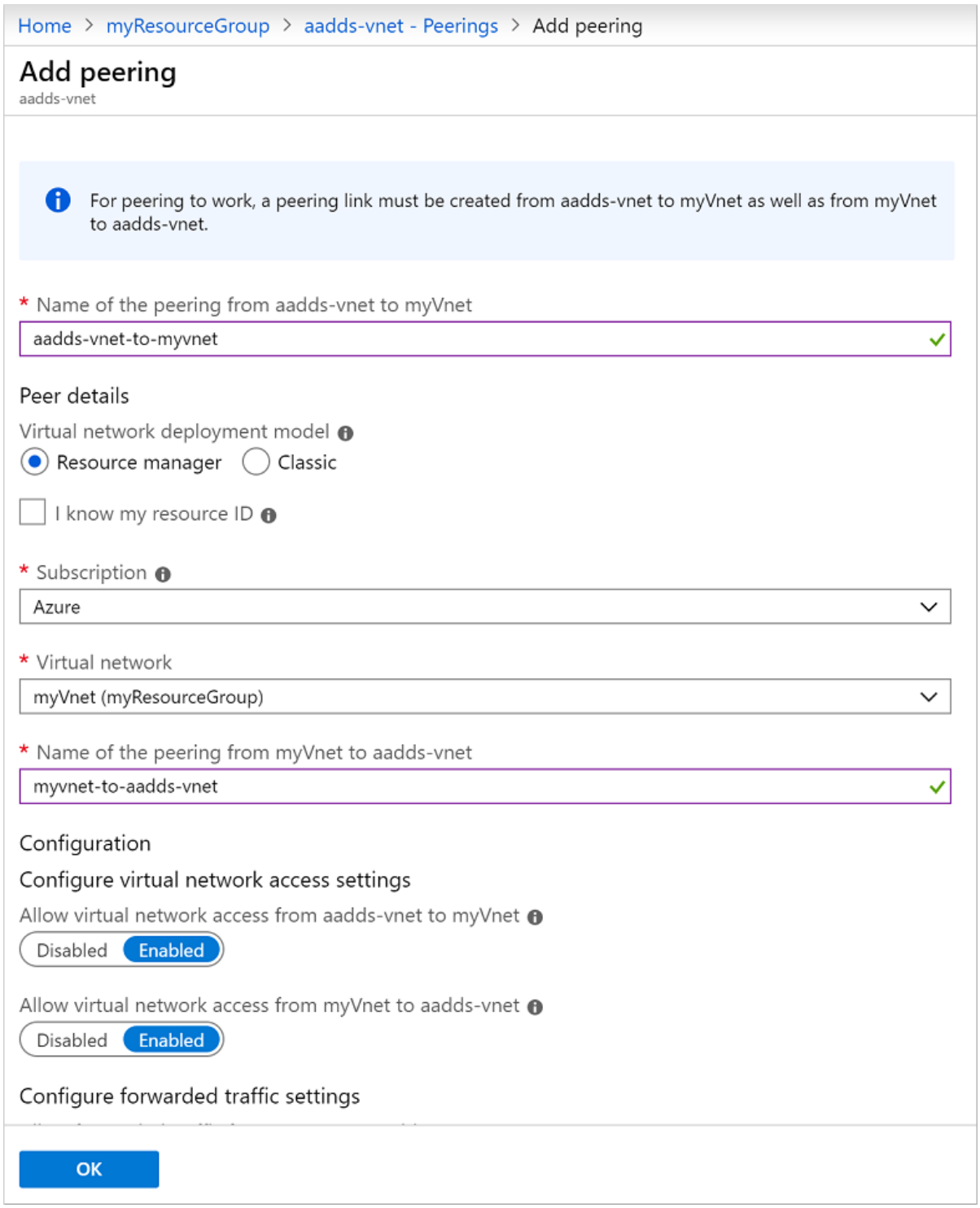
You may have an existing Azure virtual network for VMs, or wish to keep your managed domain virtual network separate. To use the managed domain, VMs in other virtual networks need a way to communicate with the Azure AD DS domain controllers. This connectivity can be provided using Azure virtual network peering.

With Azure virtual network peering, two virtual networks are connected together, without the need for a virtual private network (VPN) device. Network peering lets you quickly connect virtual networks and define traffic flows across your Azure environment.

For more information on peering, see Azure virtual network peering overview.

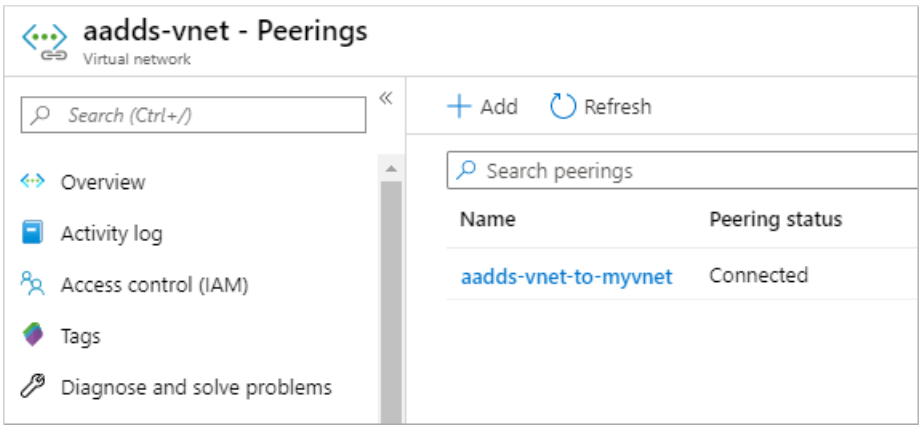
To peer a virtual network to the managed domain virtual network, complete the followings steps:

1. Choose the default virtual network created for your managed domain named *aadds-vnet*.
2. In the left-hand menu of the virtual network window, select **Peerings**.
3. To create a peering, select **+ Add**. In the following example, the default *aadds-vnet* is peered to a virtual network named *myVnet*. Configure the following settings with your own values:
   - **Name of the peering from aadds-vnet to remote virtual network**: A descriptive identifier of the two networks, such as *aadds-vnet-to-myvnet*
   - **Virtual network deployment type**: *Resource Manager*
   - **Subscription**: The subscription of the virtual network you want to peer to, such as *Azure*
   - **Virtual network**: The virtual network you want to peer to, such as *myVnet*
   - **Name of the peering from myVnet to aadds-vnet**: A descriptive identifier of the two networks, such as *myvnet-to-aadds-vnet*



Leave any other defaults for virtual network access or forwarded traffic unless you have specific requirements for your environment, then select **OK**.

4. It takes a few moments to create the peering on both the Azure AD DS virtual network and the virtual network you selected. When ready, the **Peering status** reports *Connected*, as shown in the following example:



Before VMs in the peered virtual network can use the managed domain, configure the DNS servers to allow for correct name resolution.

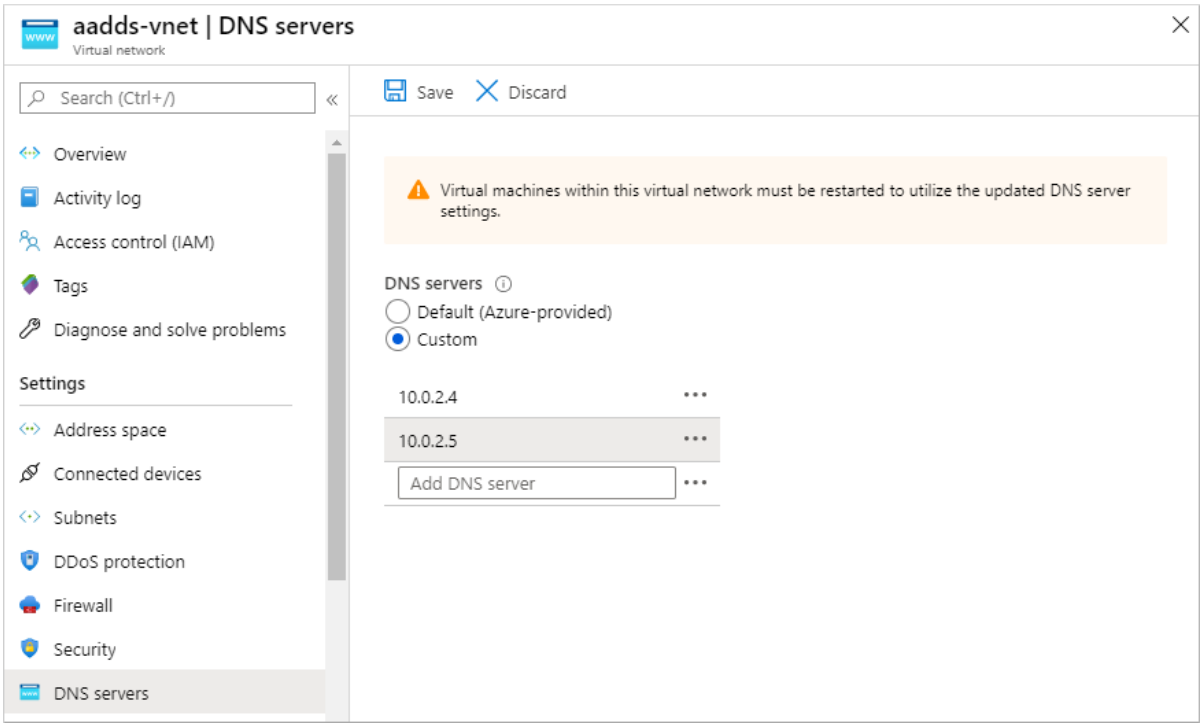
Configure DNS servers in the peered virtual network

For VMs and applications in the peered virtual network to successfully talk to the managed domain, the DNS settings must be updated. The IP addresses of the Azure AD DS domain controllers must be configured as the DNS servers on the peered virtual network. There are two ways to configure the domain controllers as DNS servers for the peered virtual network:

- Configure the Azure virtual network DNS servers to use the Azure AD DS domain controllers.
- Configure the existing DNS server in use on the peered virtual network to use conditional DNS forwarding to direct queries to the managed domain. These steps vary depending on the existing DNS server in use.

In this tutorial, let's configure the Azure virtual network DNS servers to direct all queries to the Azure AD DS domain controllers.

1. In the Azure portal, select the resource group of the peered virtual network, such as *myResourceGroup*. From the list of resources, choose the peered virtual network, such as *myVnet*.
2. In the left-hand menu of the virtual network window, select **DNS servers**.
3. By default, a virtual network uses the built-in Azure-provided DNS servers. Choose to use **Custom** DNS servers. Enter the IP addresses for the Azure AD DS domain controllers, which are usually *10.0.2.4* and *10.0.2.5*. Confirm these IP addresses on the **Overview** window of your managed domain in the portal.



4. When ready, select **Save**. It takes a few moments to update the DNS servers for the virtual network.

5. To apply the updated DNS settings to the VMs, restart VMs connected to the peered virtual network.

When you create a VM that needs to use the managed domain, make sure you select this peered virtual network. If you select a different virtual network, there's no network connectivity and DNS resolution to reach the managed domain.

Next steps

In this tutorial, you learned how to:

- Understand the virtual network connectivity options for domain-joined resources to Azure AD DS
- Create an IP address range and additional subnet in the Azure AD DS virtual network
- Configure virtual network peering to a network that's separate from Azure AD DS

To see this managed domain in action, create and join a virtual machine to the domain.

Join a Windows Server virtual machine to your managed domain

Tutorial: Join a Windows Server virtual machine to an Azure Active Directory Domain Services managed domain

7/20/2020 • 12 minutes to read • Edit Online

Azure Active Directory Domain Services (Azure AD DS) provides managed domain services such as domain join, group policy, LDAP, Kerberos/NTLM authentication that is fully compatible with Windows Server Active Directory. With an Azure AD DS managed domain, you can provide domain join features and management to virtual machines (VMs) in Azure. This tutorial shows you how to create a Windows Server VM then join it to a managed domain.

In this tutorial, you learn how to:

- Create a Windows Server VM
- Connect the Windows Server VM to an Azure virtual network
- Join the VM to the managed domain

If you don't have an Azure subscription, create an account before you begin.

Prerequisites

To complete this tutorial, you need the following resources:

- An active Azure subscription.
  - If you don't have an Azure subscription, create an account.
- An Azure Active Directory tenant associated with your subscription, either synchronized with an on-premises directory or a cloud-only directory.
  - If needed, create an Azure Active Directory tenant or associate an Azure subscription with your account.
- An Azure Active Directory Domain Services managed domain enabled and configured in your Azure AD tenant.
  - If needed, create and configure an Azure Active Directory Domain Services managed domain.
- A user account that's a part of the managed domain.
  - Make sure that Azure AD Connect password hash synchronization or self-service password reset has been performed so the account is able to sign in to managed domain.
- An Azure Bastion host deployed in your Azure AD DS virtual network.
  - If needed, create an Azure Bastion host.

If you already have a VM that you want to domain-join, skip to the section to join the VM to the managed domain.

Sign in to the Azure portal

In this tutorial, you create a Windows Server VM to join to your managed domain using the Azure portal. To get started, first sign in to the Azure portal.

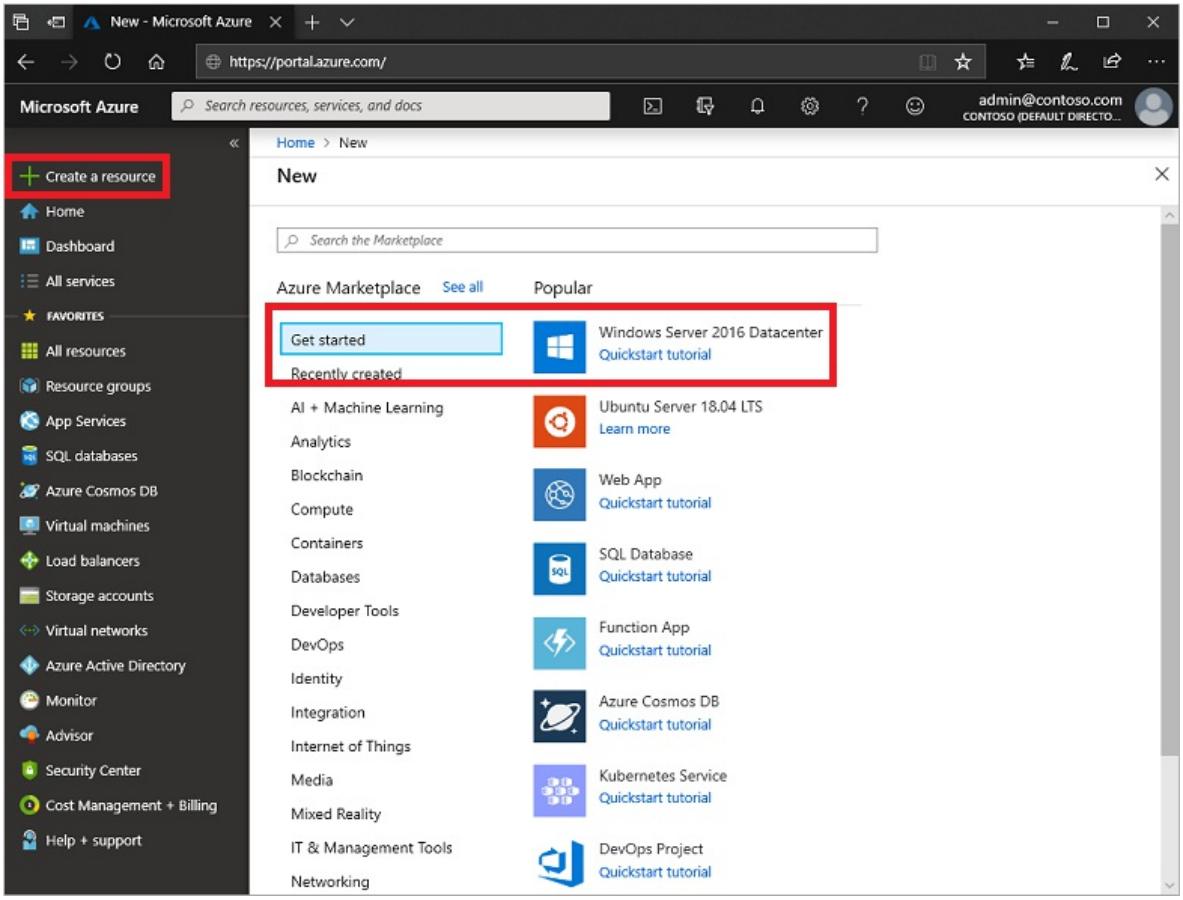
Create a Windows Server virtual machine

To see how to join a computer to a managed domain, let's create a Windows Server VM. This VM is connected to

an Azure virtual network that provides connectivity to the managed domain. The process to join a managed domain is the same as joining a regular on-premises Active Directory Domain Services domain.

If you already have a VM that you want to domain-join, skip to the section to join the VM to the managed domain.

1. From the Azure portal menu or from the **Home** page, select **Create a resource**.
2. From **Get started**, choose **Windows Server 2016 Datacenter**.



3. In the **Basics** window, configure the core settings for the virtual machine. Leave the defaults for *Availability options*, *Image*, and *Size*.

PARAMETER	SUGGESTED VALUE
Resource group	Select or create a resource group, such as *myResourceGroup*
Virtual machine name	Enter a name for the VM, such as *myVM*
Region	Choose the region to create your VM in, such as *East US*
Username	Enter a username for the local administrator account to create on the VM, such as *azureuser*
Password	Enter, and then confirm, a secure password for the local administrator to create on the VM. Don't specify a domain user account's credentials.

4. By default, VMs created in Azure are accessible from the Internet using RDP. When RDP is enabled, automated sign-in attacks are likely to occur, which may disable accounts with common names such as *admin* or *administrator* due to multiple failed successive sign-in attempts.

RDP should only be enabled when required, and limited to a set of authorized IP ranges. This configuration helps improve the security of the VM and reduces the area for potential attack. Or, create and use an Azure Bastion host that allows access only through the Azure portal over TLS. In the next step of this tutorial, you use an Azure Bastion host to securely connect to the VM.

Under **Public inbound ports**, select *None*.

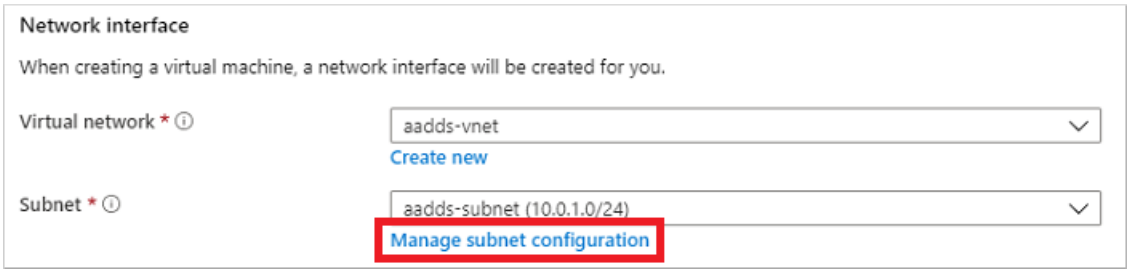
5. When done, select **Next: Disks**.
6. From the drop-down menu for **OS disk type**, choose *Standard SSD*, then select **Next: Networking**.
7. Your VM must connect to an Azure virtual network subnet that can communicate with the subnet your managed domain is deployed into. We recommend that a managed domain is deployed into its own dedicated subnet. Don't deploy your VM in the same subnet as your managed domain.

   There are two main ways to deploy your VM and connect to an appropriate virtual network subnet:

   - Create a, or select an existing, subnet in the same the virtual network as your managed domain is deployed.
   - Select a subnet in an Azure virtual network that is connected to it using Azure virtual network peering.

   If you select a virtual network subnet that isn't connected to the subnet for your managed domain, you can't join the VM to the managed domain. For this tutorial, let's create a new subnet in the Azure virtual network.

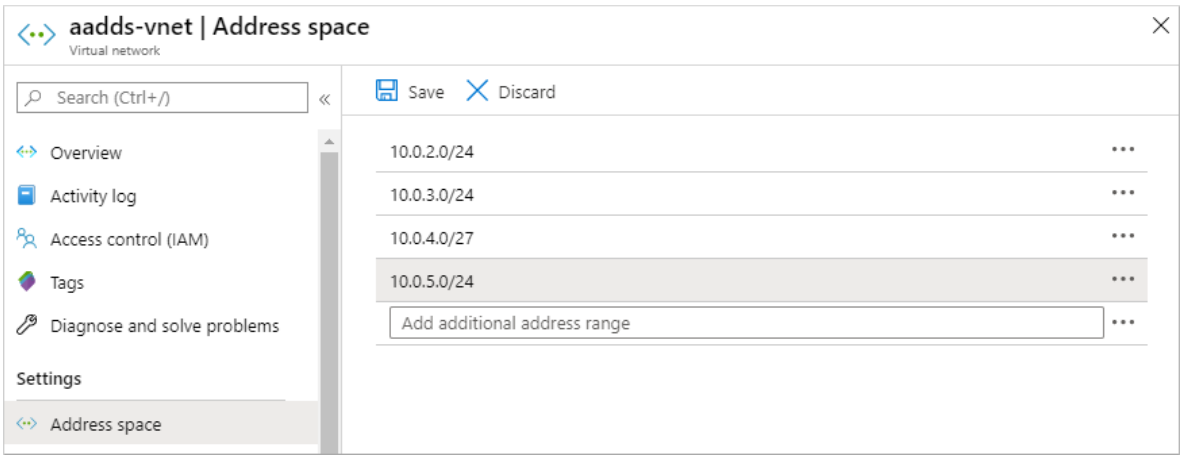
   In the **Networking** pane, select the virtual network in which your managed domain is deployed, such as *aaads-vnet*
8. In this example, the existing *aaads-subnet* is shown that the managed domain is connected to. Don't connect your VM to this subnet. To create a subnet for the VM, select **Manage subnet configuration**.



9. In the left-hand menu of the virtual network window, select **Address space**. The virtual network is created with a single address space of *10.0.2.0/24*, which is used by the default subnet. Other subnets, such as for *workloads* or Azure Bastion may also already exist.

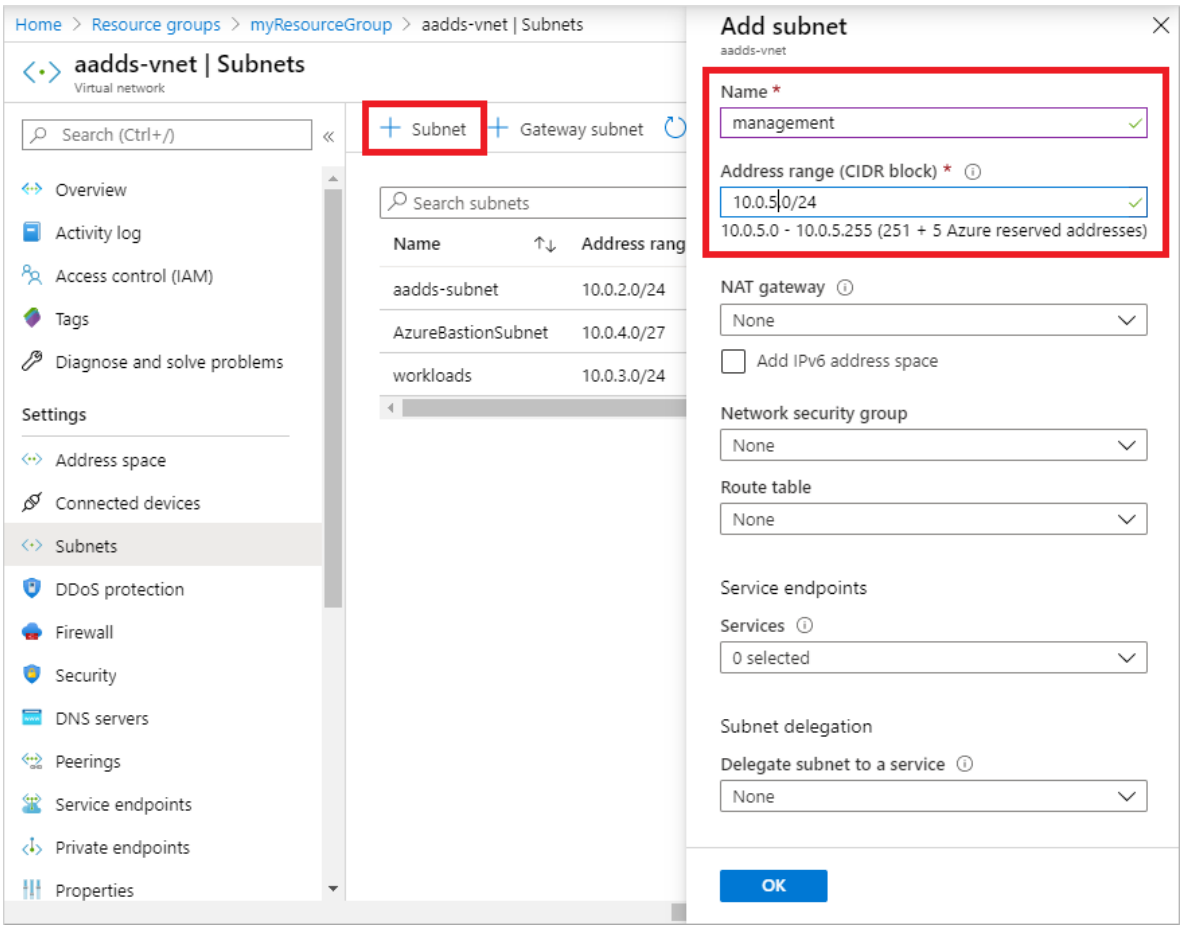
   Add an additional IP address range to the virtual network. The size of this address range and the actual IP address range to use depends on other network resources already deployed. The IP address range shouldn't overlap with any existing address ranges in your Azure or on-premises environment. Make sure that you size the IP address range large enough for the number of VMs you expect to deploy into the subnet.

   In the following example, an additional IP address range of *10.0.5.0/24* is added. When ready, select **Save**.



10. Next, in the left-hand menu of the virtual network window, select **Subnets**, then choose **+ Subnet** to add a subnet.

11. Select **+ Subnet**, then enter a name for the subnet, such as *management*. Provide an **Address range (CIDR block)**, such as *10.0.5.0/24*. Make sure that this IP address range doesn't overlap with any other existing Azure or on-premises address ranges. Leave the other options as their default values, then select **OK**.



12. It takes a few seconds to create the subnet. Once it's created, select the *X* to close the subnet window.

13. Back in the **Networking** pane to create a VM, choose the subnet you created from the drop-down menu, such as *management*. Again, make sure you choose the correct subnet and don't deploy your VM in the same subnet as your managed domain.

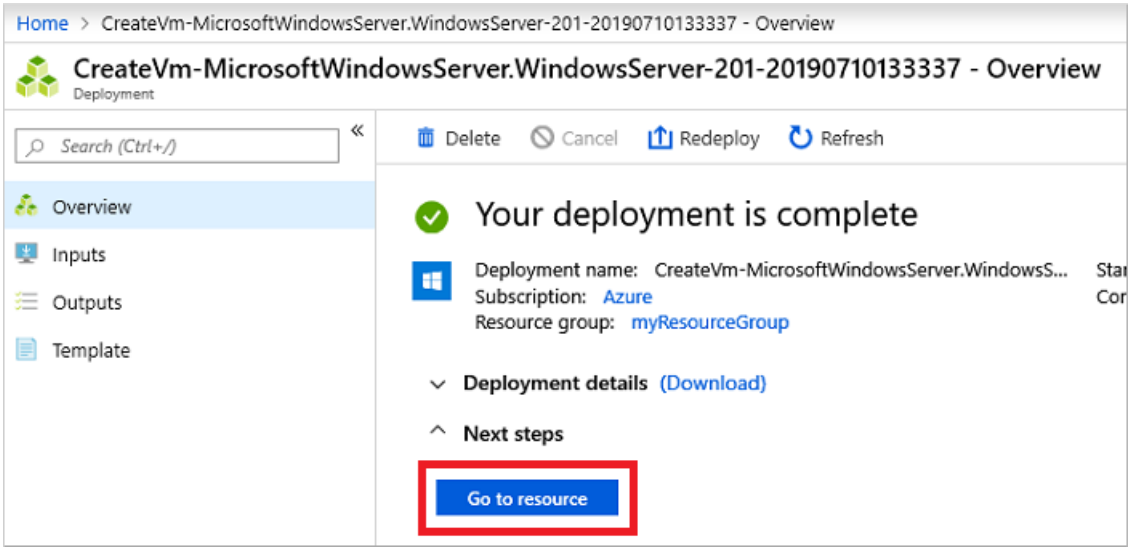
14. For **Public IP**, select *None* from the drop-down menu. As you use Azure Bastion in this tutorial to connect to the management, you don't need a public IP address assigned to the VM.

15. Leave the other options as their default values, then select **Management**.

16. Set **Boot diagnostics** to *Off*. Leave the other options as their default values, then select **Review + create**.

17. Review the VM settings, then select **Create**.

It takes a few minutes to create the VM. The Azure portal shows the status of the deployment. Once the VM is ready, select **Go to resource**.

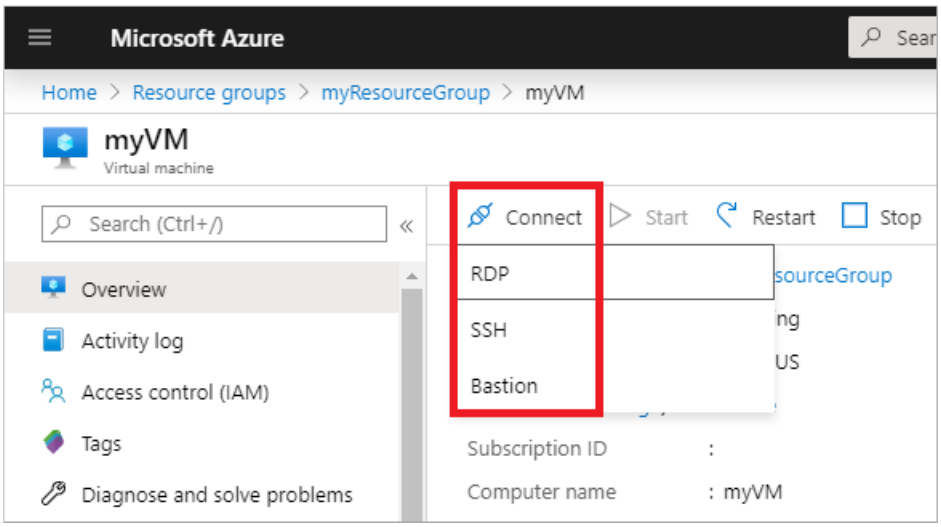


Connect to the Windows Server VM

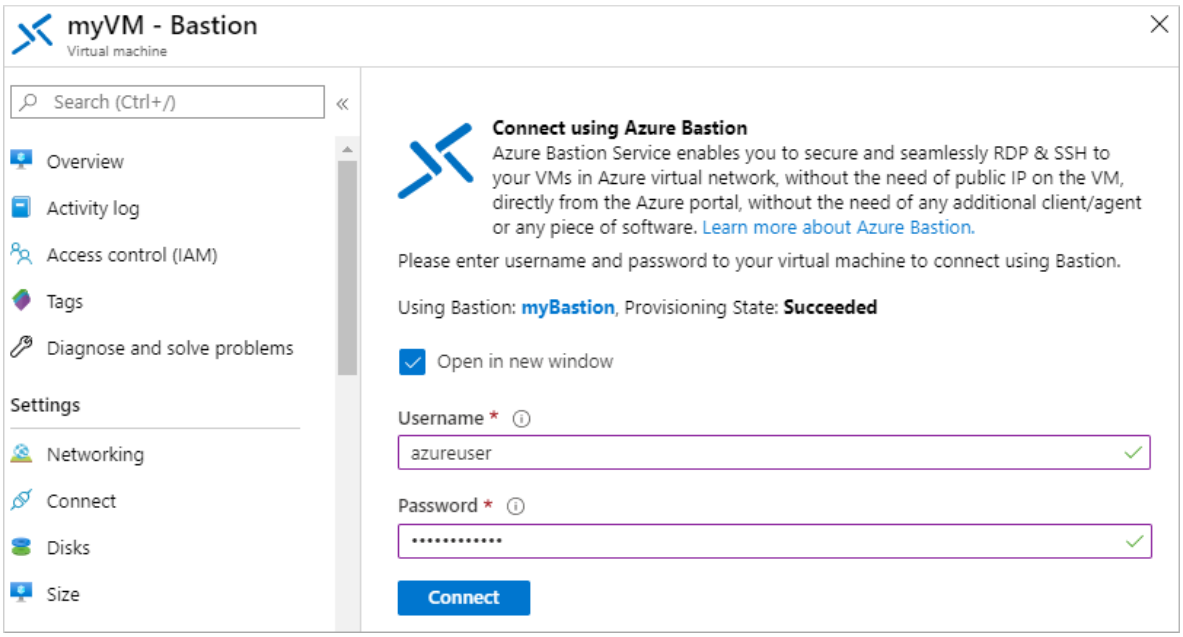
To securely connect to your VMs, use an Azure Bastion host. With Azure Bastion, a managed host is deployed into your virtual network and provides web-based RDP or SSH connections to VMs. No public IP addresses are required for the VMs, and you don't need to open network security group rules for external remote traffic. You connect to VMs using the Azure portal from your web browser. If needed, create an Azure Bastion host.

To use a Bastion host to connect to your VM, complete the following steps:

1. In the **Overview** pane for your VM, select **Connect**, then **Bastion**.



2. Enter the credentials for your VM that you specified in the previous section, then select **Connect**.

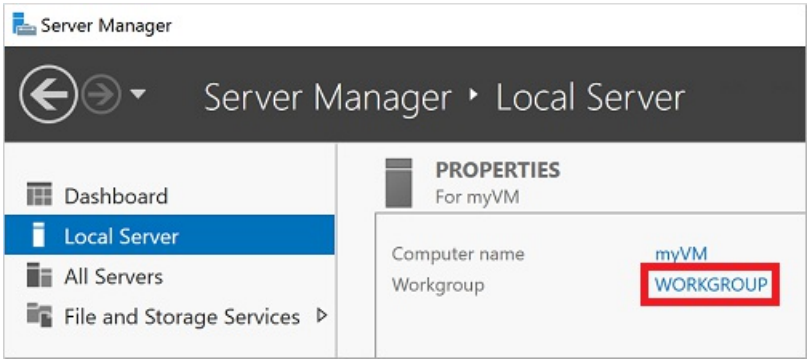


If needed, allow your web browser to open pop-ups for the Bastion connection to be displayed. It takes a few seconds to make the connection to your VM.

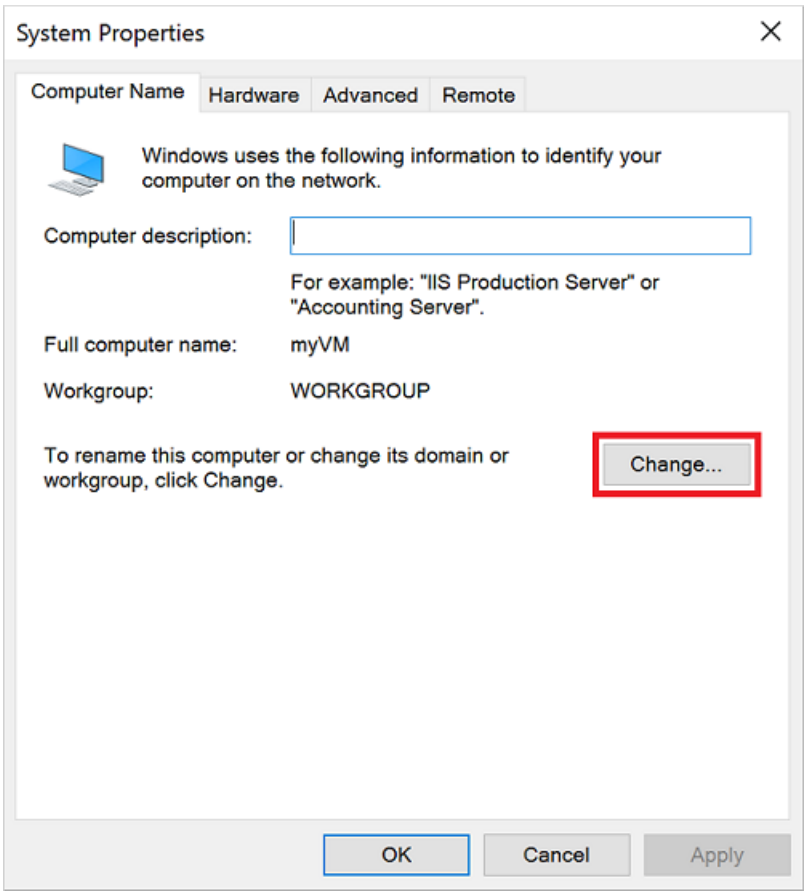
Join the VM to the managed domain

With the VM created and a web-based RDP connection established using Azure Bastion, now let's join the Windows Server virtual machine to the managed domain. This process is the same as a computer connecting to a regular on-premises Active Directory Domain Services domain.

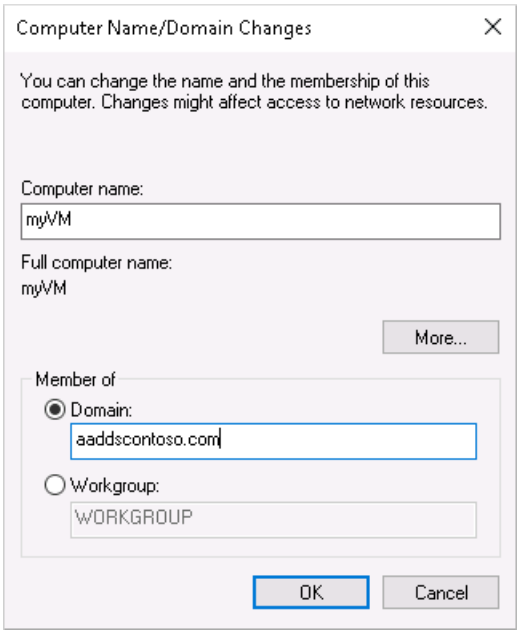
1. If **Server Manager** doesn't open by default when you sign in to the VM, select the **Start** menu, then choose **Server Manager**.
2. In the left pane of the **Server Manager** window, select **Local Server**. Under **Properties** on the right pane, choose **Workgroup**.



3. In the **System Properties** window, select **Change** to join the managed domain.



4. In the **Domain** box, specify the name of your managed domain, such as *aaddscontoso.com*, then select **OK**.



5. Enter domain credentials to join the domain. Provide credentials for a user that's a part of the managed domain. The account must be part of the managed domain or Azure AD tenant - accounts from external directories associated with your Azure AD tenant can't correctly authenticate during the domain-join process.

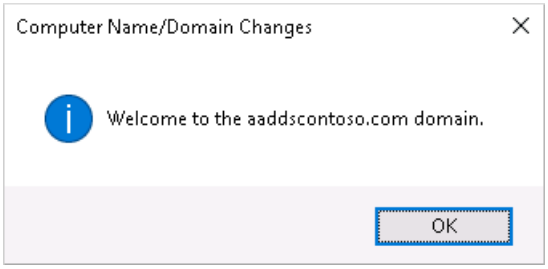
   Account credentials can be specified in one of the following ways:

   - **UPN format** (recommended) - Enter the user principal name (UPN) suffix for the user account, as configured in Azure AD. For example, the UPN suffix of the user *contosoadmin* would be `contosoadmin@aaddscontoso.onmicrosoft.com`. There are a couple of common use-cases where the UPN format can be used reliably to sign in to the domain rather than the *SAMAccountName* format:

- If a user's UPN prefix is long, such as *deehasareallylongname*, the *SAMAccountName* may be autogenerated.
- If multiple users have the same UPN prefix in your Azure AD tenant, such as *dee*, their *SAMAccountName* format might be autogenerated.

- **SAMAccountName format** - Enter the account name in the *SAMAccountName* format. For example, the *SAMAccountName* of user *contosoadmin* would be `AADDSCONTOSO\contosoadmin`.

6. It takes a few seconds to join to the managed domain. When complete, the following message welcomes you to the domain:



   Select **OK** to continue.

7. To complete the process to join to the managed domain, restart the VM.

> **TIP**
> You can domain-join a VM using PowerShell with the Add-Computer cmdlet. The following example joins the *AADDSCONTOSO* domain and then restarts the VM. When prompted, enter the credentials for a user that's a part of the managed domain:
>
> ```
> Add-Computer -DomainName AADDSCONTOSO -Restart
> ```
>
> To domain-join a VM without connecting to it and manually configuring the connection, you can use the Set-AzVmAdDomainExtension Azure PowerShell cmdlet.

Once the Windows Server VM has restarted, any policies applied in the managed domain are pushed to the VM. You can also now sign in to the Windows Server VM using appropriate domain credentials.

Clean up resources

In the next tutorial, you use this Windows Server VM to install the management tools that let you administer the managed domain. If you don't want to continue in this tutorial series, review the following clean up steps to delete the VM. Otherwise, continue to the next tutorial.

Unjoin the VM from the managed domain

To remove the VM from the managed domain, follow through the steps again to join the VM to a domain. Instead of joining the managed domain, choose to join a workgroup, such as the default *WORKGROUP*. After the VM has rebooted, the computer object is removed from the managed domain.

If you delete the VM without unjoining from the domain, an orphaned computer object is left in Azure AD DS.

Delete the VM

If you're not going use this Windows Server VM, delete the VM using the following steps:

1. From the left-hand menu, select **Resource groups**
2. Choose your resource group, such as *myResourceGroup*.
3. Choose your VM, such as *myVM*, then select **Delete**. Select **Yes** to confirm the resource deletion. It takes a few minutes to delete the VM.

4. When the VM is deleted, select the OS disk, network interface card, and any other resources with the *myVM-* prefix and delete them.

Troubleshoot domain-join issues

The Windows Server VM should successfully join to the managed domain, the same way as a regular on-premises computer would join an Active Directory Domain Services domain. If the Windows Server VM can't join the managed domain, that indicates there's a connectivity or credentials-related issue. Review the following troubleshooting sections to successfully join the managed domain.

Connectivity issues

If you don't receive a prompt that asks for credentials to join the domain, there's a connectivity problem. The VM can't reach the managed domain on the virtual network.

After trying each of these troubleshooting steps, try to join the Windows Server VM to the managed domain again.

- Verify the VM is connected to the same virtual network that Azure AD DS is enabled in, or has a peered network connection.
- Try to ping the DNS domain name of the managed domain, such as `ping aaddscontoso.com`.
  - If the ping request fails, try to ping the IP addresses for the managed domain, such as `ping 10.0.0.4`. The IP address for your environment is displayed on the *Properties* page when you select the managed domain from your list of Azure resources.
  - If you can ping the IP address but not the domain, DNS may be incorrectly configured. Confirm that the IP addresses of the managed domain are configured as DNS servers for the virtual network.
- Try to flush the DNS resolver cache on the virtual machine using the `ipconfig /flushdns` command.

Credentials-related issues

If you receive a prompt that asks for credentials to join the domain, but then an error after you enter those credentials, the VM is able to connect to the managed domain. The credentials you provided don't then let the VM join the managed domain.

After trying each of these troubleshooting steps, try to join the Windows Server VM to the managed domain again.

- Make sure that the user account you specify belongs to the managed domain.
- Confirm that the account is part of the managed domain or Azure AD tenant. Accounts from external directories associated with your Azure AD tenant can't correctly authenticate during the domain-join process.
- Try using the UPN format to specify credentials, such as `contosoadmin@aaddscontoso.onmicrosoft.com`. If there are many users with the same UPN prefix in your tenant or if your UPN prefix is overly long, the *SAMAccountName* for your account may be autogenerated. In these cases, the *SAMAccountName* format for your account may be different from what you expect or use in your on-premises domain.
- Check that you have enabled password synchronization to your managed domain. Without this configuration step, the required password hashes won't be present in the managed domain to correctly authenticate your sign-in attempt.
- Wait for password synchronization to be completed. When a user account's password is changed, an automatic background synchronization from Azure AD updates the password in Azure AD DS. It takes some time for the password to be available for domain-join use.

Next steps

In this tutorial, you learned how to:

- Create a Windows Server VM

- Connect to the Windows Server VM to an Azure virtual network
- Join the VM to the managed domain

To administer your managed domain, configure a management VM using the Active Directory Administrative Center (ADAC).

Install administration tools on a management VM

Tutorial: Create a management VM to configure and administer an Azure Active Directory Domain Services managed domain

7/20/2020 • 6 minutes to read • Edit Online

Azure Active Directory Domain Services (Azure AD DS) provides managed domain services such as domain join, group policy, LDAP, and Kerberos/NTLM authentication that is fully compatible with Windows Server Active Directory. You administer this managed domain using the same Remote Server Administration Tools (RSAT) as with an on-premises Active Directory Domain Services domain. As Azure AD DS is a managed service, there are some administrative tasks that you can't perform, such as using remote desktop protocol (RDP) to connect to the domain controllers.

This tutorial shows you how to configure a Windows Server VM in Azure and install the required tools to administer an Azure AD DS managed domain.

In this tutorial, you learn how to:

- Understand the available administrative tasks in a managed domain
- Install the Active Directory administrative tools on a Windows Server VM
- Use the Active Directory Administrative Center to perform common tasks

If you don't have an Azure subscription, create an account before you begin.

Prerequisites

To complete this tutorial, you need the following resources and privileges:

- An active Azure subscription.
  - If you don't have an Azure subscription, create an account.
- An Azure Active Directory tenant associated with your subscription, either synchronized with an on-premises directory or a cloud-only directory.
  - If needed, create an Azure Active Directory tenant or associate an Azure subscription with your account.
- An Azure Active Directory Domain Services managed domain enabled and configured in your Azure AD tenant.
  - If needed, see the first tutorial to create and configure an Azure Active Directory Domain Services managed domain.
- A Windows Server VM that is joined to the managed domain.
  - If needed, see the previous tutorial to create a Windows Server VM and join it to a managed domain.
- A user account that's a member of the *Azure AD DC administrators* group in your Azure AD tenant.
- An Azure Bastion host deployed in your Azure AD DS virtual network.
  - If needed, create an Azure Bastion host.

Sign in to the Azure portal

In this tutorial, you create and configure a management VM using the Azure portal. To get started, first sign in to the Azure portal.

Available administrative tasks in Azure AD DS

Azure AD DS provides a managed domain for your users, applications, and services to consume. This approach changes some of the available management tasks you can do, and what privileges you have within the managed domain. These tasks and permissions may be different than what you experience with a regular on-premises Active Directory Domain Services environment. You also can't connect to domain controllers on the managed domain using Remote Desktop.

Administrative tasks you can perform on a managed domain

Members of the *AAD DC Administrators* group are granted privileges on the managed domain that enables them to do tasks such as:

- Configure the built-in group policy object (GPO) for the *AADDC Computers* and *AADDC Users* containers in the managed domain.
- Administer DNS on the managed domain.
- Create and administer custom organizational units (OUs) on the managed domain.
- Gain administrative access to computers joined to the managed domain.

Administrative privileges you don't have on a managed domain

The managed domain is locked down, so you don't have privileges to do certain administrative tasks on the domain. Some of the following examples are tasks you can't do:

- Extend the schema of the managed domain.
- Connect to domain controllers for the managed domain using Remote Desktop.
- Add domain controllers to the managed domain.
- You don't have *Domain Administrator* or *Enterprise Administrator* privileges for the managed domain.

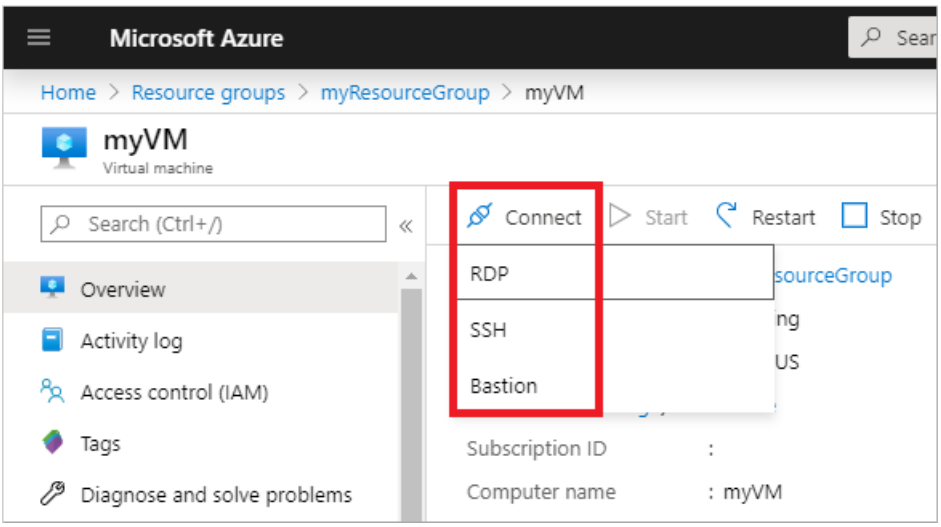
Sign in to the Windows Server VM

In the previous tutorial, a Windows Server VM was created and joined to the managed domain. Use that VM to install the management tools. If needed, follow the steps in the tutorial to create and join a Windows Server VM to a managed domain.

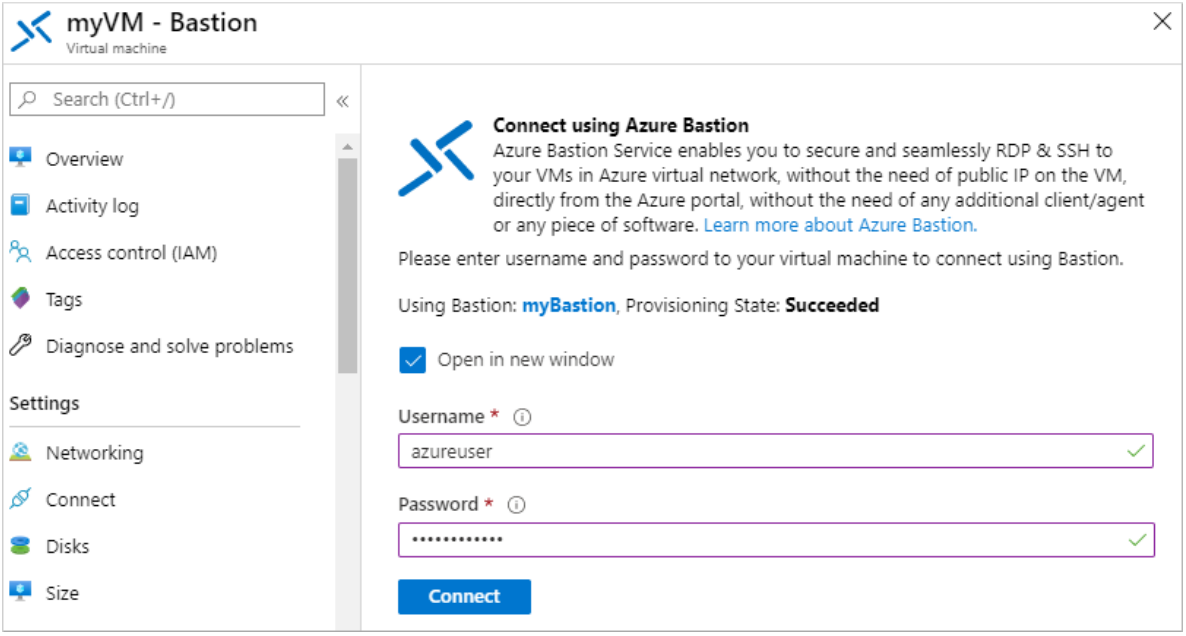
> **NOTE**
>
> In this tutorial, you use a Windows Server VM in Azure that is joined to the managed domain. You can also use a Windows client, such as Windows 10, that is joined to the managed domain.
>
> For more information on how to install the administrative tools on a Windows client, see install Remote Server Administration Tools (RSAT)

To get started, connect to the Windows Server VM as follows:

1. In the Azure portal, select **Resource groups** on the left-hand side. Choose the resource group where your VM was created, such as *myResourceGroup*, then select the VM, such as *myVM*.
2. In the **Overview** pane for your VM, select **Connect**, then **Bastion**.



3. Enter the credentials for your VM, then select **Connect**.



If needed, allow your web browser to open pop-ups for the Bastion connection to be displayed. It takes a few seconds to make the connection to your VM.

Install Active Directory administrative tools

You use the same administrative tools in a managed domain as on-premises AD DS environments, such as the Active Directory Administrative Center (ADAC) or AD PowerShell. These tools can be installed as part of the Remote Server Administration Tools (RSAT) feature on Windows Server and client computers. Members of the *AAD DC Administrators* group can then administer managed domains remotely using these AD administrative tools from a computer that is joined to the managed domain.

To install the Active Directory Administration tools on a domain-joined VM, complete the following steps:

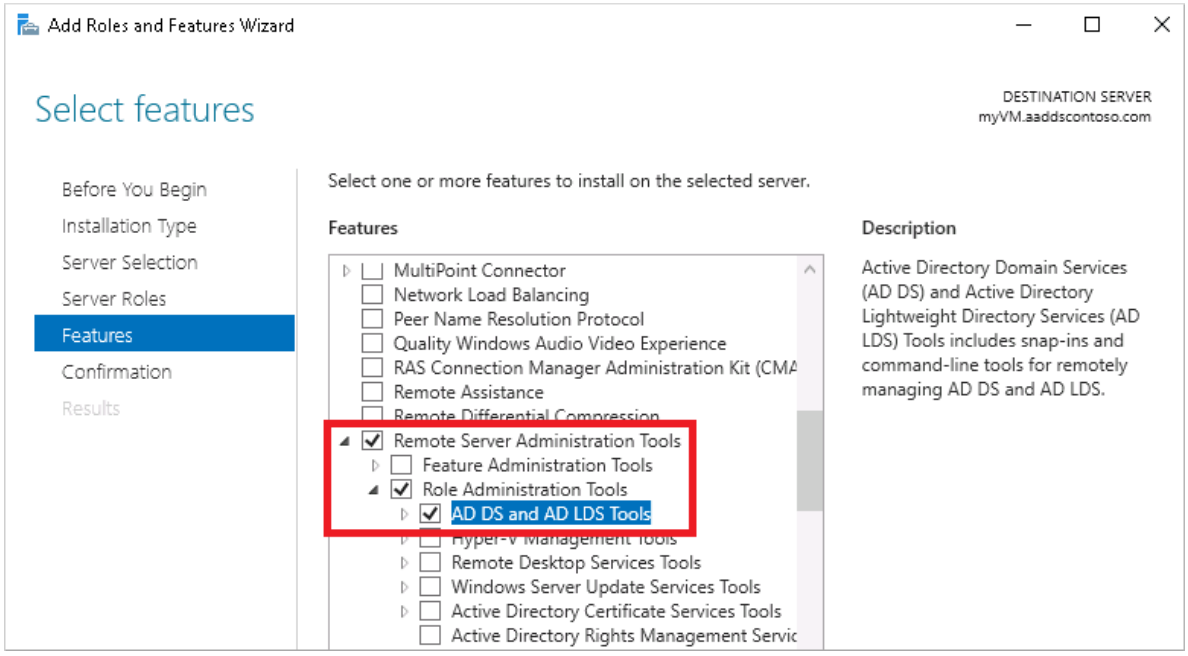
1. If **Server Manager** doesn't open by default when you sign in to the VM, select the **Start** menu, then choose **Server Manager**.
2. In the *Dashboard* pane of the **Server Manager** window, select **Add Roles and Features**.
3. On the **Before You Begin** page of the *Add Roles and Features Wizard*, select **Next**.
4. For the *Installation Type*, leave the **Role-based or feature-based installation** option checked and select **Next**.

5. On the **Server Selection** page, choose the current VM from the server pool, such as *myvm.aaddscontoso.com*, then select **Next**.

6. On the **Server Roles** page, click **Next**.

7. On the **Features** page, expand the **Remote Server Administration Tools** node, then expand the **Role Administration Tools** node.

   Choose **AD DS and AD LDS Tools** feature from the list of role administration tools, then select **Next**.



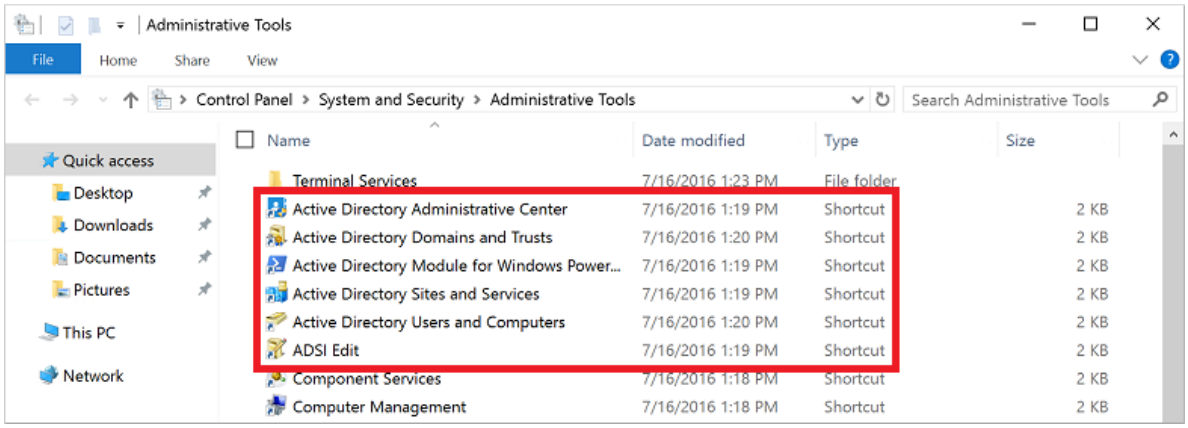
8. On the **Confirmation** page, select **Install**. It may take a minute or two to install the administrative tools.

9. When feature installation is complete, select **Close** to exit the **Add Roles and Features** wizard.

Use Active Directory administrative tools

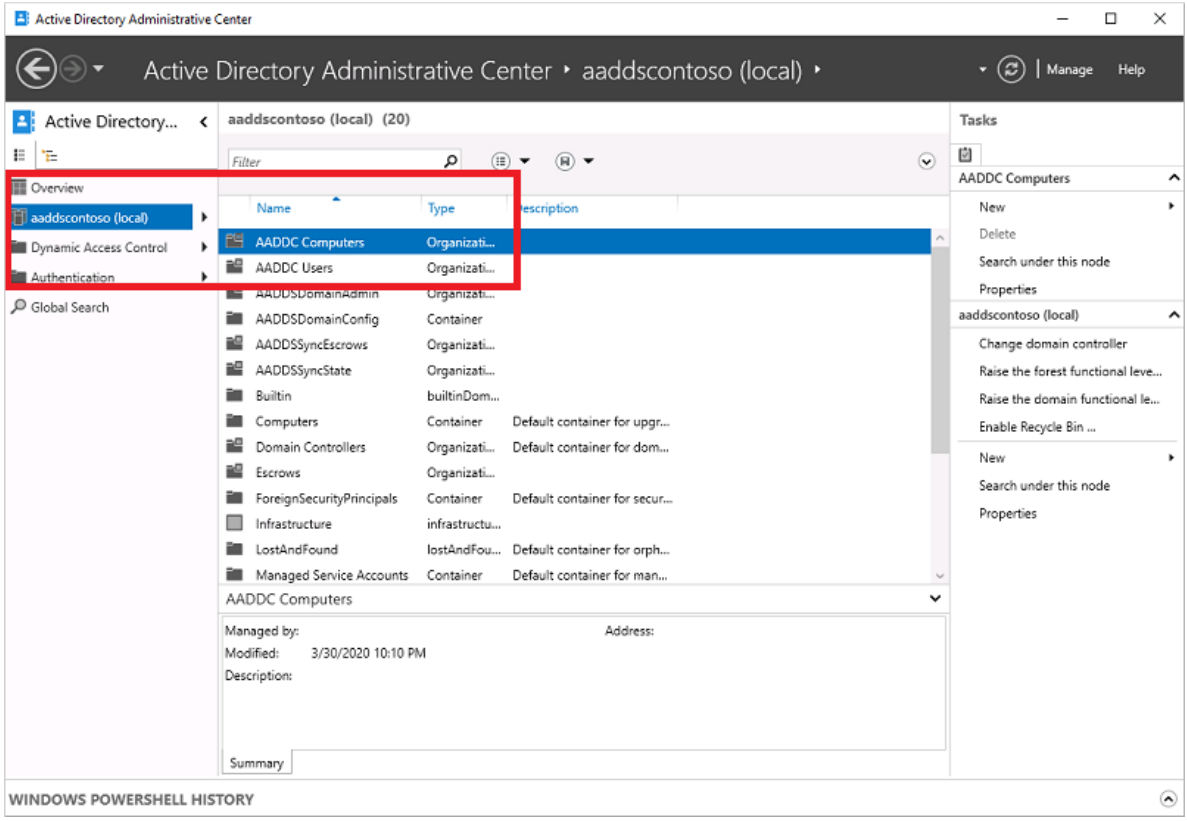
With the administrative tools installed, let's see how to use them to administer the managed domain. Make sure that you're signed in to the VM with a user account that's a member of the *AAD DC Administrators* group.

1. From the **Start** menu, select **Windows Administrative Tools**. The AD administrative tools installed in the previous step are listed.



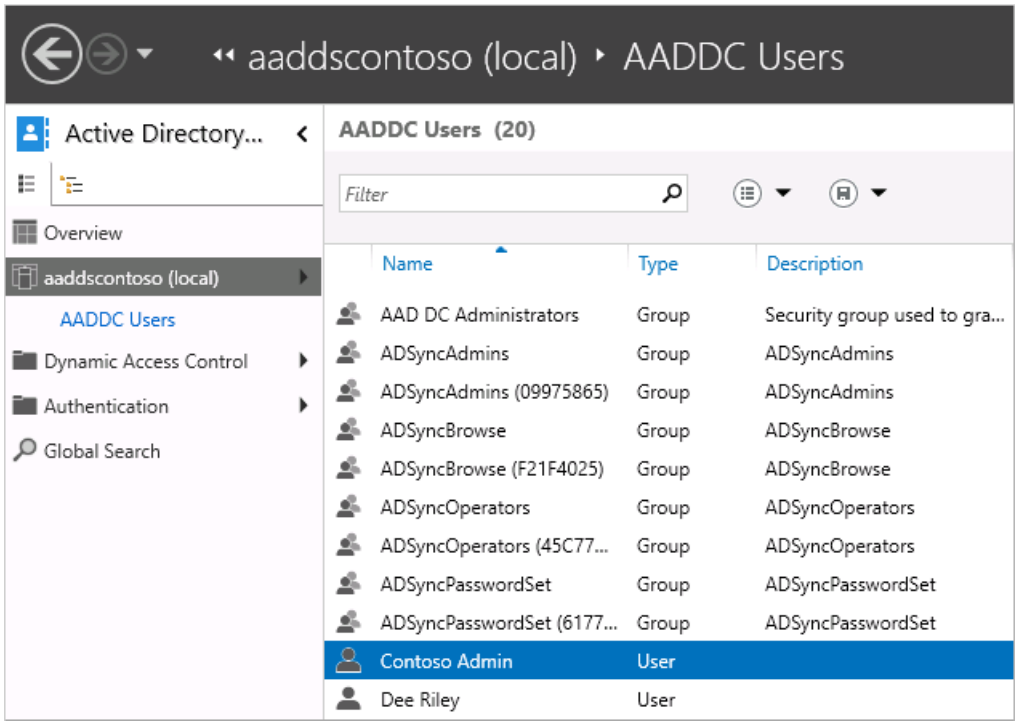
2. Select **Active Directory Administrative Center**.

3. To explore the managed domain, choose the domain name in the left pane, such as *aaddscontoso*. Two containers named *AADDC Computers* and *AADDC Users* are at the top of the list.



4. To see the users and groups that belong to the managed domain, select the **AADDC Users** container. The user accounts and groups from your Azure AD tenant are listed in this container.

   In the following example output, a user account named *Contoso Admin* and a group for *AAD DC Administrators* are shown in this container.



5. To see the computers that are joined to the managed domain, select the **AADDC Computers** container. An entry for the current virtual machine, such as *myVM*, is listed. Computer accounts for all devices that are joined to the managed domain are stored in this *AADDC Computers* container.

Common Active Directory Administrative Center actions such as resetting a user account password or managing group membership are available. These actions only work for users and groups created directly in the managed domain. Identity information only synchronizes *from* Azure AD to Azure AD DS. There's no write back from Azure

AD DS to Azure AD. You can't change passwords or managed group membership for users synchronized from Azure AD and have those changes synchronized back.

You can also use the *Active Directory Module for Windows PowerShell*, installed as part of the administrative tools, to manage common actions in your managed domain.

Next steps

In this tutorial, you learned how to:

- Understand the available administrative tasks in a managed domain
- Install the Active Directory administrative tools on a Windows Server VM
- Use the Active Directory Administrative Center to perform common tasks

To safely interact with your managed domain from other applications, enable secure Lightweight Directory Access Protocol (LDAPS).

Configure secure LDAP for your managed domain

Tutorial: Configure secure LDAP for an Azure Active Directory Domain Services managed domain

7/20/2020 • 15 minutes to read • Edit Online

To communicate with your Azure Active Directory Domain Services (Azure AD DS) managed domain, the Lightweight Directory Access Protocol (LDAP) is used. By default, the LDAP traffic isn't encrypted, which is a security concern for many environments.

With Azure AD DS, you can configure the managed domain to use secure Lightweight Directory Access Protocol (LDAPS). When you use secure LDAP, the traffic is encrypted. Secure LDAP is also known as LDAP over Secure Sockets Layer (SSL) / Transport Layer Security (TLS).

This tutorial shows you how to configure LDAPS for an Azure AD DS managed domain.

In this tutorial, you learn how to:

- Create a digital certificate for use with Azure AD DS
- Enable secure LDAP for Azure AD DS
- Configure secure LDAP for use over the public internet
- Bind and test secure LDAP for a managed domain

If you don't have an Azure subscription, create an account before you begin.

Prerequisites

To complete this tutorial, you need the following resources and privileges:

- An active Azure subscription.
  - If you don't have an Azure subscription, create an account.
- An Azure Active Directory tenant associated with your subscription, either synchronized with an on-premises directory or a cloud-only directory.
  - If needed, create an Azure Active Directory tenant or associate an Azure subscription with your account.
- An Azure Active Directory Domain Services managed domain enabled and configured in your Azure AD tenant.
  - If needed, create and configure an Azure Active Directory Domain Services managed domain.
- The *LDP.exe* tool installed on your computer.
  - If needed, install the Remote Server Administration Tools (RSAT) for *Active Directory Domain Services and LDAP*.

Sign in to the Azure portal

In this tutorial, you configure secure LDAP for the managed domain using the Azure portal. To get started, first sign in to the Azure portal.

Create a certificate for secure LDAP

To use secure LDAP, a digital certificate is used to encrypt the communication. This digital certificate is applied to your managed domain, and lets tools like *LDP.exe* use secure encrypted communication when querying data. There are two ways to create a certificate for secure LDAP access to the managed domain:

- A certificate from a public certificate authority (CA) or an enterprise CA.

    - If your organization gets certificates from a public CA, get the secure LDAP certificate from that public CA. If you use an enterprise CA in your organization, get the secure LDAP certificate from the enterprise CA.
    - A public CA only works when you use a custom DNS name with your managed domain. If the DNS domain name of your managed domain ends in *.onmicrosoft.com*, you can't create a digital certificate to secure the connection with this default domain. Microsoft owns the *.onmicrosoft.com* domain, so a public CA won't issue a certificate. In this scenario, create a self-signed certificate and use that to configure secure LDAP.
- A self-signed certificate that you create yourself.
    - This approach is good for testing purposes, and is what this tutorial shows.

The certificate you request or create must meet the following requirements. Your managed domain encounters problems if you enable secure LDAP with an invalid certificate:

- **Trusted issuer** - The certificate must be issued by an authority trusted by computers connecting to the managed domain using secure LDAP. This authority may be a public CA or an Enterprise CA trusted by these computers.
- **Lifetime** - The certificate must be valid for at least the next 3-6 months. Secure LDAP access to your managed domain is disrupted when the certificate expires.
- **Subject name** - The subject name on the certificate must be your managed domain. For example, if your domain is named *aaddscontoso.com*, the certificate's subject name must be **.aaddscontoso.com*.
    - The DNS name or subject alternate name of the certificate must be a wildcard certificate to ensure the secure LDAP works properly with the Azure AD Domain Services. Domain Controllers use random names and can be removed or added to ensure the service remains available.
- **Key usage** - The certificate must be configured for *digital signatures* and *key encipherment*.
- **Certificate purpose** - The certificate must be valid for TLS server authentication.

There are several tools available to create self-signed certificate such as OpenSSL, Keytool, MakeCert, New-SelfSignedCertificate cmdlet, etc.

In this tutorial, let's create a self-signed certificate for secure LDAP using the New-SelfSignedCertificate cmdlet.

Open a PowerShell window as **Administrator** and run the following commands. Replace the *$dnsName* variable with the DNS name used by your own managed domain, such as *aaddscontoso.com*:

```
# Define your own DNS name used by your managed domain
$dnsName="aaddscontoso.com"

# Get the current date to set a one-year expiration
$lifetime=Get-Date

# Create a self-signed certificate for use with Azure AD DS
New-SelfSignedCertificate -Subject *.$dnsName `
  -NotAfter $lifetime.AddDays(365) -KeyUsage DigitalSignature, KeyEncipherment `
  -Type SSLServerAuthentication -DnsName *.$dnsName, $dnsName
```

The following example output shows that the certificate was successfully generated and is stored in the local certificate store (*LocalMachine\MY*):

```
PS C:\WINDOWS\system32> New-SelfSignedCertificate -Subject *.$dnsName `
>>   -NotAfter $lifetime.AddDays(365) -KeyUsage DigitalSignature, KeyEncipherment `
>>   -Type SSLServerAuthentication -DnsName *.$dnsName, $dnsName.com

   PSParentPath: Microsoft.PowerShell.Security\Certificate::LocalMachine\MY

Thumbprint                                Subject
----------                                -------
959BD1531A1E674EB09E13BD8534B2C76A45B3E6  CN=aaddscontoso.com
```

Understand and export required certificates

To use secure LDAP, the network traffic is encrypted using public key infrastructure (PKI).

- A **private** key is applied to the managed domain.
  - This private key is used to *decrypt* the secure LDAP traffic. The private key should only be applied to the managed domain and not widely distributed to client computers.
  - A certificate that includes the private key uses the *.PFX* file format.
- A **public** key is applied to the client computers.
  - This public key is used to *encrypt* the secure LDAP traffic. The public key can be distributed to client computers.
  - Certificates without the private key use the *.CER* file format.

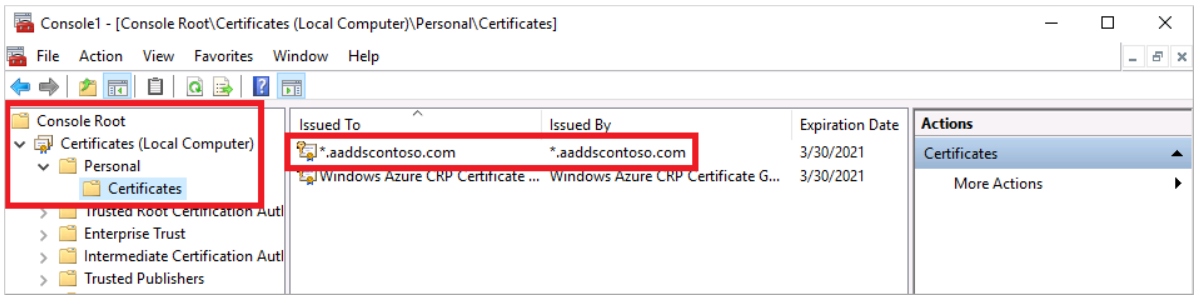
These two keys, the *private* and *public* keys, make sure that only the appropriate computers can successfully communicate with each other. If you use a public CA or enterprise CA, you are issued with a certificate that includes the private key and can be applied to a managed domain. The public key should already be known and trusted by client computers.

In this tutorial, you created a self-signed certificate with the private key, so you need to export the appropriate private and public components.

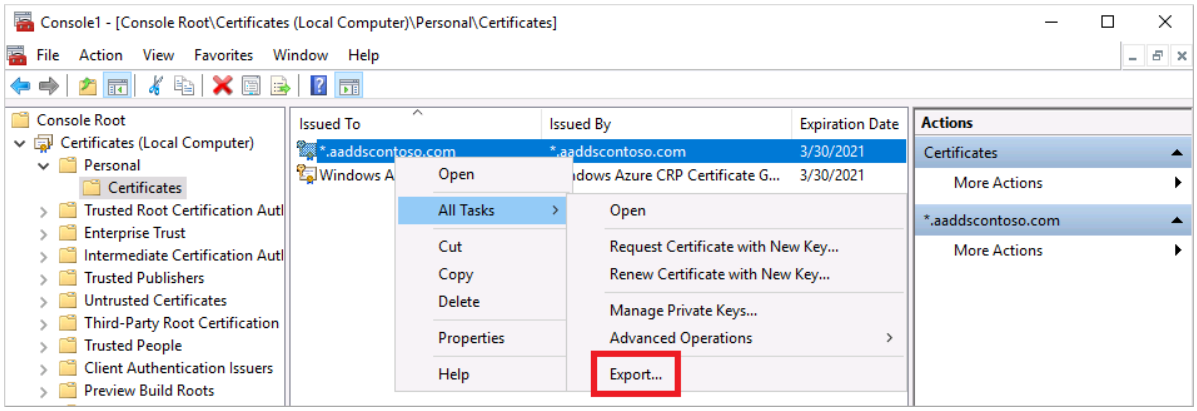
Export a certificate for Azure AD DS

Before you can use the digital certificate created in the previous step with your managed domain, export the certificate to a *.PFX* certificate file that includes the private key.

1. To open the *Run* dialog, select the **Windows** + **R** keys.
2. Open the Microsoft Management Console (MMC) by entering **mmc** in the *Run* dialog, then select **OK**.
3. On the **User Account Control** prompt, then select **Yes** to launch MMC as administrator.
4. From the **File** menu, select **Add/Remove Snap-in...**
5. In the **Certificates snap-in** wizard, choose **Computer account**, then select **Next**.
6. On the **Select Computer** page, choose **Local computer: (the computer this console is running on)**, then select **Finish**.
7. In the **Add or Remove Snap-ins** dialog, select **OK** to add the certificates snap-in to MMC.
8. In the MMC window, expand **Console Root**. Select **Certificates (Local Computer)**, then expand the **Personal** node, followed by the **Certificates** node.



9. The self-signed certificate created in the previous step is shown, such as *aaddscontoso.com*. Right-select this certificate, then choose **All Tasks** > **Export...**



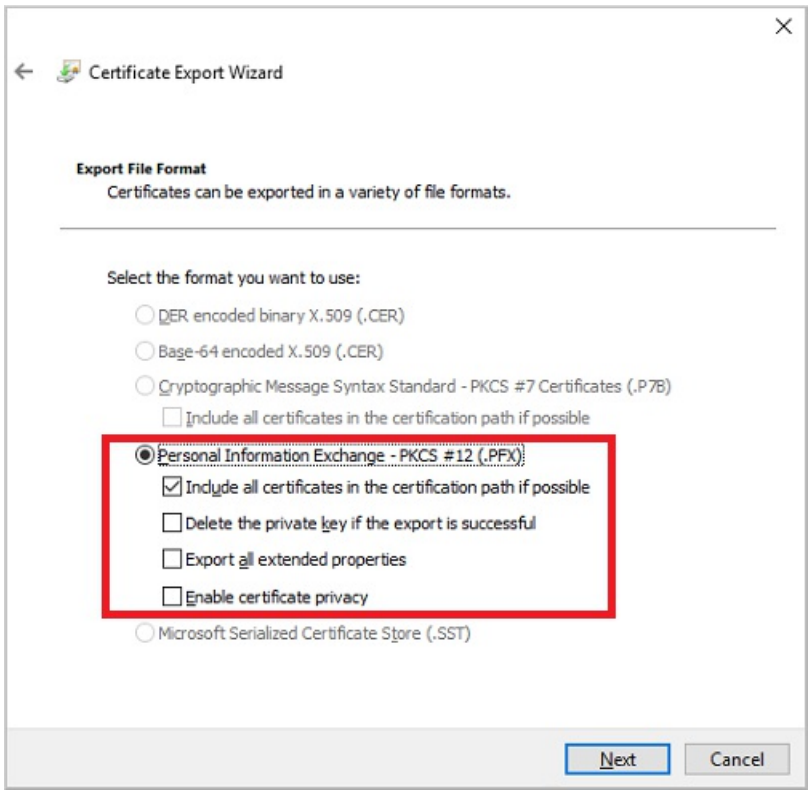
10. In the **Certificate Export Wizard**, select **Next**.

11. The private key for the certificate must be exported. If the private key is not included in the exported certificate, the action to enable secure LDAP for your managed domain fails.

    On the **Export Private Key** page, choose **Yes, export the private key**, then select **Next**.

12. Managed domains only support the *.PFX* certificate file format that includes the private key. Don't export the certificate as *.CER* certificate file format without the private key.

    On the **Export File Format** page, select **Personal Information Exchange - PKCS #12 (.PFX)** as the file format for the exported certificate. Check the box for *Include all certificates in the certification path if possible*.



13. As this certificate is used to decrypt data, you should carefully control access. A password can be used to protect the use of the certificate. Without the correct password, the certificate can't be applied to a service.

    On the **Security** page, choose the option for **Password** to protect the *.PFX* certificate file. Enter and confirm a password, then select **Next**. This password is used in the next section to enable secure LDAP for your managed domain.

14. On the **File to Export** page, specify the file name and location where you'd like to export the certificate, such as *C:\Users\accountname\azure-ad-ds.pfx*. Keep a note of the password and location of the *.PFX* file as this information would be required in next steps.

15. On the review page, select **Finish** to export the certificate to a *.PFX* certificate file. A confirmation dialog is displayed when the certificate has been successfully exported.

16. Leave the MMC open for use in the following section.

Export a certificate for client computers

Client computers must trust the issuer of the secure LDAP certificate to be able to connect successfully to the managed domain using LDAPS. The client computers need a certificate to successfully encrypt data that is decrypted by Azure AD DS. If you use a public CA, the computer should automatically trust these certificate issuers and have a corresponding certificate.

In this tutorial you use a self-signed certificate, and generated a certificate that includes the private key in the previous step. Now let's export and then install the self-signed certificate into the trusted certificate store on the client computer:

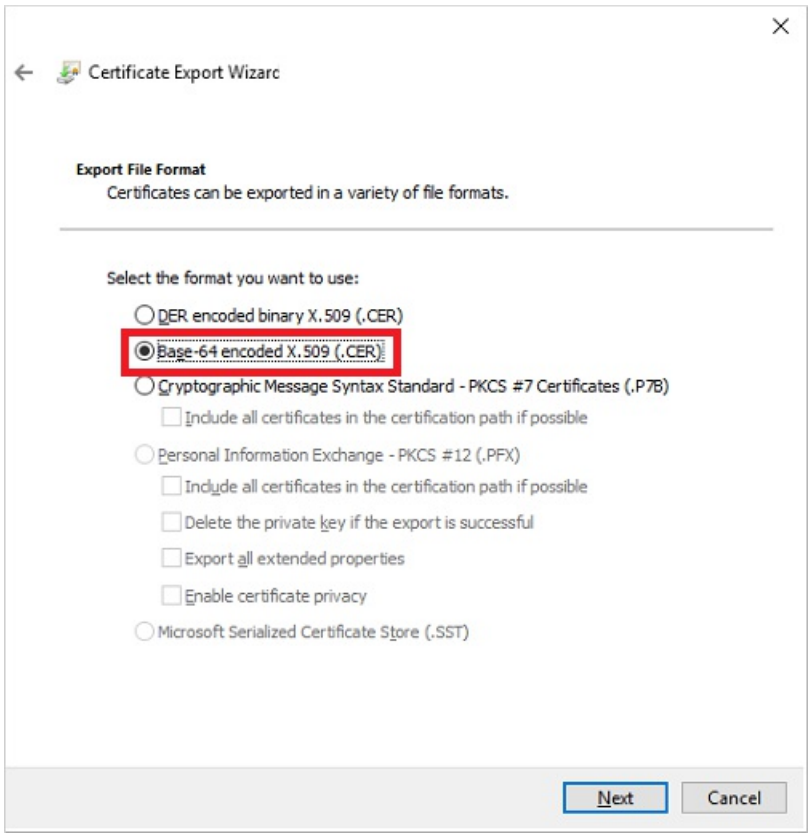
1. Go back to the MMC for *Certificates (Local Computer) > Personal > Certificates* store. The self-signed certificate created in a previous step is shown, such as *aaddscontoso.com*. Right-select this certificate, then choose **All Tasks > Export...**

2. In the **Certificate Export Wizard**, select **Next**.

3. As you don't need the private key for clients, on the **Export Private Key** page choose **No, do not export the private key**, then select **Next**.

4. On the **Export File Format** page, select **Base-64 encoded X.509 (.CER)** as the file format for the

exported certificate:



5. On the **File to Export** page, specify the file name and location where you'd like to export the certificate, such as *C:\Users\accountname\azure-ad-ds-client.cer*.

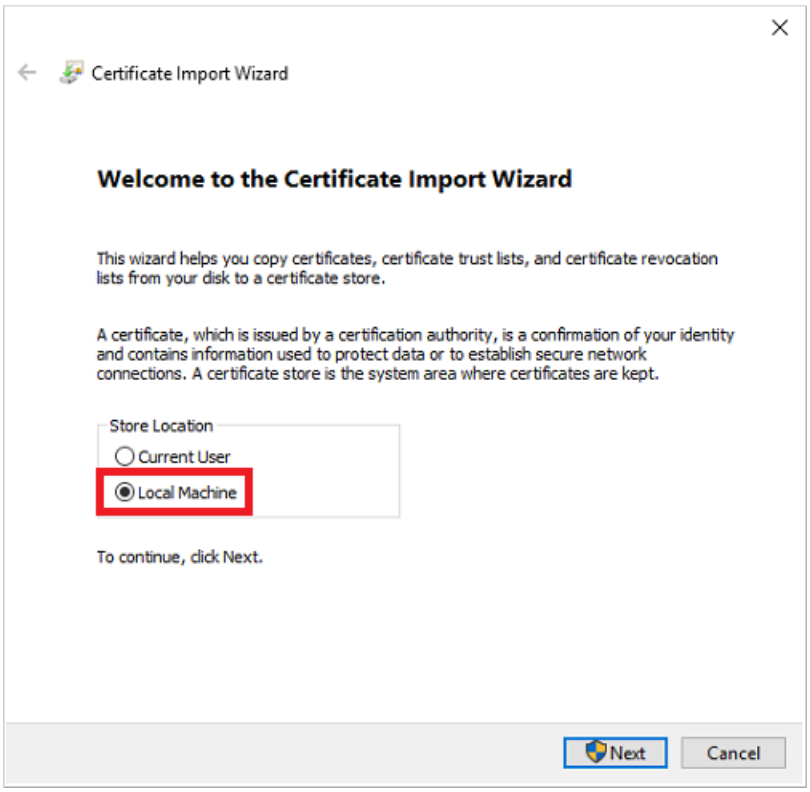
6. On the review page, select **Finish** to export the certificate to a *.CER* certificate file. A confirmation dialog is displayed when the certificate has been successfully exported.

The *.CER* certificate file can now be distributed to client computers that need to trust the secure LDAP connection to the managed domain. Let's install the certificate on the local computer.

1. Open File Explorer and browse to the location where you saved the *.CER* certificate file, such as *C:\Users\accountname\azure-ad-ds-client.cer*.

2. Right-select the *.CER* certificate file, then choose **Install Certificate**.

3. In the **Certificate Import Wizard**, choose to store the certificate in the *Local machine*, then select **Next**:



4. When prompted, choose **Yes** to allow the computer to make changes.
5. Choose to **Automatically select the certificate store based on the type of certificate**, then select **Next**.
6. On the review page, select **Finish** to import the *.CER* certificate. file A confirmation dialog is displayed when the certificate has been successfully imported.

Enable secure LDAP for Azure AD DS

With a digital certificate created and exported that includes the private key, and the client computer set to trust the connection, now enable secure LDAP on your managed domain. To enable secure LDAP on a managed domain, perform the following configuration steps:

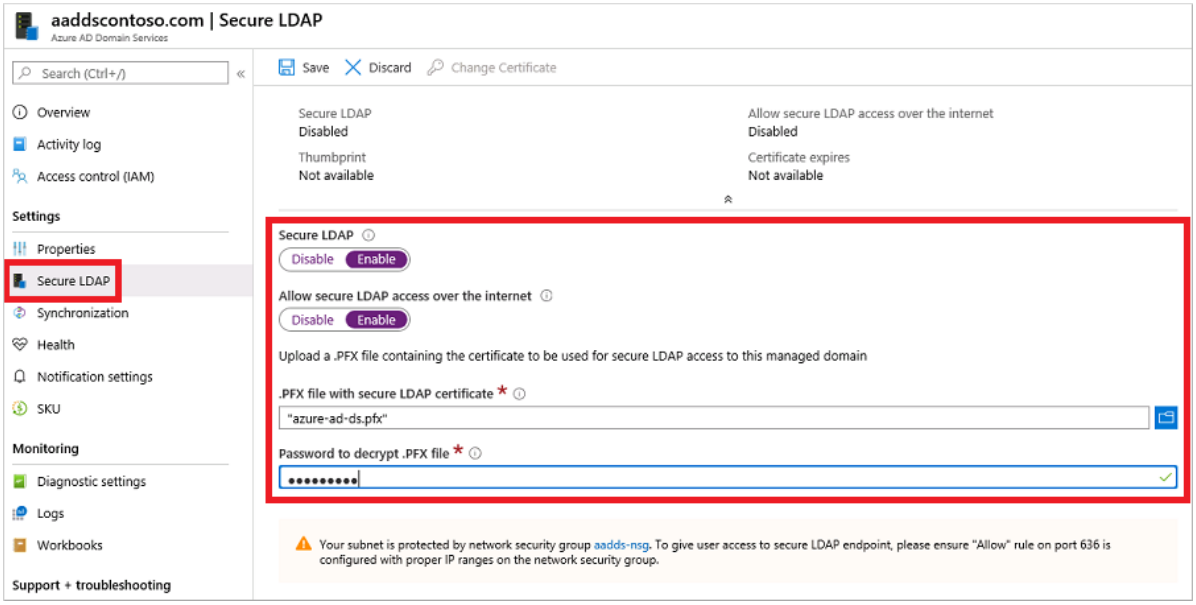
1. In the Azure portal, enter *domain services* in the **Search resources** box. Select **Azure AD Domain Services** from the search result.
2. Choose your managed domain, such as *aaddscontoso.com*.
3. On the left-hand side of the Azure AD DS window, choose **Secure LDAP**.
4. By default, secure LDAP access to your managed domain is disabled. Toggle **Secure LDAP** to **Enable**.
5. Secure LDAP access to your managed domain over the internet is disabled by default. When you enable public secure LDAP access, your domain is susceptible to password brute force attacks over the internet. In the next step, a network security group is configured to lock down access to only the required source IP address ranges.

   Toggle **Allow secure LDAP access over the internet** to **Enable**.
6. Select the folder icon next to **.PFX file with secure LDAP certificate**. Browse to the path of the *.PFX* file, then select the certificate created in a previous step that includes the private key.

> **IMPORTANT**
>
> As noted in the previous section on certificate requirements, you can't use a certificate from a public CA with the default *.onmicrosoft.com* domain. Microsoft owns the *.onmicrosoft.com* domain, so a public CA won't issue a certificate.
>
> Make sure your certificate is in the appropriate format. If it's not, the Azure platform generates certificate validation errors when you enable secure LDAP.

7. Enter the **Password to decrypt .PFX file** set in a previous step when the certificate was exported to a *.PFX* file.

8. Select **Save** to enable secure LDAP.



A notification is displayed that secure LDAP is being configured for the managed domain. You can't modify other settings for the managed domain until this operation is complete.

It takes a few minutes to enable secure LDAP for your managed domain. If the secure LDAP certificate you provide doesn't match the required criteria, the action to enable secure LDAP for the managed domain fails.

Some common reasons for failure are if the domain name is incorrect, or the certificate expires soon or has already expired. You can re-create the certificate with valid parameters, then enable secure LDAP using this updated certificate.

Lock down secure LDAP access over the internet

When you enable secure LDAP access over the internet to your managed domain, it creates a security threat. The managed domain is reachable from the internet on TCP port 636. It's recommended to restrict access to the managed domain to specific known IP addresses for your environment. An Azure network security group rule can be used to limit access to secure LDAP.

Let's create a rule to allow inbound secure LDAP access over TCP port 636 from a specified set of IP addresses. A default *DenyAll* rule with a lower priority applies to all other inbound traffic from the internet, so only the specified addresses can reach your managed domain using secure LDAP.

1. In the Azure portal, select *Resource groups* on the left-hand side navigation.

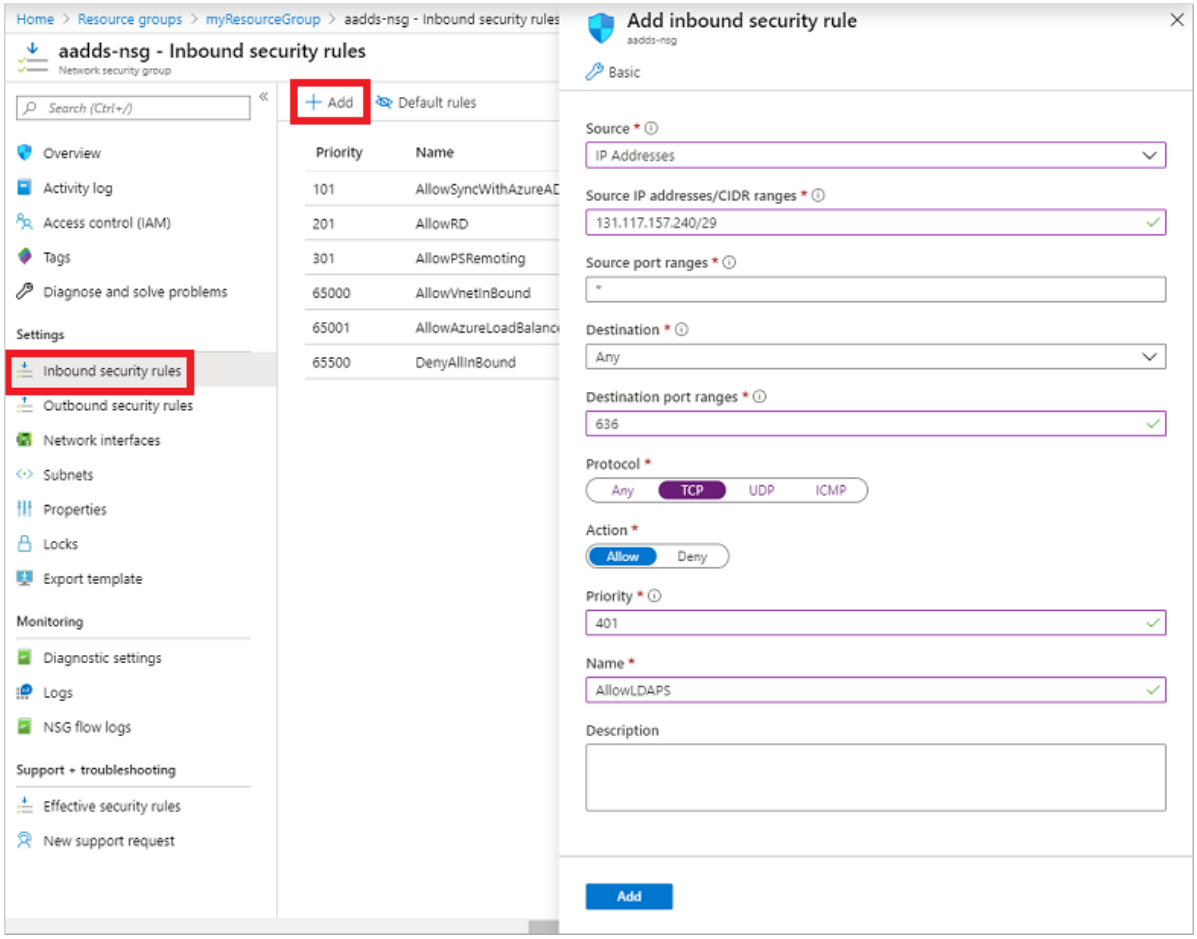
2. Choose your resource group, such as *myResourceGroup*, then select your network security group, such as *aaads-nsg*.

3. The list of existing inbound and outbound security rules are displayed. On the left-hand side of the network security group windows, choose **Settings** > **Inbound security rules**.

4. Select **Add**, then create a rule to allow *TCP* port *636*. For improved security, choose the source as *IP Addresses* and then specify your own valid IP address or range for your organization.

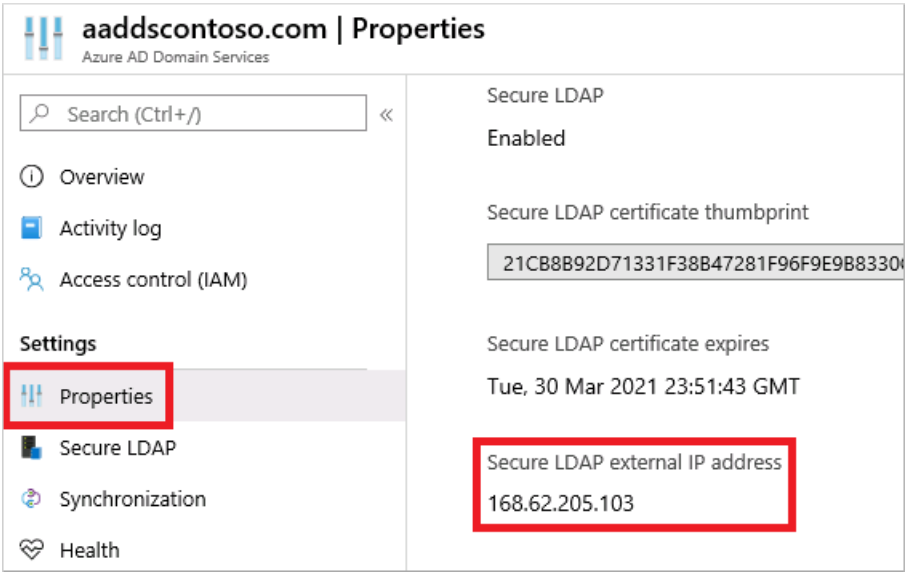
SETTING	VALUE
Source	IP Addresses
Source IP addresses / CIDR ranges	A valid IP address or range for your environment
Source port ranges	*
Destination	Any
Destination port ranges	636
Protocol	TCP
Action	Allow
Priority	401
Name	AllowLDAPS

5. When ready, select **Add** to save and apply the rule.



Configure DNS zone for external access

With secure LDAP access enabled over the internet, update the DNS zone so that client computers can find this managed domain. The *Secure LDAP external IP address* is listed on the **Properties** tab for your managed domain:



Configure your external DNS provider to create a host record, such as *ldaps*, to resolve to this external IP address. To test locally on your machine first, you can create an entry in the Windows hosts file. To successfully edit the hosts file on your local machine, open *Notepad* as an administrator, then open the file *C:\Windows\System32\drivers\etc\hosts*

The following example DNS entry, either with your external DNS provider or in the local hosts file, resolves traffic for *ldaps.aaddscontoso.com* to the external IP address of *168.62.205.103*:

```
168.62.205.103    ldaps.aaddscontoso.com
```

Test queries to the managed domain

To connect and bind to your managed domain and search over LDAP, you use the *LDP.exe* tool. This tool is included in the Remote Server Administration Tools (RSAT) package. For more information, see install Remote Server Administration Tools.

1. Open *LDP.exe* and connect to the managed domain. Select **Connection**, then choose **Connect...**.
2. Enter the secure LDAP DNS domain name of your managed domain created in the previous step, such as *ldaps.aaddscontoso.com*. To use secure LDAP, set **Port** to *636*, then check the box for **SSL**.
3. Select **OK** to connect to the managed domain.

Next, bind to your managed domain. Users (and service accounts) can't perform LDAP simple binds if you have disabled NTLM password hash synchronization on your managed domain. For more information on disabling NTLM password hash synchronization, see Secure your managed domain.

1. Select the **Connection** menu option, then choose **Bind...**.
2. Provide the credentials of a user account belonging to the *AAD DC Administrators* group, such as *contosoadmin*. Enter the user account's password, then enter your domain, such as *aaddscontoso.com*.
3. For **Bind type**, choose the option for *Bind with credentials*.
4. Select **OK** to bind to your managed domain.

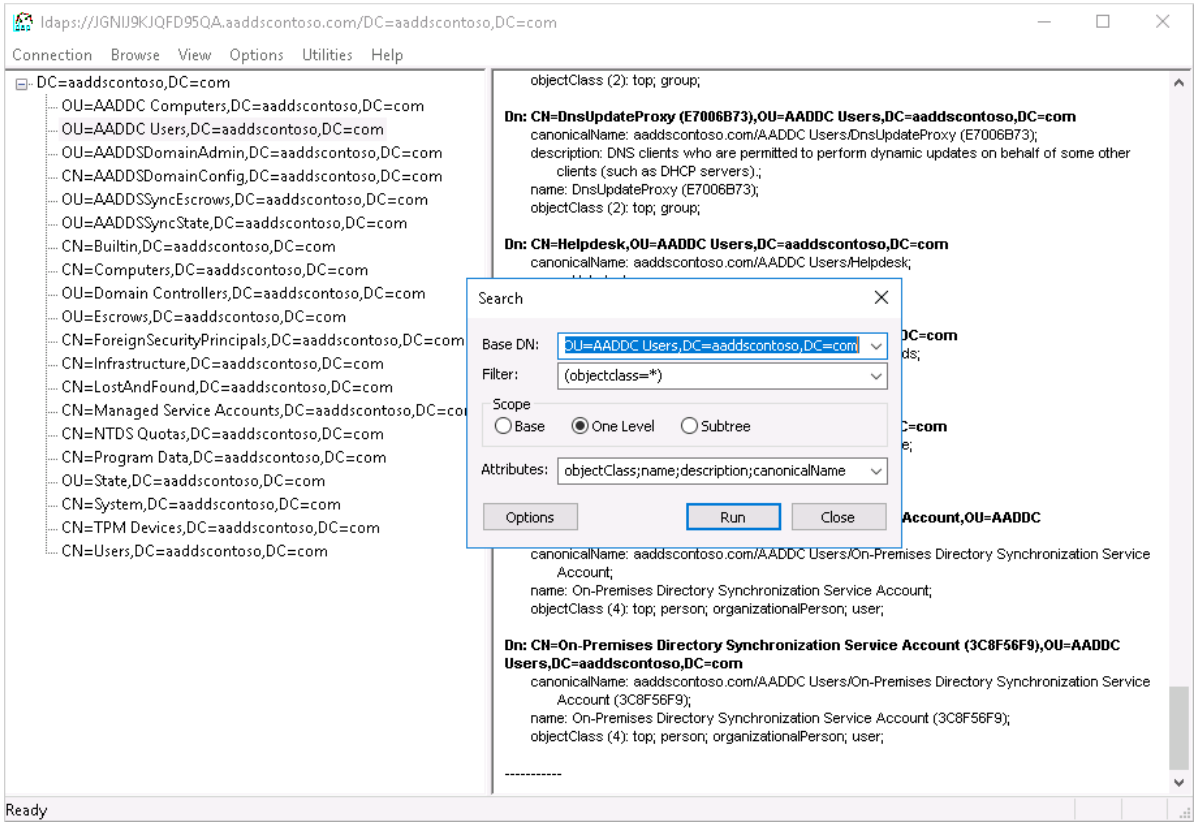
To see of the objects stored in your managed domain:

1. Select the **View** menu option, and then choose **Tree**.

2. Leave the *BaseDN* field blank, then select **OK**.

3. Choose a container, such as *AADDC Users*, then right-select the container and choose **Search**.

4. Leave the pre-populated fields set, then select **Run**. The results of the query are displayed in the right-hand window, as shown in the following example output:



To directly query a specific container, from the **View** > **Tree** menu, you can specify a **BaseDN** such as *OU=AADDC Users,DC=AADDSCONTOSO,DC=COM* or *OU=AADDC Computers,DC=AADDSCONTOSO,DC=COM*. For more information on how to format and create queries, see LDAP query basics.

Clean up resources

If you added a DNS entry to the local hosts file of your computer to test connectivity for this tutorial, remove this entry and add a formal record in your DNS zone. To remove the entry from the local hosts file, complete the following steps:

1. On your local machine, open *Notepad* as an administrator
2. Browse to and open the file *C:\Windows\System32\drivers\etc\hosts*
3. Delete the line for the record you added, such as `168.62.205.103 ldaps.aaddscontoso.com`

Next steps

In this tutorial, you learned how to:

- Create a digital certificate for use with Azure AD DS
- Enable secure LDAP for Azure AD DS
- Configure secure LDAP for use over the public internet
- Bind and test secure LDAP for a managed domain

Configure password hash synchronization for a hybrid Azure AD environment

Tutorial: Enable password synchronization in Azure Active Directory Domain Services for hybrid environments

7/20/2020 • 4 minutes to read • Edit Online

For hybrid environments, an Azure Active Directory (Azure AD) tenant can be configured to synchronize with an on-premises Active Directory Domain Services (AD DS) environment using Azure AD Connect. By default, Azure AD Connect doesn't synchronize legacy NT LAN Manager (NTLM) and Kerberos password hashes that are needed for Azure Active Directory Domain Services (Azure AD DS).

To use Azure AD DS with accounts synchronized from an on-premises AD DS environment, you need to configure Azure AD Connect to synchronize those password hashes required for NTLM and Kerberos authentication. After Azure AD Connect is configured, an on-premises account creation or password change event also then synchronizes the legacy password hashes to Azure AD.

You don't need to perform these steps if you use cloud-only accounts with no on-premises AD DS environment.

In this tutorial, you learn:

- Why legacy NTLM and Kerberos password hashes are needed
- How to configure legacy password hash synchronization for Azure AD Connect

If you don't have an Azure subscription, create an account before you begin.

Prerequisites

To complete this tutorial, you need the following resources:

- An active Azure subscription.
  - If you don't have an Azure subscription, create an account.
- An Azure Active Directory tenant associated with your subscription that's synchronized with an on-premises directory using Azure AD Connect.
  - If needed, create an Azure Active Directory tenant or associate an Azure subscription with your account.
  - If needed, enable Azure AD Connect for password hash synchronization.
- An Azure Active Directory Domain Services managed domain enabled and configured in your Azure AD tenant.
  - If needed, create and configure an Azure Active Directory Domain Services managed domain.

Password hash synchronization using Azure AD Connect

Azure AD Connect is used to synchronize objects like user accounts and groups from an on-premises AD DS environment into an Azure AD tenant. As part of the process, password hash synchronization enables accounts to use the same password in the on-prem AD DS environment and Azure AD.

To authenticate users on the managed domain, Azure AD DS needs password hashes in a format that's suitable for NTLM and Kerberos authentication. Azure AD doesn't store password hashes in the format that's required for NTLM or Kerberos authentication until you enable Azure AD DS for your tenant. For security reasons, Azure AD also doesn't store any password credentials in clear-text form. Therefore, Azure AD can't automatically generate these NTLM or Kerberos password hashes based on users' existing credentials.

Azure AD Connect can be configured to synchronize the required NTLM or Kerberos password hashes for Azure AD DS. Make sure that you have completed the steps to enable Azure AD Connect for password hash synchronization. If you had an existing instance of Azure AD Connect, download and update to the latest version to make sure you can synchronize the legacy password hashes for NTLM and Kerberos. This functionality isn't available in early releases of Azure AD Connect or with the legacy DirSync tool. Azure AD Connect version *1.1.614.0* or later is required.

> **IMPORTANT**
> Azure AD Connect should only be installed and configured for synchronization with on-premises AD DS environments. It's not supported to install Azure AD Connect in an Azure AD DS managed domain to synchronize objects back to Azure AD.

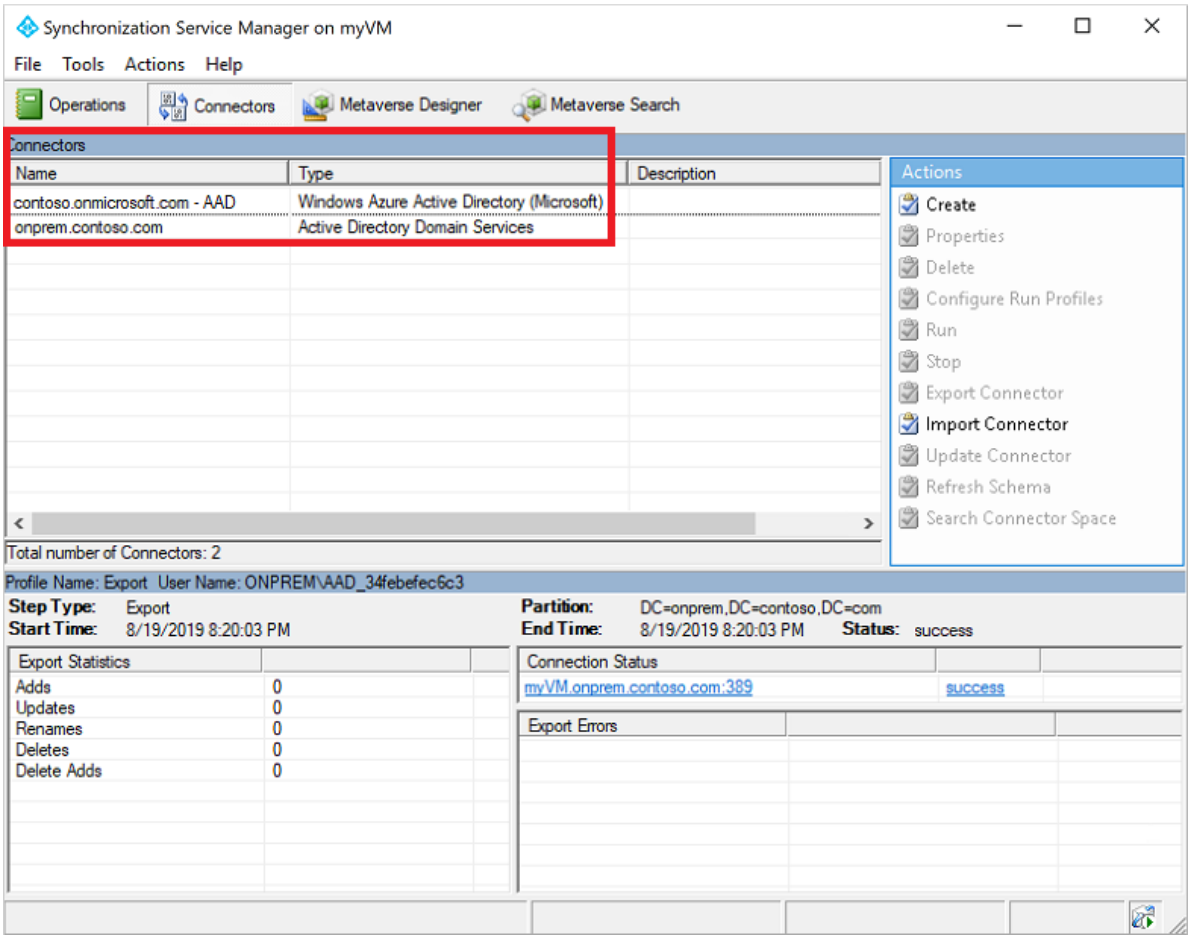
Enable synchronization of password hashes

With Azure AD Connect installed and configured to synchronize with Azure AD, now configure the legacy password hash sync for NTLM and Kerberos. A PowerShell script is used to configure the required settings and then start a full password synchronization to Azure AD. When that Azure AD Connect password hash synchronization process is complete, users can sign in to applications through Azure AD DS that use legacy NTLM or Kerberos password hashes.

1. On the computer with Azure AD Connect installed, from the Start menu, open the **Azure AD Connect** > **Synchronization Service**.

2. Select the **Connectors** tab. The connection information used to establish the synchronization between the on-premises AD DS environment and Azure AD are listed.

   The **Type** indicates either *Windows Azure Active Directory (Microsoft)* for the Azure AD connector or *Active Directory Domain Services* for the on-premises AD DS connector. Make a note of the connector names to use in the PowerShell script in the next step.



In this example screenshot, the following connectors are used:

- The Azure AD connector is named *contoso.onmicrosoft.com - AAD*
- The on-premises AD DS connector is named *onprem.contoso.com*

3. Copy and paste the following PowerShell script to the computer with Azure AD Connect installed. The script triggers a full password sync that includes legacy password hashes. Update the `$azureadConnector` and `$adConnector` variables with the connector names from the previous step.

   Run this script on each AD forest to synchronize on-premises account NTLM and Kerberos password hashes to Azure AD.

   ```
   # Define the Azure AD Connect connector names and import the required PowerShell module
   $azureadConnector = "<CASE SENSITIVE AZURE AD CONNECTOR NAME>"
   $adConnector = "<CASE SENSITIVE AD DS CONNECTOR NAME>"

   Import-Module "C:\Program Files\Microsoft Azure AD Sync\Bin\ADSync\ADSync.psd1"
   Import-Module "C:\Program Files\Microsoft Azure Active Directory
   Connect\AdSyncConfig\AdSyncConfig.psm1"

   # Create a new ForceFullPasswordSync configuration parameter object then
   # update the existing connector with this new configuration
   $c = Get-ADSyncConnector -Name $adConnector
   $p = New-Object Microsoft.IdentityManagement.PowerShell.ObjectModel.ConfigurationParameter
   "Microsoft.Synchronize.ForceFullPasswordSync", String, ConnectorGlobal, $null, $null, $null
   $p.Value = 1
   $c.GlobalParameters.Remove($p.Name)
   $c.GlobalParameters.Add($p)
   $c = Add-ADSyncConnector -Connector $c

   # Disable and re-enable Azure AD Connect to force a full password synchronization
   Set-ADSyncAADPasswordSyncConfiguration -SourceConnector $adConnector -TargetConnector $azureadConnector
   -Enable $false
   Set-ADSyncAADPasswordSyncConfiguration -SourceConnector $adConnector -TargetConnector $azureadConnector
   -Enable $true
   ```

   Depending on the size of your directory in terms of number of accounts and groups, synchronization of the legacy password hashes to Azure AD may take some time. The passwords are then synchronized to the managed domain after they've synchronized to Azure AD.

Next steps

In this tutorial, you learned:

- Why legacy NTLM and Kerberos password hashes are needed
- How to configure legacy password hash synchronization for Azure AD Connect

Learn how synchronization works in an Azure AD Domain Services managed domain

Tutorial: Create and configure an Azure Active Directory Domain Services managed domain with advanced configuration options

7/20/2020 • 15 minutes to read • Edit Online

Azure Active Directory Domain Services (Azure AD DS) provides managed domain services such as domain join, group policy, LDAP, Kerberos/NTLM authentication that is fully compatible with Windows Server Active Directory. You consume these domain services without deploying, managing, and patching domain controllers yourself. Azure AD DS integrates with your existing Azure AD tenant. This integration lets users sign in using their corporate credentials, and you can use existing groups and user accounts to secure access to resources.

You can create a managed domain using default configuration options for networking and synchronization, or manually define these settings. This tutorial shows you how to define those advanced configuration options to create and configure an Azure AD DS managed domain using the Azure portal.

In this tutorial, you learn how to:

- Configure DNS and virtual network settings for a managed domain
- Create a managed domain
- Add administrative users to domain management
- Enable password hash synchronization

If you don't have an Azure subscription, create an account before you begin.

Prerequisites

To complete this tutorial, you need the following resources and privileges:

- An active Azure subscription.
  - If you don't have an Azure subscription, create an account.
- An Azure Active Directory tenant associated with your subscription, either synchronized with an on-premises directory or a cloud-only directory.
  - If needed, create an Azure Active Directory tenant or associate an Azure subscription with your account.
- You need *global administrator* privileges in your Azure AD tenant to enable Azure AD DS.
- You need *Contributor* privileges in your Azure subscription to create the required Azure AD DS resources.

Although not required for Azure AD DS, it's recommended to configure self-service password reset (SSPR) for the Azure AD tenant. Users can change their password without SSPR, but SSPR helps if they forget their password and need to reset it.

> **IMPORTANT**
> After you create a managed domain, you can't then move the managed domain to a different resource group, virtual network, subscription, etc. Take care to select the most appropriate subscription, resource group, region, and virtual network when you deploy the managed domain.

Sign in to the Azure portal

In this tutorial, you create and configure the managed domain using the Azure portal. To get started, first sign in to the Azure portal.

Create a managed domain and configure basic settings

To launch the **Enable Azure AD Domain Services** wizard, complete the following steps:

1. On the Azure portal menu or from the **Home** page, select **Create a resource**.
2. Enter *Domain Services* into the search bar, then choose *Azure AD Domain Services* from the search suggestions.
3. On the Azure AD Domain Services page, select **Create**. The **Enable Azure AD Domain Services** wizard is launched.
4. Select the Azure **Subscription** in which you would like to create the managed domain.
5. Select the **Resource group** to which the managed domain should belong. Choose to **Create new** or select an existing resource group.

When you create a managed domain, you specify a DNS name. There are some considerations when you choose this DNS name:

- **Built-in domain name:** By default, the built-in domain name of the directory is used (a *.onmicrosoft.com* suffix). If you wish to enable secure LDAP access to the managed domain over the internet, you can't create a digital certificate to secure the connection with this default domain. Microsoft owns the *.onmicrosoft.com* domain, so a Certificate Authority (CA) won't issue a certificate.
- **Custom domain names:** The most common approach is to specify a custom domain name, typically one that you already own and is routable. When you use a routable, custom domain, traffic can correctly flow as needed to support your applications.
- **Non-routable domain suffixes:** We generally recommend that you avoid a non-routable domain name suffix, such as *contoso.local*. The *.local* suffix isn't routable and can cause issues with DNS resolution.

> **TIP**
>
> If you create a custom domain name, take care with existing DNS namespaces. It's recommended to use a domain name separate from any existing Azure or on-premises DNS name space.
>
> For example, if you have an existing DNS name space of *contoso.com*, create a managed domain with the custom domain name of *aaddscontoso.com*. If you need to use secure LDAP, you must register and own this custom domain name to generate the required certificates.
>
> You may need to create some additional DNS records for other services in your environment, or conditional DNS forwarders between existing DNS name spaces in your environment. For example, if you run a webserver that hosts a site using the root DNS name, there can be naming conflicts that require additional DNS entries.
>
> In these tutorials and how-to articles, the custom domain of *aaddscontoso.com* is used as a short example. In all commands, specify your own domain name.

The following DNS name restrictions also apply:

- **Domain prefix restrictions:** You can't create a managed domain with a prefix longer than 15 characters. The prefix of your specified domain name (such as *aaddscontoso* in the *aaddscontoso.com* domain name) must contain 15 or fewer characters.
- **Network name conflicts:** The DNS domain name for your managed domain shouldn't already exist in the virtual network. Specifically, check for the following scenarios that would lead to a name conflict:
  - If you already have an Active Directory domain with the same DNS domain name on the Azure virtual network.
  - If the virtual network where you plan to enable the managed domain has a VPN connection with your

on-premises network. In this scenario, ensure you don't have a domain with the same DNS domain name on your on-premises network.
    - If you have an existing Azure cloud service with that name on the Azure virtual network.

Complete the fields in the *Basics* window of the Azure portal to create a managed domain:

1. Enter a **DNS domain name** for your managed domain, taking into consideration the previous points.
2. Choose the Azure **Location** in which the managed domain should be created. If you choose a region that supports Availability Zones, the Azure AD DS resources are distributed across zones for additional redundancy.

    > **TIP**
    > Availability Zones are unique physical locations within an Azure region. Each zone is made up of one or more datacenters equipped with independent power, cooling, and networking. To ensure resiliency, there's a minimum of three separate zones in all enabled regions.
    >
    > There's nothing for you to configure for Azure AD DS to be distributed across zones. The Azure platform automatically handles the zone distribution of resources. For more information and to see region availability, see What are Availability Zones in Azure?

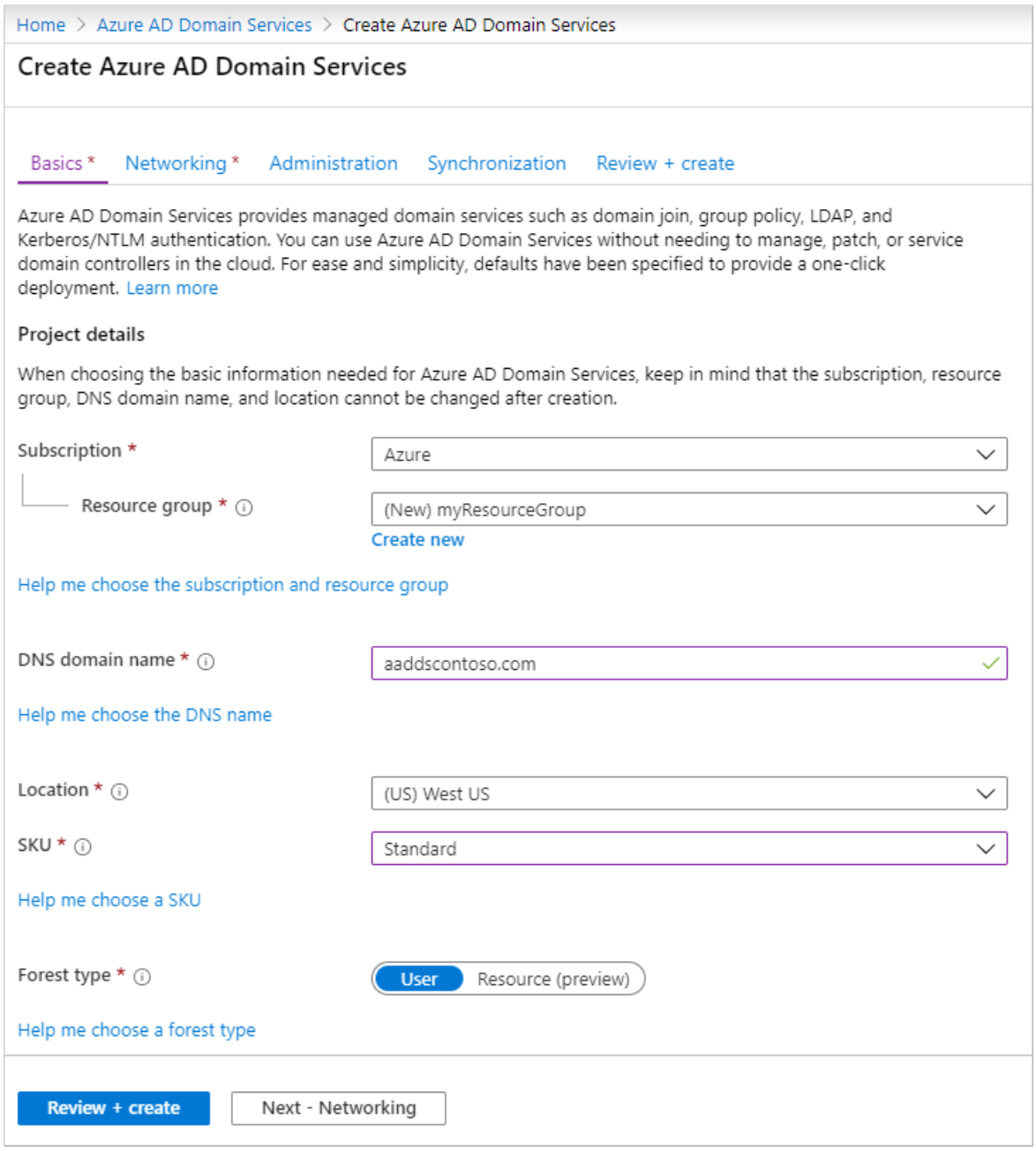
3. The **SKU** determines the performance, backup frequency, and maximum number of forest trusts you can create. You can change the SKU after the managed domain has been created if your business demands or requirements change. For more information, see Azure AD DS SKU concepts.

    For this tutorial, select the *Standard* SKU.

4. A *forest* is a logical construct used by Active Directory Domain Services to group one or more domains. By default, a managed domain is created as a *User* forest. This type of forest synchronizes all objects from Azure AD, including any user accounts created in an on-premises AD DS environment.

    A *Resource* forest only synchronizes users and groups created directly in Azure AD. Resource forests are currently in preview. For more information on *Resource* forests, including why you may use one and how to create forest trusts with on-premises AD DS domains, see Azure AD DS resource forests overview.

    For this tutorial, choose to create a *User* forest.



5. To manually configure additional options, choose **Next - Networking**. Otherwise, select **Review + create** to accept the default configuration options, then skip to the section to Deploy your managed domain. The following defaults are configured when you choose this create option:

   - Creates a virtual network named *aadds-vnet* that uses the IP address range of *10.0.1.0/24*.
   - Creates a subnet named *aadds-subnet* using the IP address range of *10.0.1.0/24*.
   - Synchronizes *All* users from Azure AD into the managed domain.

Create and configure the virtual network

To provide connectivity, an Azure virtual network and a dedicated subnet are needed. Azure AD DS is enabled in this virtual network subnet. In this tutorial, you create a virtual network, though you could instead choose to use an existing virtual network. In either approach, you must create a dedicated subnet for use by Azure AD DS.

Some considerations for this dedicated virtual network subnet include the following areas:

- The subnet must have at least 3-5 available IP addresses in its address range to support the Azure AD DS resources.
- Don't select the *Gateway* subnet for deploying Azure AD DS. It's not supported to deploy Azure AD DS into a *Gateway* subnet.

- Don't deploy any other virtual machines to the subnet. Applications and VMs often use network security groups to secure connectivity. Running these workloads in a separate subnet lets you apply those network security groups without disrupting connectivity to your managed domain.
- You can't move your managed domain to a different virtual network after you enable Azure AD DS.

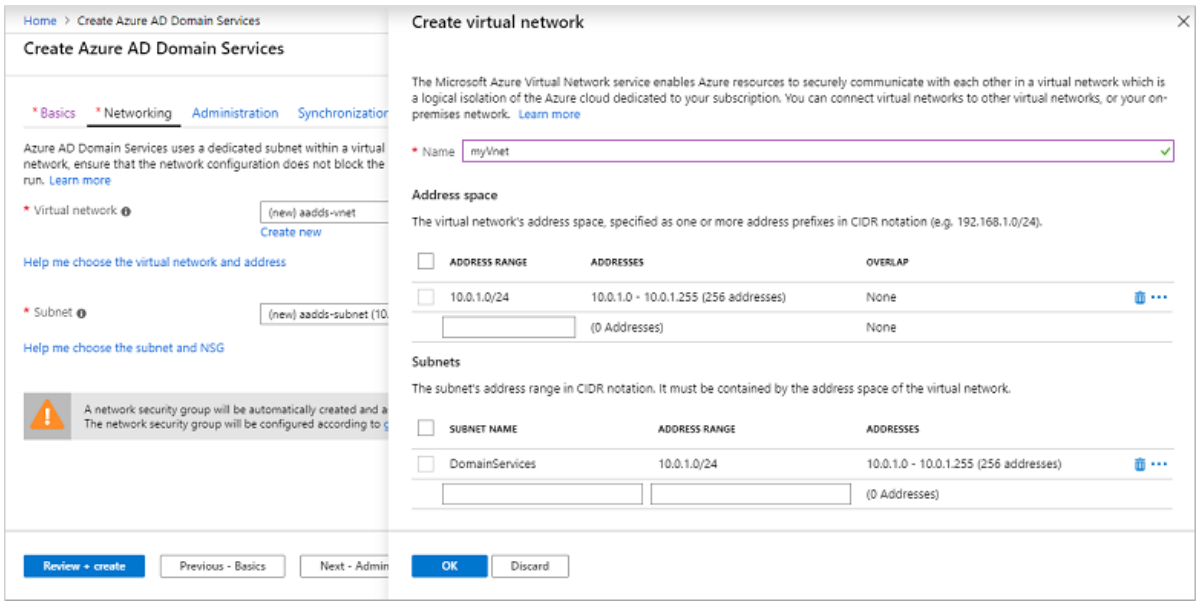
For more information on how to plan and configure the virtual network, see networking considerations for Azure Active Directory Domain Services.

Complete the fields in the *Network* window as follows:

1. On the **Network** page, choose a virtual network to deploy Azure AD DS into from the drop-down menu, or select **Create new**.

   a. If you choose to create a virtual network, enter a name for the virtual network, such as *myVnet*, then provide an address range, such as *10.0.1.0/24*.

   b. Create a dedicated subnet with a clear name, such as *DomainServices*. Provide an address range, such as *10.0.1.0/24*.



   Make sure to pick an address range that is within your private IP address range. IP address ranges you don't own that are in the public address space cause errors within Azure AD DS.

2. Select a virtual network subnet, such as *DomainServices*.

3. When ready, choose **Next - Administration**.

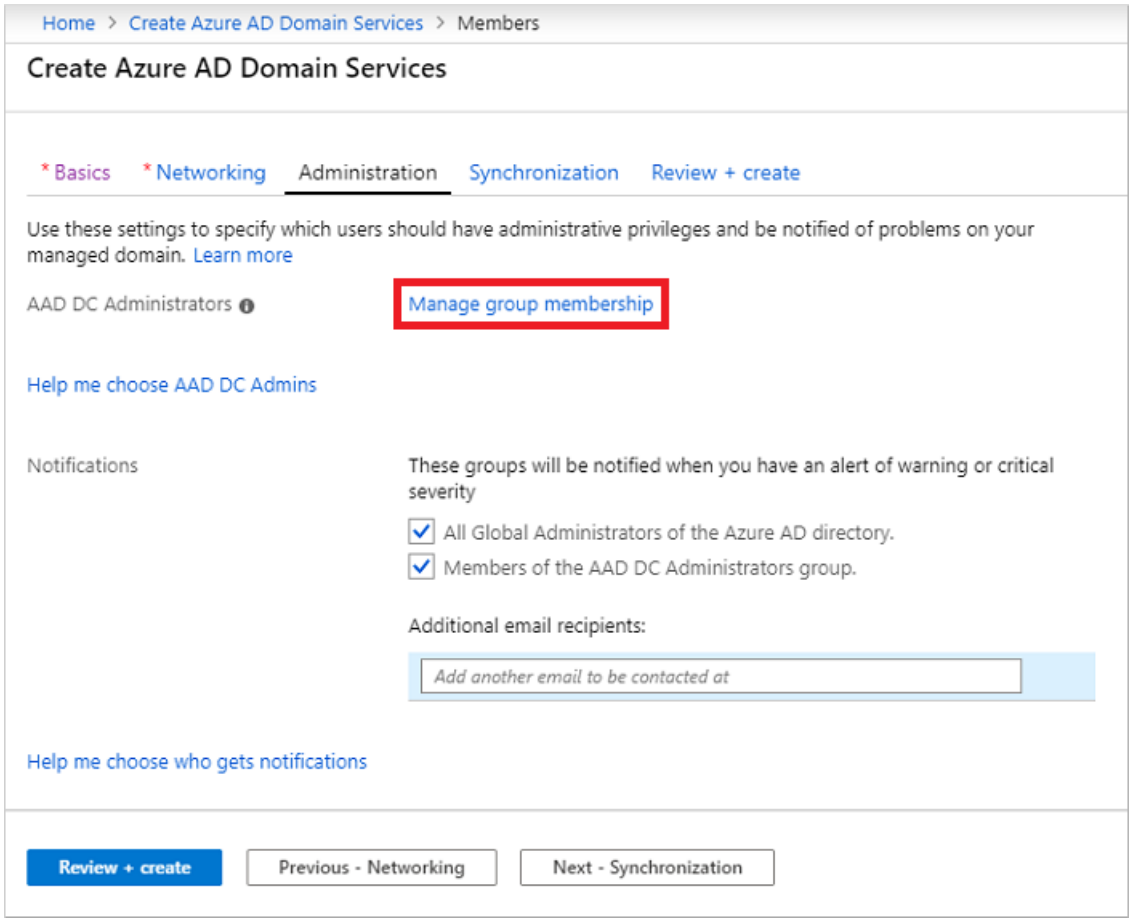
Configure an administrative group

A special administrative group named *AAD DC Administrators* is used for management of the Azure AD DS domain. Members of this group are granted administrative permissions on VMs that are domain-joined to the managed domain. On domain-joined VMs, this group is added to the local administrators group. Members of this group can also use Remote Desktop to connect remotely to domain-joined VMs.

> **IMPORTANT**
>
> You don't have *Domain Administrator* or *Enterprise Administrator* permissions on a managed domain using Azure AD DS. These permissions are reserved by the service and aren't made available to users within the tenant.
>
> Instead, the *AAD DC Administrators* group lets you perform some privileged operations. These operations include belonging to the administration group on domain-joined VMs, and configuring Group Policy.

The wizard automatically creates the *AAD DC Administrators* group in your Azure AD directory. If you have an existing group with this name in your Azure AD directory, the wizard selects this group. You can optionally choose to add additional users to this *AAD DC Administrators* group during the deployment process. These steps can be completed later.

1. To add additional users to this *AAD DC Administrators* group, select **Manage group membership**.



2. Select the **Add members** button, then search for and select users from your Azure AD directory. For example, search for your own account, and add it to the *AAD DC Administrators* group.

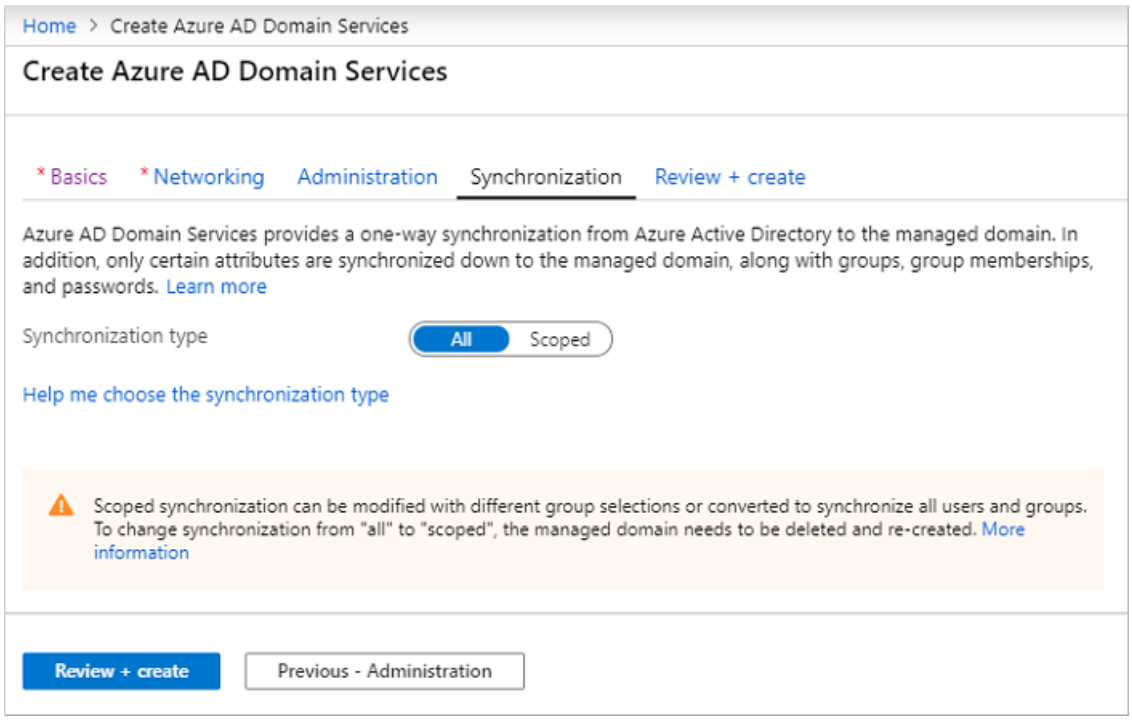
3. If desired, change or add additional recipients for notifications when there are alerts in the managed domain that require attention.

4. When ready, choose **Next - Synchronization**.

Configure synchronization

Azure AD DS lets you synchronize *all* users and groups available in Azure AD, or a *scoped* synchronization of only specific groups. If you choose to synchronize *all* users and groups, you can't later choose to only perform a scoped synchronization. For more information about scoped synchronization, see Azure AD Domain Services scoped synchronization.

1. For this tutorial, choose to synchronize **All** users and groups. This synchronization choice is the default option.



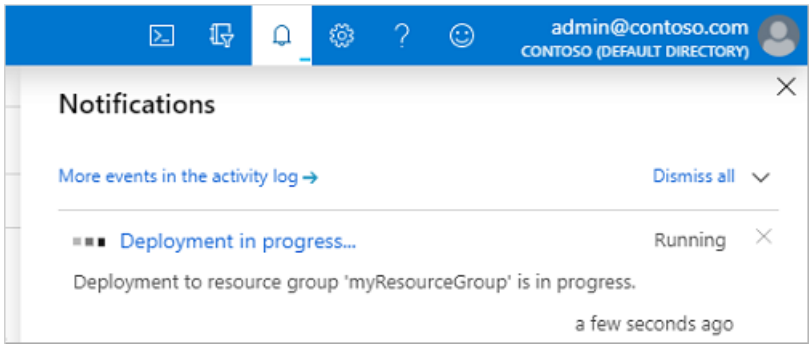
2. Select **Review + create**.

Deploy the managed domain

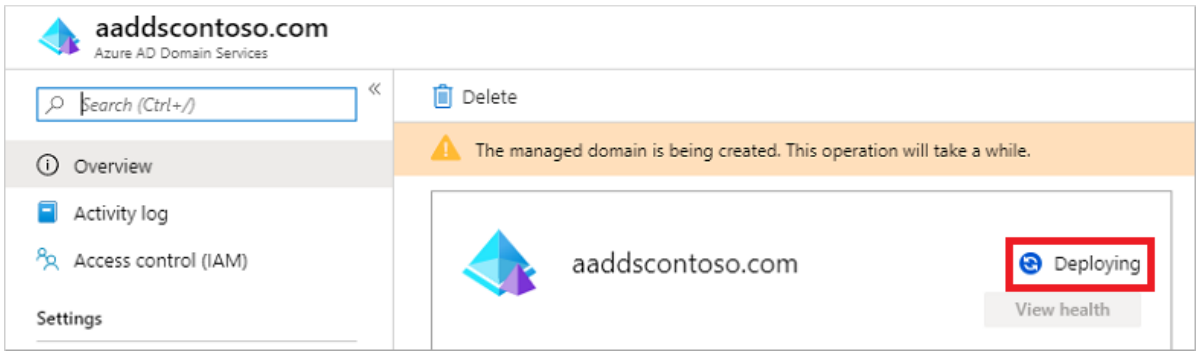
On the **Summary** page of the wizard, review the configuration settings for your managed domain. You can go back to any step of the wizard to make changes. To redeploy a managed domain to a different Azure AD tenant in a consistent way using these configuration options, you can also **Download a template for automation**.

1. To create the managed domain, select **Create**. A note is displayed that certain configuration options like DNS name or virtual network can't be changed once the Azure AD DS managed has been created. To continue, select **OK**.

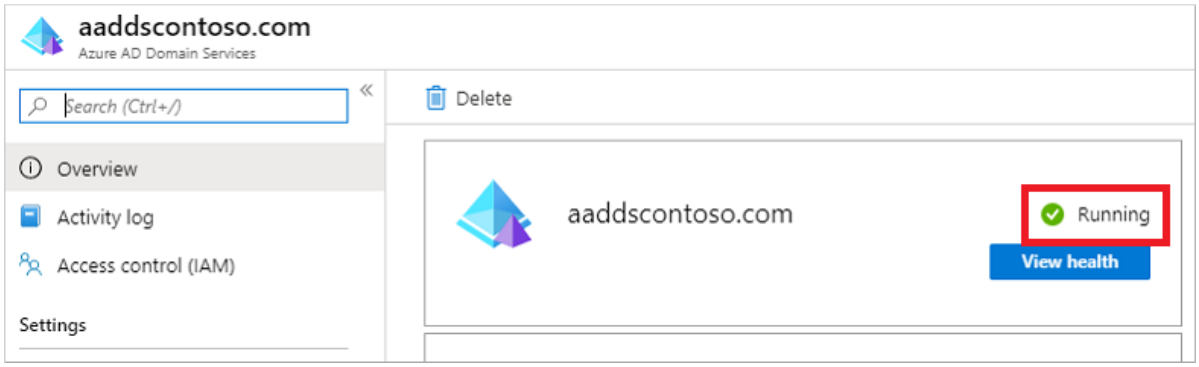
2. The process of provisioning your managed domain can take up to an hour. A notification is displayed in the portal that shows the progress of your Azure AD DS deployment. Select the notification to see detailed progress for the deployment.



3. Select your resource group, such as *myResourceGroup*, then choose your managed domain from the list of Azure resources, such as *aaddscontoso.com*. The **Overview** tab shows that the managed domain is currently *Deploying*. You can't configure the managed domain until it's fully provisioned.



4. When the managed domain is fully provisioned, the **Overview** tab shows the domain status as *Running*.



IMPORTANT

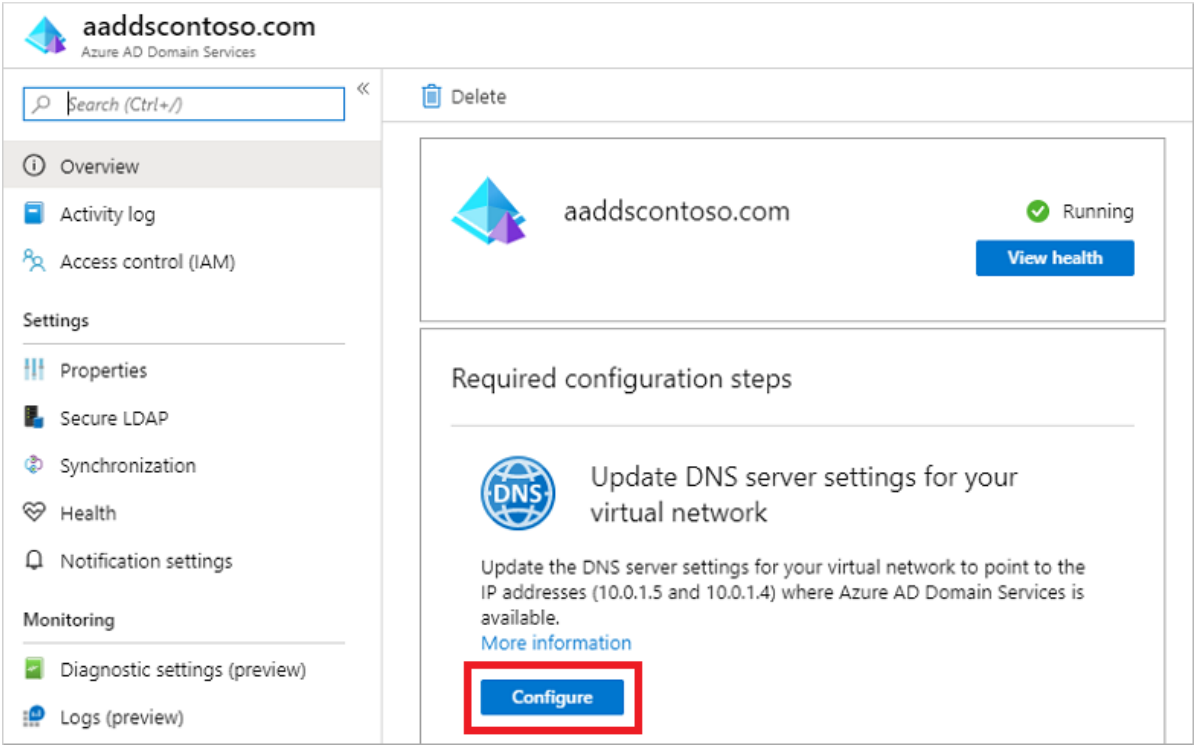
The managed domain is associated with your Azure AD tenant. During the provisioning process, Azure AD DS creates two Enterprise Applications named *Domain Controller Services* and *AzureActiveDirectoryDomainControllerServices* in the Azure AD tenant. These Enterprise Applications are needed to service your managed domain. Don't delete these applications.

Update DNS settings for the Azure virtual network

With Azure AD DS successfully deployed, now configure the virtual network to allow other connected VMs and applications to use the managed domain. To provide this connectivity, update the DNS server settings for your virtual network to point to the two IP addresses where the managed domain is deployed.

1. The **Overview** tab for your managed domain shows some **Required configuration steps**. The first configuration step is to update DNS server settings for your virtual network. Once the DNS settings are correctly configured, this step is no longer shown.

   The addresses listed are the domain controllers for use in the virtual network. In this example, those addresses are *10.0.1.4* and *10.0.1.5*. You can later find these IP addresses on the **Properties** tab.



2. To update the DNS server settings for the virtual network, select the **Configure** button. The DNS settings are automatically configured for your virtual network.

> **TIP**
> If you selected an existing virtual network in the previous steps, any VMs connected to the network only get the new DNS settings after a restart. You can restart VMs using the Azure portal, Azure PowerShell, or the Azure CLI.

Enable user accounts for Azure AD DS

To authenticate users on the managed domain, Azure AD DS needs password hashes in a format that's suitable for NT LAN Manager (NTLM) and Kerberos authentication. Azure AD doesn't generate or store password hashes in the format that's required for NTLM or Kerberos authentication until you enable Azure AD DS for your tenant. For security reasons, Azure AD also doesn't store any password credentials in clear-text form. Therefore, Azure AD can't automatically generate these NTLM or Kerberos password hashes based on users' existing credentials.

> **NOTE**
> Once appropriately configured, the usable password hashes are stored in the managed domain. If you delete the managed domain, any password hashes stored at that point are also deleted.
>
> Synchronized credential information in Azure AD can't be re-used if you later create a managed domain - you must reconfigure the password hash synchronization to store the password hashes again. Previously domain-joined VMs or users won't be able to immediately authenticate - Azure AD needs to generate and store the password hashes in the new managed domain.
>
> For more information, see Password hash sync process for Azure AD DS and Azure AD Connect.

The steps to generate and store these password hashes are different for cloud-only user accounts created in Azure AD versus user accounts that are synchronized from your on-premises directory using Azure AD Connect.

A cloud-only user account is an account that was created in your Azure AD directory using either the Azure portal or Azure AD PowerShell cmdlets. These user accounts aren't synchronized from an on-premises directory.

In this tutorial, let's work with a basic cloud-only user account. For more information on the additional steps

required to use Azure AD Connect, see Synchronize password hashes for user accounts synced from your on-premises AD to your managed domain.

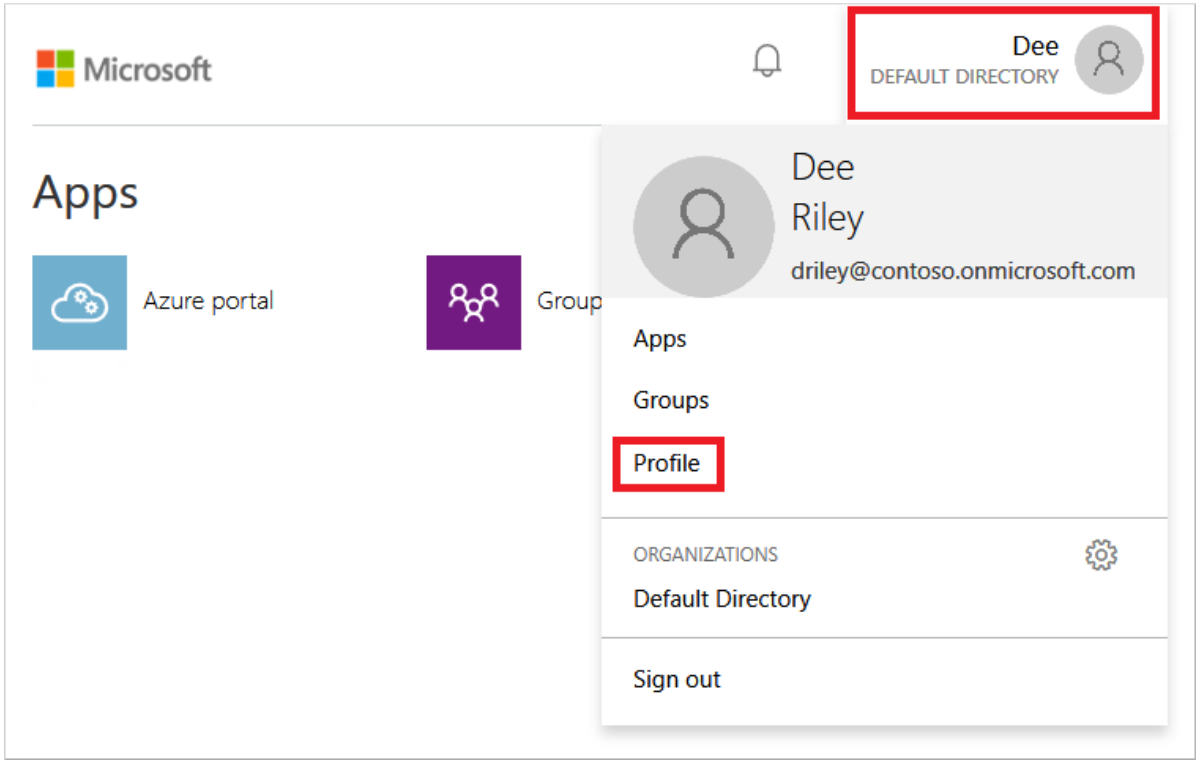
> **TIP**
> If your Azure AD tenant has a combination of cloud-only users and users from your on-premises AD, you need to complete both sets of steps.

For cloud-only user accounts, users must change their passwords before they can use Azure AD DS. This password change process causes the password hashes for Kerberos and NTLM authentication to be generated and stored in Azure AD. The account isn't synchronized from Azure AD to Azure AD DS until the password is changed. Either expire the passwords for all cloud users in the tenant who need to use Azure AD DS, which forces a password change on next sign-in, or instruct cloud users to manually change their passwords. For this tutorial, let's manually change a user password.

Before a user can reset their password, the Azure AD tenant must be configured for self-service password reset.

To change the password for a cloud-only user, the user must complete the following steps:

1. Go to the Azure AD Access Panel page at https://myapps.microsoft.com.
2. In the top-right corner, select your name, then choose **Profile** from the drop-down menu.



3. On the **Profile** page, select **Change password**.
4. On the **Change password** page, enter your existing (old) password, then enter and confirm a new password.
5. Select **Submit**.

It takes a few minutes after you've changed your password for the new password to be usable in Azure AD DS and to successfully sign in to computers joined to the managed domain.

Next steps

In this tutorial, you learned how to:

- Configure DNS and virtual network settings for a managed domain
- Create a managed domain
- Add administrative users to domain management
- Enable user accounts for Azure AD DS and generate password hashes

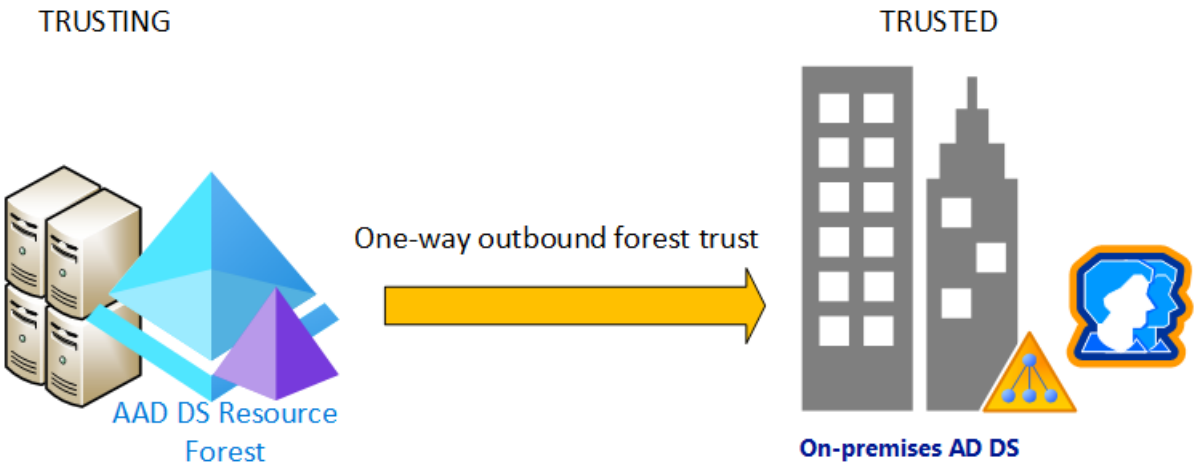
To see this managed domain in action, create and join a virtual machine to the domain.

Join a Windows Server virtual machine to your managed domain

Tutorial: Create an outbound forest trust to an on-premises domain in Azure Active Directory Domain Services (preview)

7/20/2020 • 10 minutes to read • Edit Online

In environments where you can't synchronize password hashes, or you have users that exclusively sign in using smart cards so they don't know their password, you can use a resource forest in Azure Active Directory Domain Services (Azure AD DS). A resource forest uses a one-way outbound trust from Azure AD DS to one or more on-premises AD DS environments. This trust relationship lets users, applications, and computers authenticate against an on-premises domain from the Azure AD DS managed domain. Azure AD DS resource forests are currently in preview.



In this tutorial, you learn how to:

- Configure DNS in an on-premises AD DS environment to support Azure AD DS connectivity
- Create a one-way inbound forest trust in an on-premises AD DS environment
- Create a one-way outbound forest trust in Azure AD DS
- Test and validate the trust relationship for authentication and resource access

If you don't have an Azure subscription, create an account before you begin.

Prerequisites

To complete this tutorial, you need the following resources and privileges:

- An active Azure subscription.
  - If you don't have an Azure subscription, create an account.
- An Azure Active Directory tenant associated with your subscription, either synchronized with an on-premises directory or a cloud-only directory.
  - If needed, create an Azure Active Directory tenant or associate an Azure subscription with your account.
- An Azure Active Directory Domain Services managed domain created using a resource forest and configured in your Azure AD tenant.
  - If needed, create and configure an Azure Active Directory Domain Services managed domain.

> **IMPORTANT**
>
> Make sure that you create a managed domain using a *resource* forest. The default option creates a *user* forest. Only resource forests can create trusts to on-prem AD DS environments.
>
> You also need to use a minimum of *Enterprise* SKU for your managed domain. If needed, change the SKU for a managed domain.

Sign in to the Azure portal

In this tutorial, you create and configure the outbound forest trust from Azure AD DS using the Azure portal. To get started, first sign in to the Azure portal.

Networking considerations

The virtual network that hosts the Azure AD DS resource forest needs network connectivity to your on-premises Active Directory. Applications and services also need network connectivity to the virtual network hosting the Azure AD DS resource forest. Network connectivity to the Azure AD DS resource forest must be always on and stable otherwise users may fail to authenticate or access resources.

Before you configure a forest trust in Azure AD DS, make sure your networking between Azure and on-premises environment meets the following requirements:

- Use private IP addresses. Don't rely on DHCP with dynamic IP address assignment.
- Avoid overlapping IP address spaces to allow virtual network peering and routing to successfully communicate between Azure and on-premises.
- An Azure virtual network needs a gateway subnet to configure an Azure site-to-site (S2S) VPN or ExpressRoute connection
- Create subnets with enough IP addresses to support your scenario.
- Make sure Azure AD DS has its own subnet, don't share this virtual network subnet with application VMs and services.
- Peered virtual networks are NOT transitive.
  - Azure virtual network peerings must be created between all virtual networks you want to use the Azure AD DS resource forest trust to the on-premises AD DS environment.
- Provide continuous network connectivity to your on-premises Active Directory forest. Don't use on-demand connections.
- Make sure there's continuous name resolution (DNS) between your Azure AD DS resource forest name and your on-premises Active Directory forest name.

Configure DNS in the on-premises domain

To correctly resolve the managed domain from the on-premises environment, you may need to add forwarders to the existing DNS servers. If you haven't configured the on-premises environment to communicate with the managed domain, complete the following steps from a management workstation for the on-premises AD DS domain:

1. Select **Start | Administrative Tools | DNS**
2. Right-select DNS server, such as *myAD01*, then select **Properties**
3. Choose **Forwarders**, then **Edit** to add additional forwarders.
4. Add the IP addresses of the managed domain, such as *10.0.2.4* and *10.0.2.5*.

Create inbound forest trust in the on-premises domain

The on-premises AD DS domain needs an incoming forest trust for the managed domain. This trust must be manually created in the on-premises AD DS domain, it can't be created from the Azure portal.

To configure inbound trust on the on-premises AD DS domain, complete the following steps from a management workstation for the on-premises AD DS domain:

1. Select **Start | Administrative Tools | Active Directory Domains and Trusts**
2. Right-select domain, such as *onprem.contoso.com*, then select **Properties**
3. Choose **Trusts** tab, then **New Trust**
4. Enter the name for Azure AD DS domain name, such as *aaddscontoso.com*, then select **Next**
5. Select the option to create a **Forest trust**, then to create a **One way: incoming** trust.
6. Choose to create the trust for **This domain only**. In the next step, you create the trust in the Azure portal for the managed domain.
7. Choose to use **Forest-wide authentication**, then enter and confirm a trust password. This same password is also entered in the Azure portal in the next section.
8. Step through the next few windows with default options, then choose the option for **No, do not confirm the outgoing trust**.
9. Select **Finish**

Create outbound forest trust in Azure AD DS

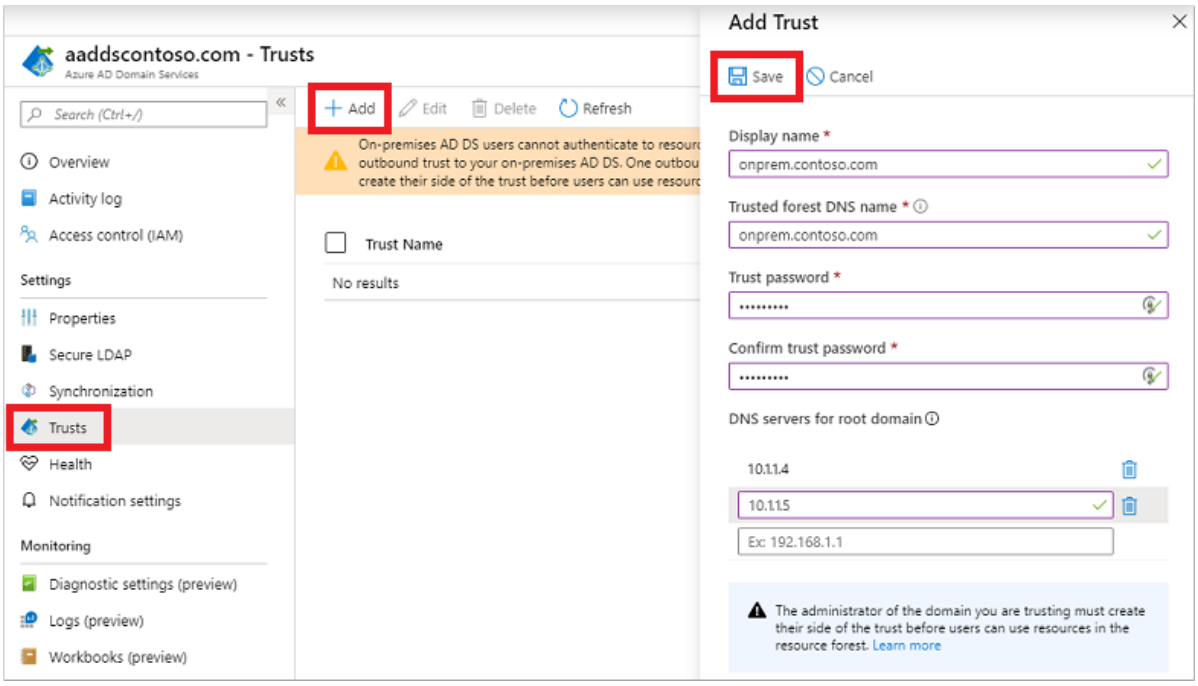
With the on-premises AD DS domain configured to resolve the managed domain and an inbound forest trust created, now create the outbound forest trust. This outbound forest trust completes the trust relationship between the on-premises AD DS domain and the managed domain.

To create the outbound trust for the managed domain in the Azure portal, complete the following steps:

1. In the Azure portal, search for and select **Azure AD Domain Services**, then select your managed domain, such as *aaddscontoso.com*
2. From the menu on the left-hand side of the managed domain, select **Trusts**, then choose to **+ Add** a trust.

   > **NOTE**
   > If you don't see the **Trusts** menu option, check under **Properties** for the *Forest type*. Only *resource* forests can create trusts. If the forest type is *User*, you can't create trusts. There's currently no way to change the forest type of a managed domain. You need to delete and recreate the managed domain as a resource forest.

3. Enter a display name that identifies your trust, then the on-premises trusted forest DNS name, such as *onprem.contoso.com*
4. Provide the same trust password that was used when configuring the inbound forest trust for the on-premises AD DS domain in the previous section.
5. Provide at least two DNS servers for the on-premises AD DS domain, such as *10.1.1.4* and *10.1.1.5*
6. When ready, **Save** the outbound forest trust



Validate resource authentication

The following common scenarios let you validate that forest trust correctly authenticates users and access to resources:

- On-premises user authentication from the Azure AD DS resource forest
- Access resources in the Azure AD DS resource forest using on-premises user
  - Enable file and printer sharing
  - Create a security group and add members
  - Create a file share for cross-forest access
  - Validate cross-forest authentication to a resource

On-premises user authentication from the Azure AD DS resource forest

You should have Windows Server virtual machine joined to the managed domain. Use this virtual machine to test your on-premises user can authenticate on a virtual machine. If needed, create a Windows VM and join it to the managed domain.

1. Connect to the Windows Server VM joined to the Azure AD DS resource forest using Azure Bastion and your Azure AD DS administrator credentials.

2. Open a command prompt and use the `whoami` command to show the distinguished name of the currently authenticated user:

   ```
   whoami /fqdn
   ```

3. Use the `runas` command to authenticate as a user from the on-premises domain. In the following command, replace `userUpn@trusteddomain.com` with the UPN of a user from the trusted on-premises domain. The command prompts you for the user's password:

   ```
   Runas /u:userUpn@trusteddomain.com cmd.exe
   ```

4. If the authentication is a successful, a new command prompt opens. The title of the new command prompt includes `running as userUpn@trusteddomain.com`.

5. Use `whoami /fqdn` in the new command prompt to view the distinguished name of the authenticated user from the on-premises Active Directory.

Access resources in the Azure AD DS resource forest using on-premises user

Using the Windows Server VM joined to the Azure AD DS resource forest, you can test the scenario where users can access resources hosted in the resource forest when they authenticate from computers in the on-premises domain with users from the on-premises domain. The following examples show you how to create and test various common scenarios.

Enable file and printer sharing

1. Connect to the Windows Server VM joined to the Azure AD DS resource forest using Azure Bastion and your Azure AD DS administrator credentials.
2. Open **Windows Settings**, then search for and select **Network and Sharing Center**.
3. Choose the option for **Change advanced sharing** settings.
4. Under the **Domain Profile**, select **Turn on file and printer sharing** and then **Save changes**.
5. Close **Network and Sharing Center**.

Create a security group and add members

1. Open **Active Directory Users and Computers**.
2. Right-select the domain name, choose **New**, and then select **Organizational Unit**.
3. In the name box, type *LocalObjects*, then select **OK**.
4. Select and right-click **LocalObjects** in the navigation pane. Select **New** and then **Group**.
5. Type *FileServerAccess* in the **Group name** box. For the **Group Scope**, select **Domain local**, then choose **OK**.
6. In the content pane, double-click **FileServerAccess**. Select **Members**, choose to **Add**, then select **Locations**.
7. Select your on-premises Active Directory from the **Location** view, then choose **OK**.
8. Type *Domain Users* in the **Enter the object names to select** box. Select **Check Names**, provide credentials for the on-premises Active Directory, then select **OK**.

   > **NOTE**
   > You must provide credentials because the trust relationship is only one way. This means users from the Azure AD DS managed domain can't access resources or search for users or groups in the trusted (on-premises) domain.

9. The **Domain Users** group from your on-premises Active Directory should be a member of the **FileServerAccess** group. Select **OK** to save the group and close the window.

Create a file share for cross-forest access

1. On the Windows Server VM joined to the Azure AD DS resource forest, create a folder and provide name such as *CrossForestShare*.
2. Right-select the folder and choose **Properties**.
3. Select the **Security** tab, then choose **Edit**.
4. In the *Permissions for CrossForestShare* dialog box, select **Add**.
5. Type *FileServerAccess* in **Enter the object names to select**, then select **OK**.
6. Select *FileServerAccess* from the **Groups or user names** list. In the **Permissions for FileServerAccess** list, choose *Allow* for the **Modify** and **Write** permissions, then select **OK**.

7. Select the **Sharing** tab, then choose **Advanced Sharing...**
8. Choose **Share this folder**, then enter a memorable name for the file share in **Share name** such as *CrossForestShare*.
9. Select **Permissions**. In the **Permissions for Everyone** list, choose **Allow** for the **Change** permission.
10. Select **OK** two times and then **Close**.

Validate cross-forest authentication to a resource

1. Sign in a Windows computer joined to your on-premises Active Directory using a user account from your on-premises Active Directory.

2. Using **Windows Explorer**, connect to the share you created using the fully qualified host name and the share such as `\\fs1.aaddscontoso.com\CrossforestShare`.

3. To validate the write permission, right-select in the folder, choose **New**, then select **Text Document**. Use the default name **New Text Document**.

   If the write permissions are set correctly, a new text document is created. The following steps will then open, edit, and delete the file as appropriate.

4. To validate the read permission, open **New Text Document**.

5. To validate the modify permission, add text to the file and close **Notepad**. When prompted to save changes, choose **Save**.

6. To validate the delete permission, right-select **New Text Document** and choose **Delete**. Choose **Yes** to confirm file deletion.

Next steps

In this tutorial, you learned how to:

- Configure DNS in an on-premises AD DS environment to support Azure AD DS connectivity
- Create a one-way inbound forest trust in an on-premises AD DS environment
- Create a one-way outbound forest trust in Azure AD DS
- Test and validate the trust relationship for authentication and resource access

For more conceptual information about forest types in Azure AD DS, see What are resource forests? and How do forest trusts work in Azure AD DS?

Enable Azure Active Directory Domain Services using PowerShell

7/20/2020 • 8 minutes to read • Edit Online

Azure Active Directory Domain Services (Azure AD DS) provides managed domain services such as domain join, group policy, LDAP, Kerberos/NTLM authentication that is fully compatible with Windows Server Active Directory. You consume these domain services without deploying, managing, and patching domain controllers yourself. Azure AD DS integrates with your existing Azure AD tenant. This integration lets users sign in using their corporate credentials, and you can use existing groups and user accounts to secure access to resources.

This article shows you how to enable Azure AD DS using PowerShell.

> **NOTE**
> This article has been updated to use the new Azure PowerShell Az module. You can still use the AzureRM module, which will continue to receive bug fixes until at least December 2020. To learn more about the new Az module and AzureRM compatibility, see Introducing the new Azure PowerShell Az module. For Az module installation instructions, see Install Azure PowerShell.

Prerequisites

To complete this article, you need the following resources:

- Install and configure Azure PowerShell.
  - If needed, follow the instructions to install the Azure PowerShell module and connect to your Azure subscription.
  - Make sure that you sign in to your Azure subscription using the Connect-AzAccount cmdlet.
- Install and configure Azure AD PowerShell.
  - If needed, follow the instructions to install the Azure AD PowerShell module and connect to Azure AD.
  - Make sure that you sign in to your Azure AD tenant using the Connect-AzureAD cmdlet.
- You need *global administrator* privileges in your Azure AD tenant to enable Azure AD DS.
- You need *Contributor* privileges in your Azure subscription to create the required Azure AD DS resources.

Create required Azure AD resources

Azure AD DS requires a service principal and an Azure AD group. These resources let the Azure AD DS managed domain synchronize data, and define which users have administrative permissions in the managed domain.

First, create an Azure AD service principal for Azure AD DS to communicate and authenticate itself. A specific application ID is used named *Domain Controller Services* with an ID of *2565bd9d-da50-47d4-8b85-4c97f669dc36*. Don't change this application ID.

Create an Azure AD service principal using the New-AzureADServicePrincipal cmdlet:

```
New-AzureADServicePrincipal -AppId "2565bd9d-da50-47d4-8b85-4c97f669dc36"
```

Now create an Azure AD group named *AAD DC Administrators*. Users added to this group are then granted permissions to perform administration tasks on the managed domain.

Create the *AAD DC Administrators* group using the New-AzureADGroup cmdlet:

```
New-AzureADGroup -DisplayName "AAD DC Administrators" `
  -Description "Delegated group to administer Azure AD Domain Services" `
  -SecurityEnabled $true -MailEnabled $false `
  -MailNickName "AADDCAdministrators"
```

With the *AAD DC Administrators* group created, add a user to the group using the Add-AzureADGroupMember cmdlet. You first get the *AAD DC Administrators* group object ID using the Get-AzureADGroup cmdlet, then the desired user's object ID using the Get-AzureADUser cmdlet.

In the following example, the user object ID for the account with a UPN of `admin@contoso.onmicrosoft.com`. Replace this user account with the UPN of the user you wish to add to the *AAD DC Administrators* group:

```
# First, retrieve the object ID of the newly created 'AAD DC Administrators' group.
$GroupObjectId = Get-AzureADGroup `
  -Filter "DisplayName eq 'AAD DC Administrators'" | `
  Select-Object ObjectId

# Now, retrieve the object ID of the user you'd like to add to the group.
$UserObjectId = Get-AzureADUser `
  -Filter "UserPrincipalName eq 'admin@contoso.onmicrosoft.com'" | `
  Select-Object ObjectId

# Add the user to the 'AAD DC Administrators' group.
Add-AzureADGroupMember -ObjectId $GroupObjectId.ObjectId -RefObjectId $UserObjectId.ObjectId
```

Create supporting Azure resources

First, register the Azure AD Domain Services resource provider using the Register-AzResourceProvider cmdlet:

```
Register-AzResourceProvider -ProviderNamespace Microsoft.AAD
```

Next, create a resource group using the New-AzResourceGroup cmdlet. In the following example, the resource group is named *myResourceGroup* and is created in the *westus* region. Use your own name and desired region:

```
$ResourceGroupName = "myResourceGroup"
$AzureLocation = "westus"

# Create the resource group.
New-AzResourceGroup `
  -Name $ResourceGroupName `
  -Location $AzureLocation
```

Create the virtual network and subnets for Azure AD Domain Services. Two subnets are created - one for *DomainServices*, and one for *Workloads*. Azure AD DS is deployed into the dedicated *DomainServices* subnet. Don't deploy other applications or workloads into this subnet. Use the separate *Workloads* or other subnets for the rest of your VMs.

Create the subnets using the New-AzVirtualNetworkSubnetConfig cmdlet, then create the virtual network using the New-AzVirtualNetwork cmdlet.

```
$VnetName = "myVnet"

# Create the dedicated subnet for AAD Domain Services.
$AaddsSubnet = New-AzVirtualNetworkSubnetConfig `
  -Name DomainServices `
  -AddressPrefix 10.0.0.0/24

$WorkloadSubnet = New-AzVirtualNetworkSubnetConfig `
  -Name Workloads `
  -AddressPrefix 10.0.1.0/24

# Create the virtual network in which you will enable Azure AD Domain Services.
$Vnet= New-AzVirtualNetwork `
  -ResourceGroupName $ResourceGroupName `
  -Location westus `
  -Name $VnetName `
  -AddressPrefix 10.0.0.0/16 `
  -Subnet $AaddsSubnet,$WorkloadSubnet
```

Create a managed domain

Now let's create a managed domain. Set your Azure subscription ID, and then provide a name for the managed domain, such as *aaddscontoso.com*. You can get your subscription ID using the Get-AzSubscription cmdlet.

If you choose a region that supports Availability Zones, the Azure AD DS resources are distributed across zones for additional redundancy.

Availability Zones are unique physical locations within an Azure region. Each zone is made up of one or more datacenters equipped with independent power, cooling, and networking. To ensure resiliency, there's a minimum of three separate zones in all enabled regions.

There's nothing for you to configure for Azure AD DS to be distributed across zones. The Azure platform automatically handles the zone distribution of resources. For more information and to see region availability, see What are Availability Zones in Azure?.

```
$AzureSubscriptionId = "YOUR_AZURE_SUBSCRIPTION_ID"
$ManagedDomainName = "aaddscontoso.com"

# Enable Azure AD Domain Services for the directory.
New-AzResource -ResourceId
"/subscriptions/$AzureSubscriptionId/resourceGroups/$ResourceGroupName/providers/Microsoft.AAD/DomainServices/
$ManagedDomainName" `
  -Location $AzureLocation `
  -Properties @{"DomainName"=$ManagedDomainName; `

"SubnetId"="/subscriptions/$AzureSubscriptionId/resourceGroups/$ResourceGroupName/providers/Microsoft.Network/
virtualNetworks/$VnetName/subnets/DomainServices"} `
  -Force -Verbose
```

It takes a few minutes to create the resource and return control to the PowerShell prompt. The managed domain continues to be provisioned in the background, and can take up to an hour to complete the deployment. In the Azure portal, the **Overview** page for your managed domain shows the current status throughout this deployment stage.

When the Azure portal shows that the managed domain has finished provisioning, the following tasks need to be completed:

- Update DNS settings for the virtual network so virtual machines can find the managed domain for domain join or authentication.
  - To configure DNS, select your managed domain in the portal. On the **Overview** window, you are

prompted to automatically configure these DNS settings.

- If you created a managed domain in a region that supports Availability Zones, create a network security group to restrict traffic in the virtual network for the managed domain. An Azure standard load balancer is created that requires these rules to be place. This network security group secures Azure AD DS and is required for the managed domain to work correctly.
  - To create the network security group and required rules, select your managed domain in the portal. On the **Overview** window, you are prompted to automatically create and configure the network security group.
- Enable password synchronization to Azure AD Domain Services so end users can sign in to the managed domain using their corporate credentials.

Complete PowerShell script

The following complete PowerShell script combines all of the tasks shown in this article. Copy the script and save it to a file with a `.ps1` extension. Run the script in a local PowerShell console or the Azure Cloud Shell.

> **NOTE**
> To enable Azure AD DS, you must be a global administrator for the Azure AD tenant. You also need at least *Contributor* privileges in the Azure subscription.

```
# Change the following values to match your deployment.
$AaddsAdminUserUpn = "admin@contoso.onmicrosoft.com"
$ResourceGroupName = "myResourceGroup"
$VnetName = "myVnet"
$AzureLocation = "westus"
$AzureSubscriptionId = "YOUR_AZURE_SUBSCRIPTION_ID"
$ManagedDomainName = "aaddscontoso.com"

# Connect to your Azure AD directory.
Connect-AzureAD

# Login to your Azure subscription.
Connect-AzAccount

# Create the service principal for Azure AD Domain Services.
New-AzureADServicePrincipal -AppId "2565bd9d-da50-47d4-8b85-4c97f669dc36"

# Create the delegated administration group for AAD Domain Services.
New-AzureADGroup -DisplayName "AAD DC Administrators" `
  -Description "Delegated group to administer Azure AD Domain Services" `
  -SecurityEnabled $true -MailEnabled $false `
  -MailNickName "AADDCAdministrators"

# First, retrieve the object ID of the newly created 'AAD DC Administrators' group.
$GroupObjectId = Get-AzureADGroup `
  -Filter "DisplayName eq 'AAD DC Administrators'" | `
  Select-Object ObjectId

# Now, retrieve the object ID of the user you'd like to add to the group.
$UserObjectId = Get-AzureADUser `
  -Filter "UserPrincipalName eq '$AaddsAdminUserUpn'" | `
  Select-Object ObjectId

# Add the user to the 'AAD DC Administrators' group.
Add-AzureADGroupMember -ObjectId $GroupObjectId.ObjectId -RefObjectId $UserObjectId.ObjectId

# Register the resource provider for Azure AD Domain Services with Resource Manager.
Register-AzResourceProvider -ProviderNamespace Microsoft.AAD

# Create the resource group
```

```
# Create the resource group.
New-AzResourceGroup `
  -Name $ResourceGroupName `
  -Location $AzureLocation

# Create the dedicated subnet for AAD Domain Services.
$AaddsSubnet = New-AzVirtualNetworkSubnetConfig `
  -Name DomainServices `
  -AddressPrefix 10.0.0.0/24

$WorkloadSubnet = New-AzVirtualNetworkSubnetConfig `
  -Name Workloads `
  -AddressPrefix 10.0.1.0/24

# Create the virtual network in which you will enable Azure AD Domain Services.
$Vnet=New-AzVirtualNetwork `
  -ResourceGroupName $ResourceGroupName `
  -Location $AzureLocation `
  -Name $VnetName `
  -AddressPrefix 10.0.0.0/16 `
  -Subnet $AaddsSubnet,$WorkloadSubnet

# Enable Azure AD Domain Services for the directory.
New-AzResource -ResourceId
"/subscriptions/$AzureSubscriptionId/resourceGroups/$ResourceGroupName/providers/Microsoft.AAD/DomainServices/
$ManagedDomainName" `
  -Location $AzureLocation `
  -Properties @{"DomainName"=$ManagedDomainName; `

"SubnetId"="/subscriptions/$AzureSubscriptionId/resourceGroups/$ResourceGroupName/providers/Microsoft.Network/
virtualNetworks/$VnetName/subnets/DomainServices"} `
  -Force -Verbose
```

It takes a few minutes to create the resource and return control to the PowerShell prompt. The managed domain continues to be provisioned in the background, and can take up to an hour to complete the deployment. In the Azure portal, the **Overview** page for your managed domain shows the current status throughout this deployment stage.

When the Azure portal shows that the managed domain has finished provisioning, the following tasks need to be completed:

- Update DNS settings for the virtual network so virtual machines can find the managed domain for domain join or authentication.
  - To configure DNS, select your managed domain in the portal. On the **Overview** window, you are prompted to automatically configure these DNS settings.
- If you created a managed domain in a region that supports Availability Zones, create a network security group to restrict traffic in the virtual network for the managed domain. An Azure standard load balancer is created that requires these rules to be place. This network security group secures Azure AD DS and is required for the managed domain to work correctly.
  - To create the network security group and required rules, select your managed domain in the portal. On the **Overview** window, you are prompted to automatically create and configure the network security group.
- Enable password synchronization to Azure AD Domain Services so end users can sign in to the managed domain using their corporate credentials.

Next steps

To see the managed domain in action, you can domain-join a Windows VM, configure secure LDAP, and configure password hash sync.

Create an Azure Active Directory Domain Services managed domain using an Azure Resource Manager template

7/20/2020 • 10 minutes to read • Edit Online

Azure Active Directory Domain Services (Azure AD DS) provides managed domain services such as domain join, group policy, LDAP, Kerberos/NTLM authentication that is fully compatible with Windows Server Active Directory. You consume these domain services without deploying, managing, and patching domain controllers yourself. Azure AD DS integrates with your existing Azure AD tenant. This integration lets users sign in using their corporate credentials, and you can use existing groups and user accounts to secure access to resources.

This article shows you how to create a managed domain using an Azure Resource Manager template. Supporting resources are created using Azure PowerShell.

Prerequisites

To complete this article, you need the following resources:

- Install and configure Azure PowerShell.
  - If needed, follow the instructions to install the Azure PowerShell module and connect to your Azure subscription.
  - Make sure that you sign in to your Azure subscription using the Connect-AzAccount cmdlet.
- Install and configure Azure AD PowerShell.
  - If needed, follow the instructions to install the Azure AD PowerShell module and connect to Azure AD.
  - Make sure that you sign in to your Azure AD tenant using the Connect-AzureAD cmdlet.
- You need *global administrator* privileges in your Azure AD tenant to enable Azure AD DS.
- You need *Contributor* privileges in your Azure subscription to create the required Azure AD DS resources.

DNS naming requirements

When you create an Azure AD DS managed domain, you specify a DNS name. There are some considerations when you choose this DNS name:

- **Built-in domain name:** By default, the built-in domain name of the directory is used (a *.onmicrosoft.com* suffix). If you wish to enable secure LDAP access to the managed domain over the internet, you can't create a digital certificate to secure the connection with this default domain. Microsoft owns the *.onmicrosoft.com* domain, so a Certificate Authority (CA) won't issue a certificate.
- **Custom domain names:** The most common approach is to specify a custom domain name, typically one that you already own and is routable. When you use a routable, custom domain, traffic can correctly flow as needed to support your applications.
- **Non-routable domain suffixes:** We generally recommend that you avoid a non-routable domain name suffix, such as *contoso.local*. The *.local* suffix isn't routable and can cause issues with DNS resolution.

TIP

If you create a custom domain name, take care with existing DNS namespaces. It's recommended to use a domain name separate from any existing Azure or on-premises DNS name space.

For example, if you have an existing DNS name space of *contoso.com*, create a managed domain with the custom domain name of *aaddscontoso.com*. If you need to use secure LDAP, you must register and own this custom domain name to generate the required certificates.

You may need to create some additional DNS records for other services in your environment, or conditional DNS forwarders between existing DNS name spaces in your environment. For example, if you run a webserver that hosts a site using the root DNS name, there can be naming conflicts that require additional DNS entries.

In this sample and how-to articles, the custom domain of *aaddscontoso.com* is used as a short example. In all commands, specify your own domain name.

The following DNS name restrictions also apply:

- **Domain prefix restrictions:** You can't create a managed domain with a prefix longer than 15 characters. The prefix of your specified domain name (such as *aaddscontoso* in the *aaddscontoso.com* domain name) must contain 15 or fewer characters.
- **Network name conflicts:** The DNS domain name for your managed domain shouldn't already exist in the virtual network. Specifically, check for the following scenarios that would lead to a name conflict:
  - If you already have an Active Directory domain with the same DNS domain name on the Azure virtual network.
  - If the virtual network where you plan to enable the managed domain has a VPN connection with your on-premises network. In this scenario, ensure you don't have a domain with the same DNS domain name on your on-premises network.
  - If you have an existing Azure cloud service with that name on the Azure virtual network.

Create required Azure AD resources

Azure AD DS requires a service principal and an Azure AD group. These resources let the managed domain synchronize data, and define which users have administrative permissions in the managed domain.

First, register the Azure AD Domain Services resource provider using the Register-AzResourceProvider cmdlet:

```
Register-AzResourceProvider -ProviderNamespace Microsoft.AAD
```

Create an Azure AD service principal using the New-AzureADServicePrincipal cmdlet for Azure AD DS to communicate and authenticate itself. A specific application ID is used named *Domain Controller Services* with an ID of *2565bd9d-da50-47d4-8b85-4c97f669dc36*. Don't change this application ID.

```
New-AzureADServicePrincipal -AppId "2565bd9d-da50-47d4-8b85-4c97f669dc36"
```

Now create an Azure AD group named *AAD DC Administrators* using the New-AzureADGroup cmdlet. Users added to this group are then granted permissions to perform administration tasks on the managed domain.

```
New-AzureADGroup -DisplayName "AAD DC Administrators" `
  -Description "Delegated group to administer Azure AD Domain Services" `
  -SecurityEnabled $true -MailEnabled $false `
  -MailNickName "AADDCAdministrators"
```

With the *AAD DC Administrators* group created, add a user to the group using the Add-AzureADGroupMember

cmdlet. You first get the *AAD DC Administrators* group object ID using the Get-AzureADGroup cmdlet, then the desired user's object ID using the Get-AzureADUser cmdlet.

In the following example, the user object ID for the account with a UPN of `admin@contoso.onmicrosoft.com`. Replace this user account with the UPN of the user you wish to add to the *AAD DC Administrators* group:

```
# First, retrieve the object ID of the newly created 'AAD DC Administrators' group.
$GroupObjectId = Get-AzureADGroup `
  -Filter "DisplayName eq 'AAD DC Administrators'" | `
  Select-Object ObjectId

# Now, retrieve the object ID of the user you'd like to add to the group.
$UserObjectId = Get-AzureADUser `
  -Filter "UserPrincipalName eq 'admin@contoso.onmicrosoft.com'" | `
  Select-Object ObjectId

# Add the user to the 'AAD DC Administrators' group.
Add-AzureADGroupMember -ObjectId $GroupObjectId.ObjectId -RefObjectId $UserObjectId.ObjectId
```

Finally, create a resource group using the New-AzResourceGroup cmdlet. In the following example, the resource group is named *myResourceGroup* and is created in the *westus* region. Use your own name and desired region:

```
New-AzResourceGroup `
  -Name "myResourceGroup" `
  -Location "WestUS"
```

If you choose a region that supports Availability Zones, the Azure AD DS resources are distributed across zones for additional redundancy. Availability Zones are unique physical locations within an Azure region. Each zone is made up of one or more datacenters equipped with independent power, cooling, and networking. To ensure resiliency, there's a minimum of three separate zones in all enabled regions.

There's nothing for you to configure for Azure AD DS to be distributed across zones. The Azure platform automatically handles the zone distribution of resources. For more information and to see region availability, see What are Availability Zones in Azure?.

Resource definition for Azure AD DS

As part of the Resource Manager resource definition, the following configuration parameters are required:

PARAMETER	VALUE
domainName	The DNS domain name for your managed domain, taking into consideration the previous points on naming prefixes and conflicts.
filteredSync	Azure AD DS lets you synchronize *all* users and groups available in Azure AD, or a *scoped* synchronization of only specific groups. For more information about scoped synchronization, see Azure AD Domain Services scoped synchronization.

PARAMETER	VALUE
notificationSettings	If there are any alerts generated in the managed domain, email notifications can be sent out. *Global administrators* of the Azure tenant and members of the *AAD DC Administrators* group can be *Enabled* for these notifications. If desired, you can add additional recipients for notifications when there are alerts that require attention.
domainConfigurationType	By default, a managed domain is created as a *User* forest. This type of forest synchronizes all objects from Azure AD, including any user accounts created in an on-premises AD DS environment. You don't need to specify a *domainConfiguration* value to create a user forest. A *Resource* forest only synchronizes users and groups created directly in Azure AD. Resource forests are currently in preview. Set the value to *ResourceTrusting* to create a resource forest. For more information on *Resource* forests, including why you may use one and how to create forest trusts with on-premises AD DS domains, see Azure AD DS resource forests overview.

The following condensed parameters definition shows how these values are declared. A user forest named *aaddscontoso.com* is created with all users from Azure AD synchronized to the managed domain:

```
"parameters": {
    "domainName": {
        "value": "aaddscontoso.com"
    },
    "filteredSync": {
        "value": "Disabled"
    },
    "notificationSettings": {
        "value": {
            "notifyGlobalAdmins": "Enabled",
            "notifyDcAdmins": "Enabled",
            "additionalRecipients": []
        }
    },
    [...]
}
```

The following condensed Resource Manager template resource type is then used to define and create the managed domain. An Azure virtual network and subnet must already exist, or be created as part of Resource Manager template. The managed domain is connected to this subnet.

```
"resources": [
    {
        "apiVersion": "2017-06-01",
        "type": "Microsoft.AAD/DomainServices",
        "name": "[parameters('domainName')]",
        "location": "[parameters('location')]",
        "dependsOn": [
            "[concat('Microsoft.Network/virtualNetworks/', parameters('vnetName'))]"
        ],
        "properties": {
            "domainName": "[parameters('domainName')]",
            "subnetId": "[concat('/subscriptions/', subscription().subscriptionId, '/resourceGroups/',
resourceGroup().name, '/providers/Microsoft.Network/virtualNetworks/', parameters('vnetName'), '/subnets/',
parameters('subnetName'))]",
            "filteredSync": "[parameters('filteredSync')]",
            "notificationSettings": "[parameters('notificationSettings')]"
        }
    },
    [...]
]
```

These parameters and resource type can be used as part of a wider Resource Manager template to deploy a managed domain, as shown in the following section.

Create a managed domain using sample template

The following complete Resource Manager sample template creates a managed domain and the supporting virtual network, subnet, and network security group rules. The network security group rules are required to secure the managed domain and make sure traffic can flow correctly. A user forest with the DNS name of *aaddscontoso.com* is created, with all users synchronized from Azure AD:

```
{
    "$schema": "http://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {
        "apiVersion": {
            "value": "2017-06-01"
        },
        "domainConfigurationType": {
            "value": "FullySynced"
        },
        "domainName": {
            "value": "aaddscontoso.com"
        },
        "filteredSync": {
            "value": "Disabled"
        },
        "location": {
            "value": "westus"
        },
        "notificationSettings": {
            "value": {
                "notifyGlobalAdmins": "Enabled",
                "notifyDcAdmins": "Enabled",
                "additionalRecipients": []
            }
        },
        "subnetName": {
            "value": "aadds-subnet"
        },
        "vnetName": {
            "value": "aadds-vnet"
        },
        "vnetAddressPrefixes": {
```

```
            "value": [
                "10.1.0.0/24"
            ]
        },
        "subnetAddressPrefix": {
            "value": "10.1.0.0/24"
        },
        "nsgName": {
            "value": "aadds-nsg"
        }
    },
    "resources": [
        {
            "apiVersion": "2017-06-01",
            "type": "Microsoft.AAD/DomainServices",
            "name": "[parameters('domainName')]",
            "location": "[parameters('location')]",
            "dependsOn": [
                "[concat('Microsoft.Network/virtualNetworks/', parameters('vnetName'))]"
            ],
            "properties": {
                "domainName": "[parameters('domainName')]",
                "subnetId": "[concat('/subscriptions/', subscription().subscriptionId, '/resourceGroups/',
resourceGroup().name, '/providers/Microsoft.Network/virtualNetworks/', parameters('vnetName'), '/subnets/',
parameters('subnetName'))]",
                "filteredSync": "[parameters('filteredSync')]",
                "domainConfigurationType": "[parameters('domainConfigurationType')]",
                "notificationSettings": "[parameters('notificationSettings')]"
            }
        },
        {
            "type": "Microsoft.Network/NetworkSecurityGroups",
            "name": "[parameters('nsgName')]",
            "location": "[parameters('location')]",
            "properties": {
                "securityRules": [
                    {
                        "name": "AllowSyncWithAzureAD",
                        "properties": {
                            "access": "Allow",
                            "priority": 101,
                            "direction": "Inbound",
                            "protocol": "Tcp",
                            "sourceAddressPrefix": "AzureActiveDirectoryDomainServices",
                            "sourcePortRange": "*",
                            "destinationAddressPrefix": "*",
                            "destinationPortRange": "443"
                        }
                    },
                    {
                        "name": "AllowPSRemoting",
                        "properties": {
                            "access": "Allow",
                            "priority": 301,
                            "direction": "Inbound",
                            "protocol": "Tcp",
                            "sourceAddressPrefix": "AzureActiveDirectoryDomainServices",
                            "sourcePortRange": "*",
                            "destinationAddressPrefix": "*",
                            "destinationPortRange": "5986"
                        }
                    },
                    {
                        "name": "AllowRD",
                        "properties": {
                            "access": "Allow",
                            "priority": 201,
                            "direction": "Inbound",
                            "protocol": "Tcp"
```

```
                            "protocol": "Tcp",
                            "sourceAddressPrefix": "CorpNetSaw",
                            "sourcePortRange": "*",
                            "destinationAddressPrefix": "*",
                            "destinationPortRange": "3389"
                        }
                    }
                ]
            },
            "apiVersion": "2018-04-01"
        },
        {
            "type": "Microsoft.Network/virtualNetworks",
            "name": "[parameters('vnetName')]",
            "location": "[parameters('location')]",
            "apiVersion": "2018-04-01",
            "dependsOn": [
                "[concat('Microsoft.Network/NetworkSecurityGroups/', parameters('nsgName'))]"
            ],
            "properties": {
                "addressSpace": {
                    "addressPrefixes": "[parameters('vnetAddressPrefixes')]"
                },
                "subnets": [
                    {
                        "name": "[parameters('subnetName')]",
                        "properties": {
                            "addressPrefix": "[parameters('subnetAddressPrefix')]",
                            "networkSecurityGroup": {
                                "id": "[concat('/subscriptions/', subscription().subscriptionId,
'/resourceGroups/', resourceGroup().name, '/providers/Microsoft.Network/NetworkSecurityGroups/',
parameters('nsgName'))]"
                            }
                        }
                    }
                ]
            }
        }
    ],
    "outputs": {}
}
```

This template can be deployed using your preferred deployment method, such as the Azure portal, Azure PowerShell, or a CI/CD pipeline. The following example uses the New-AzResourceGroupDeployment cmdlet. Specify your own resource group name and template filename:

```
New-AzResourceGroupDeployment -ResourceGroupName "myResourceGroup" -TemplateFile <path-to-template>
```

It takes a few minutes to create the resource and return control to the PowerShell prompt. The managed domain continues to be provisioned in the background, and can take up to an hour to complete the deployment. In the Azure portal, the **Overview** page for your managed domain shows the current status throughout this deployment stage.

When the Azure portal shows that the managed domain has finished provisioning, the following tasks need to be completed:

- Update DNS settings for the virtual network so virtual machines can find the managed domain for domain join or authentication.
  - To configure DNS, select your managed domain in the portal. On the **Overview** window, you are prompted to automatically configure these DNS settings.
- Enable password synchronization to Azure AD DS so end users can sign in to the managed domain using their corporate credentials.

Next steps

To see the managed domain in action, you can domain-join a Windows VM, configure secure LDAP, and configure password hash sync.

Management concepts for user accounts, passwords, and administration in Azure Active Directory Domain Services

7/20/2020 • 7 minutes to read • Edit Online

When you create and run an Azure Active Directory Domain Services (AD DS) managed domain, there are some differences in behavior compared to a traditional on-premises AD DS environment. You use the same administrative tools in Azure AD DS as a self-managed domain, but you can't directly access the domain controllers (DC). There's also some differences in behavior for password policies and password hashes depending on the source of the user account creation.

This conceptual article details how to administer a managed domain and the different behavior of user accounts depending on the way they're created.

Domain management

A managed domain is a DNS namespace and matching directory. In a managed domain, the domain controllers (DCs) that contain all the resources like users and groups, credentials, and policies are part of the managed service. For redundancy, two DCs are created as part of a managed domain. You can't sign in to these DCs to perform management tasks. Instead, you create a management VM that's joined to the managed domain, then install your regular AD DS management tools. You can use the Active Directory Administrative Center or Microsoft Management Console (MMC) snap-ins like DNS or Group Policy objects, for example.

User account creation

User accounts can be created in a managed domain in multiple ways. Most user accounts are synchronized in from Azure AD, which can also include user account synchronized from an on-premises AD DS environment. You can also manually create accounts directly in the managed domain. Some features, like initial password synchronization or password policy, behave differently depending on how and where user accounts are created.

- The user account can be synchronized in from Azure AD. This includes cloud-only user accounts created directly in Azure AD, and hybrid user accounts synchronized from an on-premises AD DS environment using Azure AD Connect.
  - The majority of user accounts in a managed domain are created through the synchronization process from Azure AD.
- The user account can be manually created in a managed domain, and doesn't exist in Azure AD.
  - If you need to create service accounts for applications that only run in the managed domain, you can manually create them in the managed domain. As synchronization is one way from Azure AD, user accounts created in the managed domain aren't synchronized back to Azure AD.

Password policy

Azure AD DS includes a default password policy that defines settings for things like account lockout, maximum password age, and password complexity. Settings like account lockout policy apply to all users in a managed domain, regardless of how the user was created as outlined in the previous section. A few settings, like minimum password length and password complexity, only apply to users created directly in a managed domain.

You can create your own custom password policies to override the default policy in a managed domain. These

custom policies can then be applied to specific groups of users as needed.

For more information on the differences in how password policies are applied depending on the source of user creation, see Password and account lockout policies on managed domains.

Password hashes

To authenticate users on the managed domain, Azure AD DS needs password hashes in a format that's suitable for NT LAN Manager (NTLM) and Kerberos authentication. Azure AD doesn't generate or store password hashes in the format that's required for NTLM or Kerberos authentication until you enable Azure AD DS for your tenant. For security reasons, Azure AD also doesn't store any password credentials in clear-text form. Therefore, Azure AD can't automatically generate these NTLM or Kerberos password hashes based on users' existing credentials.

For cloud-only user accounts, users must change their passwords before they can use the managed domain. This password change process causes the password hashes for Kerberos and NTLM authentication to be generated and stored in Azure AD. The account isn't synchronized from Azure AD to Azure AD DS until the password is changed.

For users synchronized from an on-premises AD DS environment using Azure AD Connect, enable synchronization of password hashes.

> **IMPORTANT**
> Azure AD Connect only synchronizes legacy password hashes when you enable Azure AD DS for your Azure AD tenant. Legacy password hashes aren't used if you only use Azure AD Connect to synchronize an on-premises AD DS environment with Azure AD.
>
> If your legacy applications don't use NTLM authentication or LDAP simple binds, we recommend that you disable NTLM password hash synchronization for Azure AD DS. For more information, see Disable weak cipher suites and NTLM credential hash synchronization.

Once appropriately configured, the usable password hashes are stored in the managed domain. If you delete the managed domain, any password hashes stored at that point are also deleted. Synchronized credential information in Azure AD can't be reused if you later create another managed domain - you must reconfigure the password hash synchronization to store the password hashes again. Previously domain-joined VMs or users won't be able to immediately authenticate - Azure AD needs to generate and store the password hashes in the new managed domain. For more information, see Password hash sync process for Azure AD DS and Azure AD Connect.

> **IMPORTANT**
> Azure AD Connect should only be installed and configured for synchronization with on-premises AD DS environments. It's not supported to install Azure AD Connect in a managed domain to synchronize objects back to Azure AD.

Forests and trusts

A *forest* is a logical construct used by Active Directory Domain Services (AD DS) to group one or more *domains*. The domains then store objects for user or groups, and provide authentication services.

In Azure AD DS, the forest only contains one domain. On-premises AD DS forests often contain many domains. In large organizations, especially after mergers and acquisitions, you may end up with multiple on-premises forests that each then contain multiple domains.

By default, a managed domain is created as a *user* forest. This type of forest synchronizes all objects from Azure AD, including any user accounts created in an on-premises AD DS environment. User accounts can directly authenticate against the managed domain, such as to sign in to a domain-joined VM. A user forest works when the password hashes can be synchronized and users aren't using exclusive sign-in methods like smart card

authentication.

In an Azure AD DS *resource* forest, users authenticate over a one-way forest *trust* from their on-premises AD DS. With this approach, the user objects and password hashes aren't synchronized to Azure AD DS. The user objects and credentials only exist in the on-premises AD DS. This approach lets enterprises host resources and application platforms in Azure that depend on classic authentication such LDAPS, Kerberos, or NTLM, but any authentication issues or concerns are removed. Azure AD DS resource forests are currently in preview.

For more information about forest types in Azure AD DS, see What are resource forests? and How do forest trusts work in Azure AD DS?

Azure AD DS SKUs

In Azure AD DS, the available performance and features are based on the SKU. You select a SKU when you create the managed domain, and you can switch SKUs as your business requirements change after the managed domain has been deployed. The following table outlines the available SKUs and the differences between them:

SKU NAME	MAXIMUM OBJECT COUNT	BACKUP FREQUENCY	MAXIMUM NUMBER OF OUTBOUND FOREST TRUSTS
Standard	Unlimited	Every 7 days	0
Enterprise	Unlimited	Every 3 days	5
Premium	Unlimited	Daily	10

Before these Azure AD DS SKUs, a billing model based on the number of objects (user and computer accounts) in the managed domain was used. There is no longer variable pricing based on the number of objects in the managed domain.

For more information, see the Azure AD DS pricing page.

Managed domain performance

Domain performance varies based on how authentication is implemented for an application. Additional compute resources may help improve query response time and reduce time spent in sync operations. As the SKU level increases, the compute resources available to the managed domain is increased. Monitor the performance of your applications and plan for the required resources.

If your business or application demands change and you need additional compute power for your managed domain, you can switch to a different SKU.

Backup frequency

The backup frequency determines how often a snapshot of the managed domain is taken. Backups are an automated process managed by the Azure platform. In the event of an issue with your managed domain, Azure support can assist you in restoring from backup. As synchronization only occurs one way *from* Azure AD, any issues in a managed domain won't impact Azure AD or on-premises AD DS environments and functionality.

As the SKU level increases, the frequency of those backup snapshots increases. Review your business requirements and recovery point objective (RPO) to determine the required backup frequency for your managed domain. If your business or application requirements change and you need more frequent backups, you can switch to a different SKU.

Outbound forest trusts

The previous section detailed one-way outbound forest trusts from a managed domain to an on-premises AD DS environment (currently in preview). The SKU determines the maximum number of forest trusts you can create for a managed domain. Review your business and application requirements to determine how many trusts you

actually need, and pick the appropriate Azure AD DS SKU. Again, if your business requirements change and you need to create additional forest trusts, you can switch to a different SKU.

Next steps

To get started, create an Azure AD DS managed domain.

Common use-cases and scenarios for Azure Active Directory Domain Services

7/20/2020 • 6 minutes to read • Edit Online

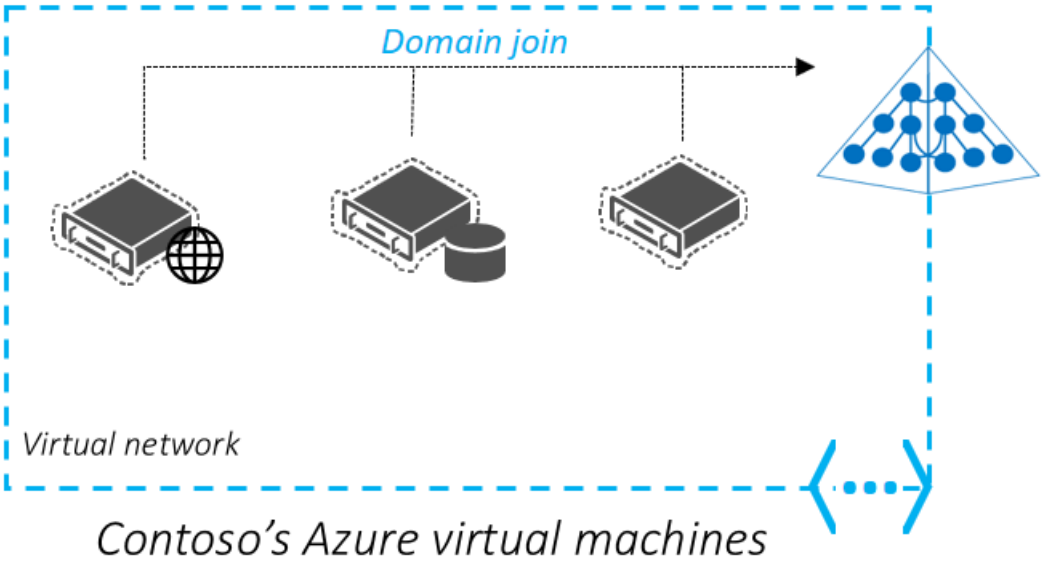
Azure Active Directory Domain Services (Azure AD DS) provides managed domain services such as domain join, group policy, lightweight directory access protocol (LDAP), and Kerberos / NTLM authentication. Azure AD DS integrates with your existing Azure AD tenant, which makes it possible for users to sign in using their existing credentials. You use these domain services without the need to deploy, manage, and patch domain controllers in the cloud, which provides a smoother lift-and-shift of on-premises resources to Azure.

This article outlines some common business scenarios where Azure AD DS provides value and meets those needs.

Secure administration of Azure virtual machines

To let you use a single set of AD credentials, Azure virtual machines (VMs) can be joined to an Azure AD DS managed domain. This approach reduces credential management issues such as maintaining local administrator accounts on each VM or separate accounts and passwords between environments.

VMs that are joined to a managed domain can also be administered and secured using group policy. Required security baselines can be applied to VMs to lock them down in accordance with corporate security guidelines. For example, you can use group policy management capabilities to restrict the types of applications that can be launched on the VM.



Let's look at a common example scenario. As servers and other infrastructure reaches end-of-life, Contoso wants to move applications currently hosted on premises to the cloud. Their current IT standard mandates that servers hosting corporate applications must be domain-joined and managed using group policy.

Contoso's IT administrator would prefer to domain join VMs deployed in Azure to make administration easier as users can then sign in using their corporate credentials. When domain-joined, VMs can also be configured to comply with required security baselines using group policy objects (GPOs). Contoso would prefer not to deploy, monitor, and manage their own domain controllers in Azure.

Azure AD DS is a great fit for this use-case. A managed domain lets you domain-join VMs, use a single set of credentials, and apply group policy. And because it's a managed domain, you don't have to configure and maintain

the domain controllers yourself.

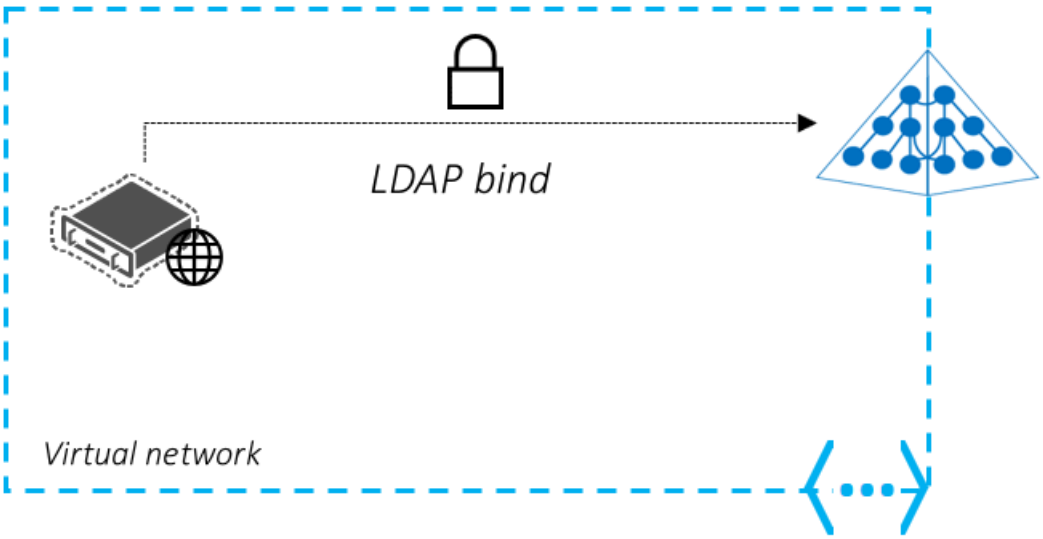
Deployment notes

The following deployment considerations apply to this example use case:

- Managed domains use a single, flat Organizational Unit (OU) structure by default. All domain-joined VMs are in a single OU. If desired, you can create custom OUs.
- Azure AD DS uses a built-in GPO each for the users and computers containers. For additional control, you can create custom GPOs and target them to custom OUs.
- Azure AD DS supports the base AD computer object schema. You can't extend the computer object's schema.

Lift-and-shift on-premises applications that use LDAP bind authentication

As a sample scenario, Contoso has an on-premises application that was purchased from an ISV many years ago. The application is currently in maintenance mode by the ISV and requesting changes to the application is prohibitively expensive. This application has a web-based frontend that collects user credentials using a web form and then authenticates users by performing an LDAP bind to the on-premises AD DS environment.



Contoso would like to migrate this application to Azure. The application should continue to works as-is, with no changes needed. Additionally, users should be able to authenticate using their existing corporate credentials and without additional training. It should be transparent to end users where the application is running.

For this scenario, Azure AD DS lets applications perform LDAP binds as part of the authentication process. Legacy on-premises applications can lift-and-shift into Azure and continue to seamlessly authenticate users without any change in configuration or user experience.

Deployment notes

The following deployment considerations apply to this example use case:

- Make sure that the application doesn't need to modify/write to the directory. LDAP write access to a managed domain isn't supported.
- You can't change passwords directly against a managed domain. End users can change their password either using Azure AD's self-service password change mechanism or against the on-premises directory. These changes are then automatically synchronized and available in the managed domain.

Lift-and-shift on-premises applications that use LDAP read to access the directory

Like the previous example scenario, let's assume Contoso has an on-premises line-of-business (LOB) application that was developed almost a decade ago. This application is directory aware and was designed to use LDAP to read information/attributes about users from AD DS. The application doesn't modify attributes or otherwise write to the directory.

Contoso wants to migrate this application to Azure and retire the aging on-premises hardware currently hosting this application. The application can't be rewritten to use modern directory APIs such as the REST-based Microsoft Graph API. A lift-and-shift option is desired where the application can be migrated to run in the cloud, without modifying code or rewriting the application.

To help with this scenario, Azure AD DS lets applications perform LDAP reads against the managed domain to get the attribute information it needs. The application doesn't need to be rewritten, so a lift-and-shift into Azure lets users continue to use the app without realizing there's a change in where it runs.

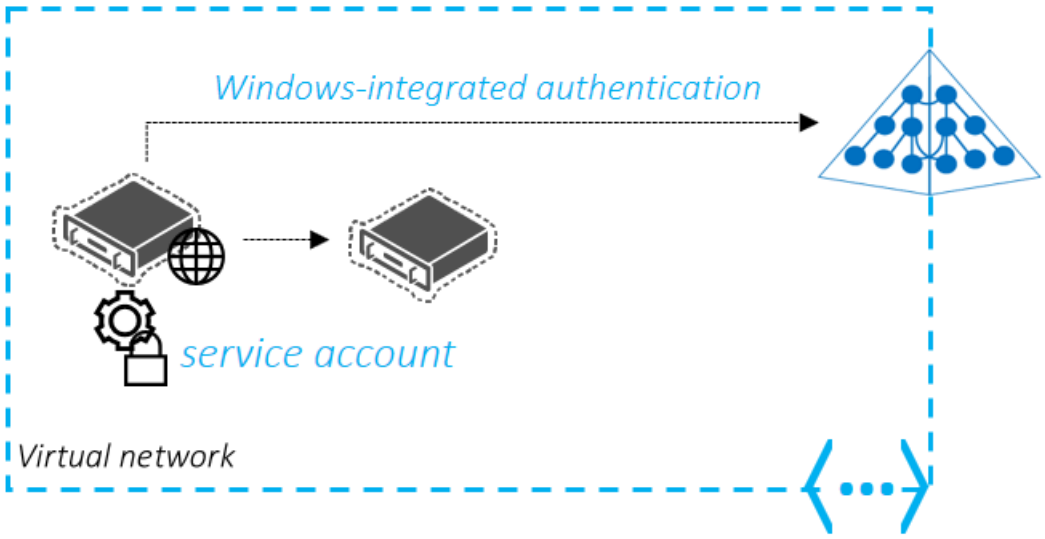
Deployment notes

The following deployment considerations apply to this example use case:

- Make sure that the application doesn't need to modify/write to the directory. LDAP write access to a managed domain isn't supported.
- Make sure that the application doesn't need a custom/extended Active Directory schema. Schema extensions aren't supported in Azure AD DS.

Migrate an on-premises service or daemon application to Azure

Some applications include multiple tiers, where one of the tiers needs to perform authenticated calls to a backend tier, such as a database. AD service accounts are commonly used in these scenarios. When you lift-and-shift applications into Azure, Azure AD DS lets you continue to use service accounts in the same way. You can choose to use the same service account that is synchronized from your on-premises directory to Azure AD or create a custom OU and then create a separate service account in that OU. With either approach, applications continue to function the same way to make authenticated calls to other tiers and services.



In this example scenario, Contoso has a custom-built software vault application that includes a web front end, a SQL server, and a backend FTP server. Windows-integrated authentication using service accounts authenticates the web front end to the FTP server. The web front end is set up to run as a service account. The backend server is configured to authorize access from the service account for the web front end. Contoso doesn't want to deploy and manage their own domain controller VMs in the cloud to move this application to Azure.

For this scenario, the servers hosting the web front end, SQL server, and the FTP server can be migrated to Azure VMs and joined to a managed domain. The VMs can then use the same service account in their on-premises directory for the app's authentication purposes, which is synchronized through Azure AD using Azure AD Connect.

Deployment notes

The following deployment considerations apply to this example use case:

- Make sure that the applications use a username and password for authentication. Certificate or smartcard-based authentication isn't supported by Azure AD DS.
- You can't change passwords directly against a managed domain. End users can change their password either using Azure AD's self-service password change mechanism or against the on-premises directory. These changes are then automatically synchronized and available in the managed domain.

Windows Server remote desktop services deployments in Azure

You can use Azure AD DS to provide managed domain services to remote desktop servers deployed in Azure.

For more information about this deployment scenario, see how to integrate Azure AD Domain Services with your RDS deployment.

Domain-joined HDInsight clusters

You can set up an Azure HDInsight cluster that is joined to a managed domain with Apache Ranger enabled. You can create and apply Hive policies through Apache Ranger, and allow users, such as data scientists, to connect to Hive using ODBC-based tools like Excel or Tableau. We continue to work to add other workloads, such as HBase, Spark, and Storm to domain-joined HDInsight.

For more information about this deployment scenario, see how to configure domain-joined HDInsight clusters

Next steps

To get started, Create and configure an Azure Active Directory Domain Services managed domain.

Resource forest concepts and features for Azure Active Directory Domain Services

7/20/2020 • 8 minutes to read • Edit Online

Azure Active Directory Domain Services (Azure AD DS) provides a sign-in experience for legacy, on-premises, line-of-business applications. Users, groups, and password hashes of on-premises and cloud users are synchronized to the Azure AD DS managed domain. These synchronized password hashes are what gives users a single set of credentials they can use for the on-premises AD DS, Office 365, and Azure Active Directory.

Although secure and provides additional security benefits, some organizations can't synchronize those user passwords hashes to Azure AD or Azure AD DS. Users in an organization may not know their password because they only use smart card authentication. These limitations prevent some organizations from using Azure AD DS to lift and shift on-premises classic applications to Azure.

To address these needs and restrictions, you can create a managed domain that uses a resource forest. This conceptual article explains what forests are, and how they trust other resources to provide a secure authentication method. Azure AD DS resource forests are currently in preview.

> **IMPORTANT**
>
> Azure AD DS resource forests don't currently support Azure HDInsight or Azure Files. The default Azure AD DS user forests do support both of these additional services.

What are forests?

A *forest* is a logical construct used by Active Directory Domain Services (AD DS) to group one or more *domains*. The domains then store objects for user or groups, and provide authentication services.

In an Azure AD DS managed domain, the forest only contains one domain. On-premises AD DS forests often contain many domains. In large organizations, especially after mergers and acquisitions, you may end up with multiple on-premises forests that each then contain multiple domains.

By default, a managed domain is created as a *user* forest. This type of forest synchronizes all objects from Azure AD, including any user accounts created in an on-premises AD DS environment. User accounts can directly authenticate against the managed domain, such as to sign in to a domain-joined VM. A user forest works when the password hashes can be synchronized and users aren't using exclusive sign-in methods like smart card authentication.

In a managed domain *resource* forest, users authenticate over a one-way forest *trust* from their on-premises AD DS. With this approach, the user objects and password hashes aren't synchronized to the managed domain. The user objects and credentials only exist in the on-premises AD DS. This approach lets enterprises host resources and application platforms in Azure that depend on classic authentication such LDAPS, Kerberos, or NTLM, but any authentication issues or concerns are removed. Azure AD DS resource forests are currently in preview.

Resource forests also provide the capability to lift-and-shift your applications one component at a time. Many legacy on-premises applications are multi-tiered, often using a web server or front end and many database-related components. These tiers make it hard to lift-and-shift the entire application to the cloud in one step. With resource forests, you can lift your application to the cloud in phased approach, which makes it easier to move your application to Azure.

What are trusts?

Organizations that have more than one domain often need users to access shared resources in a different domain. Access to these shared resources requires that users in one domain authenticate to another domain. To provide these authentication and authorization capabilities between clients and servers in different domains, there must be a *trust* between the two domains.

With domain trusts, the authentication mechanisms for each domain trust the authentications coming from the other domain. Trusts help provide controlled access to shared resources in a resource domain (the *trusting* domain) by verifying that incoming authentication requests come from a trusted authority (the *trusted* domain). Trusts act as bridges that only allow validated authentication requests to travel between domains.

How a trust passes authentication requests depends on how it's configured. Trusts can be configured in one of the following ways:

- **One-way** - provides access from the trusted domain to resources in the trusting domain.
- **Two-way** - provides access from each domain to resources in the other domain.

Trusts are also be configured to handle additional trust relationships in one of the following ways:

- **Nontransitive** - The trust exists only between the two trust partner domains.
- **Transitive** - Trust automatically extends to any other domains that either of the partners trusts.

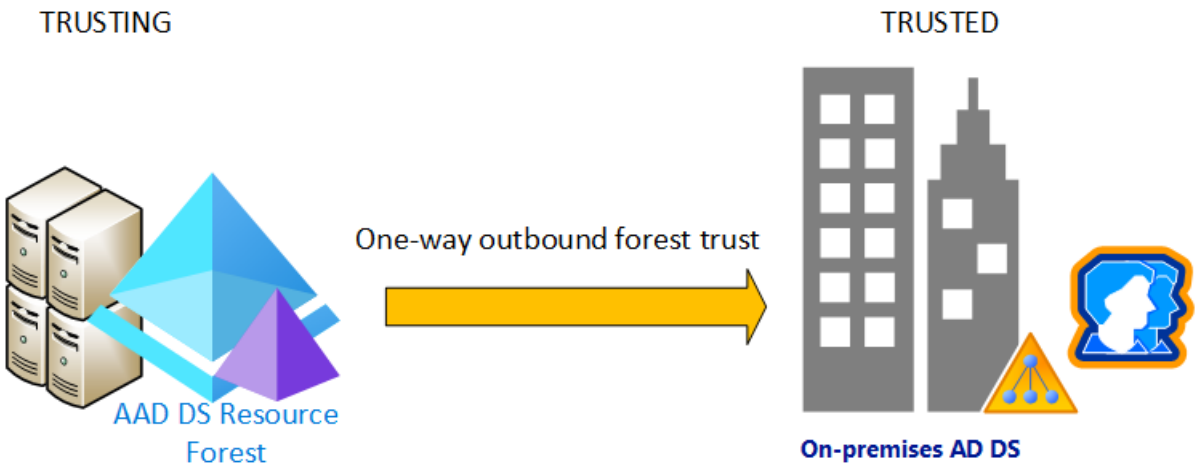
In some cases, trust relationships are automatically established when domains are created. Other times, you must choose a type of trust and explicitly establish the appropriate relationships. The specific types of trusts used and the structure of those trust relationships depend on how the AD DS directory is organized, and whether different versions of Windows coexist on the network.

Trusts between two forests

You can extend domain trusts within a single forest to another forest by manually creating a one-way or two-way forest trust. A forest trust is a transitive trust that exists only between a forest root domain and a second forest root domain.

- A one-way forest trust allows all users in one forest to trust all domains in the other forest.
- A two-way forest trust forms a transitive trust relationship between every domain in both forests.

The transitivity of forest trusts is limited to the two forest partners. The forest trust doesn't extend to additional forests trusted by either of the partners.



You can create different domain and forest trust configurations depending on the AD DS structure of the organization. Azure AD DS only supports a one-way forest trust. In this configuration, resources in the managed domain can trust all domains in an on-premises forest.

Supporting technology for trusts

Trusts use various services and features, such as DNS to locate domain controllers in partnering forests. Trusts also depend on NTLM and Kerberos authentication protocols and on Windows-based authorization and access control mechanisms to help provide a secured communications infrastructure across AD DS domains and forests. The following services and features help support successful trust relationships.

DNS

AD DS needs DNS for domain controller (DC) location and naming. The following support from DNS is provided for AD DS to work successfully:

- A name resolution service that lets network hosts and services to locate DCs.
- A naming structure that enables an enterprise to reflect its organizational structure in the names of its directory service domains.

A DNS domain namespace is usually deployed that mirrors the AD DS domain namespace. If there's an existing DNS namespace before the AD DS deployment, the DNS namespace is typically partitioned for AD DS, and a DNS subdomain and delegation for the AD DS forest root is created. Additional DNS domain names are then added for each AD DS child domain.

DNS is also used to support the location of AD DS DCs. The DNS zones are populated with DNS resource records that enable network hosts and services to locate AD DS DCs.

Applications and Net Logon

Both applications and the Net Logon service are components of the Windows distributed security channel model. Applications integrated with Windows Server and AD DS use authentication protocols to communicate with the Net Logon service so that a secured path can be established over which authentication can occur.

Authentication Protocols

AD DS DCs authenticate users and applications using one of the following protocols:

- **Kerberos version 5 authentication protocol**
  - The Kerberos version 5 protocol is the default authentication protocol used by on-premises computers running Windows and supporting third-party operating systems. This protocol is specified in RFC 1510 and is fully integrated with AD DS, server message block (SMB), HTTP, and remote procedure call (RPC), as well as the client and server applications that use these protocols.
  - When the Kerberos protocol is used, the server doesn't have to contact the DC. Instead, the client gets a ticket for a server by requesting one from a DC in the server account domain. The server then validates the ticket without consulting any other authority.
  - If any computer involved in a transaction doesn't support the Kerberos version 5 protocol, the NTLM protocol is used.
- **NTLM authentication protocol**
  - The NTLM protocol is a classic network authentication protocol used by older operating systems. For compatibility reasons, it's used by AD DS domains to process network authentication requests that come from applications designed for earlier Windows-based clients and servers, and third-party operating systems.
  - When the NTLM protocol is used between a client and a server, the server must contact a domain authentication service on a DC to verify the client credentials. The server authenticates the client by forwarding the client credentials to a DC in the client account domain.
  - When two AD DS domains or forests are connected by a trust, authentication requests made using these protocols can be routed to provide access to resources in both forests.

Authorization and access control

Authorization and trust technologies work together to provide a secured communications infrastructure across AD DS domains or forests. Authorization determines what level of access a user has to resources in a domain. Trusts facilitate cross-domain authorization of users by providing a path for authenticating users in other domains so their requests to shared resources in those domains can be authorized.

When an authentication request made in a trusting domain is validated by the trusted domain, it's passed to the target resource. The target resource then determines whether to authorize the specific request made by the user, service, or computer in the trusted domain based on its access control configuration.

Trusts provide this mechanism to validate authentication requests that are passed to a trusting domain. Access control mechanisms on the resource computer determine the final level of access granted to the requestor in the trusted domain.

Next steps

To learn more about trusts, see How do forest trusts work in Azure AD DS?

To get started with creating a managed domain with a resource forest, see Create and configure an Azure AD DS managed domain. You can then Create an outbound forest trust to an on-premises domain (preview).

How trust relationships work for resource forests in Azure Active Directory Domain Services

7/20/2020 • 19 minutes to read • Edit Online

Active Directory Domain Services (AD DS) provides security across multiple domains or forests through domain and forest trust relationships. Before authentication can occur across trusts, Windows must first check if the domain being requested by a user, computer, or service has a trust relationship with the domain of the requesting account.

To check for this trust relationship, the Windows security system computes a trust path between the domain controller (DC) for the server that receives the request and a DC in the domain of the requesting account.

The access control mechanisms provided by AD DS and the Windows distributed security model provide an environment for the operation of domain and forest trusts. For these trusts to work properly, every resource or computer must have a direct trust path to a DC in the domain in which it is located.

The trust path is implemented by the Net Logon service using an authenticated remote procedure call (RPC) connection to the trusted domain authority. A secured channel also extends to other AD DS domains through interdomain trust relationships. This secured channel is used to obtain and verify security information, including security identifiers (SIDs) for users and groups.

For an overview of how trusts apply to Azure AD DS, see Resource forest concepts and features.

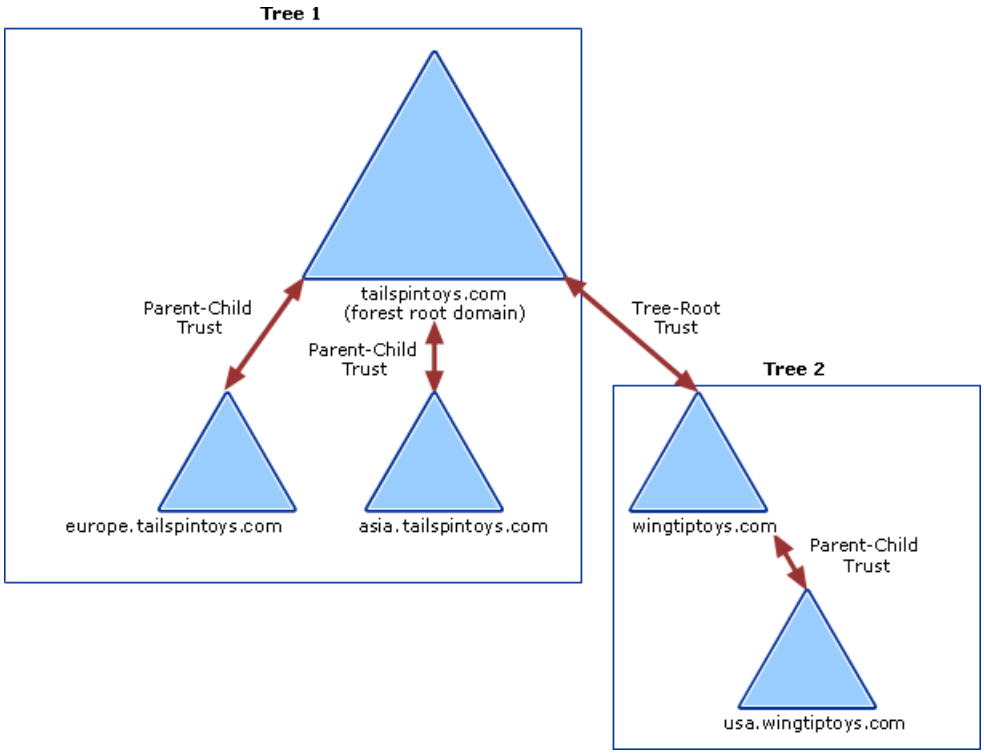
To get started using trusts in Azure AD DS, create a managed domain that uses forest trusts.

Trust relationship flows

The flow of secured communications over trusts determines the elasticity of a trust. How you create or configure a trust determines how far the communication extends within or across forests.

The flow of communication over trusts is determined by the direction of the trust. Trusts can be one-way or two-way, and can be transitive or non-transitive.

The following diagram shows that all domains in *Tree 1* and *Tree 2* have transitive trust relationships by default. As a result, users in *Tree 1* can access resources in domains in *Tree 2* and users in *Tree 1* can access resources in *Tree 2*, when the proper permissions are assigned at the resource.



One-way and two-way trusts

Trust relationships enable access to resources can be either one-way or two-way.

A one-way trust is a unidirectional authentication path created between two domains. In a one-way trust between *Domain A* and *Domain B*, users in *Domain A* can access resources in *Domain B*. However, users in *Domain B* can't access resources in *Domain A*.

Some one-way trusts can be either non-transitive or transitive depending on the type of trust being created.

In a two-way trust, *Domain A* trusts *Domain B* and *Domain B* trusts *Domain A*. This configuration means that authentication requests can be passed between the two domains in both directions. Some two-way relationships can be non-transitive or transitive depending on the type of trust being created.

All domain trusts in an AD DS forest are two-way, transitive trusts. When a new child domain is created, a two-way, transitive trust is automatically created between the new child domain and the parent domain.

Transitive and non-transitive trusts

Transitivity determines whether a trust can be extended outside of the two domains with which it was formed.

- A transitive trust can be used to extend trust relationships with other domains.
- A non-transitive trust can be used to deny trust relationships with other domains.

Each time you create a new domain in a forest, a two-way, transitive trust relationship is automatically created between the new domain and its parent domain. If child domains are added to the new domain, the trust path flows upward through the domain hierarchy extending the initial trust path created between the new domain and its parent domain. Transitive trust relationships flow upward through a domain tree as it is formed, creating transitive trusts between all domains in the domain tree.

Authentication requests follow these trust paths, so accounts from any domain in the forest can be authenticated by any other domain in the forest. With a single sign in process, accounts with the proper permissions can access resources in any domain in the forest.

Forest trusts

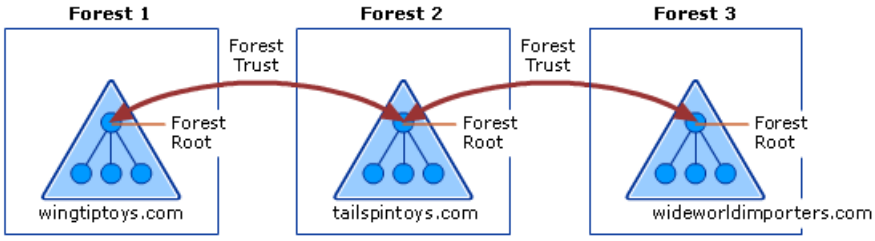
Forest trusts help you to manage a segmented AD DS infrastructures and support access to resources and other objects across multiple forests. Forest trusts are useful for service providers, companies undergoing mergers or

acquisitions, collaborative business extranets, and companies seeking a solution for administrative autonomy.

Using forest trusts, you can link two different forests to form a one-way or two-way transitive trust relationship. A forest trust allows administrators to connect two AD DS forests with a single trust relationship to provide a seamless authentication and authorization experience across the forests.

A forest trust can only be created between a forest root domain in one forest and a forest root domain in another forest. Forest trusts can only be created between two forests and can't be implicitly extended to a third forest. This behavior means that if a forest trust is created between *Forest 1* and *Forest 2*, and another forest trust is created between *Forest 2* and *Forest 3*, *Forest 1* doesn't have an implicit trust with *Forest 3*.

The following diagram shows two separate forest trust relationships between three AD DS forests in a single organization.



This example configuration provides the following access:

- Users in *Forest 2* can access resources in any domain in either *Forest 1* or *Forest 3*
- Users in *Forest 3* can access resources in any domain in *Forest 2*
- Users in *Forest 1* can access resources in any domain in *Forest 2*

This configuration doesn't allow users in *Forest 1* to access resources in *Forest 3* or vice versa. To allow users in both *Forest 1* and *Forest 3* to share resources, a two-way transitive trust must be created between the two forests.

If a one-way forest trust is created between two forests, members of the trusted forest can utilize resources located in the trusting forest. However, the trust operates in only one direction.

For example, when a one-way, forest trust is created between *Forest 1* (the trusted forest) and *Forest 2* (the trusting forest):

- Members of *Forest 1* can access resources located in *Forest 2*.
- Members of *Forest 2* can't access resources located in *Forest 1* using the same trust.

> **IMPORTANT**
> Azure AD Domain Services resource forest only supports a one-way forest trust to on-premises Active Directory.

Forest trust requirements

Before you can create a forest trust, you need to verify you have the correct Domain Name System (DNS) infrastructure in place. Forest trusts can only be created when one of the following DNS configurations is available:

- A single root DNS server is the root DNS server for both forest DNS namespaces - the root zone contains delegations for each of the DNS namespaces and the root hints of all DNS servers include the root DNS server.
- Where there is no shared root DNS server, and the root DNS servers for each forest DNS namespace use DNS conditional forwarders for each DNS namespace to route queries for names in the other namespace.

> **IMPORTANT**
> Azure AD Domain Services resource forest must use this DNS configuration. Hosting a DNS namespace other than the resource forest DNS namespace is not a feature of Azure AD Domain Services. Conditional forwarders is the proper configuration.

- Where there is no shared root DNS server, and the root DNS servers for each forest DNS namespace are use DNS secondary zones are configured in each DNS namespace to route queries for names in the other namespace.

To create a forest trust, you must be a member of the Domain Admins group (in the forest root domain) or the Enterprise Admins group in Active Directory. Each trust is assigned a password that the administrators in both forests must know. Members of Enterprise Admins in both forests can create the trusts in both forests at once and, in this scenario, a password that is cryptographically random is automatically generated and written for both forests.

The outbound forest trust for Azure AD Domain Services is created in the Azure portal. You don't manually create the trust with the managed domain itself. The incoming forest trust must be configured by a user with the privileges previously noted in the on-premises Active Directory.

Trust processes and interactions

Many inter-domain and inter-forest transactions depend on domain or forest trusts in order to complete various tasks. This section describes the processes and interactions that occur as resources are accessed across trusts and authentication referrals are evaluated.

Overview of authentication referral processing

When a request for authentication is referred to a domain, the domain controller in that domain must determine whether a trust relationship exists with the domain from which the request comes. The direction of the trust and whether the trust is transitive or nontransitive must also be determined before it authenticates the user to access resources in the domain. The authentication process that occurs between trusted domains varies according to the authentication protocol in use. The Kerberos V5 and NTLM protocols process referrals for authentication to a domain differently

Kerberos V5 referral processing

The Kerberos V5 authentication protocol is dependent on the Net Logon service on domain controllers for client authentication and authorization information. The Kerberos protocol connects to an online Key Distribution Center (KDC) and the Active Directory account store for session tickets.

The Kerberos protocol also uses trusts for cross-realm ticket-granting services (TGS) and to validate Privilege Attribute Certificates (PACs) across a secured channel. The Kerberos protocol performs cross-realm authentication only with non-Windows-brand operating system Kerberos realms such as an MIT Kerberos realm and does not need to interact with the Net Logon service.

If the client uses Kerberos V5 for authentication, it requests a ticket to the server in the target domain from a domain controller in its account domain. The Kerberos KDC acts as a trusted intermediary between the client and server and provides a session key that enables the two parties to authenticate each other. If the target domain is different from the current domain, the KDC follows a logical process to determine whether an authentication request can be referred:

1. Is the current domain trusted directly by the domain of the server that is being requested?
    - If yes, send the client a referral to the requested domain.
    - If no, go to the next step.

2. Does a transitive trust relationship exist between the current domain and the next domain on the trust path?

   - If yes, send the client a referral to the next domain on the trust path.
   - If no, send the client a sign-in denied message.

NTLM referral processing

The NTLM authentication protocol is dependent on the Net Logon service on domain controllers for client authentication and authorization information. This protocol authenticates clients that do not use Kerberos authentication. NTLM uses trusts to pass authentication requests between domains.

If the client uses NTLM for authentication, the initial request for authentication goes directly from the client to the resource server in the target domain. This server creates a challenge to which the client responds. The server then sends the user's response to a domain controller in its computer account domain. This domain controller checks the user account against its security accounts database.

If the account does not exist in the database, the domain controller determines whether to perform pass-through authentication, forward the request, or deny the request by using the following logic:

1. Does the current domain have a direct trust relationship with the user's domain?

   - If yes, the domain controller sends the credentials of the client to a domain controller in the user's domain for pass-through authentication.
   - If no, go to the next step.
2. Does the current domain have a transitive trust relationship with the user's domain?

   - If yes, pass the authentication request on to the next domain in the trust path. This domain controller repeats the process by checking the user's credentials against its own security accounts database.
   - If no, send the client a logon-denied message.

Kerberos-based processing of authentication requests over forest trusts

When two forests are connected by a forest trust, authentication requests made using the Kerberos V5 or NTLM protocols can be routed between forests to provide access to resources in both forests.

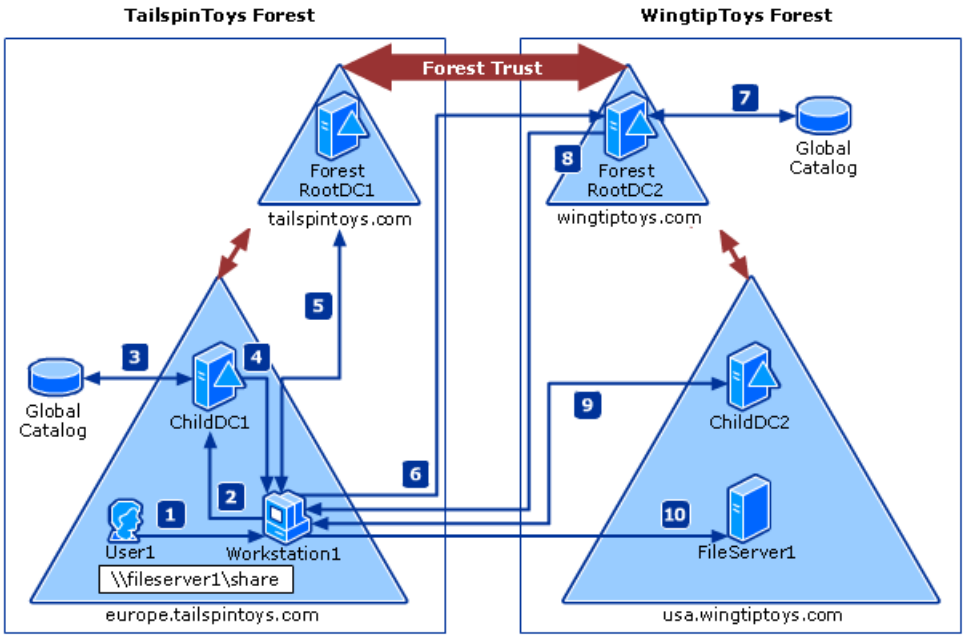
When a forest trust is first established, each forest collects all of the trusted namespaces in its partner forest and stores the information in a trusted domain object. Trusted namespaces include domain tree names, user principal name (UPN) suffixes, service principal name (SPN) suffixes, and security ID (SID) namespaces used in the other forest. TDO objects are replicated to the global catalog.

Before authentication protocols can follow the forest trust path, the service principal name (SPN) of the resource computer must be resolved to a location in the other forest. An SPN can be one of the following names:

- The DNS name of a host.
- The DNS name of a domain.
- The distinguished name of a service connection point object.

When a workstation in one forest attempts to access data on a resource computer in another forest, the Kerberos authentication process contacts the domain controller for a service ticket to the SPN of the resource computer. Once the domain controller queries the global catalog and determines that the SPN is not in the same forest as the domain controller, the domain controller sends a referral for its parent domain back to the workstation. At that point, the workstation queries the parent domain for the service ticket and continues to follow the referral chain until it reaches the domain where the resource is located.

The following diagram and steps provide a detailed description of the Kerberos authentication process that's used when computers running Windows attempt to access resources from a computer located in another forest.



1. *User1* signs in to *Workstation1* using credentials from the *europe.tailspintoys.com* domain. The user then attempts to access a shared resource on *FileServer1* located in the *usa.wingtiptoys.com* forest.
2. *Workstation1* contacts the Kerberos KDC on a domain controller in its domain, *ChildDC1*, and requests a service ticket for the *FileServer1* SPN.
3. *ChildDC1* does not find the SPN in its domain database and queries the global catalog to see if any domains in the *tailspintoys.com* forest contain this SPN. Because a global catalog is limited to its own forest, the SPN is not found.

   The global catalog then checks its database for information about any forest trusts that are established with its forest. If found, it compares the name suffixes listed in the forest trust trusted domain object (TDO) to the suffix of the target SPN to find a match. Once a match is found, the global catalog provides a routing hint back to *ChildDC1*.

   Routing hints help direct authentication requests toward the destination forest. Hints are only used when all traditional authentication channels, such as local domain controller and then global catalog, fail to locate an SPN.
4. *ChildDC1* sends a referral for its parent domain back to *Workstation1*.
5. *Workstation1* contacts a domain controller in *ForestRootDC1* (its parent domain) for a referral to a domain controller (*ForestRootDC2*) in the forest root domain of the *wingtiptoys.com* forest.
6. *Workstation1* contacts *ForestRootDC2* in the *wingtiptoys.com* forest for a service ticket to the requested service.
7. *ForestRootDC2* contacts its global catalog to find the SPN, and the global catalog finds a match for the SPN and sends it back to *ForestRootDC2*.
8. *ForestRootDC2* then sends the referral to *usa.wingtiptoys.com* back to *Workstation1*.
9. *Workstation1* contacts the KDC on *ChildDC2* and negotiates the ticket for *User1* to gain access to *FileServer1*.
10. Once *Workstation1* has a service ticket, it sends the service ticket to *FileServer1*, which reads *User1*'s security credentials and constructs an access token accordingly.

Trusted domain object

Each domain or forest trust within an organization is represented by a Trusted Domain Object (TDO) stored in the *System* container within its domain.

TDO contents

The information contained in a TDO varies depending on whether a TDO was created by a domain trust or by a forest trust.

When a domain trust is created, attributes such as the DNS domain name, domain SID, trust type, trust transitivity, and the reciprocal domain name are represented in the TDO. Forest trust TDOs store additional attributes to identify all of the trusted namespaces from the partner forest. These attributes include domain tree names, user principal name (UPN) suffixes, service principal name (SPN) suffixes, and security ID (SID) namespaces.

Because trusts are stored in Active Directory as TDOs, all domains in a forest have knowledge of the trust relationships that are in place throughout the forest. Similarly, when two or more forests are joined together through forest trusts, the forest root domains in each forest have knowledge of the trust relationships that are in place throughout all of the domains in trusted forests.

TDO password changes

Both domains in a trust relationship share a password, which is stored in the TDO object in Active Directory. As part of the account maintenance process, every 30 days the trusting domain controller changes the password stored in the TDO. Because all two-way trusts are actually two one-way trusts going in opposite directions, the process occurs twice for two-way trusts.

A trust has a trusting and a trusted side. On the trusted side, any writable domain controller can be used for the process. On the trusting side, the PDC emulator performs the password change.

To change a password, the domain controllers complete the following process:

1. The primary domain controller (PDC) emulator in the trusting domain creates a new password. A domain controller in the trusted domain never initiates the password change. It's always initiated by the trusting domain PDC emulator.
2. The PDC emulator in the trusting domain sets the *OldPassword* field of the TDO object to the current *NewPassword* field.
3. The PDC emulator in the trusting domain sets the *NewPassword* field of the TDO object to the new password. Keeping a copy of the previous password makes it possible to revert to the old password if the domain controller in the trusted domain fails to receive the change, or if the change is not replicated before a request is made that uses the new trust password.
4. The PDC emulator in the trusting domain makes a remote call to a domain controller in the trusted domain asking it to set the password on the trust account to the new password.
5. The domain controller in the trusted domain changes the trust password to the new password.
6. On each side of the trust, the updates are replicated to the other domain controllers in the domain. In the trusting domain, the change triggers an urgent replication of the trusted domain object.

The password is now changed on both domain controllers. Normal replication distributes the TDO objects to the other domain controllers in the domain. However, it's possible for the domain controller in the trusting domain to change the password without successfully updating a domain controller in the trusted domain. This scenario might occur because a secured channel, which is required to process the password change, couldn't be established. It's also possible that the domain controller in the trusted domain might be unavailable at some point during the process and might not receive the updated password.

To deal with situations in which the password change isn't successfully communicated, the domain controller in the trusting domain never changes the new password unless it has successfully authenticated (set up a secured channel) using the new password. This behavior is why both the old and new passwords are kept in the TDO

object of the trusting domain.

A password change isn't finalized until authentication using the password succeeds. The old, stored password can be used over the secured channel until the domain controller in the trusted domain receives the new password, thus enabling uninterrupted service.

If authentication using the new password fails because the password is invalid, the trusting domain controller tries to authenticate using the old password. If it authenticates successfully with the old password, it resumes the password change process within 15 minutes.

Trust password updates need to replicate to the domain controllers of both sides of the trust within 30 days. If the trust password is changed after 30 days and a domain controller only has the N-2 password, it cannot use the trust from the trusting side and cannot create a secure channel on the trusted side.

Network ports used by trusts

Because trusts must be deployed across various network boundaries, they might have to span one or more firewalls. When this is the case, you can either tunnel trust traffic across a firewall or open specific ports in the firewall to allow the traffic to pass through.

> **IMPORTANT**
> Active Directory Domain Services does not support restricting Active Directory RPC traffic to specific ports.

Read the **Windows Server 2008 and later versions** section of the Microsoft Support Article How to configure a firewall for Active Directory domains and trusts to learn about the ports needed for a forest trust.

Supporting services and tools

To support trusts and authentication, some additional features and management tools are used.

Net Logon

The Net Logon service maintains a secured channel from a Windows-based computer to a DC. It's also used in the following trust-related processes:

- Trust setup and management - Net Logon helps maintain trust passwords, gathers trust information, and verifies trusts by interacting with the LSA process and the TDO.

  For Forest trusts, the trust information includes the Forest Trust Information (*FTInfo*) record, which includes the set of namespaces that a trusted forest claims to manage, annotated with a field that indicates whether each claim is trusted by the trusting forest.

- Authentication – Supplies user credentials over a secured channel to a domain controller and returns the domain SIDs and user rights for the user.

- Domain controller location – Helps with finding or locating domain controllers in a domain or across domains.

- Pass-through validation – Credentials of users in other domains are processed by Net Logon. When a trusting domain needs to verify the identity of a user, it passes the user's credentials through Net Logon to the trusted domain for verification.

- Privilege Attribute Certificate (PAC) verification – When a server using the Kerberos protocol for authentication needs to verify the PAC in a service ticket, it sends the PAC across the secure channel to its domain controller for verification.

Local Security Authority

The Local Security Authority (LSA) is a protected subsystem that maintains information about all aspects of local security on a system. Collectively known as local security policy, the LSA provides various services for translation between names and identifiers.

The LSA security subsystem provides services in both kernel mode and user mode for validating access to objects, checking user privileges, and generating audit messages. LSA is responsible for checking the validity of all session tickets presented by services in trusted or untrusted domains.

Management tools

Administrators can use *Active Directory Domains and Trusts*, *Netdom* and *Nltest* to expose, create, remove, or modify trusts.

- *Active Directory Domains and Trusts* is the Microsoft Management Console (MMC) that is used to administer domain trusts, domain and forest functional levels, and user principal name suffixes.
- The *Netdom* and *Nltest* command-line tools can be used to find, display, create, and manage trusts. These tools communicate directly with the LSA authority on a domain controller.

Next steps

To learn more about resource forests, see How do forest trusts work in Azure AD DS?

To get started with creating a managed domain with a resource forest, see Create and configure an Azure AD DS managed domain. You can then Create an outbound forest trust to an on-premises domain (preview).

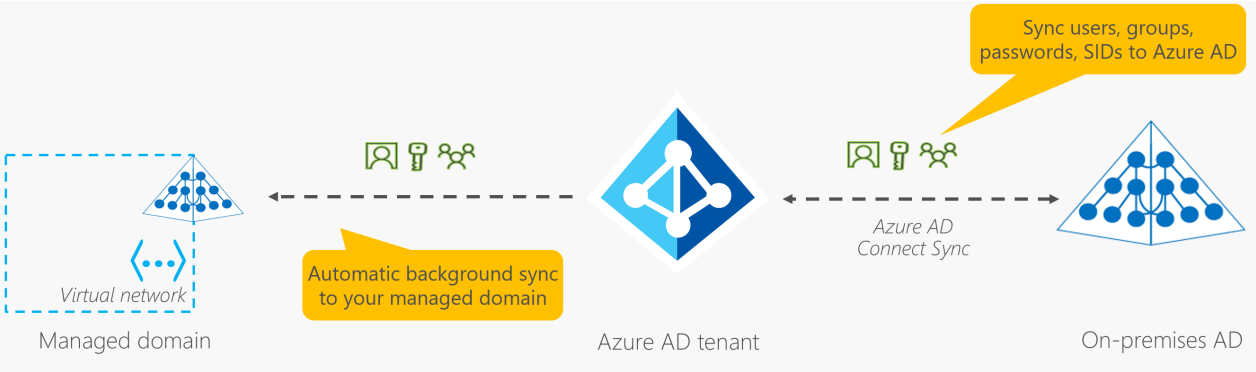
How objects and credentials are synchronized in an Azure Active Directory Domain Services managed domain

7/20/2020 • 9 minutes to read • Edit Online

Objects and credentials in an Azure Active Directory Domain Services (Azure AD DS) managed domain can either be created locally within the domain, or synchronized from an Azure Active Directory (Azure AD) tenant. When you first deploy Azure AD DS, an automatic one-way synchronization is configured and started to replicate the objects from Azure AD. This one-way synchronization continues to run in the background to keep the Azure AD DS managed domain up-to-date with any changes from Azure AD. No synchronization occurs from Azure AD DS back to Azure AD.

In a hybrid environment, objects and credentials from an on-premises AD DS domain can be synchronized to Azure AD using Azure AD Connect. Once those objects are successfully synchronized to Azure AD, the automatic background sync then makes those objects and credentials available to applications using the managed domain.

The following diagram illustrates how synchronization works between Azure AD DS, Azure AD, and an optional on-premises AD DS environment:



Synchronization from Azure AD to Azure AD DS

User accounts, group memberships, and credential hashes are synchronized one way from Azure AD to Azure AD DS. This synchronization process is automatic. You don't need to configure, monitor, or manage this synchronization process. The initial synchronization may take a few hours to a couple of days, depending on the number of objects in the Azure AD directory. After the initial synchronization is complete, changes that are made in Azure AD, such as password or attribute changes, are then automatically synchronized to Azure AD DS.

When a user is created in Azure AD, they're not synchronized to Azure AD DS until they change their password in Azure AD. This password change process causes the password hashes for Kerberos and NTLM authentication to be generated and stored in Azure AD. The password hashes are needed to successfully authenticate a user in Azure AD DS.

The synchronization process is one way / unidirectional by design. There's no reverse synchronization of changes from Azure AD DS back to Azure AD. A managed domain is largely read-only except for custom OUs that you can create. You can't make changes to user attributes, user passwords, or group memberships within a managed domain.

Attribute synchronization and mapping to Azure AD DS

The following table lists some common attributes and how they're synchronized to Azure AD DS.

ATTRIBUTE IN AZURE AD DS	SOURCE	NOTES
UPN	User's *UPN* attribute in Azure AD tenant	The UPN attribute from the Azure AD tenant is synchronized as-is to Azure AD DS. The most reliable way to sign in to a managed domain is using the UPN.
SAMAccountName	User's *mailNickname* attribute in Azure AD tenant or autogenerated	The *SAMAccountName* attribute is sourced from the *mailNickname* attribute in the Azure AD tenant. If multiple user accounts have the same *mailNickname* attribute, the *SAMAccountName* is autogenerated. If the user's *mailNickname* or *UPN* prefix is longer than 20 characters, the *SAMAccountName* is autogenerated to meet the 20 character limit on *SAMAccountName* attributes.
Passwords	User's password from the Azure AD tenant	Legacy password hashes required for NTLM or Kerberos authentication are synchronized from the Azure AD tenant. If the Azure AD tenant is configured for hybrid synchronization using Azure AD Connect, these password hashes are sourced from the on-premises AD DS environment.
Primary user/group SID	Autogenerated	The primary SID for user/group accounts is autogenerated in Azure AD DS. This attribute doesn't match the primary user/group SID of the object in an on-premises AD DS environment. This mismatch is because the managed domain has a different SID namespace than the on-premises AD DS domain.
SID history for users and groups	On-premises primary user and group SID	The *SidHistory* attribute for users and groups in Azure AD DS is set to match the corresponding primary user or group SID in an on-premises AD DS environment. This feature helps make lift-and-shift of on-premises applications to Azure AD DS easier as you don't need to re-ACL resources.

TIP

Sign in to the managed domain using the UPN format The *SAMAccountName* attribute, such as `AADDSCONTOSO\driley`, may be auto-generated for some user accounts in a managed domain. Users' auto-generated *SAMAccountName* may differ from their UPN prefix, so isn't always a reliable way to sign in.

For example, if multiple users have the same *mailNickname* attribute or users have overly long UPN prefixes, the *SAMAccountName* for these users may be auto-generated. Use the UPN format, such as `driley@aaddscontoso.com`, to reliably sign in to a managed domain.

Attribute mapping for user accounts

The following table illustrates how specific attributes for user objects in Azure AD are synchronized to corresponding attributes in Azure AD DS.

USER ATTRIBUTE IN AZURE AD	USER ATTRIBUTE IN AZURE AD DS
accountEnabled	userAccountControl (sets or clears the ACCOUNT_DISABLED bit)
city	l
country	co
department	department
displayName	displayName
employeedId	employeeId
facsimileTelephoneNumber	facsimileTelephoneNumber
givenName	givenName
jobTitle	title
mail	mail
mailNickname	msDS-AzureADMailNickname
mailNickname	SAMAccountName (may sometimes be autogenerated)
manager	manager
mobile	mobile
objectid	msDS-AzureADObjectId
onPremiseSecurityIdentifier	sidHistory
passwordPolicies	userAccountControl (sets or clears the DONT_EXPIRE_PASSWORD bit)
physicalDeliveryOfficeName	physicalDeliveryOfficeName
postalCode	postalCode
preferredLanguage	preferredLanguage
proxyAddresses	proxyAddresses
state	st
streetAddress	streetAddress

USER ATTRIBUTE IN AZURE AD	USER ATTRIBUTE IN AZURE AD DS
surname	sn
telephoneNumber	telephoneNumber
userPrincipalName	userPrincipalName

Attribute mapping for groups

The following table illustrates how specific attributes for group objects in Azure AD are synchronized to corresponding attributes in Azure AD DS.

GROUP ATTRIBUTE IN AZURE AD	GROUP ATTRIBUTE IN AZURE AD DS
displayName	displayName
displayName	SAMAccountName (may sometimes be autogenerated)
mail	mail
mailNickname	msDS-AzureADMailNickname
objectid	msDS-AzureADObjectId
onPremiseSecurityIdentifier	sidHistory
proxyAddresses	proxyAddresses
securityEnabled	groupType

Synchronization from on-premises AD DS to Azure AD and Azure AD DS

Azure AD Connect is used to synchronize user accounts, group memberships, and credential hashes from an on-premises AD DS environment to Azure AD. Attributes of user accounts such as the UPN and on-premises security identifier (SID) are synchronized. To sign in using Azure AD DS, legacy password hashes required for NTLM and Kerberos authentication are also synchronized to Azure AD.

> **IMPORTANT**
> Azure AD Connect should only be installed and configured for synchronization with on-premises AD DS environments. It's not supported to install Azure AD Connect in a managed domain to synchronize objects back to Azure AD.

If you configure write-back, changes from Azure AD are synchronized back to the on-premises AD DS environment. For example, if a user changes their password using Azure AD self-service password management, the password is updated back in the on-premises AD DS environment.

> **NOTE**
> Always use the latest version of Azure AD Connect to ensure you have fixes for all known bugs.

Synchronization from a multi-forest on-premises environment

Many organizations have a fairly complex on-premises AD DS environment that includes multiple forests. Azure AD Connect supports synchronizing users, groups, and credential hashes from multi-forest environments to Azure AD.

Azure AD has a much simpler and flat namespace. To enable users to reliably access applications secured by Azure AD, resolve UPN conflicts across user accounts in different forests. Managed domains use a flat OU structure, similar to Azure AD. All user accounts and groups are stored in the *AADDC Users* container, despite being synchronized from different on-premises domains or forests, even if you've configured a hierarchical OU structure on-premises. The managed domain flattens any hierarchical OU structures.

As previously detailed, there's no synchronization from Azure AD DS back to Azure AD. You can create a custom Organizational Unit (OU) in Azure AD DS and then users, groups, or service accounts within those custom OUs. None of the objects created in custom OUs are synchronized back to Azure AD. These objects are available only within the managed domain, and aren't visible using Azure AD PowerShell cmdlets, Microsoft Graph API, or using the Azure AD management UI.

What isn't synchronized to Azure AD DS

The following objects or attributes aren't synchronized from an on-premises AD DS environment to Azure AD or Azure AD DS:

- **Excluded attributes:** You can choose to exclude certain attributes from synchronizing to Azure AD from an on-premises AD DS environment using Azure AD Connect. These excluded attributes aren't then available in Azure AD DS.
- **Group Policies:** Group Policies configured in an on-premises AD DS environment aren't synchronized to Azure AD DS.
- **Sysvol folder:** The contents of the *Sysvol* folder in an on-premises AD DS environment aren't synchronized to Azure AD DS.
- **Computer objects:** Computer objects for computers joined to an on-premises AD DS environment aren't synchronized to Azure AD DS. These computers don't have a trust relationship with the managed domain and only belong to the on-premises AD DS environment. In Azure AD DS, only computer objects for computers that have explicitly domain-joined to the managed domain are shown.
- **SidHistory attributes for users and groups:** The primary user and primary group SIDs from an on-premises AD DS environment are synchronized to Azure AD DS. However, existing *SidHistory* attributes for users and groups aren't synchronized from the on-premises AD DS environment to Azure AD DS.
- **Organization Units (OU) structures:** Organizational Units defined in an on-premises AD DS environment don't synchronize to Azure AD DS. There are two built-in OUs in Azure AD DS - one for users, and one for computers. The managed domain has a flat OU structure. You can choose to create a custom OU in your managed domain.

Password hash synchronization and security considerations

When you enable Azure AD DS, legacy password hashes for NTLM + Kerberos authentication are required. Azure AD doesn't store clear-text passwords, so these hashes can't be automatically generated for existing user accounts. Once generated and stored, NTLM and Kerberos compatible password hashes are always stored in an encrypted manner in Azure AD.

The encryption keys are unique to each Azure AD tenant. These hashes are encrypted such that only Azure AD DS has access to the decryption keys. No other service or component in Azure AD has access to the decryption keys.

Legacy password hashes are then synchronized from Azure AD into the domain controllers for a managed domain. The disks for these managed domain controllers in Azure AD DS are encrypted at rest. These password hashes are stored and secured on these domain controllers similar to how passwords are stored and secured in

an on-premises AD DS environment.

For cloud-only Azure AD environments, users must reset/change their password in order for the required password hashes to be generated and stored in Azure AD. For any cloud user account created in Azure AD after enabling Azure AD Domain Services, the password hashes are generated and stored in the NTLM and Kerberos compatible formats. All cloud user accounts must change their password before they're synchronized to Azure AD DS.

For hybrid user accounts synced from on-premises AD DS environment using Azure AD Connect, you must configure Azure AD Connect to synchronize password hashes in the NTLM and Kerberos compatible formats.

Next steps

For more information on the specifics of password synchronization, see How password hash synchronization works with Azure AD Connect.

To get started with Azure AD DS, create a managed domain.

Implement password hash synchronization with Azure AD Connect sync

7/20/2020 • 14 minutes to read • Edit Online

This article provides information that you need to synchronize your user passwords from an on-premises Active Directory instance to a cloud-based Azure Active Directory (Azure AD) instance.

How password hash synchronization works

The Active Directory domain service stores passwords in the form of a hash value representation, of the actual user password. A hash value is a result of a one-way mathematical function (the *hashing algorithm*). There is no method to revert the result of a one-way function to the plain text version of a password.

To synchronize your password, Azure AD Connect sync extracts your password hash from the on-premises Active Directory instance. Extra security processing is applied to the password hash before it is synchronized to the Azure Active Directory authentication service. Passwords are synchronized on a per-user basis and in chronological order.

The actual data flow of the password hash synchronization process is similar to the synchronization of user data. However, passwords are synchronized more frequently than the standard directory synchronization window for other attributes. The password hash synchronization process runs every 2 minutes. You cannot modify the frequency of this process. When you synchronize a password, it overwrites the existing cloud password.

The first time you enable the password hash synchronization feature, it performs an initial synchronization of the passwords of all in-scope users. You cannot explicitly define a subset of user passwords that you want to synchronize. However, if there are multiple connectors, it is possible to disable password hash sync for some connectors but not others using the Set-ADSyncAADPasswordSyncConfiguration cmdlet.

When you change an on-premises password, the updated password is synchronized, most often in a matter of minutes. The password hash synchronization feature automatically retries failed synchronization attempts. If an error occurs during an attempt to synchronize a password, an error is logged in your event viewer.

The synchronization of a password has no impact on the user who is currently signed in. Your current cloud service session is not immediately affected by a synchronized password change that occurs, while you are signed in, to a cloud service. However, when the cloud service requires you to authenticate again, you need to provide your new password.

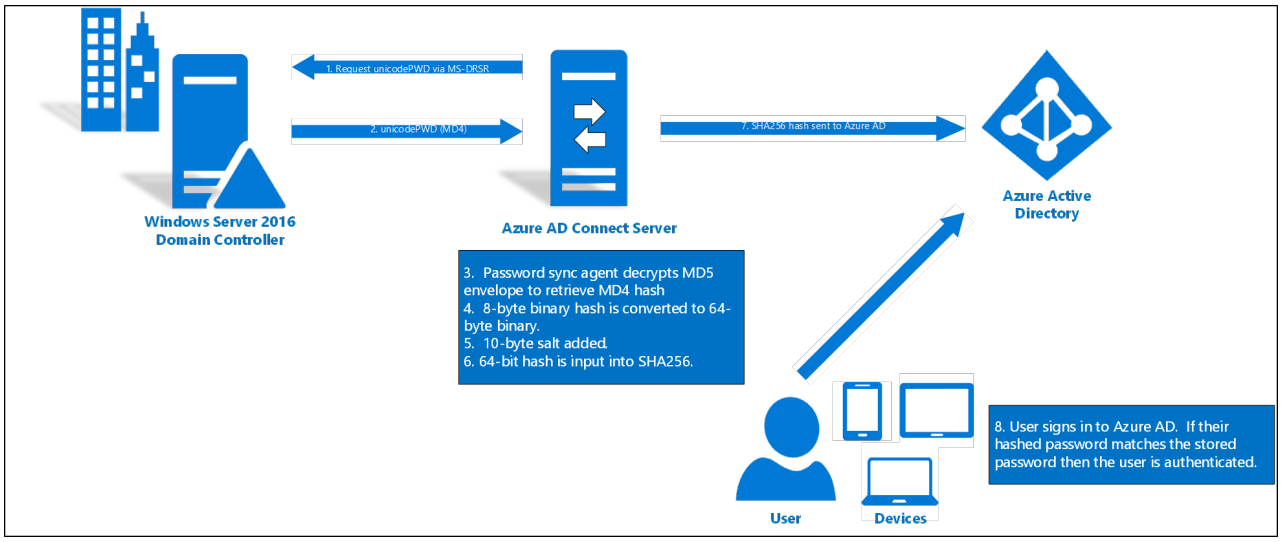
A user must enter their corporate credentials a second time to authenticate to Azure AD, regardless of whether they're signed in to their corporate network. This pattern can be minimized, however, if the user selects the Keep me signed in (KMSI) check box at sign-in. This selection sets a session cookie that bypasses authentication for 180 days. KMSI behavior can be enabled or disabled by the Azure AD administrator. In addition, you can reduce password prompts by turning on Seamless SSO, which automatically signs users in when they are on their corporate devices connected to your corporate network.

> **NOTE**
>
> Password sync is only supported for the object type user in Active Directory. It is not supported for the iNetOrgPerson object type.

Detailed description of how password hash synchronization works

The following section describes, in-depth, how password hash synchronization works between Active Directory and

Azure AD.



1. Every two minutes, the password hash synchronization agent on the AD Connect server requests stored password hashes (the unicodePwd attribute) from a DC. This request is via the standard MS-DRSR replication protocol used to synchronize data between DCs. The service account must have Replicate Directory Changes and Replicate Directory Changes All AD permissions (granted by default on installation) to obtain the password hashes.
2. Before sending, the DC encrypts the MD4 password hash by using a key that is a MD5 hash of the RPC session key and a salt. It then sends the result to the password hash synchronization agent over RPC. The DC also passes the salt to the synchronization agent by using the DC replication protocol, so the agent will be able to decrypt the envelope.
3. After the password hash synchronization agent has the encrypted envelope, it uses MD5CryptoServiceProvider and the salt to generate a key to decrypt the received data back to its original MD4 format. The password hash synchronization agent never has access to the clear text password. The password hash synchronization agent's use of MD5 is strictly for replication protocol compatibility with the DC, and it is only used on premises between the DC and the password hash synchronization agent.
4. The password hash synchronization agent expands the 16-byte binary password hash to 64 bytes by first converting the hash to a 32-byte hexadecimal string, then converting this string back into binary with UTF-16 encoding.
5. The password hash synchronization agent adds a per user salt, consisting of a 10-byte length salt, to the 64-byte binary to further protect the original hash.
6. The password hash synchronization agent then combines the MD4 hash plus the per user salt, and inputs it into the PBKDF2 function. 1000 iterations of the HMAC-SHA256 keyed hashing algorithm are used.
7. The password hash synchronization agent takes the resulting 32-byte hash, concatenates both the per user salt and the number of SHA256 iterations to it (for use by Azure AD), then transmits the string from Azure AD Connect to Azure AD over TLS.
8. When a user attempts to sign in to Azure AD and enters their password, the password is run through the same MD4+salt+PBKDF2+HMAC-SHA256 process. If the resulting hash matches the hash stored in Azure AD, the user has entered the correct password and is authenticated.

> **NOTE**
>
> The original MD4 hash is not transmitted to Azure AD. Instead, the SHA256 hash of the original MD4 hash is transmitted. As a result, if the hash stored in Azure AD is obtained, it cannot be used in an on-premises pass-the-hash attack.

Security considerations

When synchronizing passwords, the plain-text version of your password is not exposed to the password hash

synchronization feature, to Azure AD, or any of the associated services.

User authentication takes place against Azure AD rather than against the organization's own Active Directory instance. The SHA256 password data stored in Azure AD--a hash of the original MD4 hash--is more secure than what is stored in Active Directory. Further, because this SHA256 hash cannot be decrypted, it cannot be brought back to the organization's Active Directory environment and presented as a valid user password in a pass-the-hash attack.

Password policy considerations

There are two types of password policies that are affected by enabling password hash synchronization:

- Password complexity policy
- Password expiration policy

Password complexity policy

When password hash synchronization is enabled, the password complexity policies in your on-premises Active Directory instance override complexity policies in the cloud for synchronized users. You can use all of the valid passwords from your on-premises Active Directory instance to access Azure AD services.

> **NOTE**
> Passwords for users that are created directly in the cloud are still subject to password policies as defined in the cloud.

Password expiration policy

If a user is in the scope of password hash synchronization, by default the cloud account password is set to *Never Expire*.

You can continue to sign in to your cloud services by using a synchronized password that is expired in your on-premises environment. Your cloud password is updated the next time you change the password in the on-premises environment.

EnforceCloudPasswordPolicyForPasswordSyncedUsers

If there are synchronized users that only interact with Azure AD integrated services and must also comply with a password expiration policy, you can force them to comply with your Azure AD password expiration policy by enabling the *EnforceCloudPasswordPolicyForPasswordSyncedUsers* feature.

When *EnforceCloudPasswordPolicyForPasswordSyncedUsers* is disabled (which is the default setting), Azure AD Connect sets thePasswordPoliciesattribute of synchronized users to "DisablePasswordExpiration". This is done every time a user's password is synchronized and instructs Azure AD to ignore the cloud password expiration policy for that user. You can check the value of the attribute using theAzure ADPowerShell module with the following command:

```
(Get-AzureADUser -objectID <User Object ID>).passwordpolicies
```

To enable the EnforceCloudPasswordPolicyForPasswordSyncedUsers feature, run the following command using the MSOnline PowerShell module as shown below. You would have to type yes for the Enable parameter as shown below :

```
Set-MsolDirSyncFeature -Feature EnforceCloudPasswordPolicyForPasswordSyncedUsers
cmdlet Set-MsolDirSyncFeature at command pipeline position 1
Supply values for the following parameters:
Enable: yes
Confirm
Continue with this operation?
[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"): y
```

Once enabled, Azure AD does not go to each synchronized user to remove the `DisablePasswordExpiration` value

from thePasswordPoliciesattribute. Instead, the value is set to `None` during thenext password syncfor each user when they next change their password in on-premises AD.

It is recommended to enable EnforceCloudPasswordPolicyForPasswordSyncedUsers, prior to enabling password hash sync, so that the initial sync of password hashes does not add the `DisablePasswordExpiration` value to the PasswordPolicies attribute for the users.

The default Azure AD password policy requires users to change their passwords every 90 days. Ifyour policy in AD is also 90 days, the two policies should match. However, if the AD policy is not 90 days, you can update the Azure AD password policy to match by using the Set-MsolPasswordPolicyPowerShell command.

Azure AD supports a separate password expiration policy per registered domain.

Caveat: If there are synchronized accounts that need to have non-expiring passwords in Azure AD, you must explicitly add the `DisablePasswordExpiration` value to the PasswordPolicies attribute of the user object in Azure AD. You can do this by running the following command.

```
Set-AzureADUser -ObjectID <User Object ID> -PasswordPolicies "DisablePasswordExpiration"
```

> **NOTE**
> The Set-MsolPasswordPolicy PowerShell command will not work on federated domains.

Synchronizing temporary passwords and "Force Password Change on Next Logon"

It is typical to force a user to change their password during their first logon, especially after an admin password reset occurs. It is commonly known as setting a "temporary" password and is completed by checking the "User must change password at next logon" flag on a user object in Active Directory (AD).

The temporary password functionality helps to ensure that the transfer of ownership of the credential is completed on first use, to minimize the duration of time in which more than one individual has knowledge of that credential.

To support temporary passwords in Azure AD for synchronized users, you can enable the *ForcePasswordChangeOnLogOn* feature, by running the following command on your Azure AD Connect server:

```
Set-ADSyncAADCompanyFeature -ForcePasswordChangeOnLogOn$true
```

> **NOTE**
> Forcing a user to change their password on next logon requires a password change at the same time. Azure AD Connect will not pick up the force password change flag by itself; it is supplemental to the detected password change that occurs during password hash sync.

Caution

You should only use this feature when SSPR and Password Writeback are enabled on the tenant. This is so that if a user changes their password via SSPR, it will be synchronized to Active Directory.

Account expiration

If your organization uses the accountExpires attribute as part of user account management, this attribute is not synchronized to Azure AD. As a result, an expired Active Directory account in an environment configured for password hash synchronization will still be active in Azure AD. We recommend that if the account is expired, a workflow action should trigger a PowerShell script that disables the user's Azure AD account (use the Set-AzureADUser cmdlet). Conversely, when the account is turned on, the Azure AD instance should be turned on.

Overwrite synchronized passwords

An administrator can manually reset your password by using Windows PowerShell.

In this case, the new password overrides your synchronized password, and all password policies defined in the

cloud are applied to the new password.

If you change your on-premises password again, the new password is synchronized to the cloud, and it overrides the manually updated password.

The synchronization of a password has no impact on the Azure user who is signed in. Your current cloud service session is not immediately affected by a synchronized password change that occurs while you're signed in to a cloud service. KMSI extends the duration of this difference. When the cloud service requires you to authenticate again, you need to provide your new password.

Additional advantages

- Generally, password hash synchronization is simpler to implement than a federation service. It doesn't require any additional servers, and eliminates dependence on a highly available federation service to authenticate users.
- Password hash synchronization can also be enabled in addition to federation. It may be used as a fallback if your federation service experiences an outage.

Password hash sync process for Azure AD Domain Services

If you use Azure AD Domain Services to provide legacy authentication for applications and services that need to use Kerberos, LDAP, or NTLM, some additional processes are part of the password hash synchronization flow. Azure AD Connect uses the additional following process to synchronize password hashes to Azure AD for use in Azure AD Domain Services:

> **IMPORTANT**
>
> Azure AD Connect should only be installed and configured for synchronization with on-premises AD DS environments. It's not supported to install Azure AD Connect in an Azure AD DS managed domain to synchronize objects back to Azure AD.
>
> Azure AD Connect only synchronizes legacy password hashes when you enable Azure AD DS for your Azure AD tenant. The following steps aren't used if you only use Azure AD Connect to synchronize an on-premises AD DS environment with Azure AD.
>
> If your legacy applications don't use NTLM authentication or LDAP simple binds, we recommend that you disable NTLM password hash synchronization for Azure AD DS. For more information, see Disable weak cipher suites and NTLM credential hash synchronization.

1. Azure AD Connect retrieves the public key for the tenant's instance of Azure AD Domain Services.
2. When a user changes their password, the on-premises domain controller stores the result of the password change (hashes) in two attributes:
   - *unicodePwd* for the NTLM password hash.
   - *supplementalCredentials* for the Kerberos password hash.
3. Azure AD Connect detects password changes through the directory replication channel (attribute changes needing to replicate to other domain controllers).
4. For each user whose password has changed, Azure AD Connect performs the following steps:
   - Generates a random AES 256-bit symmetric key.
   - Generates a random initialization vector needed for the first round of encryption.
   - Extracts Kerberos password hashes from the *supplementalCredentials* attributes.
   - Checks the Azure AD Domain Services security configuration *SyncNtlmPasswords* setting.
     - If this setting is disabled, generates a random, high-entropy NTLM hash (different from the user's password). This hash is then combined with the exacted Kerberos password hashes from the *supplementalCrendetials* attribute into one data structure.
     - If enabled, combines the value of the *unicodePwd* attribute with the extracted Kerberos password

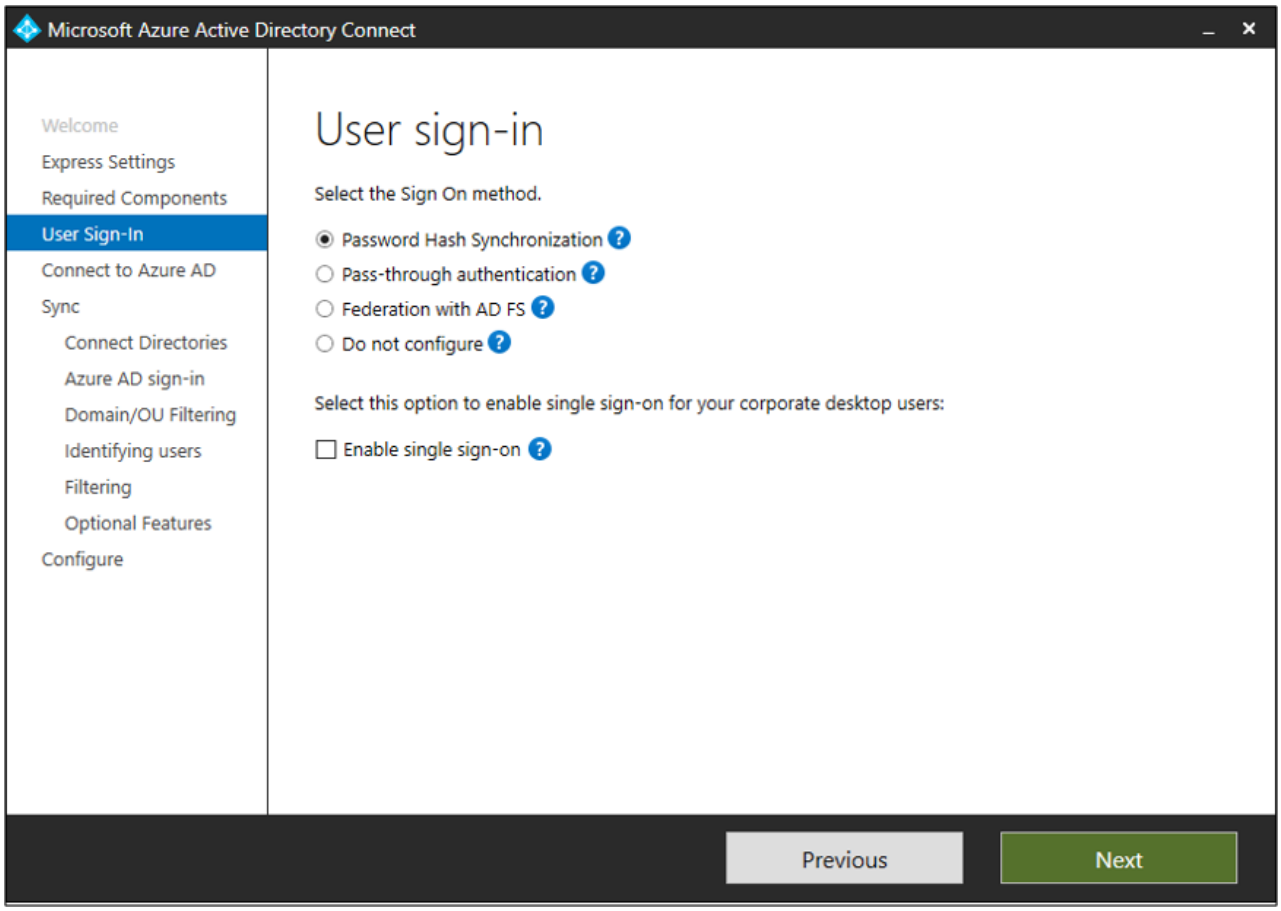
hashes from the *supplementalCredentials* attribute into one data structure.
   - Encrypts the single data structure using the AES symmetric key.
   - Encrypts the AES symmetric key using the tenant's Azure AD Domain Services public key.
5. Azure AD Connect transmits the encrypted AES symmetric key, the encrypted data structure containing the password hashes, and the initialization vector to Azure AD.
6. Azure AD stores the encrypted AES symmetric key, the encrypted data structure, and the initialization vector for the user.
7. Azure AD pushes the encrypted AES symmetric key, the encrypted data structure, and the initialization vector using an internal synchronization mechanism over an encrypted HTTP session to Azure AD Domain Services.
8. Azure AD Domain Services retrieves the private key for the tenant's instance from Azure Key vault.
9. For each encrypted set of data (representing a single user's password change), Azure AD Domain Services then performs the following steps:
   - Uses its private key to decrypt the AES symmetric key.
   - Uses the AES symmetric key with the initialization vector to decrypt the encrypted data structure that contains the password hashes.
   - Writes the Kerberos password hashes it receives to the Azure AD Domain Services domain controller. The hashes are saved into the user object's *supplementalCredentials* attribute that is encrypted to the Azure AD Domain Services domain controller's public key.
   - Azure AD Domain Services writes the NTLM password hash it received to the Azure AD Domain Services domain controller. The hash is saved into the user object's *unicodePwd* attribute that is encrypted to the Azure AD Domain Services domain controller's public key.

Enable password hash synchronization

> **IMPORTANT**
> If you are migrating from AD FS (or other federation technologies) to Password Hash Synchronization, we highly recommend that you follow our detailed deployment guide published here.

When you install Azure AD Connect by using the **Express Settings** option, password hash synchronization is automatically enabled. For more information, see Getting started with Azure AD Connect using express settings.

If you use custom settings when you install Azure AD Connect, password hash synchronization is available on the user sign-in page. For more information, see Custom installation of Azure AD Connect.



Password hash synchronization and FIPS

If your server has been locked down according to Federal Information Processing Standard (FIPS), then MD5 is disabled.

To enable MD5 for password hash synchronization, perform the following steps:

1. Go to %programfiles%\Azure AD Sync\Bin.
2. Open miiserver.exe.config.
3. Go to the configuration/runtime node at the end of the file.
4. Add the following node: `<enforceFIPSPolicy enabled="false"/>`
5. Save your changes.

For reference, this snippet is what it should look like:

```
<configuration>
    <runtime>
        <enforceFIPSPolicy enabled="false"/>
    </runtime>
</configuration>
```

For information about security and FIPS, see Azure AD password hash sync, encryption, and FIPS compliance.

Troubleshoot password hash synchronization

If you have problems with password hash synchronization, see Troubleshoot password hash synchronization.

Next steps

- Azure AD Connect sync: Customizing synchronization options
- Integrating your on-premises identities with Azure Active Directory

- Get a step-by-step deployment plan for migrating from ADFS to Password Hash Synchronization

Benefits of migration from the Classic to Resource Manager deployment model in Azure Active Directory Domain Services

7/20/2020 • 2 minutes to read • Edit Online

Azure Active Directory Domain Services (Azure AD DS) lets you migrate an existing managed domain that uses the Classic deployment model to the Resource Manager deployment model. Azure AD DS managed domains that use the Resource Manager deployment model provide additional features such as fine-grained password policy, audit logs, and account lockout protection.

This article outlines the benefits for migration. To get started, see Migrate Azure AD Domain Services from the Classic virtual network model to Resource Manager.

> **NOTE**
> In 2017, Azure AD Domain Services became available to host in an Azure Resource Manager network. Since then, we have been able to build a more secure service using the Azure Resource Manager's modern capabilities. Because Azure Resource Manager deployments fully replace classic deployments, Azure AD DS classic virtual network deployments will be retired on March 1, 2023.
>
> For more information, see the official deprecation notice

Migration benefits

The migration process takes an existing managed domain that uses the Classic deployment model and moves to use the Resource Manager deployment model. When you migrate a managed domain from the Classic to Resource Manager deployment model, you avoid the need to rejoin machines to the managed domain or delete the managed domain and create one from scratch. VMs continue to be joined to the managed domain at the end of the migration process.

After migration, Azure AD DS provides many features that are only available for domains using Resource Manager deployment model, such as the following:

- Fine-grained password policy support.
- Faster synchronization speeds between Azure AD and Azure AD Domain Services.
- Two new attributes that synchronize from Azure AD - *manager* and *employeeID*.
- Access to higher-powered domain controllers when you upgrade the SKU.
- AD account lockout protection.
- Email notifications for alerts on your managed domain.
- Use Azure Workbooks and Azure monitor to view audit logs and sign-in activity.
- In supported regions, Azure Availability Zones.
- Integrations with other Azure products such as Azure Files, HD Insights, and Windows Virtual Desktop.
- Support has access to more telemetry and can help troubleshoot more effectively.
- Encryption at rest using Azure Managed Disks for the data on the managed domain controllers.

Managed domains that use a Resource Manager deployment model help you stay up-to-date with the latest new features. New features aren't available for managed domains that use the Classic deployment model.

Next steps

To get started, see Migrate Azure AD Domain Services from the Classic virtual network model to Resource Manager.

What is Azure Active Directory?

7/20/2020 • 9 minutes to read • Edit Online

Azure Active Directory (Azure AD) is Microsoft's cloud-based identity and access management service, which helps your employees sign in and access resources in:

- External resources, such as Microsoft Office 365, the Azure portal, and thousands of other SaaS applications.
- Internal resources, such as apps on your corporate network and intranet, along with any cloud apps developed by your own organization. For more information about creating a tenant for your organization, see Quickstart: Create a new tenant in Azure Active Directory.

To learn the difference between Azure AD and Active Directory Domain Services, see Compare Active Directory to Azure Active Directory. You can also use the various Microsoft Cloud for Enterprise Architects Series posters to better understand the core identity services in Azure, Azure AD, and Office 365.

Who uses Azure AD?

Azure AD is intended for:

- **IT admins.** As an IT admin, you can use Azure AD to control access to your apps and your app resources, based on your business requirements. For example, you can use Azure AD to require multi-factor authentication when accessing important organizational resources. Additionally, you can use Azure AD to automate user provisioning between your existing Windows Server AD and your cloud apps, including Office 365. Finally, Azure AD gives you powerful tools to automatically help protect user identities and credentials and to meet your access governance requirements. To get started, sign up for a free 30-day Azure Active Directory Premium trial.
- **App developers.** As an app developer, you can use Azure AD as a standards-based approach for adding single sign-on (SSO) to your app, allowing it to work with a user's pre-existing credentials. Azure AD also provides APIs that can help you build personalized app experiences using existing organizational data. To get started, sign up for a free 30-day Azure Active Directory Premium trial. For more information, you can also see Azure Active Directory for developers.
- **Microsoft 365, Office 365, Azure, or Dynamics CRM Online subscribers.** As a subscriber, you're already using Azure AD. Each Microsoft 365, Office 365, Azure, and Dynamics CRM Online tenant is automatically an Azure AD tenant. You can immediately start to manage access to your integrated cloud apps.

What are the Azure AD licenses?

Microsoft Online business services, such as Office 365 or Microsoft Azure, require Azure AD for sign-in and to help with identity protection. If you subscribe to any Microsoft Online business service, you automatically get Azure AD with access to all the free features.

To enhance your Azure AD implementation, you can also add paid capabilities by upgrading to Azure Active Directory Premium P1 or Premium P2 licenses. Azure AD paid licenses are built on top of your existing free directory, providing self-service, enhanced monitoring, security reporting, and secure access for your mobile users.

> **NOTE**
>
> For the pricing options of these licenses, see Azure Active Directory Pricing.
>
> Azure Active Directory Premium P1 and Premium P2 are not currently supported in China. For more information about Azure AD pricing, contact the Azure Active Directory Forum.

- **Azure Active Directory Free.** Provides user and group management, on-premises directory synchronization, basic reports, self-service password change for cloud users, and single sign-on across Azure, Office 365, and many popular SaaS apps.
- **Azure Active Directory Premium P1.** In addition to the Free features, P1 also lets your hybrid users access both on-premises and cloud resources. It also supports advanced administration, such as dynamic groups, self-service group management, Microsoft Identity Manager (an on-premises identity and access management suite) and cloud write-back capabilities, which allow self-service password reset for your on-premises users.
- **Azure Active Directory Premium P2.** In addition to the Free and P1 features, P2 also offers Azure Active Directory Identity Protection to help provide risk-based Conditional Access to your apps and critical company data and Privileged Identity Management to help discover, restrict, and monitor administrators and their access to resources and to provide just-in-time access when needed.
- **"Pay as you go" feature licenses.** You can also get additional feature licenses, such as Azure Active Directory Business-to-Customer (B2C). B2C can help you provide identity and access management solutions for your customer-facing apps. For more information, see Azure Active Directory B2C documentation.

For more information about associating an Azure subscription to Azure AD, see Associate or add an Azure subscription to Azure Active Directory and for more information about assigning licenses to your users, see How to: Assign or remove Azure Active Directory licenses.

Which features work in Azure AD?

After you choose your Azure AD license, you'll get access to some or all of the following features for your organization:

CATEGORY	DESCRIPTION
Application management	Manage your cloud and on-premises apps using Application Proxy, single sign-on, the My Apps portal (also known as the Access panel), and Software as a Service (SaaS) apps. For more information, see How to provide secure remote access to on-premises applications and Application Management documentation.
Authentication	Manage Azure Active Directory self-service password reset, Multi-Factor Authentication, custom banned password list, and smart lockout. For more information, see Azure AD Authentication documentation.
Azure Active Directory for developers	Build apps that sign in all Microsoft identities, get tokens to call Microsoft Graph, other Microsoft APIs, or custom APIs. For more information, see Microsoft identity platform (Azure Active Directory for developers).

CATEGORY	DESCRIPTION
Business-to-Business (B2B)	Manage your guest users and external partners, while maintaining control over your own corporate data. For more information, see Azure Active Directory B2B documentation.
Business-to-Customer (B2C)	Customize and control how users sign up, sign in, and manage their profiles when using your apps. For more information, see Azure Active Directory B2C documentation.
Conditional Access	Manage access to your cloud apps. For more information, see Azure AD Conditional Access documentation.
Device Management	Manage how your cloud or on-premises devices access your corporate data. For more information, see Azure AD Device Management documentation.
Domain services	Join Azure virtual machines to a domain without using domain controllers. For more information, see Azure AD Domain Services documentation.
Enterprise users	Manage license assignment, access to apps, and set up delegates using groups and administrator roles. For more information, see Azure Active Directory user management documentation.
Hybrid identity	Use Azure Active Directory Connect and Connect Health to provide a single user identity for authentication and authorization to all resources, regardless of location (cloud or on-premises). For more information, see Hybrid identity documentation.
Identity governance	Manage your organization's identity through employee, business partner, vendor, service, and app access controls. You can also perform access reviews. For more information, see Azure AD identity governance documentation and Azure AD access reviews.
Identity protection	Detect potential vulnerabilities affecting your organization's identities, configure policies to respond to suspicious actions, and then take appropriate action to resolve them. For more information, see Azure AD Identity Protection.
Managed identities for Azure resources	Provides your Azure services with an automatically managed identity in Azure AD that can authenticate any Azure AD-supported authentication service, including Key Vault. For more information, see What is managed identities for Azure resources?.
Privileged identity management (PIM)	Manage, control, and monitor access within your organization. This feature includes access to resources in Azure AD and Azure, and other Microsoft Online Services, like Office 365 or Intune. For more information, see Azure AD Privileged Identity Management.
Reports and monitoring	Gain insights into the security and usage patterns in your environment. For more information, see Azure Active Directory reports and monitoring.

Terminology

To better understand Azure AD and its documentation, we recommend reviewing the following terms.

TERM OR CONCEPT	DESCRIPTION
Identity	A thing that can get authenticated. An identity can be a user with a username and password. Identities also include applications or other servers that might require authentication through secret keys or certificates.
Account	An identity that has data associated with it. You cannot have an account without an identity.
Azure AD account	An identity created through Azure AD or another Microsoft cloud service, such as Office 365. Identities are stored in Azure AD and accessible to your organization's cloud service subscriptions. This account is also sometimes called a Work or school account.
Account Administrator	This classic subscription administrator role is conceptually the billing owner of a subscription. This role has access to the Azure Account Center and enables you to manage all subscriptions in an account. For more information, see Classic subscription administrator roles, Azure Role-based access control (RBAC) roles, and Azure AD administrator roles.
Service Administrator	This classic subscription administrator role enables you to manage all Azure resources, including access. This role has the equivalent access of a user who is assigned the Owner role at the subscription scope. For more information, see Classic subscription administrator roles, Azure RBAC roles, and Azure AD administrator roles.
Owner	This role helps you manage all Azure resources, including access. This role is built on a newer authorization system called role-base access control (RBAC) that provides fine-grained access management to Azure resources. For more information, see Classic subscription administrator roles, Azure RBAC roles, and Azure AD administrator roles.
Azure AD Global administrator	This administrator role is automatically assigned to whomever created the Azure AD tenant. Global administrators can do all of the administrative functions for Azure AD and any services that federate to Azure AD, such as Exchange Online, SharePoint Online, and Skype for Business Online. You can have multiple Global administrators, but only Global administrators can assign administrator roles (including assigning other Global administrators) to users. Note that this administrator role is called Global administrator in the Azure portal, but it's called **Company administrator** in the Microsoft Graph API and Azure AD PowerShell.For more information about the various administrator roles, see Administrator role permissions in Azure Active Directory.
Azure subscription	Used to pay for Azure cloud services. You can have many subscriptions and they're linked to a credit card.

TERM OR CONCEPT	DESCRIPTION
Azure tenant	A dedicated and trusted instance of Azure AD that's automatically created when your organization signs up for a Microsoft cloud service subscription, such as Microsoft Azure, Microsoft Intune, or Office 365. An Azure tenant represents a single organization.
Single tenant	Azure tenants that access other services in a dedicated environment are considered single tenant.
Multi-tenant	Azure tenants that access other services in a shared environment, across multiple organizations, are considered multi-tenant.
Azure AD directory	Each Azure tenant has a dedicated and trusted Azure AD directory. The Azure AD directory includes the tenant's users, groups, and apps and is used to perform identity and access management functions for tenant resources.
Custom domain	Every new Azure AD directory comes with an initial domain name, domainname.onmicrosoft.com. In addition to that initial name, you can also add your organization's domain names, which include the names you use to do business and your users use to access your organization's resources, to the list. Adding custom domain names helps you to create user names that are familiar to your users, such as alain@contoso.com.
Microsoft account (also called, MSA)	Personal accounts that provide access to your consumer-oriented Microsoft products and cloud services, such as Outlook, OneDrive, Xbox LIVE, or Office 365. Your Microsoft account is created and stored in the Microsoft consumer identity account system that's run by Microsoft.

Next steps

- Sign up for Azure Active Directory Premium
- Associate an Azure subscription to your Azure Active Directory
- Azure Active Directory Premium P2 feature deployment checklist

What is the Azure Active Directory architecture?

2/14/2020 • 6 minutes to read • Edit Online

Azure Active Directory (Azure AD) enables you to securely manage access to Azure services and resources for your users. Included with Azure AD is a full suite of identity management capabilities. For information about Azure AD features, see What is Azure Active Directory?

With Azure AD, you can create and manage users and groups, and enable permissions to allow and deny access to enterprise resources. For information about identity management, see The fundamentals of Azure identity management.

Azure AD architecture

Azure AD's geographically distributed architecture combines extensive monitoring, automated rerouting, failover, and recovery capabilities, which deliver company-wide availability and performance to customers.

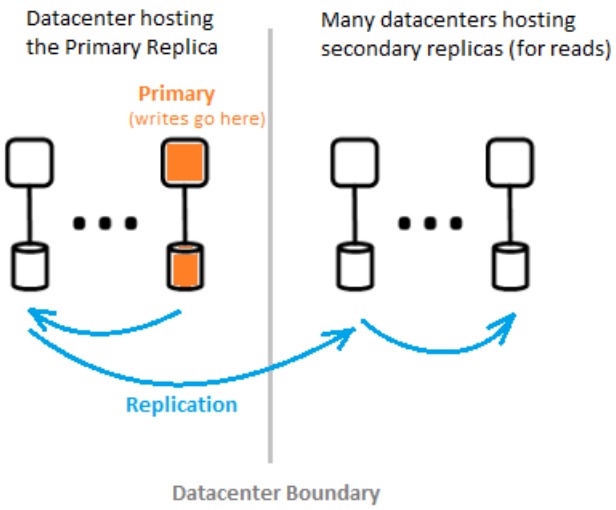
The following architecture elements are covered in this article:

- Service architecture design
- Scalability
- Continuous availability
- Datacenters

Service architecture design

The most common way to build an accessible and usable, data-rich system is through independent building blocks or scale units. For the Azure AD data tier, scale units are called *partitions*.

The data tier has several front-end services that provide read-write capability. The diagram below shows how the components of a single-directory partition are delivered throughout geographically distributed datacenters.



The components of Azure AD architecture include a primary replica and secondary replicas.

Primary replica

The *primary replica* receives all *writes* for the partition it belongs to. Any write operation is immediately replicated to a secondary replica in a different datacenter before returning success to the caller, thus ensuring geo-redundant durability of writes.

Secondary replicas

All directory *reads* are serviced from *secondary replicas*, which are at datacenters that are physically located across different geographies. There are many secondary replicas, as data is replicated asynchronously. Directory reads, such as authentication requests, are serviced from datacenters that are close to customers. The secondary replicas are responsible for read scalability.

Scalability

Scalability is the ability of a service to expand to meet increasing performance demands. Write scalability is achieved by partitioning the data. Read scalability is achieved by replicating data from one partition to multiple secondary replicas distributed throughout the world.

Requests from directory applications are routed to the datacenter that they are physically closest to. Writes are transparently redirected to the primary replica to provide read-write consistency. Secondary replicas significantly extend the scale of partitions because the directories are typically serving reads most of the time.

Directory applications connect to the nearest datacenters. This connection improves performance, and therefore scaling out is possible. Since a directory partition can have many secondary replicas, secondary replicas can be placed closer to the directory clients. Only internal directory service components that are write-intensive target the active primary replica directly.

Continuous availability

Availability (or uptime) defines the ability of a system to perform uninterrupted. The key to Azure AD's high-availability is that the services can quickly shift traffic across multiple geographically distributed datacenters. Each datacenter is independent, which enables de-correlated failure modes. Through this high availability design, Azure AD requires no downtime for maintenance activities.

Azure AD's partition design is simplified compared to the enterprise AD design, using a single-master design that includes a carefully orchestrated and deterministic primary replica failover process.

Fault tolerance

A system is more available if it is tolerant to hardware, network, and software failures. For each partition on the directory, a highly available master replica exists: The primary replica. Only writes to the partition are performed at this replica. This replica is being continuously and closely monitored, and writes can be immediately shifted to another replica (which becomes the new primary) if a failure is detected. During failover, there could be a loss of write availability typically of 1-2 minutes. Read availability is not affected during this time.

Read operations (which outnumber writes by many orders of magnitude) only go to secondary replicas. Since secondary replicas are idempotent, loss of any one replica in a given partition is easily compensated by directing the reads to another replica, usually in the same datacenter.

Data durability

A write is durably committed to at least two datacenters prior to it being acknowledged. This happens by first committing the write on the primary, and then immediately replicating the write to at least one other datacenter. This write action ensures that a potential catastrophic loss of the datacenter hosting the primary does not result in data loss.

Azure AD maintains a zero Recovery Time Objective (RTO) to not lose data on failovers. This includes:

- Token issuance and directory reads
- Allowing only about 5 minutes RTO for directory writes

Datacenters

Azure AD's replicas are stored in datacenters located throughout the world. For more information, see Azure global infrastructure.

Azure AD operates across datacenters with the following characteristics:

- Authentication, Graph, and other AD services reside behind the Gateway service. The Gateway manages load balancing of these services. It will fail over automatically if any unhealthy servers are detected using transactional health probes. Based on these health probes, the Gateway dynamically routes traffic to healthy datacenters.
- For *reads*, the directory has secondary replicas and corresponding front-end services in an active-active configuration operating in multiple datacenters. In case of a failure of an entire datacenter, traffic will be automatically routed to a different datacenter. *For *writes*, the directory will fail over primary (master) replica across datacenters via planned (new primary is synchronized to old primary) or emergency failover procedures. Data durability is achieved by replicating any commit to at least two datacenters.

Data consistency

The directory model is one of eventual consistencies. One typical problem with distributed asynchronously replicating systems is that the data returned from a "particular" replica may not be up-to-date.

Azure AD provides read-write consistency for applications targeting a secondary replica by routing its writes to the primary replica, and synchronously pulling the writes back to the secondary replica.

Application writes using the Microsoft Graph API of Azure AD are abstracted from maintaining affinity to a directory replica for read-write consistency. The Microsoft Graph API service maintains a logical session, which has affinity to a secondary replica used for reads; affinity is captured in a "replica token" that the service caches using a distributed cache in the secondary replica datacenter. This token is then used for subsequent operations in the same logical session. To continue using the same logical session, subsequent requests must be routed to the same Azure AD datacenter. It is not possible to continue a logical session if the directory client requests are being routed to multiple Azure AD datacenters; if this happens then the client has multiple logical sessions which have independent read-write consistencies.

> **NOTE**
> Writes are immediately replicated to the secondary replica to which the logical session's reads were issued.

Backup protection

The directory implements soft deletes, instead of hard deletes, for users and tenants for easy recovery in case of accidental deletes by a customer. If your tenant administrator accidental deletes users, they can easily undo and restore the deleted users.

Azure AD implements daily backups of all data, and therefore can authoritatively restore data in case of any logical deletions or corruptions. The data tier employs error correcting codes, so that it can check for errors and automatically correct particular types of disk errors.

Metrics and monitors

Running a high availability service requires world-class metrics and monitoring capabilities. Azure AD continually analyzes and reports key service health metrics and success criteria for each of its services. There is also continuous development and tuning of metrics and monitoring and alerting for each scenario, within each Azure AD service and across all services.

If any Azure AD service is not working as expected, action is immediately taken to restore functionality as quickly as possible. The most important metric Azure AD tracks is how quickly live site issues can be detected and mitigated for customers. We invest heavily in monitoring and alerts to minimize time to detect (TTD Target: <5 minutes) and operational readiness to minimize time to mitigate (TTM Target: <30 minutes).

Secure operations

Using operational controls such as multi-factor authentication (MFA) for any operation, as well as auditing of all operations. In addition, using a just-in-time elevation system to grant necessary temporary access for any operational task-on-demand on an ongoing basis. For more information, see The Trusted Cloud.

Next steps

Azure Active Directory developer's guide

Configure scoped synchronization from Azure AD to Azure Active Directory Domain Services

7/20/2020 • 9 minutes to read • Edit Online

To provide authentication services, Azure Active Directory Domain Services (Azure AD DS) synchronizes users and groups from Azure AD. In a hybrid environment, users and groups from an on-premises Active Directory Domain Services (AD DS) environment can be first synchronized to Azure AD using Azure AD Connect, and then synchronized to Azure AD DS.

By default, all users and groups from an Azure AD directory are synchronized to an Azure AD DS managed domain. If you have specific needs, you can instead choose to synchronize only a defined set of users.

This article shows you how to create a managed domain that uses scoped synchronization and then change or disable the set of scoped users.

Scoped synchronization overview

By default, all users and groups from an Azure AD directory are synchronized to a managed domain. If only a few users need to access the managed domain, you can synchronize only those user accounts. This scoped synchronization is group-based. When you configure group-based scoped synchronization, only the user accounts that belong to the groups you specify are synchronized to the managed domain.

The following table outlines how to use scoped synchronization:

CURRENT STATE	DESIRED STATE	REQUIRED CONFIGURATION
An existing managed domain is configured to synchronize all user accounts and groups.	You want to synchronize only user accounts that belong to specific groups.	You can't change from synchronizing all users to using scoped synchronization. Delete the existing managed domain, then follow the steps in this article to re-create a managed domain with scoped synchronization configured.
No existing managed domain.	You want to create a new managed domain and synchronize only user accounts belonging to specific groups.	Follow the steps in this article to create a managed domain with scoped synchronization configured.
An existing managed domain is configured to synchronize only accounts that belong to specific groups.	You want to modify the list of groups whose users should be synchronized to the managed domain.	Follow the steps in this article to modify scoped synchronization.

You use the Azure portal or PowerShell to configure the scoped synchronization settings:

ACTION	USE AZURE PORTAL	USE POWERSHELL
Create a managed domain and configure scoped synchronization	Azure portal	PowerShell
Modify scoped synchronization	Azure portal	PowerShell
Disable scoped synchronization	Azure portal	PowerShell

ACTION	USE AZURE PORTAL	USE POWERSHELL

WARNING

Changing the scope of synchronization causes the managed domain to resynchronize all data. The following considerations apply:

- When you change the synchronization scope for a managed domain, a full resynchronization occurs.
- Objects that are no longer required in the managed domain are deleted. New objects are created in the managed domain.
- Resynchronization may take a long time to complete. The synchronization time depends on the number of objects such as users, groups, and group memberships in the managed domain and Azure AD directory. For large directories with many hundreds of thousands of objects, resynchronization may take a few days.

Enable scoped synchronization using the Azure portal

To enable scoped synchronization in the Azure portal, complete the the following steps:

1. Follow the tutorial to create and configure a managed domain. Complete all prerequisites and deployment steps other than for synchronization scope.
2. Choose **Scoped** at the synchronization step, then select the Azure AD groups to synchronize to the managed domain.

The managed domain can take up to an hour to complete the deployment. In the Azure portal, the **Overview** page for your managed domain shows the current status throughout this deployment stage.

When the Azure portal shows that the managed domain has finished provisioning, the following tasks need to be completed:

- Update DNS settings for the virtual network so virtual machines can find the managed domain for domain join or authentication.
  - To configure DNS, select your managed domain in the portal. On the **Overview** window, you are prompted to automatically configure these DNS settings.
- Enable password synchronization to Azure AD Domain Services so end users can sign in to the managed domain using their corporate credentials.

Modify scoped synchronization using the Azure portal

To modify the list of groups whose users should be synchronized to the managed domain, complete the following steps:

1. In the Azure portal, search for and select **Azure AD Domain Services**. Choose your managed domain, such as *aaddscontoso.com*.
2. Select **Synchronization** from the menu on the left-hand side.
3. To add a group, choose **+ Select groups** at the top, then choose the groups to add.
4. To remove a group from the synchronization scope, select it from the list of currently synchronized groups and choose **Remove groups**.
5. When all changes are made, select **Save synchronization scope**.

Changing the scope of synchronization causes the managed domain to resynchronize all data. Objects that are no longer required in the managed domain are deleted, and resynchronization may take a long time to complete.

Disable scoped synchronization using the Azure portal

To disable group-based scoped synchronization for a managed domain, complete the following steps:

1. In the Azure portal, search for and select **Azure AD Domain Services**. Choose your managed domain, such as *aaddscontoso.com*.
2. Select **Synchronization** from the menu on the left-hand side.
3. Set the synchronization scope from **Scoped** to **All**, then select **Save synchronization scope**.

Changing the scope of synchronization causes the managed domain to resynchronize all data. Objects that are no longer required in the managed domain are deleted, and resynchronization may take a long time to complete.

PowerShell script for scoped synchronization

To configure scoped synchronization using PowerShell, first save the following script to a file named `Select-GroupsToSync.ps1`. This script configures Azure AD DS to synchronize selected groups from Azure AD. All user accounts that are part of the specified groups are synchronized to the managed domain.

This script is used in the additional steps in this article.

```
param (
    [Parameter(Position = 0)]
    [String[]]$groupsToAdd
)

Connect-AzureAD
$sp = Get-AzureADServicePrincipal -Filter "AppId eq '2565bd9d-da50-47d4-8b85-4c97f669dc36'"
$role = $sp.AppRoles | where-object -FilterScript {$_.DisplayName -eq "User"}

Write-Output "`n*****************************************************************************"

Write-Output "Total group-assignments need to be added: $($groupsToAdd.Count)"
$newGroupIds = New-Object 'System.Collections.Generic.HashSet[string]'
foreach ($groupName in $groupsToAdd)
{
    try
    {
        $group = Get-AzureADGroup -Filter "DisplayName eq '$groupName'"
        $newGroupIds.Add($group.ObjectId)

        Write-Output "Group-Name: $groupName, Id: $($group.ObjectId)"
    }
    catch
    {
        Write-Error "Failed to find group: $groupName. Exception: $($_.Exception)."
    }
}

Write-Output "*****************************************************************************`n"
Write-Output "`n*****************************************************************************"

$currentAssignments = Get-AzureADServiceAppRoleAssignment -ObjectId $sp.ObjectId
Write-Output "Total current group-assignments: $($currentAssignments.Count), SP-ObjectId: $($sp.ObjectId)"

$currAssignedObjectIds = New-Object 'System.Collections.Generic.HashSet[string]'
foreach ($assignment in $currentAssignments)
{
    Write-Output "Assignment-ObjectId: $($assignment.PrincipalId)"

    if ($newGroupIds.Contains($assignment.PrincipalId) -eq $false)
    {
        Write-Output "This assignment is not needed anymore. Removing it! Assignment-ObjectId:
$($assignment.PrincipalId)"
        Remove-AzureADServiceAppRoleAssignment -ObjectId $sp.ObjectId -AppRoleAssignmentId
```

```
$assignment.ObjectId
    }
    else
    {
        $currAssignedObjectIds.Add($assignment.PrincipalId)
    }
}

Write-Output "********************************************************************************`n"
Write-Output "`n********************************************************************************"

foreach ($id in $newGroupIds)
{
    try
    {
        if ($currAssignedObjectIds.Contains($id) -eq $false)
        {
            Write-Output "Adding new group-assignment. Role-Id: $($role.Id), Group-Object-Id: $id, ResourceId:
$($sp.ObjectId)"
            New-AzureADGroupAppRoleAssignment -Id $role.Id -ObjectId $id -PrincipalId $id -ResourceId
$sp.ObjectId
        }
        else
        {
            Write-Output "Group-ObjectId: $id is already assigned."
        }
    }
    catch
    {
        Write-Error "Exception occurred assigning Object-ID: $id. Exception: $($_.Exception)."
    }
}

Write-Output "********************************************************************************`n"
```

Enable scoped synchronization using PowerShell

Use PowerShell to complete the following set of steps. Refer to the instructions to enable Azure Active Directory Domain Services using PowerShell. A couple of steps in this article are modified slightly to configure scoped synchronization.

1. Complete the following tasks from the article to enable Azure AD DS using PowerShell. Stop at the step to actually create the managed domain. You configure the scoped synchronization you create the managed domain.
    - Install the required PowerShell modules.
    - Create the required service principal and Azure AD group for administrative access.
    - Create supporting Azure resources like a virtual network and subnets.
2. Determine the groups and users they contain that you want to synchronize from Azure AD. Make a list of the display names of the groups to synchronize to Azure AD DS.
3. Run the script from the previous section and use the *-groupsToAdd* parameter to pass the list of groups to synchronize.

> **WARNING**
> You must include the *AAD DC Administrators* group in the list of groups for scoped synchronization. If you don't include this group, the managed domain is unusable.

```
.\Select-GroupsToSync.ps1 -groupsToAdd @("AAD DC Administrators", "GroupName1", "GroupName2")
```

4. Now create the managed domain and enable group-based scoped synchronization. Include *"filteredSync" = "Enabled"* in the *-Properties* parameter.

   Set your Azure subscription ID, and then provide a name for the managed domain, such as *aaddscontoso.com*. You can get your subscription ID using the Get-AzSubscription cmdlet. Set the resource group name, virtual network name, and region to the values used in the previous steps to create the supporting Azure resources:

```
$AzureSubscriptionId = "YOUR_AZURE_SUBSCRIPTION_ID"
$ManagedDomainName = "aaddscontoso.com"
$ResourceGroupName = "myResourceGroup"
$VnetName = "myVnet"
$AzureLocation = "westus"

# Enable Azure AD Domain Services for the directory.
New-AzResource -ResourceId
"/subscriptions/$AzureSubscriptionId/resourceGroups/$ResourceGroupName/providers/Microsoft.AAD/DomainSe
rvices/$ManagedDomainName" `
-Location $AzureLocation `
-Properties @{"DomainName"=$ManagedDomainName; "filteredSync" = "Enabled"; `
 "SubnetId"="/subscriptions/$AzureSubscriptionId/resourceGroups/$ResourceGroupName/providers/Microsoft.
Network/virtualNetworks/$VnetName/subnets/DomainServices"} `
-Force -Verbose
```

It takes a few minutes to create the resource and return control to the PowerShell prompt. The managed domain continues to be provisioned in the background, and can take up to an hour to complete the deployment. In the Azure portal, the **Overview** page for your managed domain shows the current status throughout this deployment stage.

When the Azure portal shows that the managed domain has finished provisioning, the following tasks need to be completed:

- Update DNS settings for the virtual network so virtual machines can find the managed domain for domain join or authentication.
  - To configure DNS, select your managed domain in the portal. On the **Overview** window, you are prompted to automatically configure these DNS settings.
- If you created a managed domain in a region that supports Availability Zones, create a network security group to restrict traffic in the virtual network for the managed domain. An Azure standard load balancer is created that requires these rules to be place. This network security group secures Azure AD DS and is required for the managed domain to work correctly.
  - To create the network security group and required rules, select your managed domain in the portal. On the **Overview** window, you are prompted to automatically create and configure the network security group.
- Enable password synchronization to Azure AD Domain Services so end users can sign in to the managed domain using their corporate credentials.

Modify scoped synchronization using PowerShell

To modify the list of groups whose users should be synchronized to the managed domain, re-run the PowerShell script and specify the new list of groups. In the following example, the groups to synchronize no longer includes *GroupName2*, and now includes *GroupName3*.

WARNING

You must include the *AAD DC Administrators* group in the list of groups for scoped synchronization. If you don't include this group, the managed domain is unusable.

```
.\Select-GroupsToSync.ps1 -groupsToAdd @("AAD DC Administrators", "GroupName1", "GroupName3")
```

Changing the scope of synchronization causes the managed domain to resynchronize all data. Objects that are no longer required in the managed domain are deleted, and resynchronization may take a long time to complete.

Disable scoped synchronization using PowerShell

To disable group-based scoped synchronization for a managed domain, set *"filteredSync" = "Disabled"* on the Azure AD DS resource, then update the managed domain. When complete, all users and groups are set to synchronize from Azure AD.

```
// Retrieve the Azure AD DS resource.
$DomainServicesResource = Get-AzResource -ResourceType "Microsoft.AAD/DomainServices"

// Disable group-based scoped synchronization.
$disableScopedSync = @{"filteredSync" = "Disabled"}

// Update the Azure AD DS resource
Set-AzResource -Id $DomainServicesResource.ResourceId -Properties $disableScopedSync
```

Changing the scope of synchronization causes the managed domain to resynchronize all data. Objects that are no longer required in the managed domain are deleted, and resynchronization may take a long time to complete.

Next steps

To learn more about the synchronization process, see Understand synchronization in Azure AD Domain Services.

Create an Organizational Unit (OU) in an Azure Active Directory Domain Services managed domain

7/20/2020 • 4 minutes to read • Edit Online

Organizational units (OUs) in an Active Directory Domain Services (AD DS) managed domain let you logically group objects such as user accounts, service accounts, or computer accounts. You can then assign administrators to specific OUs, and apply group policy to enforce targeted configuration settings.

Azure AD DS managed domains include the following two built-in OUs:

- *AADDC Computers* - contains computer objects for all computers that are joined to the managed domain.
- *AADDC Users* - includes users and groups synchronized in from the Azure AD tenant.

As you create and run workloads that use Azure AD DS, you may need to create service accounts for applications to authenticate themselves. To organize these service accounts, you often create a custom OU in the managed domain and then create service accounts within that OU.

In a hybrid environment, OUs created in an on-premises AD DS environment aren't synchronized to the managed domain. Managed domains use a flat OU structure. All user accounts and groups are stored in the *AADDC Users* container, despite being synchronized from different on-premises domains or forests, even if you've configured a hierarchical OU structure there.

This article shows you how to create an OU in your managed domain.

Before you begin

To complete this article, you need the following resources and privileges:

- An active Azure subscription.
  - If you don't have an Azure subscription, create an account.
- An Azure Active Directory tenant associated with your subscription, either synchronized with an on-premises directory or a cloud-only directory.
  - If needed, create an Azure Active Directory tenant or associate an Azure subscription with your account.
- An Azure Active Directory Domain Services managed domain enabled and configured in your Azure AD tenant.
  - If needed, complete the tutorial to create and configure an Azure Active Directory Domain Services managed domain.
- A Windows Server management VM that is joined to the Azure AD DS managed domain.
  - If needed, complete the tutorial to create a management VM.
- A user account that's a member of the *Azure AD DC administrators* group in your Azure AD tenant.

Custom OU considerations and limitations

When you create custom OUs in a managed domain, you gain additional management flexibility for user management and applying group policy. Compared to an on-premises AD DS environment, there are some limitations and considerations when creating and managing a custom OU structure in a managed domain:

- To create custom OUs, users must be a member of the *AAD DC Administrators* group.
- A user that creates a custom OU is granted administrative privileges (full control) over that OU and is the resource owner.

- By default, the *AAD DC Administrators* group also has full control of the custom OU.

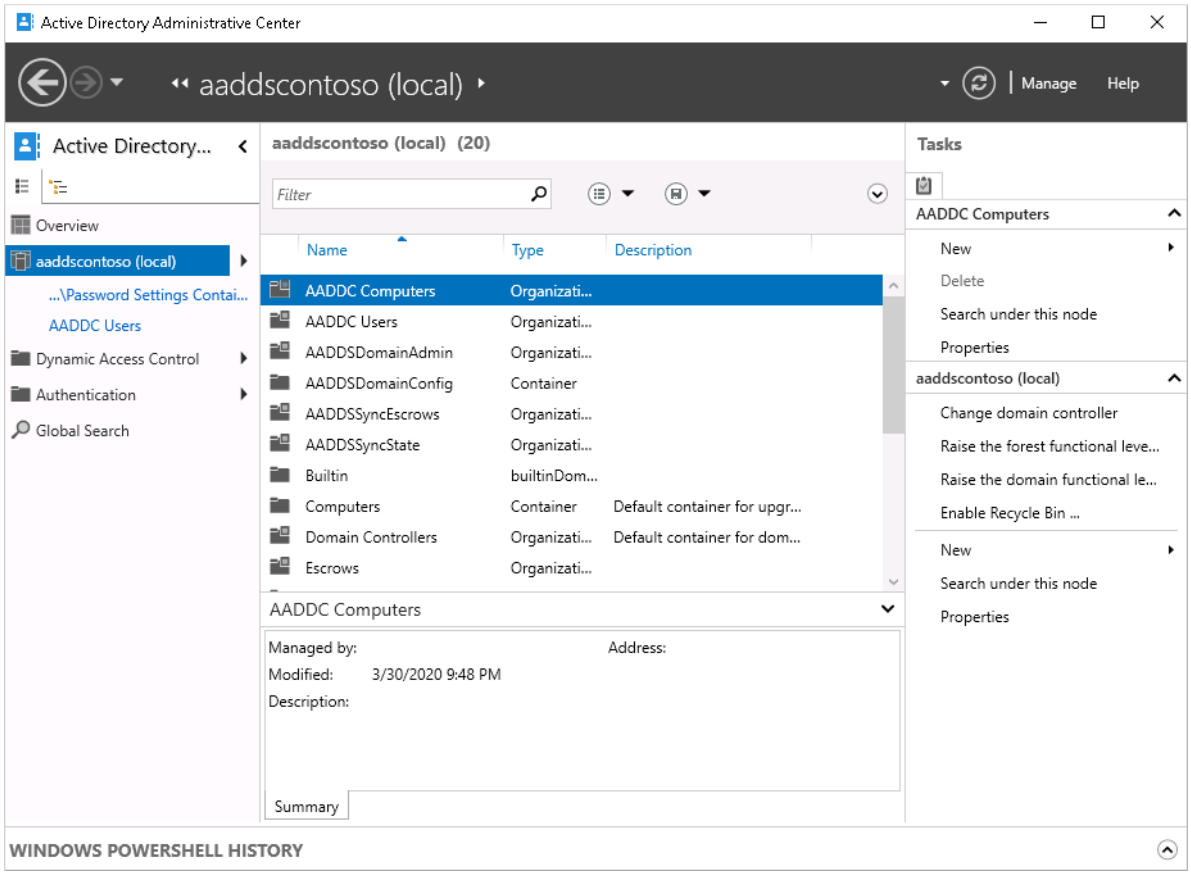
- A default OU for *AADDC Users* is created that contains all the synchronized user accounts from your Azure AD tenant.
  - You can't move users or groups from the *AADDC Users* OU to custom OUs that you create. Only user accounts or resources created in the managed domain can be moved into custom OUs.
- User accounts, groups, service accounts, and computer objects that you create under custom OUs aren't available in your Azure AD tenant.
  - These objects don't show up using the Microsoft Graph API or in the Azure AD UI; they're only available in your managed domain.

Create a custom OU

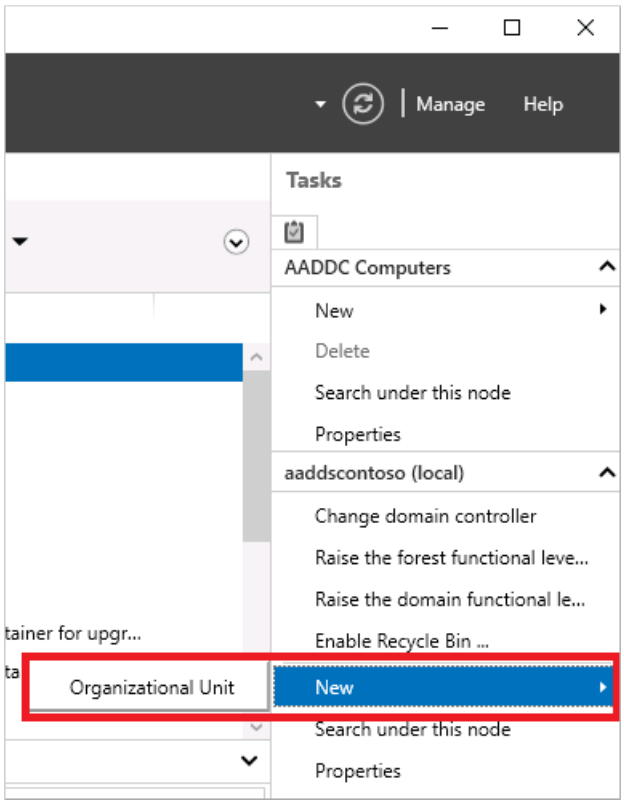
To create a custom OU, you use the Active Directory Administrative Tools from a domain-joined VM. The Active Directory Administrative Center lets you view, edit, and create resources in a managed domain, including OUs.

> **NOTE**
> To create a custom OU in a managed domain, you must be signed in to a user account that's a member of the *AAD DC Administrators* group.

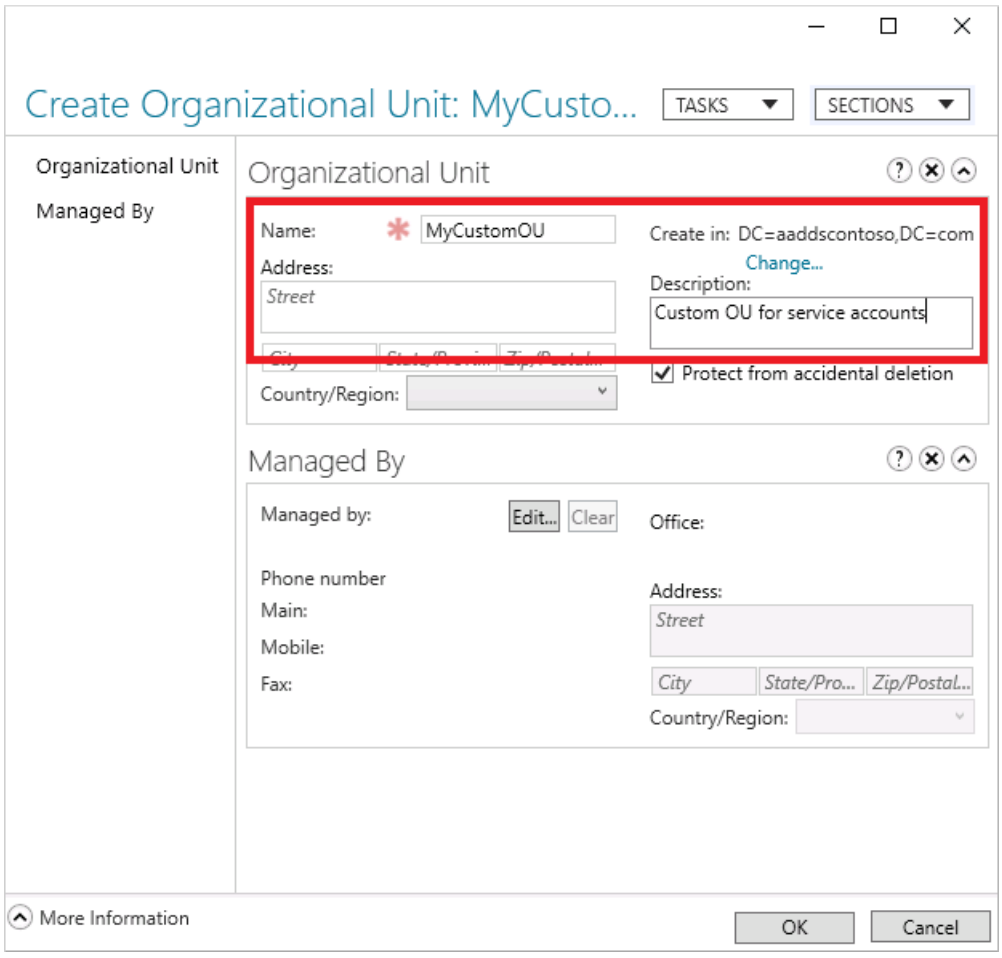
1. Sign in to your management VM. For steps on how to connect using the Azure portal, see Connect to a Windows Server VM.
2. From the Start screen, select **Administrative Tools**. A list of available management tools is shown that were installed in the tutorial to create a management VM.
3. To create and manage OUs, select **Active Directory Administrative Center** from the list of administrative tools.
4. In the left pane, choose your managed domain, such as *aaddscontoso.com*. A list of existing OUs and resources is shown:



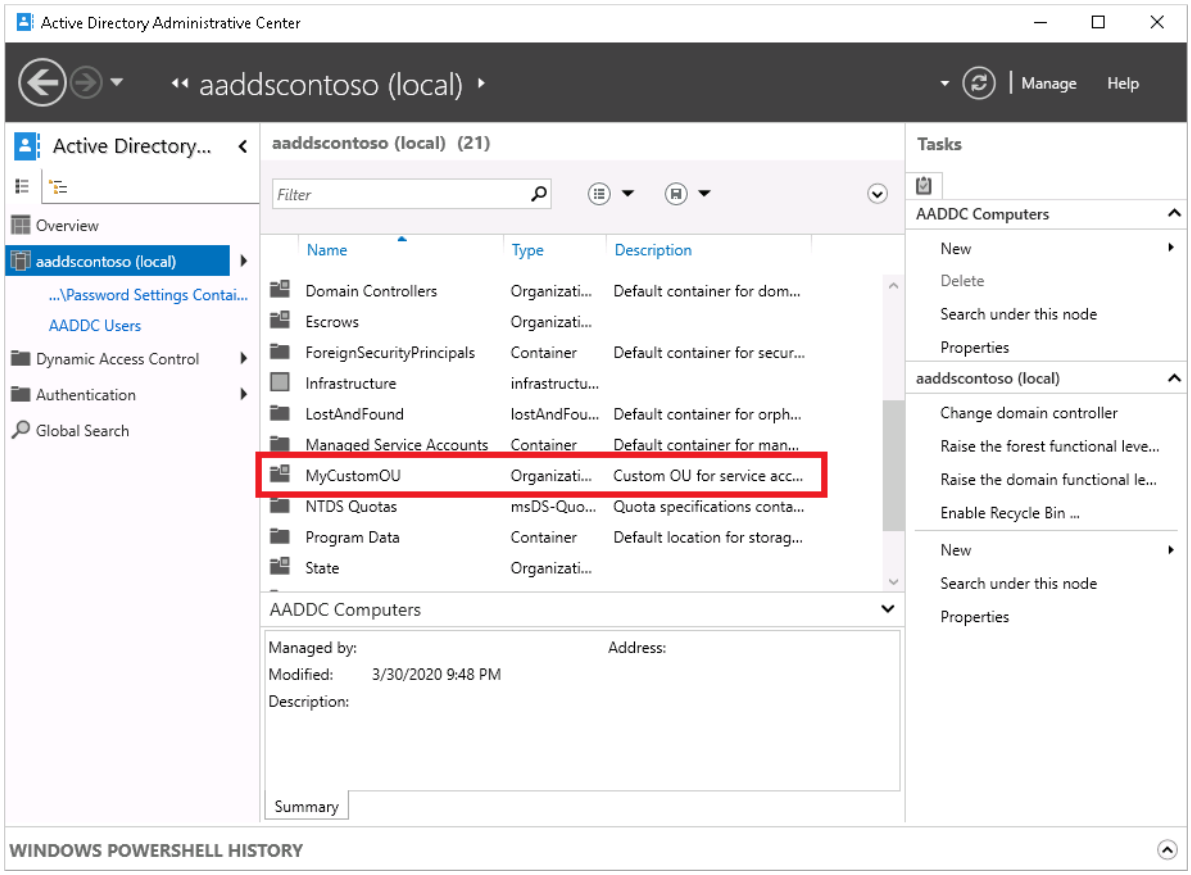
5. The **Tasks** pane is shown on the right side of the Active Directory Administrative Center. Under the domain, such as *aaddscontoso.com*, select **New** > **Organizational Unit**.



6. In the **Create Organizational Unit** dialog, specify a **Name** for the new OU, such as *MyCustomOu*. Provide a short description for the OU, such as *Custom OU for service accounts*. If desired, you can also set the **Managed By** field for the OU. To create the custom OU, select **OK**.



7. Back in the Active Directory Administrative Center, the custom OU is now listed and is available for use:



Next steps

For more information on using the administrative tools or creating and using service accounts, see the following articles:

- Active Directory Administrative Center: Getting Started
- Service Accounts Step-by-Step Guide

Create a group managed service account (gMSA) in Azure Active Directory Domain Services

7/20/2020 • 3 minutes to read • Edit Online

Applications and services often need an identity to authenticate themselves with other resources. For example, a web service may need to authenticate with a database service. If an application or service has multiple instances, such as a web server farm, manually creating and configuring the identities for those resources gets time consuming.

Instead, a group managed service account (gMSA) can be created in the Azure Active Directory Domain Services (Azure AD DS) managed domain. The Windows OS automatically manages the credentials for a gMSA, which simplifies the management of large groups of resources.

This article shows you how to create a gMSA in a managed domain using Azure PowerShell.

Before you begin

To complete this article, you need the following resources and privileges:

- An active Azure subscription.
  - If you don't have an Azure subscription, create an account.
- An Azure Active Directory tenant associated with your subscription, either synchronized with an on-premises directory or a cloud-only directory.
  - If needed, create an Azure Active Directory tenant or associate an Azure subscription with your account.
- An Azure Active Directory Domain Services managed domain enabled and configured in your Azure AD tenant.
  - If needed, complete the tutorial to create and configure an Azure Active Directory Domain Services managed domain.
- A Windows Server management VM that is joined to the Azure AD DS managed domain.
  - If needed, complete the tutorial to create a management VM.

Managed service accounts overview

A standalone managed service account (sMSA) is a domain account whose password is automatically managed. This approach simplifies service principal name (SPN) management, and enables delegated management to other administrators. You don't need to manually create and rotate credentials for the account.

A group managed service account (gMSA) provides the same management simplification, but for multiple servers in the domain. A gMSA lets all instances of a service hosted on a server farm use the same service principal for mutual authentication protocols to work. When a gMSA is used as service principal, the Windows operating system again manages the account's password instead of relying on the administrator.

For more information, see group managed service accounts (gMSA) overview.

Using service accounts in Azure AD DS

As managed domains are locked down and managed by Microsoft, there are some considerations when using service accounts:

- Create service accounts in custom organizational units (OU) on the managed domain.
  - You can't create a service account in the built-in *AADDC Users* or *AADDC Computers* OUs.

- Instead, create a custom OU in the managed domain and then create service accounts in that custom OU.
- The Key Distribution Services (KDS) root key is pre-created.
  - The KDS root key is used to generate and retrieve passwords for gMSAs. In Azure AD DS, the KDS root is created for you.
  - You don't have privileges to create another, or view the default, KDS root key.

Create a gMSA

First, create a custom OU using the New-ADOrganizationalUnit cmdlet. For more information on creating and managing custom OUs, see Custom OUs in Azure AD DS.

> **TIP**
> To complete these steps to create a gMSA, use your management VM. This management VM should already have the required AD PowerShell cmdlets and connection to the managed domain.

The following example creates a custom OU named *myNewOU* in the managed domain named *aaddscontoso.com*. Use your own OU and managed domain name:

```
New-ADOrganizationalUnit -Name "myNewOU" -Path "DC=aaddscontoso,DC=COM"
```

Now create a gMSA using the New-ADServiceAccount cmdlet. The following example parameters are defined:

- **-Name** is set to *WebFarmSvc*
- **-Path** parameter specifies the custom OU for the gMSA created in the previous step.
- DNS entries and service principal names are set for *WebFarmSvc.aaddscontoso.com*
- Principals in *AADDSCONTOSO-SERVER$* are allowed to retrieve the password use the identity.

Specify your own names and domain names.

```
New-ADServiceAccount -Name WebFarmSvc `
    -DNSHostName WebFarmSvc.aaddscontoso.com `
    -Path "OU=MYNEWOU,DC=aaddscontoso,DC=com" `
    -KerberosEncryptionType AES128, AES256 `
    -ManagedPasswordIntervalInDays 30 `
    -ServicePrincipalNames http/WebFarmSvc.aaddscontoso.com/aaddscontoso.com, `
        http/WebFarmSvc.aaddscontoso.com/aaddscontoso, `
        http/WebFarmSvc/aaddscontoso.com, `
        http/WebFarmSvc/aaddscontoso `
    -PrincipalsAllowedToRetrieveManagedPassword AADDSCONTOSO-SERVER$
```

Applications and services can now be configured to use the gMSA as needed.

Next steps

For more information about gMSAs, see Getting started with group managed service accounts.

Virtual network design considerations and configuration options for Azure Active Directory Domain Services

7/20/2020 • 10 minutes to read • Edit Online

Azure Active Directory Domain Services (Azure AD DS) provides authentication and management services to other applications and workloads. Network connectivity is a key component. Without correctly configured virtual network resources, applications and workloads can't communicate with and use the features provided by Azure AD DS. Plan your virtual network requirements to make sure that Azure AD DS can serve your applications and workloads as needed.

This article outlines design considerations and requirements for an Azure virtual network to support Azure AD DS.

Azure virtual network design

To provide network connectivity and allow applications and services to authenticate against an Azure AD DS managed domain, you use an Azure virtual network and subnet. Ideally, the managed domain should be deployed into its own virtual network.

You can include a separate application subnet in the same virtual network to host your management VM or light application workloads. A separate virtual network for larger or complex application workloads, peered to the Azure AD DS virtual network, is usually the most appropriate design.

Other designs choices are valid, provided you meet the requirements outlined in the following sections for the virtual network and subnet.

As you design the virtual network for Azure AD DS, the following considerations apply:

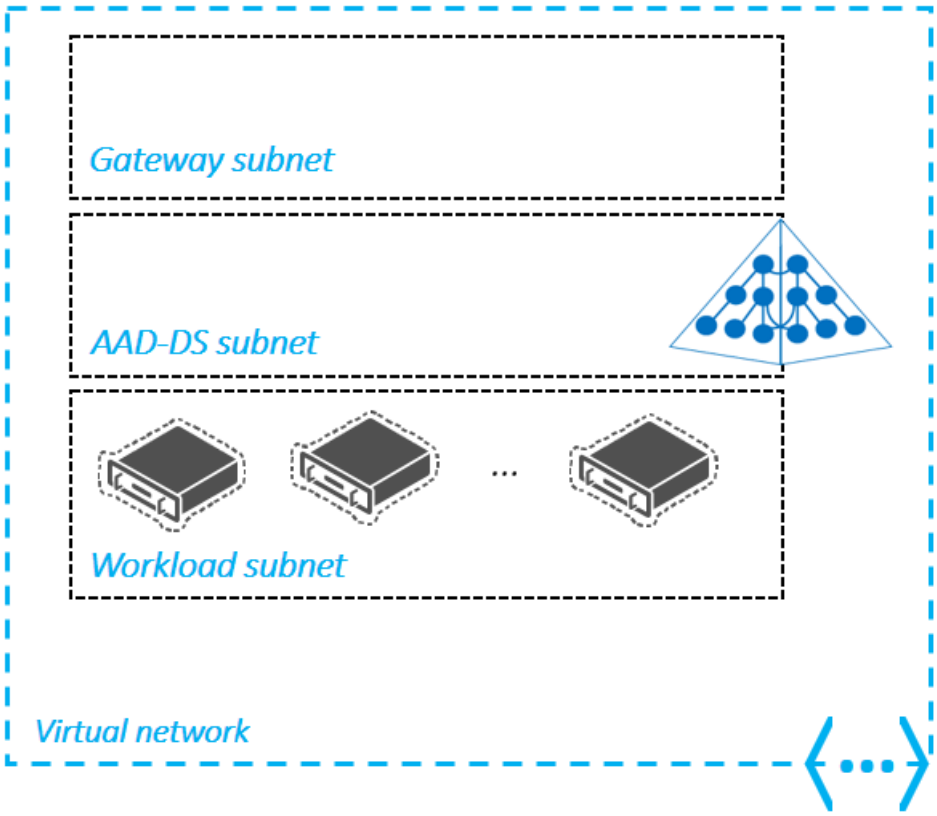
- Azure AD DS must be deployed into the same Azure region as your virtual network.
  - At this time, you can only deploy one managed domain per Azure AD tenant. The managed domain is deployed to single region. Make sure that you create or select a virtual network in a region that supports Azure AD DS.
- Consider the proximity of other Azure regions and the virtual networks that host your application workloads.
  - To minimize latency, keep your core applications close to, or in the same region as, the virtual network subnet for your managed domain. You can use virtual network peering or virtual private network (VPN) connections between Azure virtual networks. These connection options are discussed in a following section.
- The virtual network can't rely on DNS services other than those services provided by the managed domain.
  - Azure AD DS provides its own DNS service. The virtual network must be configured to use these DNS service addresses. Name resolution for additional namespaces can be accomplished using conditional forwarders.
  - You can't use custom DNS server settings to direct queries from other DNS servers, including on VMs. Resources in the virtual network must use the DNS service provided by the managed domain.

> **IMPORTANT**
> You can't move Azure AD DS to a different virtual network after you've enabled the service.

A managed domain connects to a subnet in an Azure virtual network. Design this subnet for Azure AD DS with the following considerations:

- A managed domain must be deployed in its own subnet. Don't use an existing subnet or a gateway subnet.
- A network security group is created during the deployment of a managed domain. This network security group contains the required rules for correct service communication.
  - Don't create or use an existing network security group with your own custom rules.
- A managed domain requires 3-5 IP addresses. Make sure that your subnet IP address range can provide this number of addresses.
  - Restricting the available IP addresses can prevent the managed domain from maintaining two domain controllers.

The following example diagram outlines a valid design where the managed domain has its own subnet, there's a gateway subnet for external connectivity, and application workloads are in a connected subnet within the virtual network:



Connections to the Azure AD DS virtual network

As noted in the previous section, you can only create a managed domain in a single virtual network in Azure, and only one managed domain can be created per Azure AD tenant. Based on this architecture, you may need to connect one or more virtual networks that host your application workloads to your managed domain's virtual network.

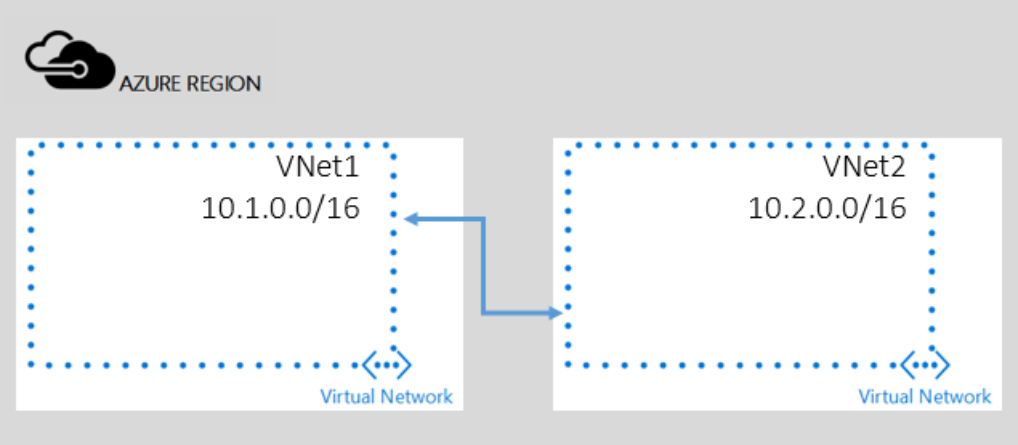
You can connect application workloads hosted in other Azure virtual networks using one of the following methods:

- Virtual network peering
- Virtual private networking (VPN)

Virtual network peering

Virtual network peering is a mechanism that connects two virtual networks in the same region through the Azure backbone network. Global virtual network peering can connect virtual network across Azure regions. Once peered, the two virtual networks let resources, such as VMs, communicate with each other directly using private IP

addresses. Using virtual network peering lets you deploy a managed domain with your application workloads deployed in other virtual networks.



For more information, see Azure virtual network peering overview.

Virtual Private Networking (VPN)

You can connect a virtual network to another virtual network (VNet-to-VNet) in the same way that you can configure a virtual network to an on-premises site location. Both connections use a VPN gateway to create a secure tunnel using IPsec/IKE. This connection model lets you deploy the managed domain into an Azure virtual network and then connect on-premises locations or other clouds.

For more information on using virtual private networking, read Configure a VNet-to-VNet VPN gateway connection by using the Azure portal.

Name resolution when connecting virtual networks

Virtual networks connected to the managed domain's virtual network typically have their own DNS settings. When you connect virtual networks, it doesn't automatically configure name resolution for the connecting virtual network to resolve services provided by the managed domain. Name resolution on the connecting virtual networks must be configured to enable application workloads to locate the managed domain.

You can enable name resolution using conditional DNS forwarders on the DNS server supporting the connecting virtual networks, or by using the same DNS IP addresses from the managed domain's virtual network.

Network resources used by Azure AD DS

A managed domain creates some networking resources during deployment. These resources are needed for successful operation and management of the managed domain, and shouldn't be manually configured.

AZURE RESOURCE	DESCRIPTION
Network interface card	Azure AD DS hosts the managed domain on two domain controllers (DCs) that run on Windows Server as Azure VMs. Each VM has a virtual network interface that connects to your virtual network subnet.

AZURE RESOURCE	DESCRIPTION
Dynamic standard public IP address	Azure AD DS communicates with the synchronization and management service using a standard SKU public IP address. For more information about public IP addresses, see IP address types and allocation methods in Azure.
Azure standard load balancer	Azure AD DS uses a standard SKU load balancer for network address translation (NAT) and load balancing (when used with secure LDAP). For more information about Azure load balancers, see What is Azure Load Balancer?
Network address translation (NAT) rules	Azure AD DS creates and uses three NAT rules on the load balancer - one rule for secure HTTP traffic, and two rules for secure PowerShell remoting.
Load balancer rules	When a managed domain is configured for secure LDAP on TCP port 636, three rules are created and used on a load balancer to distribute the traffic.

WARNING

Don't delete or modify any of the network resource created by Azure AD DS, such as manually configuring the load balancer or rules. If you delete or modify any of the network resources, an Azure AD DS service outage may occur.

Network security groups and required ports

A network security group (NSG) contains a list of rules that allow or deny network traffic to traffic in an Azure virtual network. A network security group is created when you deploy a managed domain that contains a set of rules that let the service provide authentication and management functions. This default network security group is associated with the virtual network subnet your managed domain is deployed into.

The following network security group rules are required for the managed domain to provide authentication and management services. Don't edit or delete these network security group rules for the virtual network subnet your managed domain is deployed into.

PORT NUMBER	PROTOCOL	SOURCE	DESTINATION	ACTION	REQUIRED	PURPOSE
443	TCP	AzureActiveDirectoryDomainServices	Any	Allow	Yes	Synchronization with your Azure AD tenant.
3389	TCP	CorpNetSaw	Any	Allow	Yes	Management of your domain.
5986	TCP	AzureActiveDirectoryDomainServices	Any	Allow	Yes	Management of your domain.

> **WARNING**
>
> Don't manually edit these network resources and configurations. When you associate a misconfigured network security group or a user defined route table with the subnet in which the managed domain is deployed, you may disrupt Microsoft's ability to service and manage the domain. Synchronization between your Azure AD tenant and your managed domain is also disrupted.
>
> If you use secure LDAP, you can add the required TCP port 636 rule to allow external traffic if needed. Adding this rule doesn't place your network security group rules in an unsupported state. For more information, see Lock down secure LDAP access over the internet
>
> Default rules for *AllowVnetInBound*, *AllowAzureLoadBalancerInBound*, *DenyAllInBound*, *AllowVnetOutBound*, *AllowInternetOutBound*, and *DenyAllOutBound* also exist for the network security group. Don't edit or delete these default rules.
>
> The Azure SLA doesn't apply to deployments where an improperly configured network security group and/or user defined route tables have been applied that blocks Azure AD DS from updating and managing your domain.

Port 443 - synchronization with Azure AD

- Used to synchronize your Azure AD tenant with your managed domain.
- Without access to this port, your managed domain can't sync with your Azure AD tenant. Users may not be able to sign in as changes to their passwords wouldn't be synchronized to your managed domain.
- Inbound access to this port to IP addresses is restricted by default using the **AzureActiveDirectoryDomainServices** service tag.
- Do not restrict outbound access from this port.

Port 3389 - management using remote desktop

- Used for remote desktop connections to domain controllers in your managed domain.
- The default network security group rule uses the *CorpNetSaw* service tag to further restrict traffic.
  - This service tag permits only secure access workstations on the Microsoft corporate network to use remote desktop to the managed domain.
  - Access is only allowed with business justification, such as for management or troubleshooting scenarios.
- This rule can be set to *Deny*, and only set to *Allow* when required. Most management and monitoring tasks are performed using PowerShell remoting. RDP is only used in the rare event that Microsoft needs to connect remotely to your managed domain for advanced troubleshooting.

> **NOTE**
>
> You can't manually select the *CorpNetSaw* service tag from the portal if you try to edit this network security group rule. You must use Azure PowerShell or the Azure CLI to manually configure a rule that uses the *CorpNetSaw* service tag.

Port 5986 - management using PowerShell remoting

- Used to perform management tasks using PowerShell remoting in your managed domain.
- Without access to this port, your managed domain can't be updated, configured, backed-up, or monitored.
- For managed domains that use a Resource Manager-based virtual network, you can restrict inbound access to this port to the *AzureActiveDirectoryDomainServices* service tag.
  - For legacy managed domains using a Classic-based virtual network, you can restrict inbound access to this port to the following source IP addresses: *52.180.183.8*, *23.101.0.70*, *52.225.184.198*, *52.179.126.223*, *13.74.249.156*, *52.187.117.83*, *52.161.13.95*, *104.40.156.18*, and *104.40.87.209*.

> **NOTE**
>
> In 2017, Azure AD Domain Services became available to host in an Azure Resource Manager network. Since then, we have been able to build a more secure service using the Azure Resource Manager's modern capabilities. Because Azure Resource Manager deployments fully replace classic deployments, Azure AD DS classic virtual network deployments will be retired on March 1, 2023.
>
> For more information, see the official deprecation notice

User-defined routes

User-defined routes aren't created by default, and aren't needed for Azure AD DS to work correctly. If you're required to use route tables, avoid making any changes to the *0.0.0.0* route. Changes to this route disrupt Azure AD DS and puts the managed domain in an unsupported state.

You must also route inbound traffic from the IP addresses included in the respective Azure service tags to the managed domain's subnet. For more information on service tags and their associated IP address from, see Azure IP Ranges and Service Tags - Public Cloud.

Caution

These Azure datacenter IP ranges can change without notice. Ensure you have processes to validate you have the latest IP addresses.

Next steps

For more information about some of the network resources and connection options used by Azure AD DS, see the following articles:

- Azure virtual network peering
- Azure VPN gateways
- Azure network security groups

Administer Group Policy in an Azure Active Directory Domain Services managed domain

7/20/2020 • 5 minutes to read • Edit Online

Settings for user and computer objects in Azure Active Directory Domain Services (Azure AD DS) are often managed using Group Policy Objects (GPOs). Azure AD DS includes built-in GPOs for the *AADDC Users* and *AADDC Computers* containers. You can customize these built-in GPOs to configure Group Policy as needed for your environment. Members of the *Azure AD DC administrators* group have Group Policy administration privileges in the Azure AD DS domain, and can also create custom GPOs and organizational units (OUs). More more information on what Group Policy is and how it works, see Group Policy overview.

In a hybrid environment, group policies configured in an on-premises AD DS environment aren't synchronized to Azure AD DS. To define configuration settings for users or computers in Azure AD DS, edit one of the default GPOs or create a custom GPO.

This article shows you how to install the Group Policy Management tools, then edit the built-in GPOs and create custom GPOs.

Before you begin

To complete this article, you need the following resources and privileges:

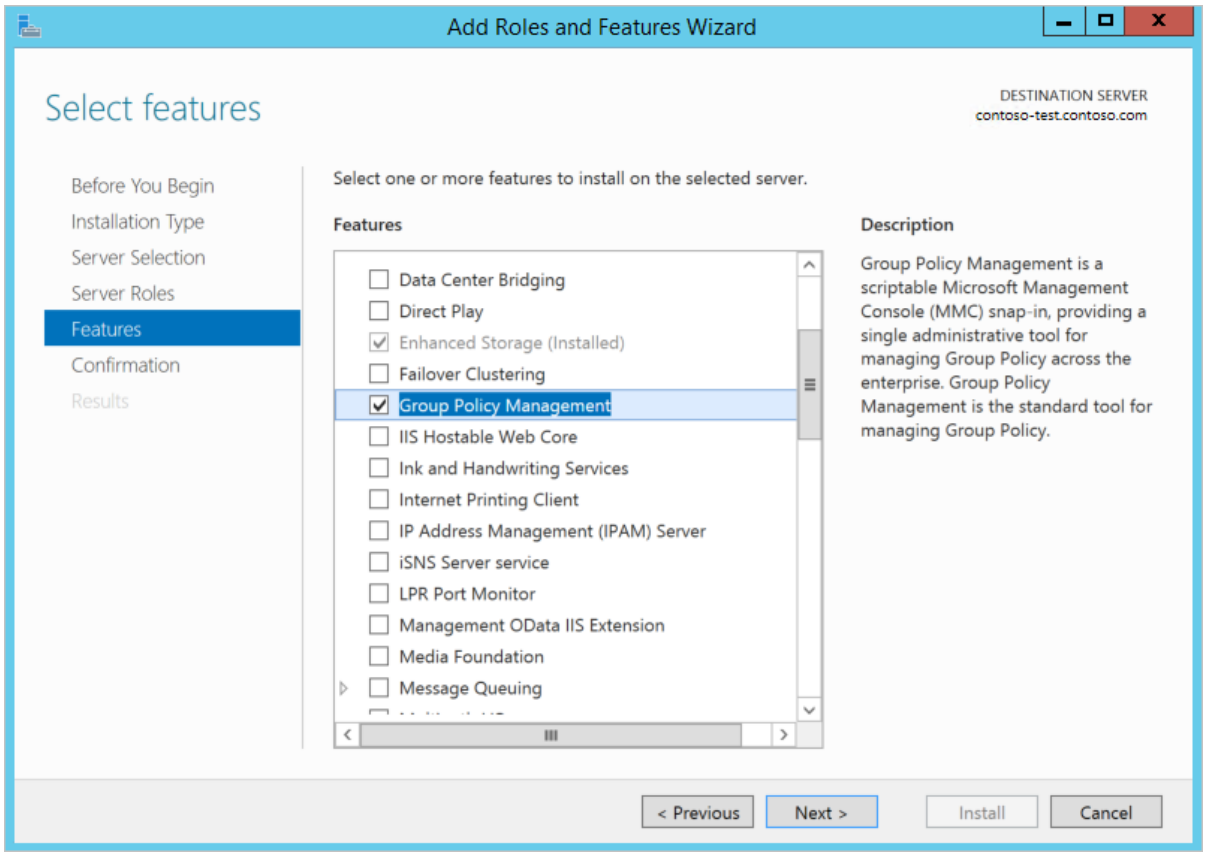
- An active Azure subscription.
  - If you don't have an Azure subscription, create an account.
- An Azure Active Directory tenant associated with your subscription, either synchronized with an on-premises directory or a cloud-only directory.
  - If needed, create an Azure Active Directory tenant or associate an Azure subscription with your account.
- An Azure Active Directory Domain Services managed domain enabled and configured in your Azure AD tenant.
  - If needed, complete the tutorial to create and configure an Azure Active Directory Domain Services managed domain.
- A Windows Server management VM that is joined to the Azure AD DS managed domain.
  - If needed, complete the tutorial to create a Windows Server VM and join it to a managed domain.
- A user account that's a member of the *Azure AD DC administrators* group in your Azure AD tenant.

> **NOTE**
>
> You can use Group Policy Administrative Templates by copying the new templates to the management workstation. Copy the *.admx* files into `%SYSTEMROOT%\PolicyDefinitions` and copy the locale-specific *.adml* files to `%SYSTEMROOT%\PolicyDefinitions\[Language-CountryRegion]`, where `Language-CountryRegion` matches the language and region of the *.adml* files.
>
> For example, copy the English, United States version of the *.adml* files into the `\en-us` folder.
>
> Alternatively, you can centrally store your Group Policy Administrative Template on the domain controllers that are part of the managed domain. For more information, see How to create and manage the Central Store for Group Policy Administrative Templates in Windows.

Install Group Policy Management tools

To create and configure Group Policy Object (GPOs), you need to install the Group Policy Management tools. These tools can be installed as a feature in Windows Server. For more information on how to install the administrative tools on a Windows client, see install Remote Server Administration Tools (RSAT).

1. Sign in to your management VM. For steps on how to connect using the Azure portal, see Connect to a Windows Server VM.
2. **Server Manager** should open by default when you sign in to the VM. If not, on the **Start** menu, select **Server Manager**.
3. In the *Dashboard* pane of the **Server Manager** window, select **Add Roles and Features**.
4. On the **Before You Begin** page of the *Add Roles and Features Wizard*, select **Next**.
5. For the *Installation Type*, leave the **Role-based or feature-based installation** option checked and select **Next**.
6. On the **Server Selection** page, choose the current VM from the server pool, such as *myvm.aaddscontoso.com*, then select **Next**.
7. On the **Server Roles** page, click **Next**.
8. On the **Features** page, select the **Group Policy Management** feature.



9. On the **Confirmation** page, select **Install**. It may take a minute or two to install the Group Policy Management tools.
10. When feature installation is complete, select **Close** to exit the **Add Roles and Features** wizard.

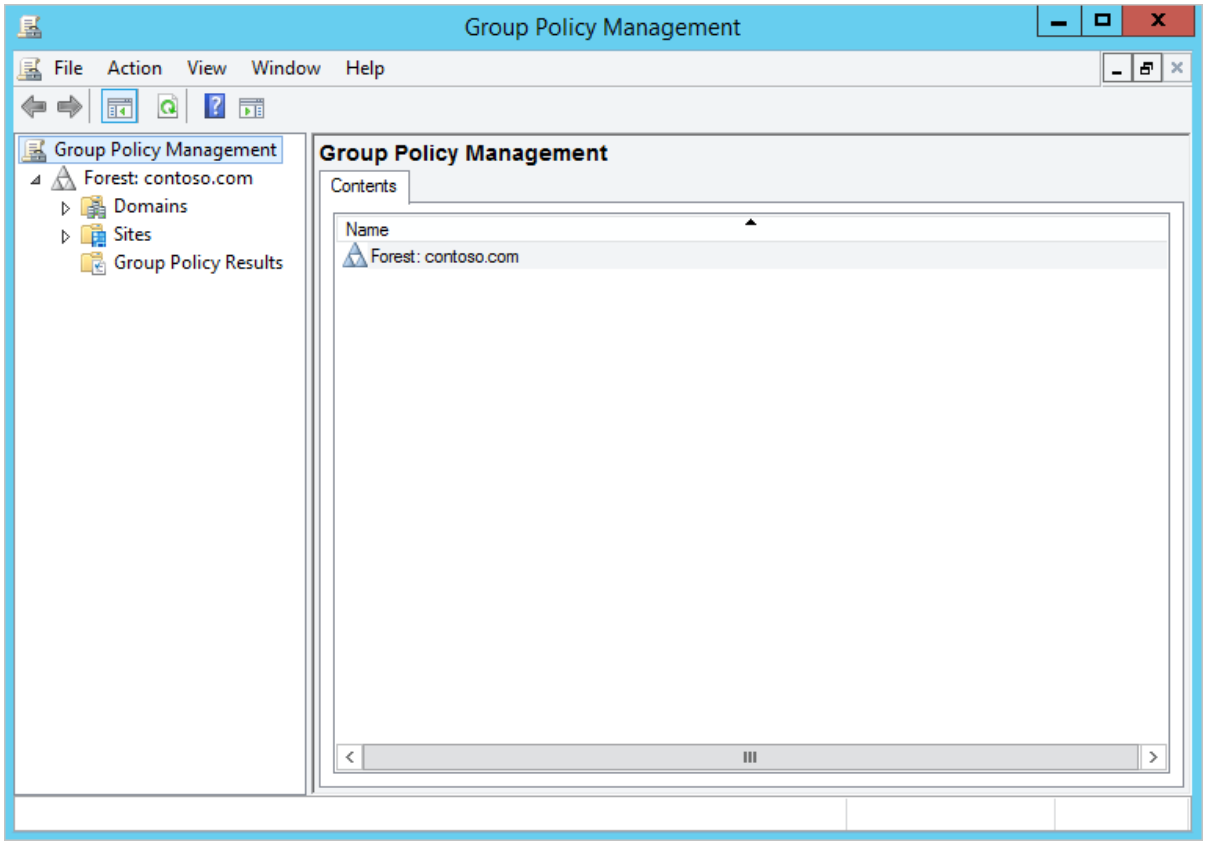
Open the Group Policy Management Console and edit an object

Default group policy objects (GPOs) exist for users and computers in a managed domain. With the Group Policy Management feature installed from the previous section, let's view and edit an existing GPO. In the next section, you create a custom GPO.

> **NOTE**
>
> To administer group policy in a managed domain, you must be signed in to a user account that's a member of the *AAD DC Administrators* group.

1. From the Start screen, select **Administrative Tools**. A list of available management tools is shown, including **Group Policy Management** installed in the previous section.

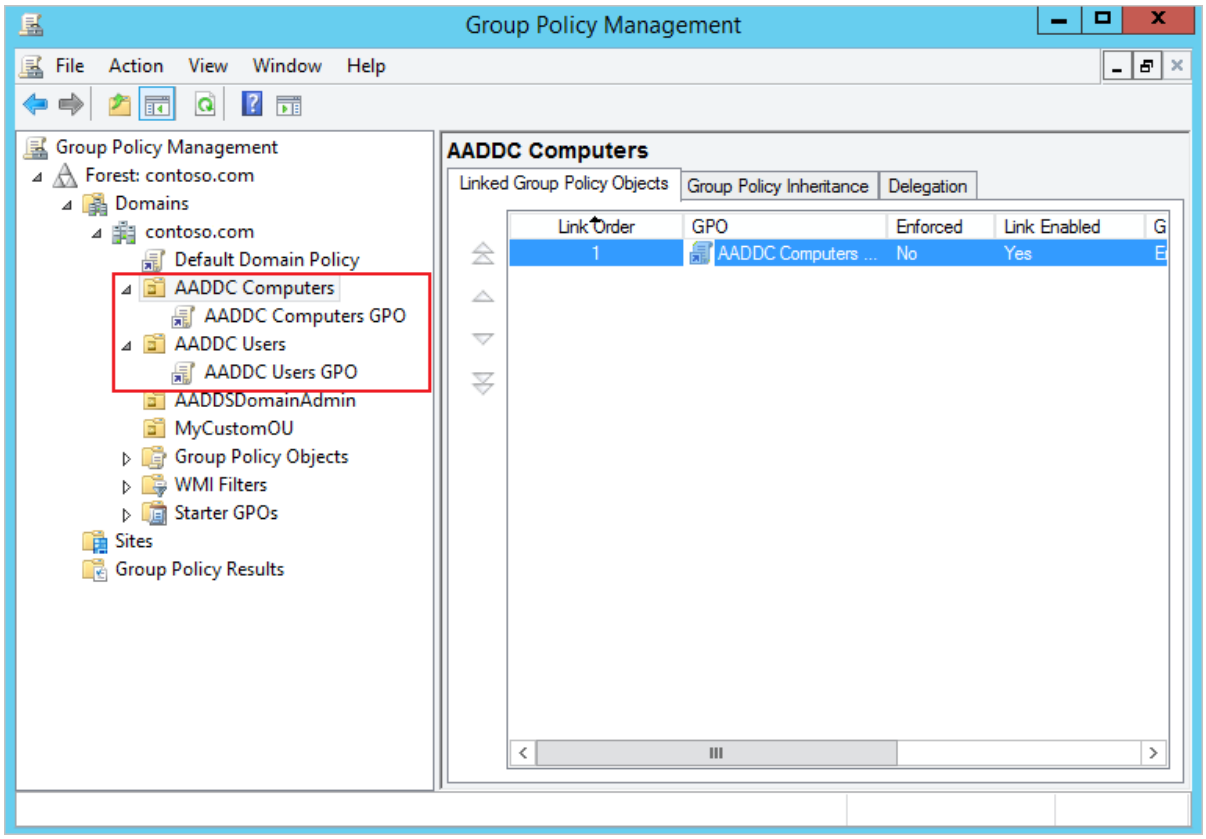
2. To open the Group Policy Management Console (GPMC), choose **Group Policy Management**.



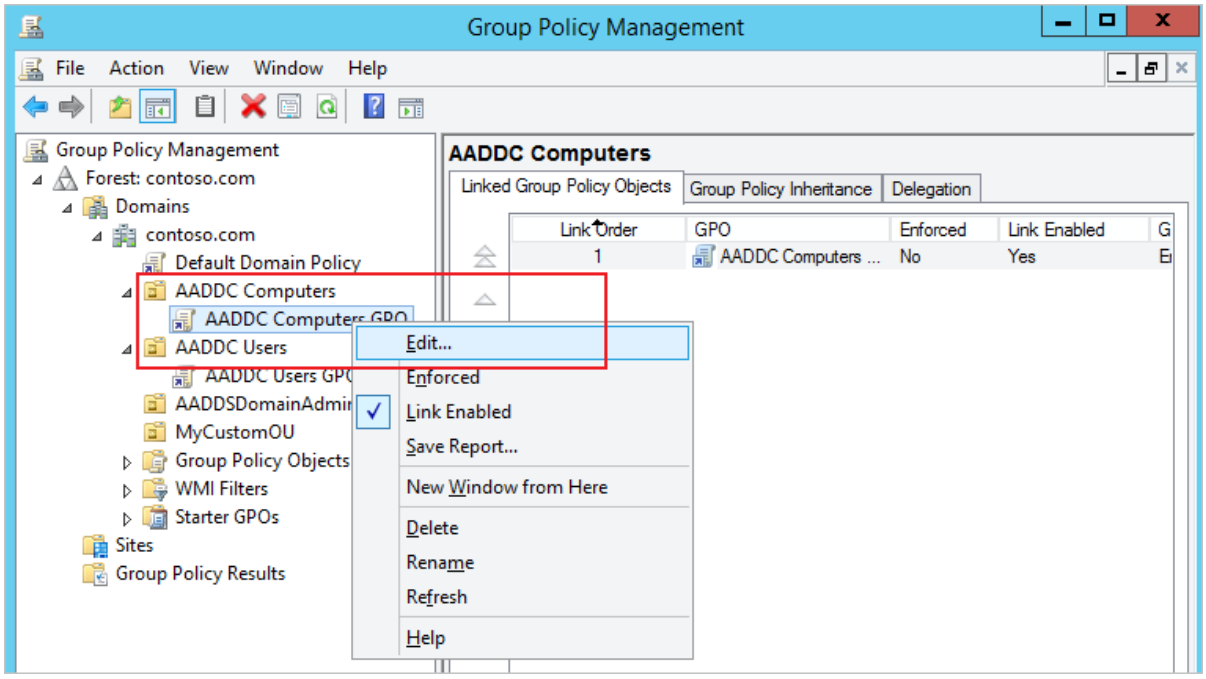
There are two built-in Group Policy Objects (GPOs) in a managed domain - one for the *AADDC Computers* container, and one for the *AADDC Users* container. You can customize these GPOs to configure group policy as needed within your managed domain.

1. In the **Group Policy Management** console, expand the **Forest: aaddscontoso.com** node. Next, expand the **Domains** nodes.

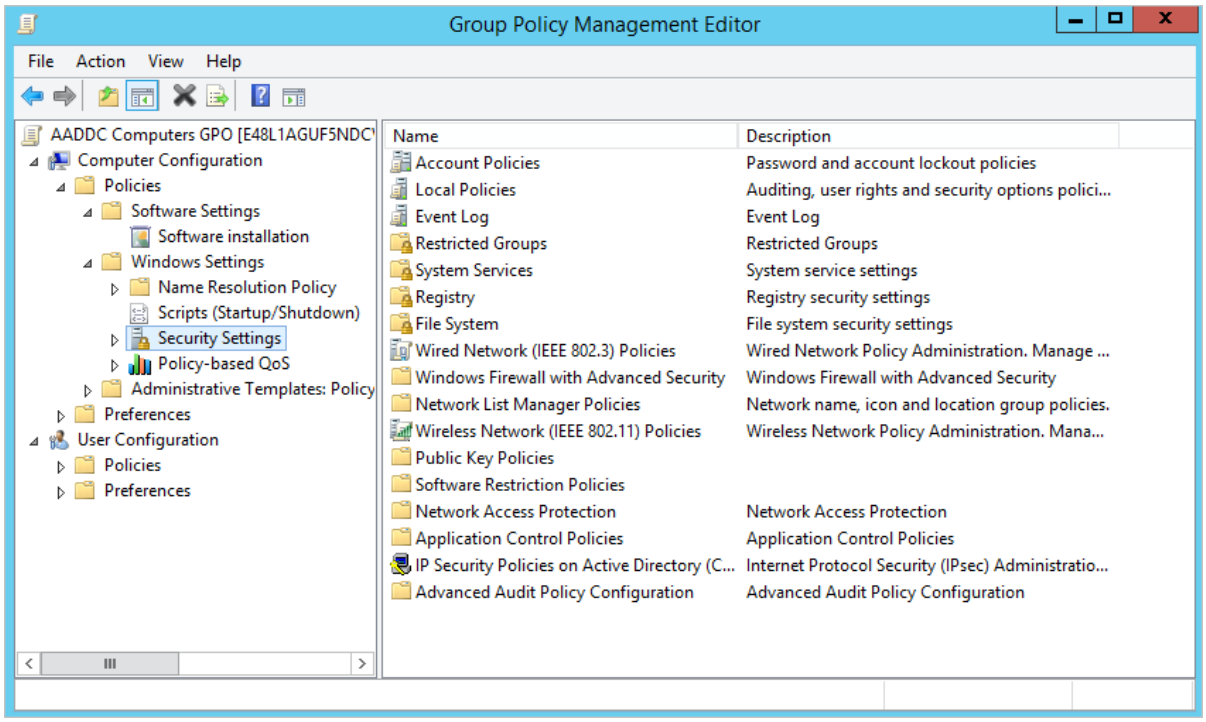
   Two built-in containers exist for *AADDC Computers* and *AADDC Users*. Each of these containers has a default GPO applied to them.



2. These built-in GPOs can be customized to configure specific group policies on your managed domain. Right-select one of the GPOs, such as *AADDC Computers GPO*, then choose **Edit...**.



3. The Group Policy Management Editor tool opens to let you customize the GPO, such as *Account Policies*:

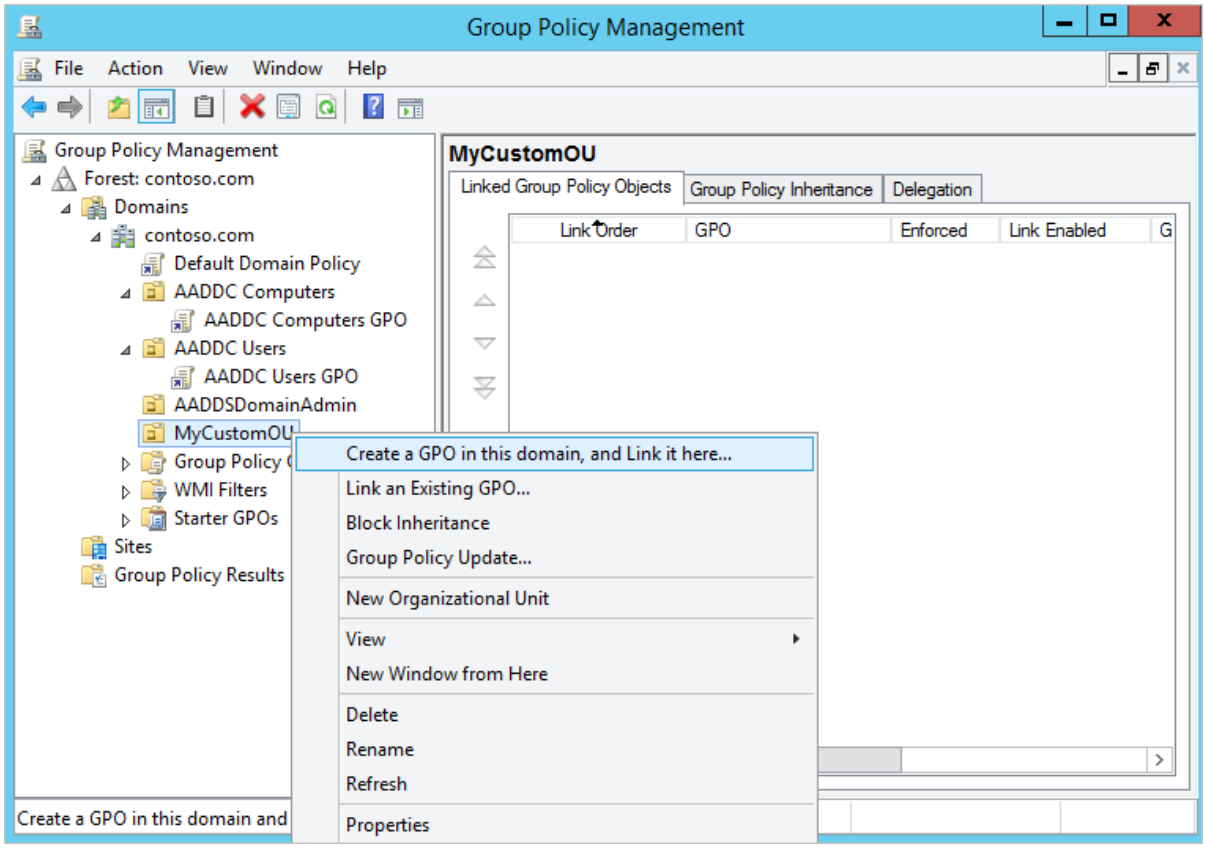


When done, choose **File > Save** to save the policy. Computers refresh Group Policy by default every 90 minutes and apply the changes you made.

Create a custom Group Policy Object

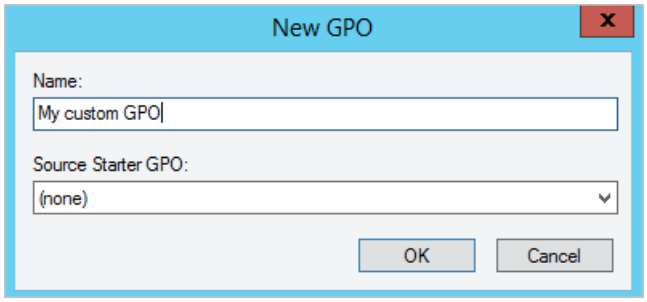
To group similar policy settings, you often create additional GPOs instead of applying all of the required settings in the single, default GPO. With Azure AD DS, you can create or import your own custom group policy objects and link them to a custom OU. If you need to first create a custom OU, see create a custom OU in a managed domain.

1. In the **Group Policy Management** console, select your custom organizational unit (OU), such as *MyCustomOU*. Right-select the OU and choose **Create a GPO in this domain, and Link it here...**:

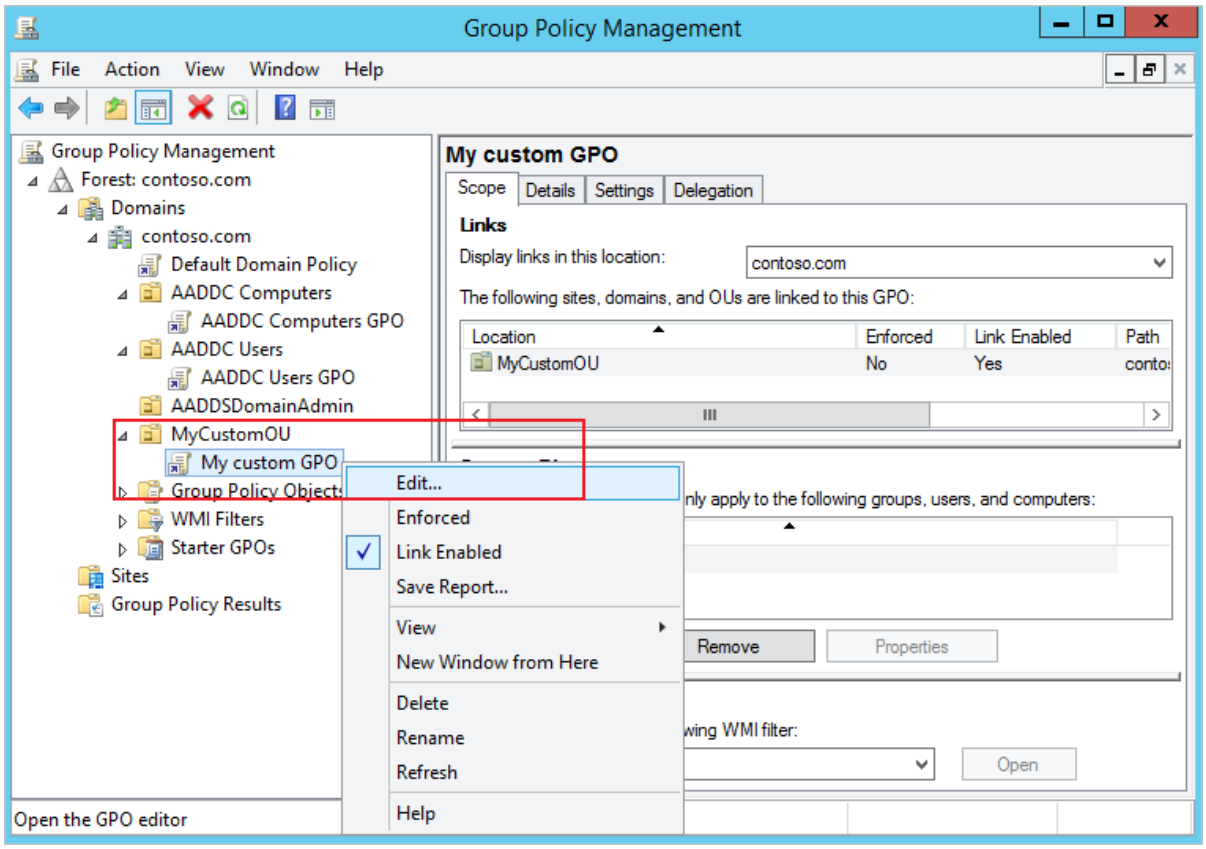


2. Specify a name for the new GPO, such as *My custom GPO*, then select **OK**. You can optionally base this

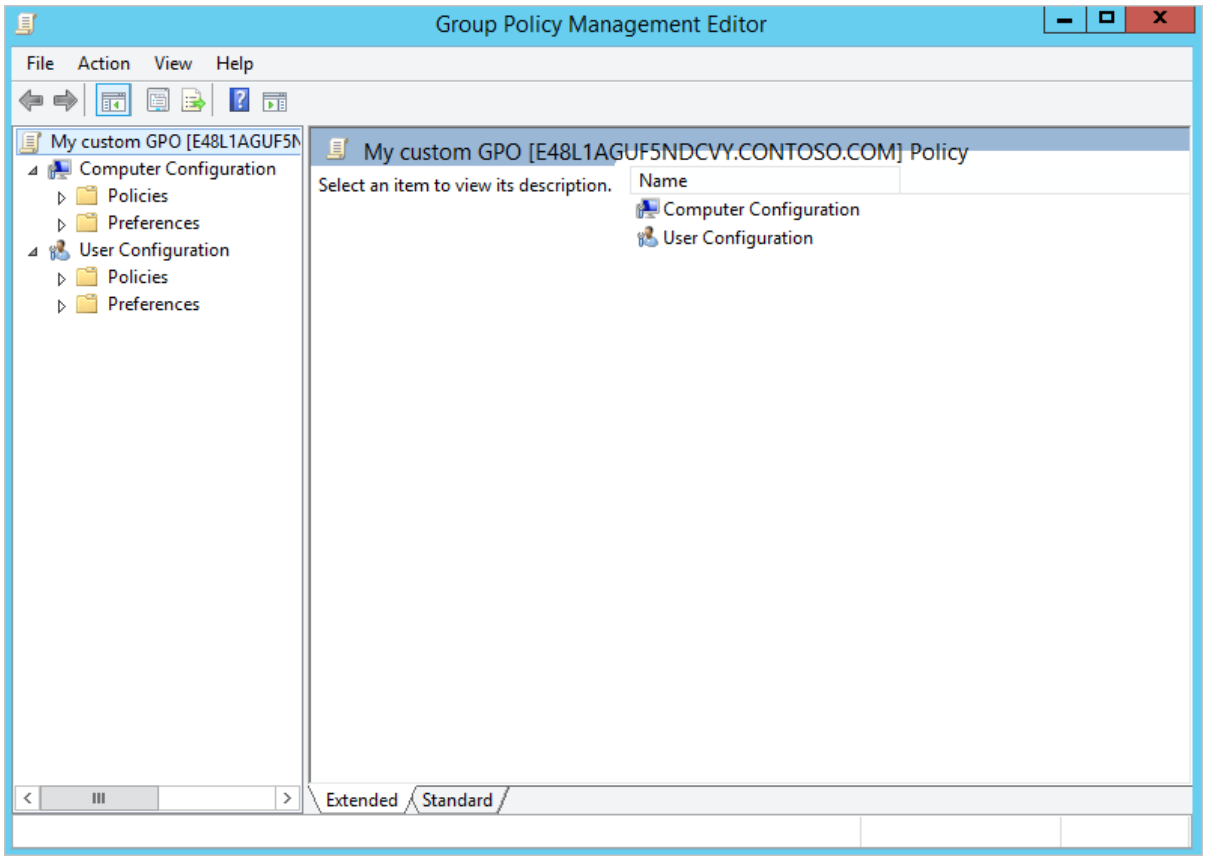
custom GPO on an existing GPO and set of policy options.



3. The custom GPO is created and linked to your custom OU. To now configure the policy settings, right-select the custom GPO and choose **Edit...**:



4. The **Group Policy Management Editor** opens to let you customize the GPO:



When done, choose **File > Save** to save the policy. Computers refresh Group Policy by default every 90 minutes and apply the changes you made.

Next steps

For more information on the available Group Policy settings that you can configure using the Group Policy Management Console, see Work with Group Policy preference items.

Administer DNS and create conditional forwarders in an Azure Active Directory Domain Services managed domain

7/20/2020 • 5 minutes to read • Edit Online

In Azure Active Directory Domain Services (Azure AD DS), a key component is DNS (Domain Name Resolution). Azure AD DS includes a DNS server that provides name resolution for the managed domain. This DNS server includes built-in DNS records and updates for the key components that allow the service to run.

As you run your own applications and services, you may need to create DNS records for machines that aren't joined to the domain, configure virtual IP addresses for load balancers, or set up external DNS forwarders. Users who belong to the *AAD DC Administrators* group are granted DNS administration privileges on the Azure AD DS managed domain and can create and edit custom DNS records.

In a hybrid environment, DNS zones and records configured in other DNS namespaces, such as an on-premises AD DS environment, aren't synchronized to the managed domain. To resolve named resources in other DNS namespaces, create and use conditional forwarders that point to existing DNS servers in your environment.

This article shows you how to install the DNS Server tools then use the DNS console to manage records and create conditional forwarders in Azure AD DS.

Before you begin

To complete this article, you need the following resources and privileges:

- An active Azure subscription.
  - If you don't have an Azure subscription, create an account.
- An Azure Active Directory tenant associated with your subscription, either synchronized with an on-premises directory or a cloud-only directory.
  - If needed, create an Azure Active Directory tenant or associate an Azure subscription with your account.
- An Azure Active Directory Domain Services managed domain enabled and configured in your Azure AD tenant.
  - If needed, complete the tutorial to create and configure an Azure Active Directory Domain Services managed domain.
- Connectivity from your Azure AD DS virtual network to where your other DNS namespaces are hosted.
  - This connectivity can be provided with an Azure ExpressRoute or Azure VPN Gateway connection.
- A Windows Server management VM that is joined to the managed domain.
  - If needed, complete the tutorial to create a Windows Server VM and join it to a managed domain.
- A user account that's a member of the *Azure AD DC administrators* group in your Azure AD tenant.

Install DNS Server tools

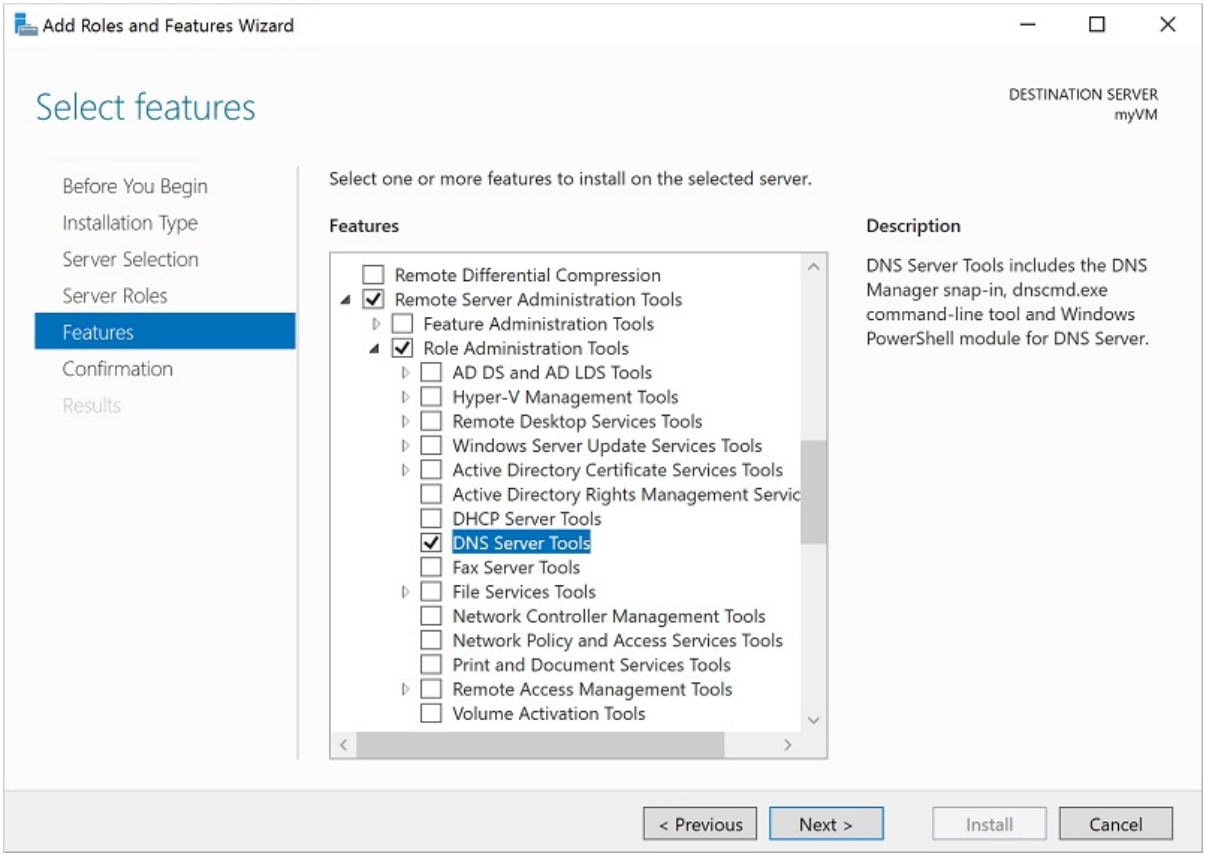
To create and modify DNS records in a managed domain, you need to install the DNS Server tools. These tools can be installed as a feature in Windows Server. For more information on how to install the administrative tools on a Windows client, see install Remote Server Administration Tools (RSAT).

1. Sign in to your management VM. For steps on how to connect using the Azure portal, see Connect to a Windows Server VM.

2. If **Server Manager** doesn't open by default when you sign in to the VM, select the **Start** menu, then choose

Server Manager.

3. In the *Dashboard* pane of the **Server Manager** window, select **Add Roles and Features**.
4. On the **Before You Begin** page of the *Add Roles and Features Wizard*, select **Next**.
5. For the *Installation Type*, leave the **Role-based or feature-based installation** option checked and select **Next**.
6. On the **Server Selection** page, choose the current VM from the server pool, such as *myvm.aaddscontoso.com*, then select **Next**.
7. On the **Server Roles** page, click **Next**.
8. On the **Features** page, expand the **Remote Server Administration Tools** node, then expand the **Role Administration Tools** node. Select **DNS Server Tools** feature from the list of role administration tools.



9. On the **Confirmation** page, select **Install**. It may take a minute or two to install the DNS Server Tools.
10. When feature installation is complete, select **Close** to exit the **Add Roles and Features** wizard.

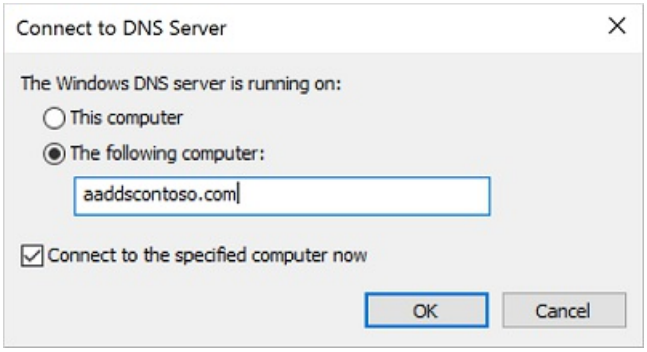
Open the DNS management console to administer DNS

With the DNS Server tools installed, you can administer DNS records on the managed domain.

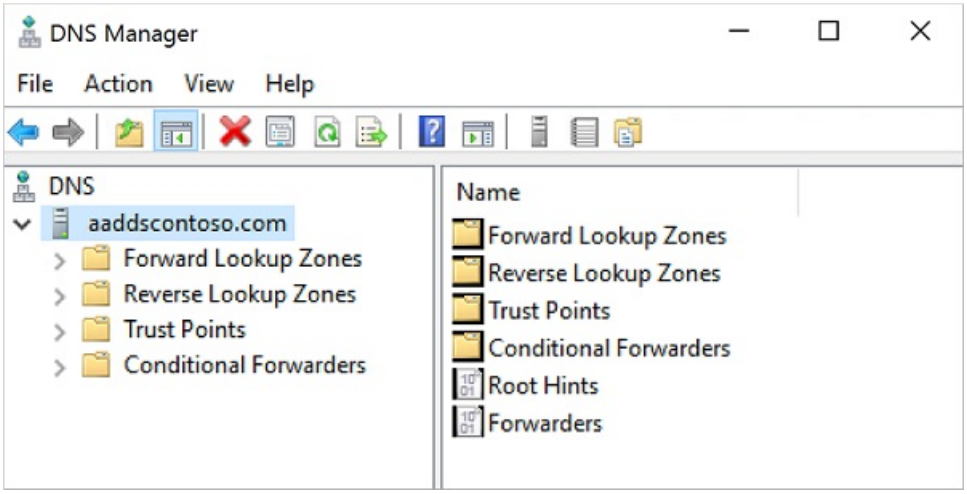
> **NOTE**
> To administer DNS in a managed domain, you must be signed in to a user account that's a member of the *AAD DC Administrators* group.

1. From the Start screen, select **Administrative Tools**. A list of available management tools is shown, including **DNS** installed in the previous section. Select **DNS** to launch the DNS Management console.
2. In the **Connect to DNS Server** dialog, select **The following computer**, then enter the DNS domain name

of the managed domain, such as *aaddscontoso.com*:



3. The DNS Console connects to the specified managed domain. Expand the **Forward Lookup Zones** or **Reverse Lookup Zones** to create your required DNS entries or edit existing records as needed.



WARNING

When you manage records using the DNS Server tools, make sure that you don't delete or modify the built-in DNS records that are used by Azure AD DS. Built-in DNS records include domain DNS records, name server records, and other records used for DC location. If you modify these records, domain services are disrupted on the virtual network.

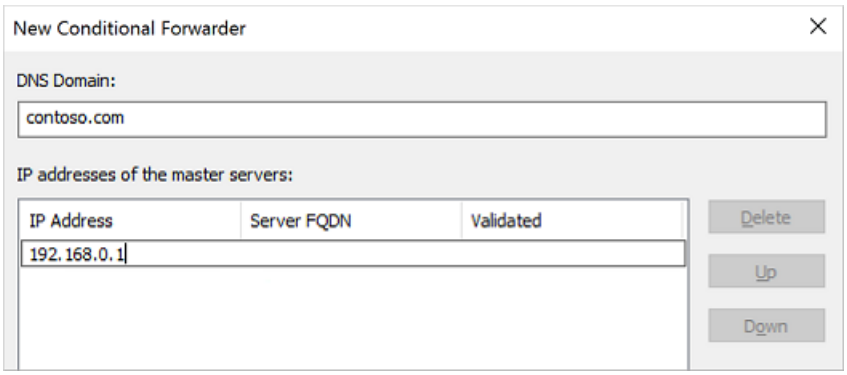
Create conditional forwarders

An Azure AD DS DNS zone should only contain the zone and records for the managed domain itself. Don't create additional zones in the managed domain to resolve named resources in other DNS namespaces. Instead, use conditional forwarders in the managed domain to tell the DNS server where to go in order to resolve addresses for those resources.

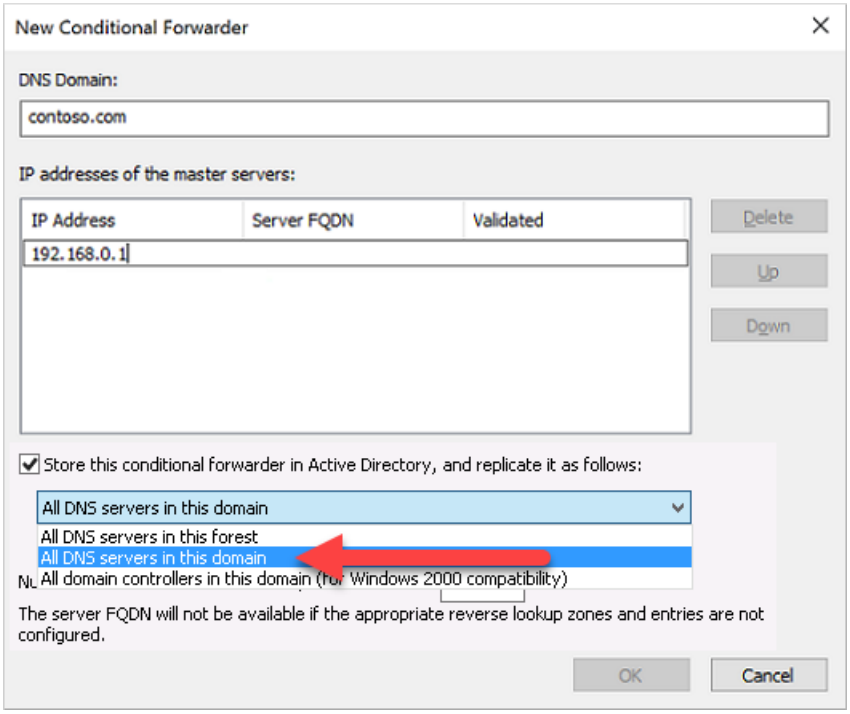
A conditional forwarder is a configuration option in a DNS server that lets you define a DNS domain, such as *contoso.com*, to forward queries to. Instead of the local DNS server trying to resolve queries for records in that domain, DNS queries are forwarded to the configured DNS for that domain. This configuration makes sure that the correct DNS records are returned, as you don't create a local a DNS zone with duplicate records in the managed domain to reflect those resources.

To create a conditional forwarder in your managed domain, complete the following steps:

1. Select your DNS zone, such as *aaddscontoso.com*.
2. Select **Conditional Forwarders**, then right-select and choose **New Conditional Forwarder...**
3. Enter your other **DNS Domain**, such as *contoso.com*, then enter the IP addresses of the DNS servers for that namespace, as shown in the following example:



4. Check the box for **Store this conditional forwarder in Active Directory, and replicate it as follows**, then select the option for *All DNS servers in this domain*, as shown in the following example:



> **IMPORTANT**
>
> If the conditional forwarder is stored in the *forest* instead of the *domain*, the conditional forwarder fails.

5. To create the conditional forwarder, select **OK**.

Name resolution of the resources in other namespaces from VMs connected to the managed domain should now resolve correctly. Queries for the DNS domain configured in the conditional forwarder are passed to the relevant DNS servers.

Next steps

For more information about managing DNS, see the DNS tools article on Technet.

Check the health of an Azure Active Directory Domain Services managed domain

7/20/2020 • 4 minutes to read • Edit Online

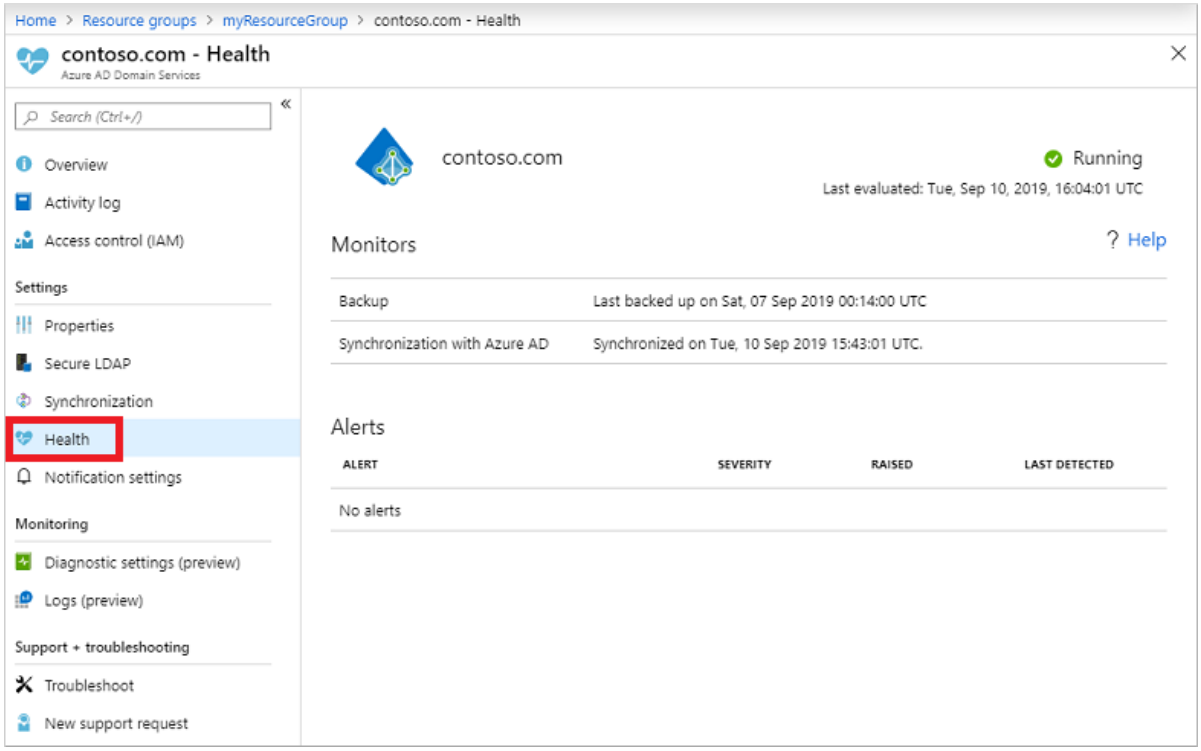
Azure Active Directory Domain Services (Azure AD DS) runs some background tasks to keep the managed domain healthy and up-to-date. These tasks include taking backups, applying security updates, and synchronizing data from Azure AD. If there are issues with the Azure AD DS managed domain, these tasks may not successfully complete. To review and resolve any issues, you can check the health status of a managed domain using the Azure portal.

This article shows you how to view the Azure AD DS health status and understand the information or alerts shown.

View the health status

The health status for a managed domain is viewed using the Azure portal. Information on the last backup time and synchronization with Azure AD can be seen, along with any alerts that indicate a problem with the managed domain's health. To view the health status for a managed domain, complete the following steps:

1. In the Azure portal, search for and select **Azure AD Domain Services**.
2. Select your managed domain, such as *aaddscontoso.com*.
3. On the left-hand side of the Azure AD DS resource window, select **Health**. The following example screenshot shows a healthy managed domain and the status of the last backup and Azure AD synchronization:



The *Last evaluated* timestamp of the health page shows when the managed domain was last checked. The health of a managed domain is evaluated every hour. If you make any changes to a managed domain, wait until the next evaluation cycle to view the updated health status.

The status in the top right indicates the overall health of the managed domain. The status factors all of the existing

alerts on your domain. The following table details the available status indicators:

STATUS	ICON	EXPLANATION
Running		The managed domain is running correctly and doesn't have any critical or warning alerts. The domain may have informational alerts.
Needs attention (warning)		There are no critical alerts on the managed domain, but there are one or more warning alerts that should be addressed.
Needs attention (critical)		There are one or more critical alerts on the managed domain that must be addressed. You may also have warning and / or informational alerts.
Deploying		The managed domain is being deployed.

Understand monitors and alerts

The health status for a managed domain show two types of information - *monitors*, and *alerts*. Monitors show the time that core background tasks were completed. Alerts provide information or suggestions to improve the stability of the managed domain.

Monitors

Monitors are areas of a managed domain that are checked on a regular basis. If there are any active alerts for the managed domain, it may cause one of the monitors to report an issue. Azure AD DS currently has monitors for the following areas:

- Backup
- Synchronization with Azure AD

Backup monitor

The backup monitor checks that automated regular backups of the managed domain successfully run. The following table details the available backup monitor status:

DETAIL VALUE	EXPLANATION
Never backed up	This state is normal for new managed domains. The first backup should be created 24 hours after the managed domain is deployed. If this status persists, open an Azure support request.
Last backup was taken 1 to 14 days ago	This time range is the expected status for the backup monitor. Automated regular backups should occur in this period.
Last backup was taken more than 14 days ago.	A timespan longer than two weeks indicates there's an issue with the automated regular backups. Active critical alerts may prevent the managed domain from being backed up. Resolve any active alerts for the managed domain. If the backup monitor doesn't then update the status to report a recent backup, open an Azure support request.

Synchronization with Azure AD monitor

A managed domain regularly synchronizes with Azure Active Directory. The number of users and group objects, and the number of changes made in the Azure AD directory since the last sync, affects how long it takes to synchronize. If the managed domain was last synchronized over three days ago, check for and resolve any active alerts. If the synchronization monitor doesn't update the status to show a recent sync after you address any active alerts, open an Azure support request.

Alerts

Alerts are generated for issues in a managed domain that need to be addressed for the service to run correctly. Each alert explains the problem and gives a URL that outlines specific steps to resolve the issue. For more information on the possible alerts and their resolutions, see Troubleshooting alerts.

Health status alerts are categorized into the following levels of severity:

- **Critical alerts** are issues that severely impact the managed domain. These alerts should be addressed immediately. The Azure platform can't monitor, manage, patch, and synchronize the managed domain until the issues are resolved.
- **Warning alerts** notify you of issues that may impact the managed domain operations if the problem persists. These alerts also offer recommendations to secure the managed domain.
- **Informational alerts** are notifications that don't negatively impact the managed domain. Informational alerts provide some insight as to what's happening in the managed domain.

Next steps

For more information on alerts that are shown in the health status page, see Resolve alerts on your managed domain

Configure email notifications for issues in Azure Active Directory Domain Services

7/20/2020 • 3 minutes to read • Edit Online

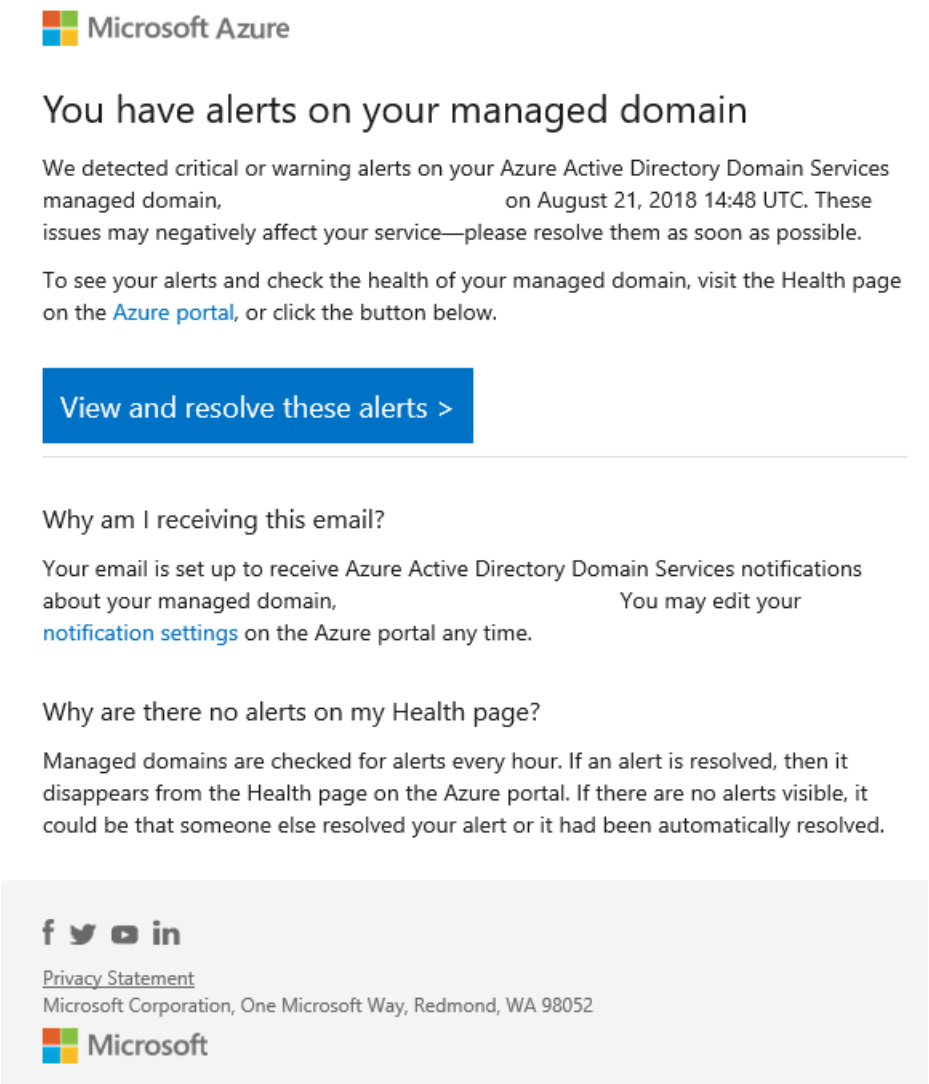
The health of an Azure Active Directory Domain Services (Azure AD DS) managed domain is monitored by the Azure platform. The health status page in the Azure portal shows any alerts for the managed domain. To make sure issues are responded to in a timely manner, email notifications can be configured to report on health alerts as soon as they're detected in the Azure AD DS managed domain.

This article shows you how to configure email notification recipients for a managed domain.

Email notification overview

To alert you of issues with a managed domain, you can configure email notifications. These email notifications specify the managed domain that the alert is present on, as well as giving the time of detection and a link to the health page in the Azure portal. You can then follow the provided troubleshooting advice to resolve the issues.

The following example email notification indicates a critical warning or alert was generated on the managed domain:



> **WARNING**
> Always make sure that the email comes from a verified Microsoft sender before you click the links in the message. The email notifications always come from the `azure-noreply@microsoft.com` address.

Why would I receive email notifications?

Azure AD DS sends email notifications for important updates about the managed domain. These notifications are only for urgent issues that impact the service and should be addressed immediately. Each email notification is triggered by an alert on the managed domain. The alerts also appear in the Azure portal and can be viewed on the Azure AD DS health page.

Azure AD DS doesn't send emails for advertisement, updates, or sales purposes.

When will I receive email notifications?

A notification is sent immediately when a new alert is found on a managed domain. If the alert isn't resolved, additional email notifications are sent as a reminder every four days.

Who should receive the email notifications?

The list of email recipients for Azure AD DS should be composed of people who are able to administer and make changes to the managed domain. This email list should be thought of as your "first responders" to any alerts and issues.

You can add up to five additional emails recipients for email notifications. If you want more than five recipients for email notifications, create a distribution list and add that to the notification list instead.

You can also choose to have all *Global Administrators* of the Azure AD directory and every member of the *AAD DC Administrators* group receive email notifications. Azure AD DS only sends notification to up to 100 email addresses, including the list of global administrators and AAD DC administrators.

Configure email notifications

To review the existing email notification recipients or add additional recipients, complete the following steps:

1. In the Azure portal, search for and select **Azure AD Domain Services**.
2. Select your managed domain, such as *aaddscontoso.com*.
3. On the left-hand side of the Azure AD DS resource window, select **Notification settings**. The existing recipients for email notifications are shown.
4. To add an email recipient, enter the email address in the additional recipients table.
5. When done, select **Save** on the top-hand navigation.

> **WARNING**
> When you change the notification settings, the notification settings for the entire managed domain are updated, not just yourself.

Frequently asked questions

I received an email notification for an alert but when I logged on to the Azure portal there was no alert. What happened?

If an alert is resolved, the alert is cleared from the Azure portal. The most likely reason is that someone else who receives email notifications resolved the alert on the managed domain, or it was autoresolved by Azure platform.

Why can I not edit the notification settings?

If you're unable to access the notification settings page in the Azure portal, you don't have the permissions to edit the managed domain. Contact a global administrator to either get permissions to edit Azure AD DS resource or be removed from the recipient list.

I don't seem to be receiving email notifications even though I provided my email address. Why?

Check your spam or junk folder in your email for the notification and make sure to allow the sender of `azure-noreply@microsoft.com`.

Next steps

For more information on troubleshooting some of the issues that may be reported, see Resolve alerts on a managed domain.

Delete an Azure Active Directory Domain Services managed domain using the Azure portal

7/20/2020 • 2 minutes to read • Edit Online

If you no longer need an Azure Active Directory Domain Services (Azure AD DS) managed domain, you can delete it. There's no option to turn off or temporarily disable an Azure AD DS managed domain. Deleting the managed domain doesn't delete or otherwise adversely impact the Azure AD tenant.

This article shows you how to use the Azure portal to delete a managed domain.

WARNING

Deletion is permanent and can't be reversed.

When you delete a managed domain, the following steps occur:

- Domain controllers for the managed domain are de-provisioned and removed from the virtual network.
- Data on the managed domain is deleted permanently. This data includes custom OUs, GPOs, custom DNS records, service principals, GMSAs, etc. that you created.
- Machines joined to the managed domain lose their trust relationship with the domain and need to be unjoined from the domain.
  - You can't sign in to these machines using corporate AD credentials. Instead, you must use the local administrator credentials for the machine.

Delete the managed domain

To delete a managed domain, complete the following steps:

1. In the Azure portal, search for and select **Azure AD Domain Services**.
2. Select the name of your managed domain, such as *aaddscontoso.com*.
3. On the **Overview** page, select **Delete**. To confirm the deletion, type the domain name of the managed domain again, then select **Delete**.

It can take 15-20 minutes or more to delete the managed domain.

Next steps

Consider sharing feedback for the features that you would like to see in Azure AD DS.

If you want to get started with Azure AD DS again, see Create and configure an Azure Active Directory Domain Services managed domain.

Migrate Azure Active Directory Domain Services from the Classic virtual network model to Resource Manager

7/20/2020 • 18 minutes to read • Edit Online

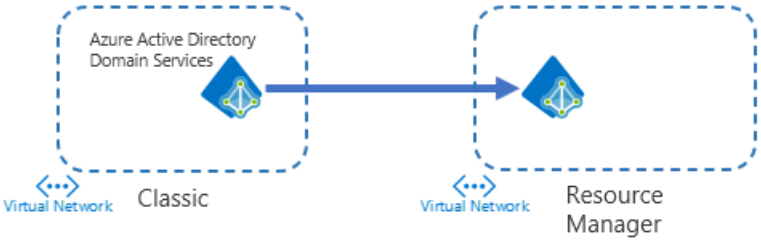
Azure Active Directory Domain Services (Azure AD DS) supports a one-time move for customers currently using the Classic virtual network model to the Resource Manager virtual network model. Azure AD DS managed domains that use the Resource Manager deployment model provide additional features such as fine-grained password policy, audit logs, and account lockout protection.

This article outlines considerations for migration, then the required steps to successfully migrate an existing managed domain. For some of the benefits, see Benefits of migration from the Classic to Resource Manager deployment model in Azure AD DS.

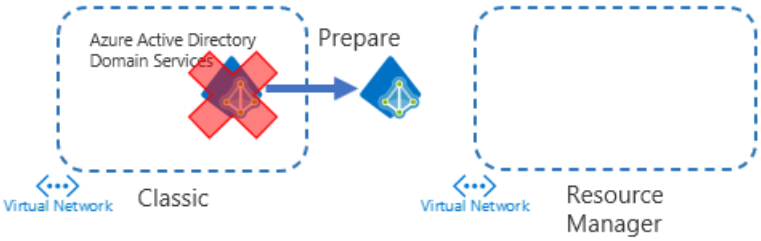
> **NOTE**
>
> In 2017, Azure AD Domain Services became available to host in an Azure Resource Manager network. Since then, we have been able to build a more secure service using the Azure Resource Manager's modern capabilities. Because Azure Resource Manager deployments fully replace classic deployments, Azure AD DS classic virtual network deployments will be retired on March 1, 2023.
>
> For more information, see the official deprecation notice.

Overview of the migration process

The migration process takes an existing managed domain that runs in a Classic virtual network and moves it to an existing Resource Manager virtual network. The migration is performed using PowerShell, and has two main stages of execution: *preparation* and *migration*.

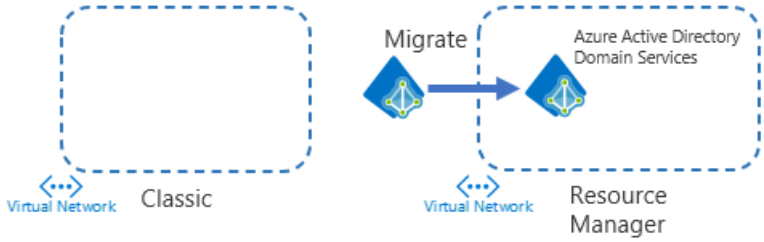


In the *preparation* stage, Azure AD DS takes a backup of the domain to get the latest snapshot of users, groups, and passwords synchronized to the managed domain. Synchronization is then disabled, and the cloud service that hosts the managed domain is deleted. During the preparation stage, the managed domain is unable to authenticate users.



In the *migration* stage, the underlying virtual disks for the domain controllers from the Classic managed domain

are copied to create the VMs using the Resource Manager deployment model. The managed domain is then recreated, which includes the LDAPS and DNS configuration. Synchronization to Azure AD is restarted, and LDAP certificates are restored. There's no need to rejoin any machines to a managed domain – they continue to be joined to the managed domain and run without changes.



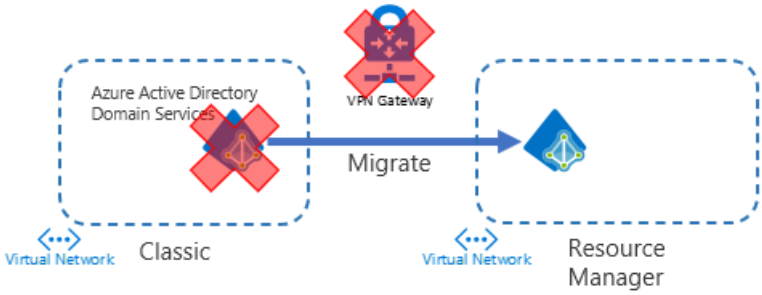
Example scenarios for migration

Some common scenarios for migrating a managed domain include the following examples.

> **NOTE**
> Don't convert the Classic virtual network until you have confirmed a successful migration. Converting the virtual network removes the option to roll back or restore the managed domain if there are any problems during the migration and verification stages.

Migrate Azure AD DS to an existing Resource Manager virtual network (recommended)

A common scenario is where you've already moved other existing Classic resources to a Resource Manager deployment model and virtual network. Peering is then used from the Resource Manager virtual network to the Classic virtual network that continues to run Azure AD DS. This approach lets the Resource Manager applications and services use the authentication and management functionality of the managed domain in the Classic virtual network. Once migrated, all resources run using the Resource Manager deployment model and virtual network.

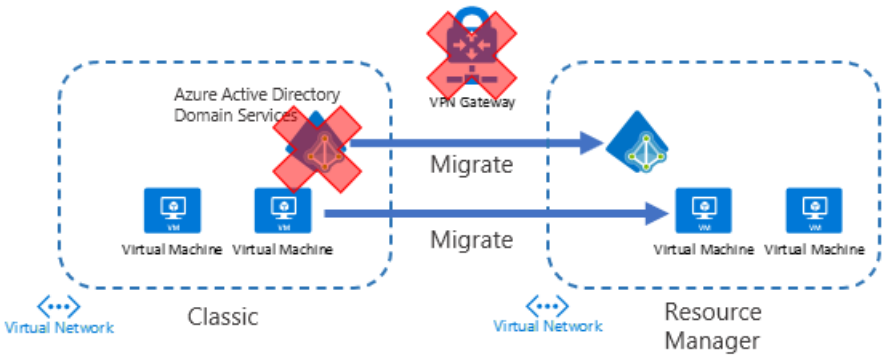


High-level steps involved in this example migration scenario include the following parts:

1. Remove existing VPN gateways or virtual network peering configured on the Classic virtual network.
2. Migrate the managed domain using the steps outlined in this article.
3. Test and confirm a successful migration, then delete the Classic virtual network.

Migrate multiple resources including Azure AD DS

In this example scenario, you migrate Azure AD DS and other associated resources from the Classic deployment model to the Resource Manager deployment model. If some resources continued to run in the Classic virtual network alongside the managed domain, they can all benefit from migrating to the Resource Manager deployment model.

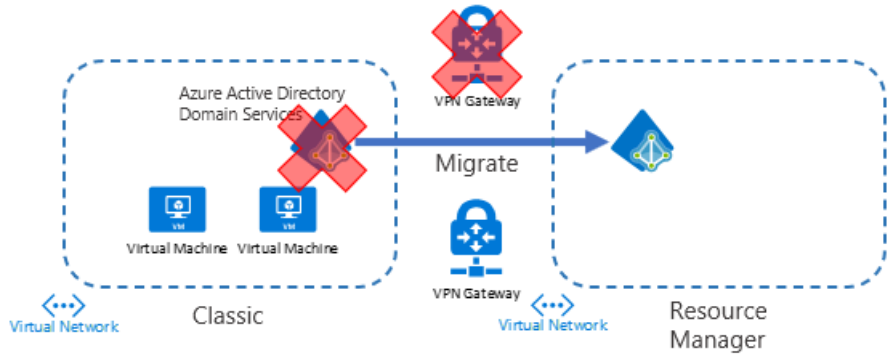


High-level steps involved in this example migration scenario include the following parts:

1. Remove existing VPN gateways or virtual network peering configured on the Classic virtual network.
2. Migrate the managed domain using the steps outlined in this article.
3. Set up virtual network peering between the Classic virtual network and Resource Manager network.
4. Test and confirm a successful migration.
5. Move additional Classic resources like VMs.

Migrate Azure AD DS but keep other resources on the Classic virtual network

With this example scenario, you have the minimum amount of downtime in one session. You only migrate Azure AD DS to a Resource Manager virtual network, and keep existing resources on the Classic deployment model and virtual network. In a following maintenance period, you can migrate the additional resources from the Classic deployment model and virtual network as desired.



High-level steps involved in this example migration scenario include the following parts:

1. Remove existing VPN gateways or virtual network peering configured on the Classic virtual network.
2. Migrate the managed domain using the steps outlined in this article.
3. Set up virtual network peering between the Classic virtual network and the new Resource Manager virtual network.
4. Later, migrate the additional resources from the Classic virtual network as needed.

Before you begin

As you prepare and then migrate a managed domain, there are some considerations around the availability of authentication and management services. The managed domain is unavailable for a period of time during migration. Applications and services that rely on Azure AD DS experience downtime during migration.

> **IMPORTANT**
> Read all of this migration article and guidance before you start the migration process. The migration process affects the availability of the Azure AD DS domain controllers for periods of time. Users, services, and applications can't authenticate against the managed domain during the migration process.

IP addresses

The domain controller IP addresses for a managed domain change after migration. This change includes the public IP address for the secure LDAP endpoint. The new IP addresses are inside the address range for the new subnet in the Resource Manager virtual network.

If you need to roll back, the IP addresses may change after rolling back.

Azure AD DS typically uses the first two available IP addresses in the address range, but this isn't guaranteed. You can't currently specify the IP addresses to use after migration.

Downtime

The migration process involves the domain controllers being offline for a period of time. Domain controllers are inaccessible while Azure AD DS is migrated to the Resource Manager deployment model and virtual network.

On average, the downtime is around 1 to 3 hours. This time period is from when the domain controllers are taken offline to the moment the first domain controller comes back online. This average doesn't include the time it takes for the second domain controller to replicate, or the time it may take to migrate additional resources to the Resource Manager deployment model.

Account lockout

Managed domains that run on Classic virtual networks don't have AD account lockout policies in place. If VMs are exposed to the internet, attackers could use password-spray methods to brute-force their way into accounts. There's no account lockout policy to stop those attempts. For managed domains that use the Resource Manager deployment model and virtual networks, AD account lockout policies protect against these password-spray attacks.

By default, 5 bad password attempts in 2 minutes lock out an account for 30 minutes.

A locked out account can't be used to sign in, which may interfere with the ability to manage the managed domain or applications managed by the account. After a managed domain is migrated, accounts can experience what feels like a permanent lockout due to repeated failed attempts to sign in. Two common scenarios after migration include the following:

- A service account that's using an expired password.
  - The service account repeatedly tries to sign in with an expired password, which locks out the account. To fix this, locate the application or VM with expired credentials and update the password.
- A malicious entity is using brute-force attempts to sign in to accounts.
  - When VMs are exposed to the internet, attackers often try common username and password combinations as they attempt to sign. These repeated failed sign-in attempts can lock out the accounts. It's not recommended to use administrator accounts with generic names such as *admin* or *administrator*, for example, to minimize administrative accounts from being locked out.
  - Minimize the number of VMs that are exposed to the internet. You can use Azure Bastion to securely connect to VMs using the Azure portal.

If you suspect that some accounts may be locked out after migration, the final migration steps outline how to enable auditing or change the fine-grained password policy settings.

Roll back and restore

If the migration isn't successful, there's process to roll back or restore a managed domain. Rollback is a self-service option to immediately return the state of the managed domain to before the migration attempt. Azure support engineers can also restore a managed domain from backup as a last resort. For more information, see how to roll back or restore from a failed migration.

Restrictions on available virtual networks

There are some restrictions on the virtual networks that a managed domain can be migrated to. The destination

Resource Manager virtual network must meet the following requirements:

- The Resource Manager virtual network must be in the same Azure subscription as the Classic virtual network that Azure AD DS is currently deployed in.
- The Resource Manager virtual network must be in the same region as the Classic virtual network that Azure AD DS is currently deployed in.
- The Resource Manager virtual network's subnet should have at least 3-5 available IP addresses.
- The Resource Manager virtual network's subnet should be a dedicated subnet for Azure AD DS, and shouldn't host any other workloads.

For more information on virtual network requirements, see Virtual network design considerations and configuration options.

Migration steps

The migration to the Resource Manager deployment model and virtual network is split into 5 main steps:

STEP	PERFORMED THROUGH	ESTIMATED TIME	DOWNTIME	ROLL BACK/RESTORE?
Step 1 - Update and locate the new virtual network	Azure portal	15 minutes	No downtime required	N/A
Step 2 - Prepare the managed domain for migration	PowerShell	15 – 30 minutes on average	Downtime of Azure AD DS starts after this command is completed.	Roll back and restore available.
Step 3 - Move the managed domain to an existing virtual network	PowerShell	1 – 3 hours on average	One domain controller is available once this command is completed, downtime ends.	On failure, both rollback (self-service) and restore are available.
Step 4 - Test and wait for the replica domain controller	PowerShell and Azure portal	1 hour or more, depending on the number of tests	Both domain controllers are available and should function normally.	N/A. Once the first VM is successfully migrated, there's no option for rollback or restore.
Step 5 - Optional configuration steps	Azure portal and VMs	N/A	No downtime required	N/A

IMPORTANT

To avoid additional downtime, read all of this migration article and guidance before you start the migration process. The migration process affects the availability of the Azure AD DS domain controllers for a period of time. Users, services, and applications can't authenticate against the managed domain during the migration process.

Update and verify virtual network settings

Before you begin the migration process, complete the following initial checks and updates. These steps can happen at any time before the migration and don't affect the operation of the managed domain.

1. Update your local Azure PowerShell environment to the latest version. To complete the migration steps, you

need at least version *2.3.2*.

For information on how to check and update your PowerShell version, see Azure PowerShell overview.

2. Create, or choose an existing, Resource Manager virtual network.

   Make sure that network settings don't block necessary ports required for Azure AD DS. Ports must be open on both the Classic virtual network and the Resource Manager virtual network. These settings include route tables (although it's not recommended to use route tables) and network security groups.

   To view the ports required, see Network security groups and required ports. To minimize network communication problems, it's recommended to wait and apply a network security group or route table to the Resource Manager virtual network after the migration successfully completed.

   Make a note of this target resource group, target virtual network, and target virtual network subnet. These resource names are used during the migration process.

3. Check the managed domain health in the Azure portal. If you have any alerts for the managed domain, resolve them before you start the migration process.

4. Optionally, if you plan to move other resources to the Resource Manager deployment model and virtual network, confirm that those resources can be migrated. For more information, see Platform-supported migration of IaaS resources from Classic to Resource Manager.

   > **NOTE**
   > Don't convert the Classic virtual network to a Resource Manager virtual network. If you do, there's no option to roll back or restore the managed domain.

Prepare the managed domain for migration

Azure PowerShell is used to prepare the managed domain for migration. These steps include taking a backup, pausing synchronization, and deleting the cloud service that hosts Azure AD DS. When this step completes, Azure AD DS is taken offline for a period of time. If the preparation step fails, you can roll back to the previous state.

To prepare the managed domain for migration, complete the following steps:

1. Install the `Migrate-Aaads` script from the PowerShell Gallery. This PowerShell migration script is a digitally signed by the Azure AD engineering team.

   ```
   Install-Script -Name Migrate-Aadds
   ```

2. Create a variable to hold the credentials for by the migration script using the Get-Credential cmdlet.

   The user account you specify needs *global administrator* privileges in your Azure AD tenant to enable Azure AD DS and then *Contributor* privileges in your Azure subscription to create the required Azure AD DS resources.

   When prompted, enter an appropriate user account and password:

   ```
   $creds = Get-Credential
   ```

3. Now run the `Migrate-Aadds` cmdlet using the *-Prepare* parameter. Provide the *-ManagedDomainFqdn* for your own managed domain, such as *aaddscontoso.com*:

```
Migrate-Aadds `
    -Prepare `
    -ManagedDomainFqdn aaddscontoso.com `
    -Credentials $creds
```

Migrate the managed domain

With the managed domain prepared and backed up, the domain can be migrated. This step recreates the Azure AD DS domain controller VMs using the Resource Manager deployment model. This step can take 1 to 3 hours to complete.

Run the `Migrate-Aadds` cmdlet using the *-Commit* parameter. Provide the *-ManagedDomainFqdn* for your own managed domain prepared in the previous section, such as *aaddscontoso.com*:

Specify the target resource group that contains the virtual network you want to migrate Azure AD DS to, such as *myResourceGroup*. Provide the target virtual network, such as *myVnet*, and the subnet, such as *DomainServices*.

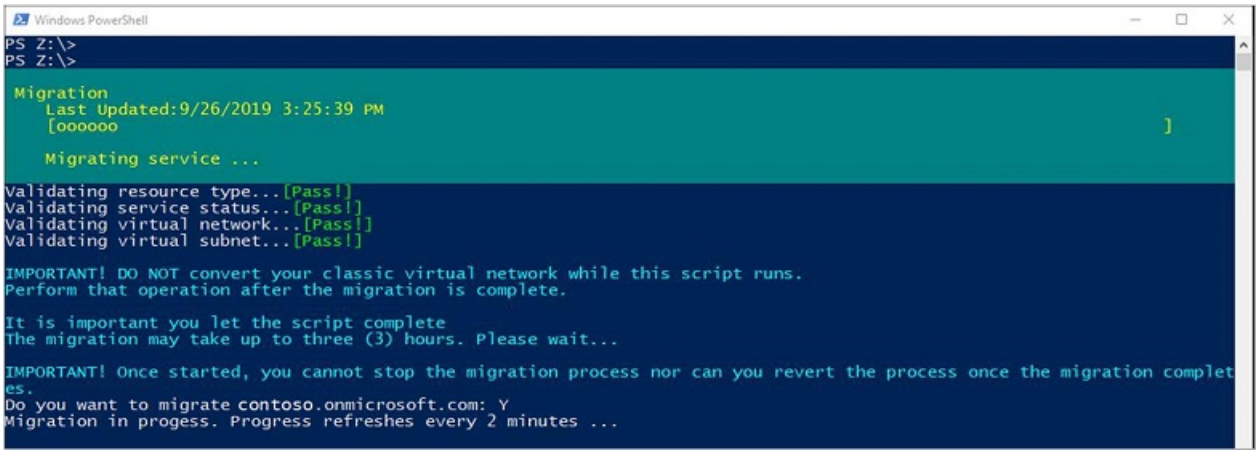
After this command runs, you can't then roll back:

```
Migrate-Aadds `
    -Commit `
    -ManagedDomainFqdn aaddscontoso.com `
    -VirtualNetworkResourceGroupName myResourceGroup `
    -VirtualNetworkName myVnet `
    -VirtualSubnetName DomainServices `
    -Credentials $creds
```

After the script validates the managed domain is prepared for migration, enter *Y* to start the migration process.

> **IMPORTANT**
> Don't convert the Classic virtual network to a Resource Manager virtual network during the migration process. If you convert the virtual network, you can't then rollback or restore the managed domain as the original virtual network won't exist anymore.

Every two minutes during the migration process, a progress indicator reports the current status, as shown in the following example output:



The migration process continues to run, even if you close out the PowerShell script. In the Azure portal, the status of the managed domain reports as *Migrating*.

When the migration successfully completes, you can view your first domain controller's IP address in the Azure portal or through Azure PowerShell. A time estimate on the second domain controller being available is also

shown.

At this stage, you can optionally move other existing resources from the Classic deployment model and virtual network. Or, you can keep the resources on the Classic deployment model and peer the virtual networks to each other after the Azure AD DS migration is complete.

Test and verify connectivity after the migration

It can take some time for the second domain controller to successfully deploy and be available for use in the managed domain.

With the Resource Manager deployment model, the network resources for the managed domain are shown in the Azure portal or Azure PowerShell. To learn more about what these network resources are and do, see Network resources used by Azure AD DS.

When at least one domain controller is available, complete the following configuration steps for network connectivity with VMs:

- **Update DNS server settings** To let other resources on the Resource Manager virtual network resolve and use the managed domain, update the DNS settings with the IP addresses of the new domain controllers. The Azure portal can automatically configure these settings for you.

  To learn more about how to configure the Resource Manager virtual network, see Update DNS settings for the Azure virtual network.

- **Restart domain-joined VMs** - As the DNS server IP addresses for the Azure AD DS domain controllers change, restart any domain-joined VMs so they then use the new DNS server settings. If applications or VMs have manually configured DNS settings, manually update them with the new DNS server IP addresses of the domain controllers that are shown in the Azure portal.

Now test the virtual network connection and name resolution. On a VM that's connected to the Resource Manager virtual network, or peered to it, try the following network communication tests:

1. Check if you can ping the IP address of one of the domain controllers, such as `ping 10.1.0.4`
   - The IP addresses of the domain controllers are shown on the **Properties** page for the managed domain in the Azure portal.
2. Verify name resolution of the managed domain, such as `nslookup aaddscontoso.com`
   - Specify the DNS name for your own managed domain to verify that the DNS settings are correct and resolves.

The second domain controller should be available 1-2 hours after the migration cmdlet finishes. To check if the second domain controller is available, look at the **Properties** page for the managed domain in the Azure portal. If two IP addresses shown, the second domain controller is ready.

Optional post-migration configuration steps

When the migration process is successfully complete, some optional configuration steps include enabling audit logs or e-mail notifications, or updating the fine-grained password policy.

Subscribe to audit logs using Azure Monitor

Azure AD DS exposes audit logs to help troubleshoot and view events on the domain controllers. For more information, see Enable and use audit logs.

You can use templates to monitor important information exposed in the logs. For example, the audit log workbook template can monitor possible account lockouts on the managed domain.

Configure email notifications

To be notified when a problem is detected on the managed domain, update the email notification settings in the Azure portal. For more information, see Configure notification settings.

Update fine-grained password policy

If needed, you can update the fine-grained password policy to be less restrictive than the default configuration. You can use the audit logs to determine if a less restrictive setting makes sense, then configure the policy as needed. Use the following high-level steps to review and update the policy settings for accounts that are repeatedly locked out after migration:

1. Configure password policy for fewer restrictions on the managed domain and observe the events in the audit logs.
2. If any service accounts are using expired passwords as identified in the audit logs, update those accounts with the correct password.
3. If a VM is exposed to the internet, review for generic account names like *administrator*, *user*, or *guest* with high sign-in attempts. Where possible, update those VMs to use less generically named accounts.
4. Use a network trace on the VM to locate the source of the attacks and block those IP addresses from being able to attempt sign-ins.
5. When there are minimal lockout issues, update the fine-grained password policy to be as restrictive as necessary.

Creating a network security group

Azure AD DS needs a network security group to secure the ports needed for the managed domain and block all other incoming traffic. This network security group acts as an extra layer of protection to lock down access to the managed domain, and isn't automatically created. To create the network security group and open the required ports, review the following steps:

1. In the Azure portal, select your Azure AD DS resource. On the overview page, a button is displayed to create a network security group if there's none associated with Azure AD Domain Services.
2. If you use secure LDAP, add a rule to the network security group to allow incoming traffic for *TCP* port *636*. For more information, see Configure secure LDAP.

Roll back and restore from migration

Up to a certain point in the migration process, you can choose to roll back or restore the managed domain.

Roll back

If there's an error when you run the PowerShell cmdlet to prepare for migration in step 2 or for the migration itself in step 3, the managed domain can roll back to the original configuration. This roll back requires the original Classic virtual network. The IP addresses may still change after rollback.

Run the `Migrate-Aadds` cmdlet using the *-Abort* parameter. Provide the *-ManagedDomainFqdn* for your own managed domain prepared in a previous section, such as *aaddscontoso.com*, and the Classic virtual network name, such as *myClassicVnet*.

```
Migrate-Aadds `
    -Abort `
    -ManagedDomainFqdn aaddscontoso.com `
    -ClassicVirtualNetworkName myClassicVnet `
    -Credentials $creds
```

Restore

As a last resort, Azure AD Domain Services can be restored from the last available backup. A backup is taken in step 1 of the migration to make sure that the most current backup is available. This backup is stored for 30 days.

To restore the managed domain from backup, open a support case ticket using the Azure portal. Provide your directory ID, domain name, and reason for restore. The support and restore process may take multiple days to complete.

Troubleshooting

If you have problems after migration to the Resource Manager deployment model, review some of the following common troubleshooting areas:

- Troubleshoot domain-join problems
- Troubleshoot account lockout problems
- Troubleshoot account sign-in problems
- Troubleshoot secure LDAP connectivity problems

Next steps

With your managed domain migrated to the Resource Manager deployment model, create and domain-join a Windows VM and then install management tools.

Change the SKU for an existing Azure Active Directory Domain Services managed domain

7/20/2020 • 2 minutes to read • Edit Online

In Azure Active Directory Domain Services (Azure AD DS), the available performance and features are based on the SKU type. These feature differences include the backup frequency or maximum number of one-way outbound forest trusts (currently in preview).

You select a SKU when you create the managed domain, and you can switch SKUs up or down as your business needs change after the managed domain has been deployed. Changes in business requirements could include the need for more frequent backups or to create additional forest trusts. For more information on the limits and pricing of the different SKUs, see Azure AD DS SKU concepts and Azure AD DS pricing pages.

This article shows you how to change the SKU for an existing Azure AD DS managed domain using the Azure portal.

Before you begin

To complete this article, you need the following resources and privileges:

- An active Azure subscription.
  - If you don't have an Azure subscription, create an account.
- An Azure Active Directory tenant associated with your subscription, either synchronized with an on-premises directory or a cloud-only directory.
  - If needed, create an Azure Active Directory tenant or associate an Azure subscription with your account.
- An Azure Active Directory Domain Services managed domain enabled and configured in your Azure AD tenant.
  - If needed, complete the tutorial to create and configure a managed domain.

SKU change limitations

You can change SKUs up or down after the managed domain has been deployed. However, if you use a resource forest (currently in preview) and have created one-way outbound forest trusts from Azure AD DS to an on-premises AD DS environment, there are some limitations for the SKU change operation. The *Premium* and *Enterprise* SKUs define a limit on the number of trusts you can create. You can't change to a SKU with a lower maximum limit than you currently have configured.

For example:

- If you have created two forest trusts on the *Premium* SKU, you can't change down to the *Standard* SKU. The *Standard* SKU doesn't support forest trusts.
- Or, if you have created seven trusts on the *Premium* SKU, you can't change down to the *Enterprise* SKU. The *Enterprise* SKU supports a maximum of five trusts.

For more information on these limits, see Azure AD DS SKU features and limits.

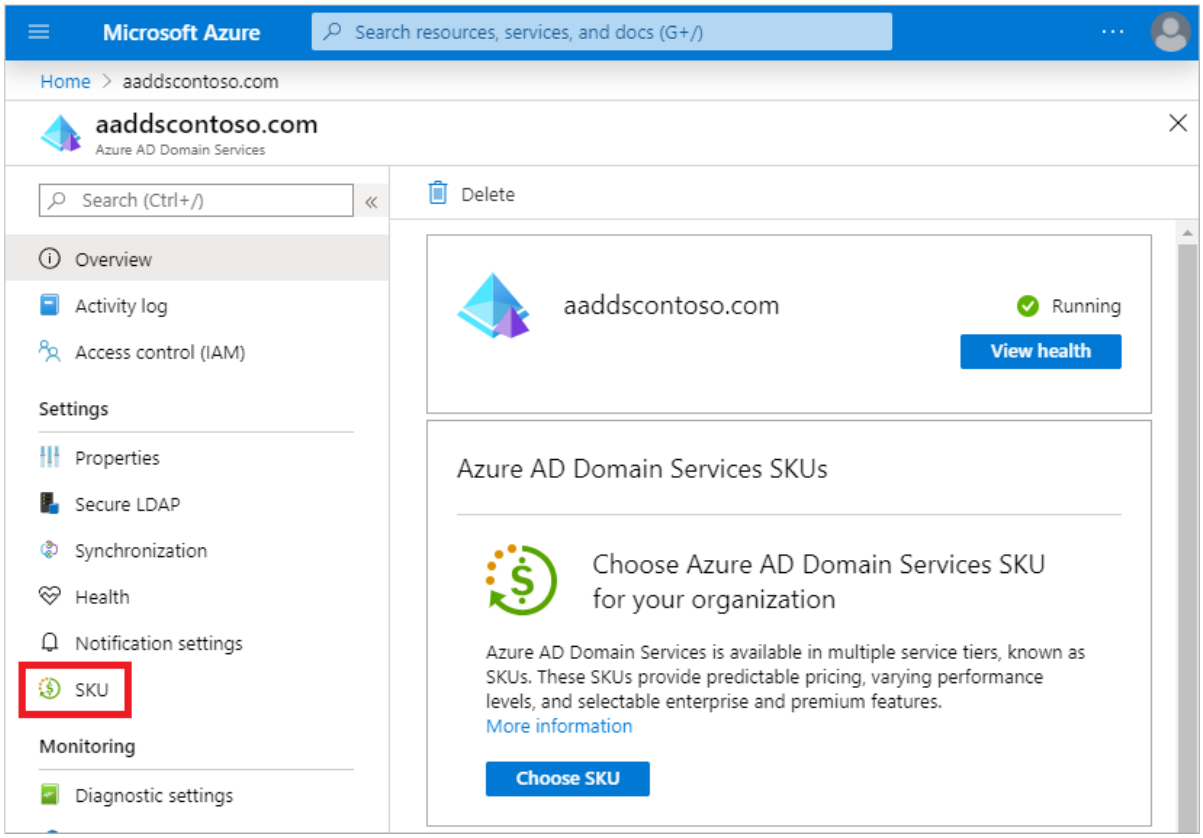
Select a new SKU

To change the SKU for a managed domain using the Azure portal, complete the following steps:

1. At the top of the Azure portal, search for and select **Azure AD Domain Services**. Choose your managed

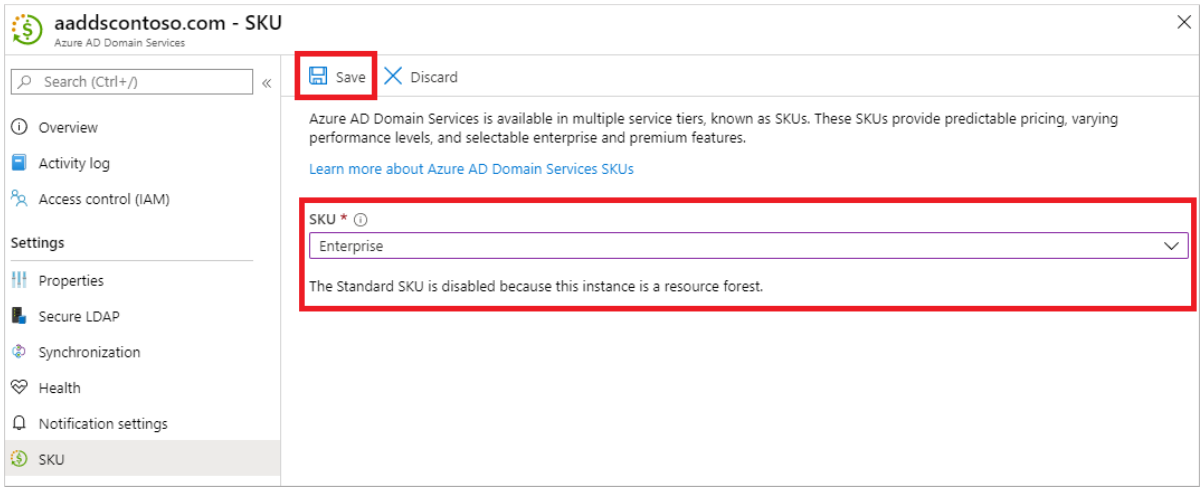
domain from the list, such as *aaddscontoso.com*.

2. In the menu on the left-hand side of the Azure AD DS page, select **Settings > SKU**.



3. From the drop-down menu, select the SKU you wish for your managed domain. If you have a resource forest, you can't select *Standard* SKU as forest trusts are only available on the *Enterprise* SKU or higher.

   Choose the SKU you want from the drop-down menu, then select **Save**.



It can take a minute or two to change the SKU type.

Next steps

If you have a resource forest and want to create additional trusts after the SKU change, see Create an outbound forest trust to an on-premises domain in Azure AD DS (preview).

Disable weak ciphers and password hash synchronization to secure an Azure Active Directory Domain Services managed domain

7/20/2020 • 2 minutes to read • Edit Online

By default, Azure Active Directory Domain Services (Azure AD DS) enables the use of ciphers such as NTLM v1 and TLS v1. These ciphers may be required for some legacy applications, but are considered weak and can be disabled if you don't need them. If you have on-premises hybrid connectivity using Azure AD Connect, you can also disable the synchronization of NTLM password hashes.

This article shows you how to disable NTLM v1 and TLS v1 ciphers and disable NTLM password hash synchronization.

Prerequisites

To complete this article, you need the following resources:

- An active Azure subscription.
  - If you don't have an Azure subscription, create an account.
- An Azure Active Directory tenant associated with your subscription, either synchronized with an on-premises directory or a cloud-only directory.
  - If needed, create an Azure Active Directory tenant or associate an Azure subscription with your account.
- An Azure Active Directory Domain Services managed domain enabled and configured in your Azure AD tenant.
  - If needed, create and configure an Azure Active Directory Domain Services managed domain.
- Install and configure Azure PowerShell.
  - If needed, follow the instructions to install the Azure PowerShell module and connect to your Azure subscription.
  - Make sure that you sign in to your Azure subscription using the Connect-AzAccount cmdlet.
- Install and configure Azure AD PowerShell.
  - If needed, follow the instructions to install the Azure AD PowerShell module and connect to Azure AD.
  - Make sure that you sign in to your Azure AD tenant using the Connect-AzureAD cmdlet.

Disable weak ciphers and NTLM password hash sync

To disable weak cipher suites and NTLM credential hash synchronization, sign in to your Azure account, then get the Azure AD DS resource using the Get-AzResource cmdlet:

> **TIP**
> If you receive an error using the Get-AzResource command that the *Microsoft.AAD/DomainServices* resource doesn't exist, elevate your access to manage all Azure subscriptions and management groups.

```
Login-AzAccount

$DomainServicesResource = Get-AzResource -ResourceType "Microsoft.AAD/DomainServices"
```

Next, define *DomainSecuritySettings* to configure the following security options:

1. Disable NTLM v1 support.
2. Disable the synchronization of NTLM password hashes from your on-premises AD.
3. Disable TLS v1.

> **IMPORTANT**
> Users and service accounts can't perform LDAP simple binds if you disable NTLM password hash synchronization in the Azure AD DS managed domain. If you need to perform LDAP simple binds, don't set the *"SyncNtlmPasswords"="Disabled";* security configuration option in the following command.

```
$securitySettings =
@{"DomainSecuritySettings"=@{"NtlmV1"="Disabled";"SyncNtlmPasswords"="Disabled";"TlsV1"="Disabled"}}
```

Finally, apply the defined security settings to the managed domain using the Set-AzResource cmdlet. Specify the Azure AD DS resource from the first step, and the security settings from the previous step.

```
Set-AzResource -Id $DomainServicesResource.ResourceId -Properties $securitySettings -Verbose -Force
```

It takes a few moments for the security settings to be applied to the managed domain.

Next steps

To learn more about the synchronization process, see How objects and credentials are synchronized in a managed domain.

Configure Kerberos constrained delegation (KCD) in Azure Active Directory Domain Services

7/20/2020 • 4 minutes to read • Edit Online

As you run applications, there may be a need for those applications to access resources in the context of a different user. Active Directory Domain Services (AD DS) supports a mechanism called *Kerberos delegation* that enables this use-case. Kerberos *constrained* delegation (KCD) then builds on this mechanism to define specific resources that can be accessed in the context of the user.

Azure Active Directory Domain Services (Azure AD DS) managed domains are more securely locked down than traditional on-premises AD DS environments, so use a more secure *resource-based* KCD.

This article shows you how to configure resource-based Kerberos constrained delegation in an Azure AD DS managed domain.

Prerequisites

To complete this article, you need the following resources:

- An active Azure subscription.
  - If you don't have an Azure subscription, create an account.
- An Azure Active Directory tenant associated with your subscription, either synchronized with an on-premises directory or a cloud-only directory.
  - If needed, create an Azure Active Directory tenant or associate an Azure subscription with your account.
- An Azure Active Directory Domain Services managed domain enabled and configured in your Azure AD tenant.
  - If needed, create and configure an Azure Active Directory Domain Services managed domain.
- A Windows Server management VM that is joined to the Azure AD DS managed domain.
  - If needed, complete the tutorial to create a Windows Server VM and join it to a managed domain then install the AD DS management tools.
- A user account that's a member of the *Azure AD DC administrators* group in your Azure AD tenant.

Kerberos constrained delegation overview

Kerberos delegation lets one account impersonate another account to access resources. For example, a web application that accesses a back-end web component can impersonate itself as a different user account when it makes the back-end connection. Kerberos delegation is insecure as it doesn't limit what resources the impersonating account can access.

Kerberos *constrained* delegation (KCD) restricts the services or resources that a specified server or application can connect when impersonating another identity. Traditional KCD requires domain administrator privileges to configure a domain account for a service, and it restricts the account to run on a single domain.

Traditional KCD also has a few issues. For example, in earlier operating systems, the service administrator had no useful way to know which front-end services delegated to the resource services they owned. Any front-end service that could delegate to a resource service was a potential attack point. If a server that hosted a front-end service configured to delegate to resource services was compromised, the resource services could also be compromised.

In a managed domain, you don't have domain administrator privileges. As a result, traditional account-based KCD can't be configured in a managed domain. Resource-based KCD can instead be used, which is also more secure.

Resource-based KCD

Windows Server 2012 and later gives service administrators the ability to configure constrained delegation for their service. This model is known as resource-based KCD. With this approach, the back-end service administrator can allow or deny specific front-end services from using KCD.

Resource-based KCD is configured using PowerShell. You use the Set-ADComputer or Set-ADUser cmdlets, depending on whether the impersonating account is a computer account or a user account / service account.

Configure resource-based KCD for a computer account

In this scenario, let's assume you have a web app that runs on the computer named *contoso-webapp.aaddscontoso.com*.

The web app needs to access a web API that runs on the computer named *contoso-api.aaddscontoso.com* in the context of domain users.

Complete the following steps to configure this scenario:

1. Create a custom OU. You can delegate permissions to manage this custom OU to users within the managed domain.
2. Domain-join the virtual machines, both the one that runs the web app, and the one that runs the web API, to the managed domain. Create these computer accounts in the custom OU from the previous step.

   > **NOTE**
   > The computer accounts for the web app and the web API must be in a custom OU where you have permissions to configure resource-based KCD. You can't configure resource-based KCD for a computer account in the built-in *AAD DC Computers* container.

3. Finally, configure resource-based KCD using the Set-ADComputer PowerShell cmdlet.

   From your domain-joined management VM and logged in as user account that's a member of the *Azure AD DC administrators* group, run the following cmdlets. Provide your own computer names as needed:

   ```
   $ImpersonatingAccount = Get-ADComputer -Identity contoso-webapp.aaddscontoso.com
   Set-ADComputer contoso-api.aaddscontoso.com -PrincipalsAllowedToDelegateToAccount $ImpersonatingAccount
   ```

Configure resource-based KCD for a user account

In this scenario, let's assume you have a web app that runs as a service account named *appsvc*. The web app needs to access a web API that runs as a service account named *backendsvc* in the context of domain users. Complete the following steps to configure this scenario:

1. Create a custom OU. You can delegate permissions to manage this custom OU to users within the managed domain.
2. Domain-join the virtual machines that run the backend web API/resource to the managed domain. Create its computer account within the custom OU.
3. Create the service account (for example, *appsvc*) used to run the web app within the custom OU.

> **NOTE**
>
> Again, the computer account for the web API VM, and the service account for the web app, must be in a custom OU where you have permissions to configure resource-based KCD. You can't configure resource-based KCD for accounts in the built-in *AAD DC Computers* or *AAD DC Users* containers. This also means that you can't use user accounts synchronized from Azure AD to set up resource-based KCD. You must create and use service accounts specifically created in Azure AD DS.

4. Finally, configure resource-based KCD using the Set-ADUser PowerShell cmdlet.

   From your domain-joined management VM and logged in as user account that's a member of the *Azure AD DC administrators* group, run the following cmdlets. Provide your own service names as needed:

   ```
   $ImpersonatingAccount = Get-ADUser -Identity appsvc
   Set-ADUser backendsvc -PrincipalsAllowedToDelegateToAccount $ImpersonatingAccount
   ```

Next steps

To learn more about how delegation works in Active Directory Domain Services, see Kerberos Constrained Delegation Overview.

Password and account lockout policies on Active Directory Domain Services managed domains

7/20/2020 • 6 minutes to read • Edit Online

To manage user security in Azure Active Directory Domain Services (Azure AD DS), you can define fine-grained password policies that control account lockout settings or minimum password length and complexity. A default fine grained password policy is created and applied to all users in an Azure AD DS managed domain. To provide granular control and meet specific business or compliance needs, additional policies can be created and applied to specific groups of users.

This article shows you how to create and configure a fine-grained password policy in Azure AD DS using the Active Directory Administrative Center.

> **NOTE**
> Password policies are only available for managed domains created using the Resource Manager deployment model. For older managed domains created using Classic, migrate from the Classic virtual network model to Resource Manager.

Before you begin

To complete this article, you need the following resources and privileges:

- An active Azure subscription.
  - If you don't have an Azure subscription, create an account.
- An Azure Active Directory tenant associated with your subscription, either synchronized with an on-premises directory or a cloud-only directory.
  - If needed, create an Azure Active Directory tenant or associate an Azure subscription with your account.
- An Azure Active Directory Domain Services managed domain enabled and configured in your Azure AD tenant.
  - If needed, complete the tutorial to create and configure an Azure Active Directory Domain Services managed domain.
  - The managed domain must have been created using the Resource Manager deployment model. If needed, Migrate from the Classic virtual network model to Resource Manager.
- A Windows Server management VM that is joined to the managed domain.
  - If needed, complete the tutorial to create a management VM.
- A user account that's a member of the *Azure AD DC administrators* group in your Azure AD tenant.

Default password policy settings

Fine-grained password policies (FGPPs) let you apply specific restrictions for password and account lockout policies to different users in a domain. For example, to secure privileged accounts you can apply stricter account lockout settings than regular non-privileged accounts. You can create multiple FGPPs within a managed domain and specify the order of priority to apply them to users.

For more information about password policies and using the Active Directory Administration Center, see the following articles:

- Learn about fine-grained password policies

- Configure fine-grained password policies using AD Administration Center

Policies are distributed through group association in a managed domain, and any changes you make are applied at the next user sign-in. Changing the policy doesn't unlock a user account that's already locked out.

Password policies behave a little differently depending on how the user account they're applied to was created. There are two ways a user account can be created in Azure AD DS:

- The user account can be synchronized in from Azure AD. This includes cloud-only user accounts created directly in Azure, and hybrid user accounts synchronized from an on-premises AD DS environment using Azure AD Connect.
  - The majority of user accounts in Azure AD DS are created through the synchronization process from Azure AD.
- The user account can be manually created in a managed domain, and doesn't exist in Azure AD.

All users, regardless of how they're created, have the following account lockout policies applied by the default password policy in Azure AD DS:

- **Account lockout duration:** 30
- **Number of failed logon attempts allowed:** 5
- **Reset failed logon attempts count after:** 30 minutes
- **Maximum password age (lifetime):** 90 days

With these default settings, user accounts are locked out for 30 minutes if five invalid passwords are used within 2 minutes. Accounts are automatically unlocked after 30 minutes.

Account lockouts only occur within the managed domain. User accounts are only locked out in Azure AD DS, and only due to failed sign-in attempts against the managed domain. User accounts that were synchronized in from Azure AD or on-premises aren't locked out in their source directories, only in Azure AD DS.

If you have an Azure AD password policy that specifies a maximum password age greater than 90 days, that password age is applied to the default policy in Azure AD DS. You can configure a custom password policy to define a different maximum password age in Azure AD DS. Take care if you have a shorter maximum password age configured in an Azure AD DS password policy than in Azure AD or an on-premises AD DS environment. In that scenario, a user's password may expire in Azure AD DS before they're prompted to change in Azure AD or an on-premises AD DS environment.

For user accounts created manually in a managed domain, the following additional password settings are also applied from the default policy. These settings don't apply to user accounts synchronized in from Azure AD, as a user can't update their password directly in Azure AD DS.

- **Minimum password length (characters):** 7
- **Passwords must meet complexity requirements**

You can't modify the account lockout or password settings in the default password policy. Instead, members of the *AAD DC Administrators* group can create custom password policies and configure it to override (take precedence over) the default built-in policy, as shown in the next section.

Create a custom password policy

As you build and run applications in Azure, you may want to configure a custom password policy. For example, you could create a policy to set different account lockout policy settings.

Custom password policies are applied to groups in a managed domain. This configuration effectively overrides the default policy.

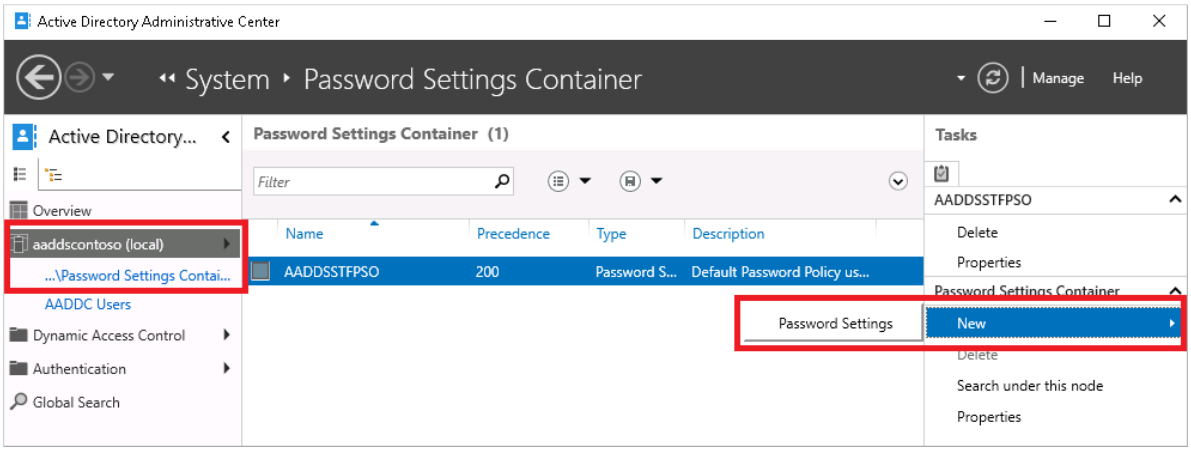
To create a custom password policy, you use the Active Directory Administrative Tools from a domain-joined VM.

The Active Directory Administrative Center lets you view, edit, and create resources in a managed domain, including OUs.

> **NOTE**
> To create a custom password policy in a managed domain, you must be signed in to a user account that's a member of the *AAD DC Administrators* group.

1. From the Start screen, select **Administrative Tools**. A list of available management tools is shown that were installed in the tutorial to create a management VM.
2. To create and manage OUs, select **Active Directory Administrative Center** from the list of administrative tools.
3. In the left pane, choose your managed domain, such as *aaddscontoso.com*.
4. Open the **System** container, then the **Password Settings Container**.

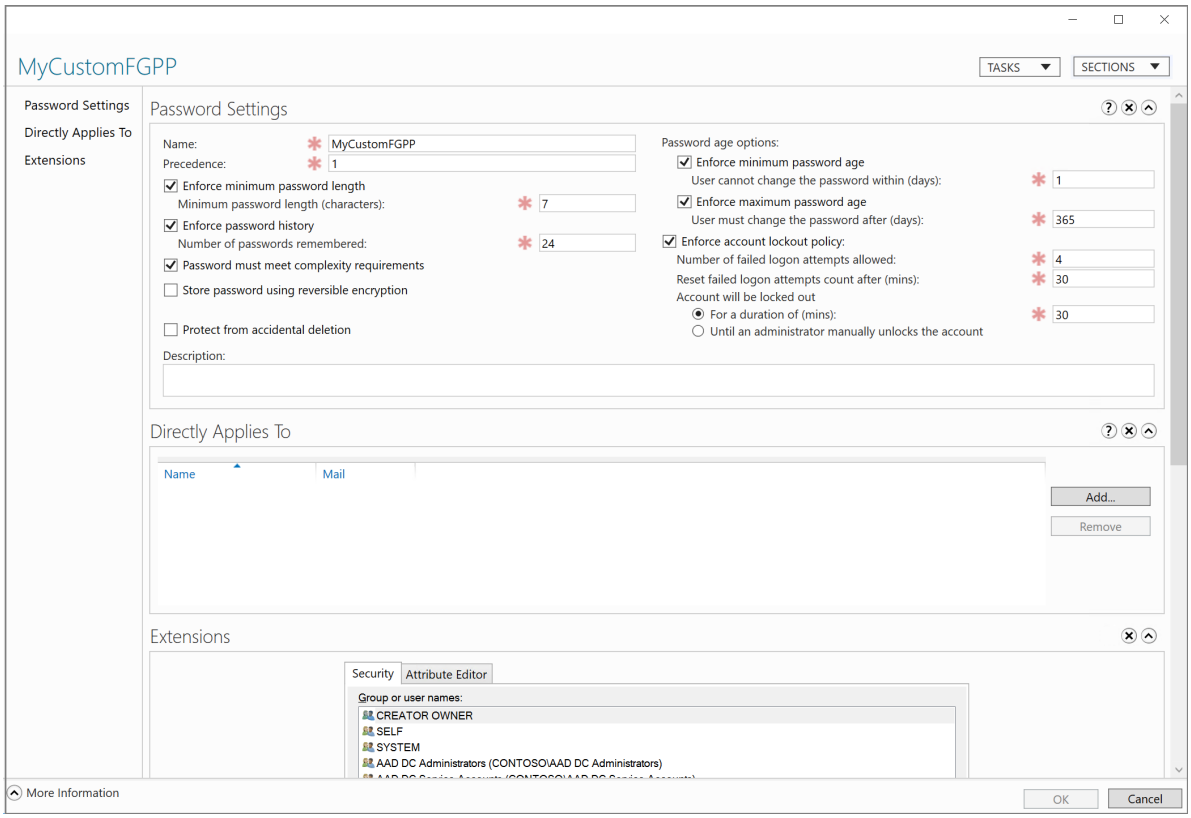
   A built-in password policy for the managed domain is shown. You can't modify this built-in policy. Instead, create a custom password policy to override the default policy.



5. In the **Tasks** panel on the right, select **New** > **Password Settings**.
6. In the **Create Password Settings** dialog, enter a name for the policy, such as *MyCustomFGPP*.
7. When multiple password policies exist, the policy with the highest precedence, or priority, is applied to a user. The lower the number, the higher the priority. The default password policy has a priority of *200*.

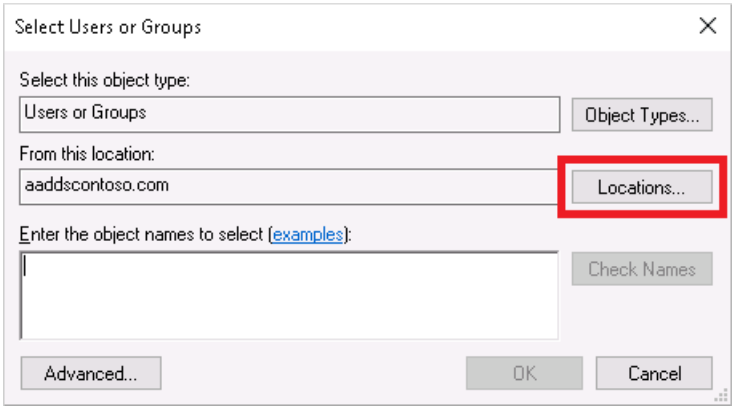
   Set the precedence for your custom password policy to override the default, such as *1*.
8. Edit other password policy settings as desired. Remember the following key points:

   - Settings like password complexity, age, or expiration time only to users manually created in a managed domain.
   - Account lockout settings apply to all users, but only take effect within the managed domain and not in Azure AD itself.

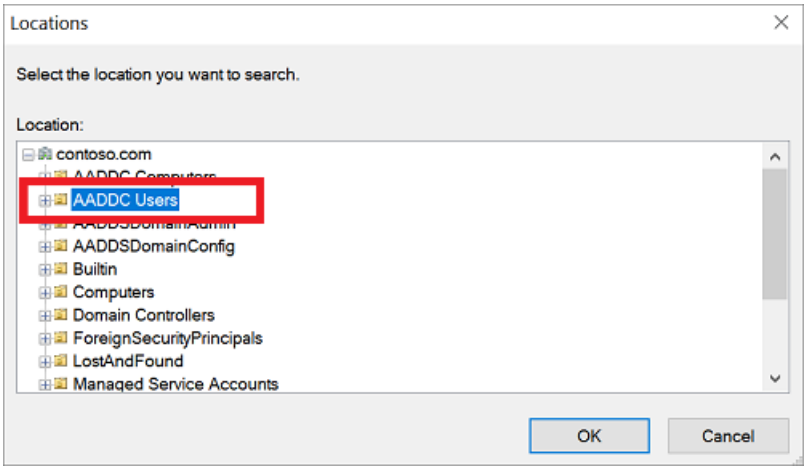


9. Uncheck **Protect from accidental deletion**. If this option is selected, you can't save the FGPP.

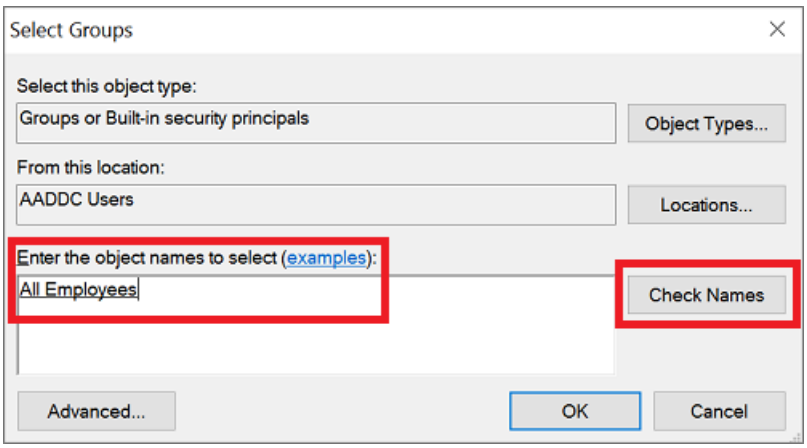
10. In the **Directly Applies To** section, select the **Add** button. In the **Select Users or Groups** dialog, select the **Locations** button.



11. Password policies can only be applied to groups. In the **Locations** dialog, expand the domain name, such as *aaddscontoso.com*, then select an OU, such as **AADDC Users**. If you have a custom OU that contains a group of users you wish to apply, select that OU.



12. Type the name of the group you wish to apply the policy to, then select **Check Names** to validate that the group exists.



13. With the name of the group you selected now displayed in **Directly Applies To** section, select **OK** to save your custom password policy.

Next steps

For more information about password policies and using the Active Directory Administration Center, see the following articles:

- Learn about fine-grained password policies
- Configure fine-grained password policies using AD Administration Center

Enable security audits for Azure Active Directory Domain Services

7/20/2020 • 10 minutes to read • Edit Online

Azure Active Directory Domain Services (Azure AD DS) security audits lets Azure stream security events to targeted resources. These resources include Azure Storage, Azure Log Analytics workspaces, or Azure Event Hub. After you enable security audit events, Azure AD DS sends all the audited events for the selected category to the targeted resource.

You can archive events into Azure storage and stream events into security information and event management (SIEM) software (or equivalent) using Azure Event Hubs, or do your own analysis and using Azure Log Analytics workspaces from the Azure portal.

> **IMPORTANT**
> Azure AD DS security audits are only available for Azure Resource Manager-based managed domains. For information on how to migrate, see Migrate Azure AD DS from the Classic virtual network model to Resource Manager.

Security audit destinations

You can use Azure Storage, Azure Event Hubs, or Azure Log Analytics workspaces as a target resource for Azure AD DS security audits. These destinations can be combined. For example, you could use Azure Storage for archiving security audit events, but an Azure Log Analytics workspace to analyze and report on the information in the short term.

The following table outlines scenarios for each destination resource type.

> **IMPORTANT**
> You need to create the target resource before you enable Azure AD DS security audits. You can create these resources using the Azure portal, Azure PowerShell, or the Azure CLI.

TARGET RESOURCE	SCENARIO
Azure Storage	This target should be used when your primary need is to store security audit events for archival purposes. Other targets can be used for archival purposes, however those targets provide capabilities beyond the primary need of archiving. Before you enable Azure AD DS security audit events, first Create an Azure Storage account.
Azure Event Hubs	This target should be used when your primary need is to share security audit events with additional software such as data analysis software or security information & event management (SIEM) software. Before you enable Azure AD DS security audit events, Create an event hub using Azure portal

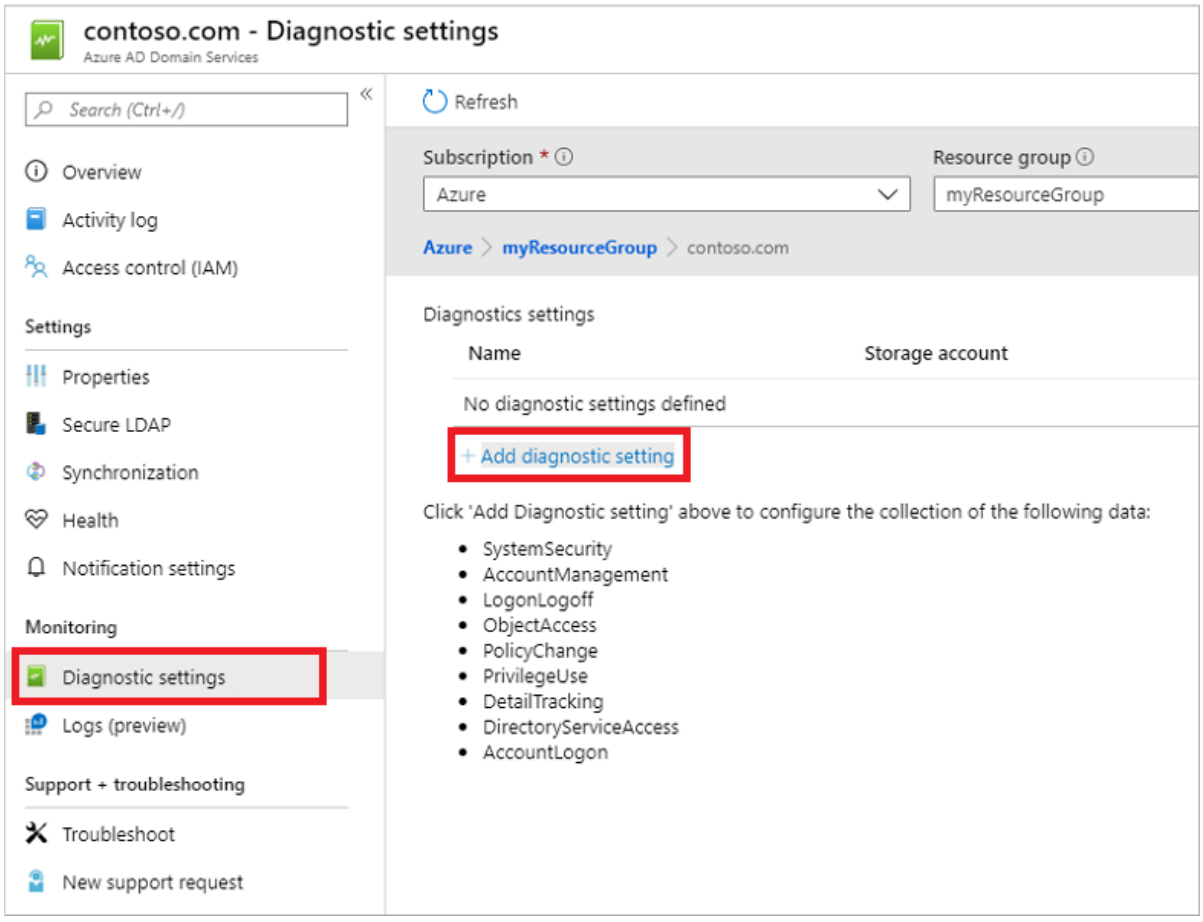
TARGET RESOURCE	SCENARIO
Azure Log Analytics Workspace	This target should be used when your primary need is to analyze and review secure audits from the Azure portal directly. Before you enable Azure AD DS security audit events, Create a Log Analytics workspace in the Azure portal.

Enable security audit events using the Azure portal

To enable Azure AD DS security audit events using the Azure portal, complete the following steps.

> **IMPORTANT**
> Azure AD DS security audits aren't retroactive. You can't retrieve or replay events from the past. Azure AD DS can only send events that occur after security audits are enabled.

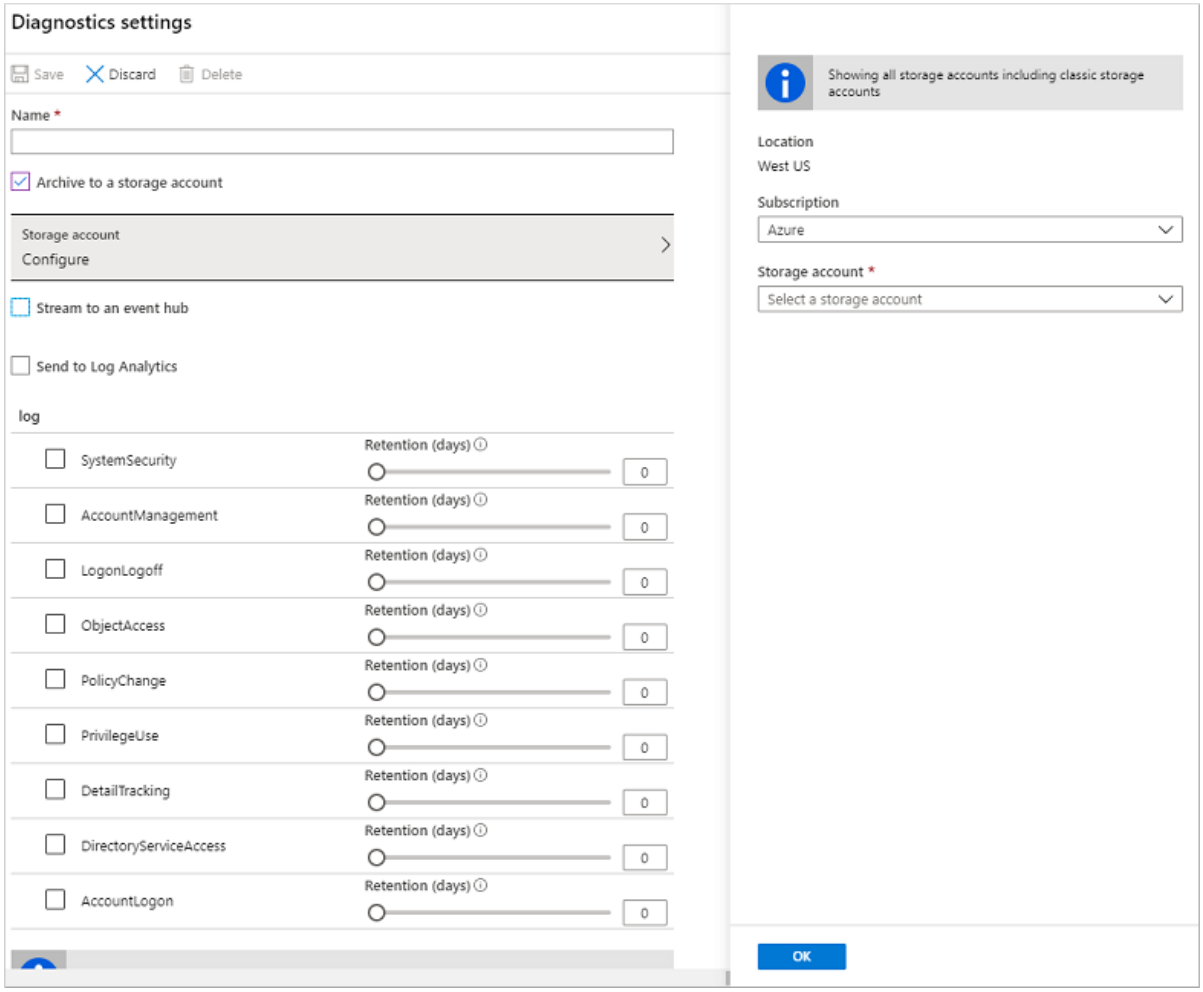
1. Sign in to the Azure portal at https://portal.azure.com.
2. At the top of the Azure portal, search for and select **Azure AD Domain Services**. Choose your managed domain, such as *aaddscontoso.com*.
3. In the Azure AD DS window, select **Diagnostic settings** on the left-hand side.
4. No diagnostics are configured by default. To get started, select **Add diagnostic setting**.



5. Enter a name for the diagnostic configuration, such as *aadds-auditing*.

   Check the box for the security audit destination you want. You can choose from an Azure Storage account, an Azure event hub, or a Log Analytics workspace. These destination resources must already exist in your

Azure subscription. You can't create the destination resources in this wizard.



- **Azure storage**
  - Select **Archive to a storage account**, then choose **Configure**.
  - Select the **Subscription** and the **Storage account** you want to use to archive security audit events.
  - When ready, choose **OK**.
- **Azure event hubs**
  - Select **Stream to an event hub**, then choose **Configure**.
  - Select the **Subscription** and the **Event hub namespace**. If needed, also choose an **Event hub name** and then **Event hub policy name**.
  - When ready, choose **OK**.
- **Azure Log Analytic workspaces**
  - Select **Send to Log Analytics**, then choose the **Subscription** and **Log Analytics Workspace** you want to use to store security audit events.

6. Select the log categories you want included for the particular target resource. If you send the audit events to an Azure Storage account, you can also configure a retention policy that defines the number of days to retain data. A default setting of *0* retains all data and doesn't rotate events after a period of time.

   You can select different log categories for each targeted resource within a single configuration. This ability lets you choose which logs categories you want to keep for Log Analytics and which logs categories your want to archive, for example.

7. When done, select **Save** to commit your changes. The target resources start to receive Azure AD DS security audit events soon after the configuration is saved.

Enable security audit events using Azure PowerShell

To enable Azure AD DS security audit events using Azure PowerShell, complete the following steps. If needed, first install the Azure PowerShell module and connect to your Azure subscription.

> **IMPORTANT**
> Azure AD DS security audits aren't retroactive. You can't retrieve or replay events from the past. Azure AD DS can only send events that occur after security audits are enabled.

1. Authenticate to your Azure subscription using the Connect-AzAccount cmdlet. When prompted, enter your account credentials.

   ```
   Connect-AzAccount
   ```

2. Create the target resource for the security audit events.

   - **Azure storage** - Create a storage account using Azure PowerShell
   - **Azure event hubs** - Create an event hub using Azure PowerShell. You may also need to use the New-AzEventHubAuthorizationRule cmdlet to create an authorization rule that grants Azure AD DS permissions to the event hub *namespace*. The authorization rule must include the **Manage**, **Listen**, and **Send** rights.

     > **IMPORTANT**
     > Ensure you set the authorization rule on the event hub namespace and not the event hub itself.

   - **Azure Log Analytic workspaces** - Create a Log Analytics workspace with Azure PowerShell.

3. Get the resource ID for your Azure AD DS managed domain using the Get-AzResource cmdlet. Create a variable named *$aadds.ResourceId* to hold the value:

   ```
   $aadds = Get-AzResource -name aaddsDomainName
   ```

4. Configure the Azure Diagnostic settings using the Set-AzDiagnosticSetting cmdlet to use the target resource for Azure AD Domain Services security audit events. In the following examples, the variable *$aadds.ResourceId* is used from the previous step.

   - **Azure storage** - Replace *storageAccountId* with your storage account name:

     ```
     Set-AzDiagnosticSetting `
         -ResourceId $aadds.ResourceId `
         -StorageAccountId storageAccountId `
         -Enabled $true
     ```

   - **Azure event hubs** - Replace *eventHubName* with the name of your event hub and *eventHubRuleId* with your authorization rule ID:

     ```
     Set-AzDiagnosticSetting -ResourceId $aadds.ResourceId `
         -EventHubName eventHubName `
         -EventHubAuthorizationRuleId eventHubRuleId `
         -Enabled $true
     ```

- **Azure Log Analytic workspaces** - Replace *workspaceId* with the ID of the Log Analytics workspace:

```
Set-AzureRmDiagnosticSetting -ResourceId $aadds.ResourceId `
    -WorkspaceID workspaceId `
    -Enabled $true
```

Query and view security audit events using Azure Monitor

Log Analytic workspaces let you view and analyze the security audit events using Azure Monitor and the Kusto query language. This query language is designed for read-only use that boasts power analytic capabilities with an easy-to-read syntax. For more information to get started with Kusto query languages, see the following articles:

- Azure Monitor documentation
- Get started with Log Analytics in Azure Monitor
- Get started with log queries in Azure Monitor
- Create and share dashboards of Log Analytics data

The following sample queries can be used to start analyzing security audit events from Azure AD DS.

Sample query 1

View all the account lockout events for the last seven days:

```
AADDomainServicesAccountManagement
| where TimeGenerated >= ago(7d)
| where OperationName has "4740"
```

Sample query 2

View all the account lockout events (*4740*) between June 3, 2020 at 9 a.m. and June 10, 2020 midnight, sorted ascending by the date and time:

```
AADDomainServicesAccountManagement
| where TimeGenerated >= datetime(2020-06-03 09:00) and TimeGenerated <= datetime(2020-06-10)
| where OperationName has "4740"
| sort by TimeGenerated asc
```

Sample query 3

View account sign-in events seven days ago (from now) for the account named user:

```
AADDomainServicesAccountLogon
| where TimeGenerated >= ago(7d)
| where "user" == tolower(extract("Logon Account:\t(.+[0-9A-Za-z])",1,tostring(ResultDescription)))
```

Sample query 4

View account sign-in events seven days ago from now for the account named user that attempted to sign in using a bad password (*0xC0000006a*):

```
AADDomainServicesAccountLogon
| where TimeGenerated >= ago(7d)
| where "user" == tolower(extract("Logon Account:\t(.+[0-9A-Za-z])",1,tostring(ResultDescription)))
| where "0xc000006a" == tolower(extract("Error Code:\t(.+[0-9A-Za-z])",1,tostring(ResultDescription)))
```

Sample query 5

View account sign-in events seven days ago from now for the account named user that attempted to sign in while the account was locked out (*0xC0000234*):

```
AADDomainServicesAccountLogon
| where TimeGenerated >= ago(7d)
| where "user" == tolower(extract("Logon Account:\t(.+[0-9A-Za-z])",1,tostring(ResultDescription)))
| where "0xc0000234" == tolower(extract("Error Code:\t(.+[0-9A-Za-z])",1,tostring(ResultDescription)))
```

Sample query 6

View the number of account sign-in events seven days ago from now for all sign-in attempts that occurred for all locked out users:

```
AADDomainServicesAccountLogon
| where TimeGenerated >= ago(7d)
| where "0xc0000234" == tolower(extract("Error Code:\t(.+[0-9A-Za-z])",1,tostring(ResultDescription)))
| summarize count()
```

Audit event categories

Azure AD DS security audits align with traditional auditing for traditional AD DS domain controllers. In hybrid environments, you can reuse existing audit patterns so the same logic may be used when analyzing the events. Depending on the scenario you need to troubleshoot or analyze, the different audit event categories need to be targeted.

The following audit event categories are available:

AUDIT CATEGORY NAME	DESCRIPTION
Account Logon	Audits attempts to authenticate account data on a domain controller or on a local Security Accounts Manager (SAM). Logon and Logoff policy settings and events track attempts to access a particular computer. Settings and events in this category focus on the account database that is used. This category includes the following subcategories: • Audit Credential Validation • Audit Kerberos Authentication Service • Audit Kerberos Service Ticket Operations • Audit Other Logon/Logoff Events
Account Management	Audits changes to user and computer accounts and groups. This category includes the following subcategories: • Audit Application Group Management • Audit Computer Account Management • Audit Distribution Group Management • Audit Other Account Management • Audit Security Group Management • Audit User Account Management

AUDIT CATEGORY NAME	DESCRIPTION
Detail Tracking	Audits activities of individual applications and users on that computer, and to understand how a computer is being used. This category includes the following subcategories: • Audit DPAPI Activity • Audit PNP activity • Audit Process Creation • Audit Process Termination • Audit RPC Events
Directory Services Access	Audits attempts to access and modify objects in Active Directory Domain Services (AD DS). These audit events are logged only on domain controllers. This category includes the following subcategories: • Audit Detailed Directory Service Replication • Audit Directory Service Access • Audit Directory Service Changes • Audit Directory Service Replication
Logon-Logoff	Audits attempts to log on to a computer interactively or over a network. These events are useful for tracking user activity and identifying potential attacks on network resources. This category includes the following subcategories: • Audit Account Lockout • Audit User/Device Claims • Audit IPsec Extended Mode • Audit Group Membership • Audit IPsec Main Mode • Audit IPsec Quick Mode • Audit Logoff • Audit Logon • Audit Network Policy Server • Audit Other Logon/Logoff Events • Audit Special Logon
Object Access	Audits attempts to access specific objects or types of objects on a network or computer. This category includes the following subcategories: • Audit Application Generated • Audit Certification Services • Audit Detailed File Share • Audit File Share • Audit File System • Audit Filtering Platform Connection • Audit Filtering Platform Packet Drop • Audit Handle Manipulation • Audit Kernel Object • Audit Other Object Access Events • Audit Registry • Audit Removable Storage • Audit SAM • Audit Central Access Policy Staging

AUDIT CATEGORY NAME	DESCRIPTION
Policy Change	Audits changes to important security policies on a local system or network. Policies are typically established by administrators to help secure network resources. Monitoring changes or attempts to change these policies can be an important aspect of security management for a network. This category includes the following subcategories: • Audit Audit Policy Change • Audit Authentication Policy Change • Audit Authorization Policy Change • Audit Filtering Platform Policy Change • Audit MPSSVC Rule-Level Policy Change • Audit Other Policy Change
Privilege Use	Audits the use of certain permissions on one or more systems. This category includes the following subcategories: • Audit Non-Sensitive Privilege Use • Audit Sensitive Privilege Use • Audit Other Privilege Use Events
System	Audits system-level changes to a computer not included in other categories and that have potential security implications. This category includes the following subcategories: • Audit IPsec Driver • Audit Other System Events • Audit Security State Change • Audit Security System Extension • Audit System Integrity

Event IDs per category

Azure AD DS security audits record the following event IDs when the specific action triggers an auditable event:

EVENT CATEGORY NAME	EVENT IDS
Account Logon security	4767, 4774, 4775, 4776, 4777
Account Management security	4720, 4722, 4723, 4724, 4725, 4726, 4727, 4728, 4729, 4730, 4731, 4732, 4733, 4734, 4735, 4737, 4738, 4740, 4741, 4742, 4743, 4754, 4755, 4756, 4757, 4758, 4764, 4765, 4766, 4780, 4781, 4782, 4793, 4798, 4799, 5376, 5377
Detail Tracking security	None
DS Access security	5136, 5137, 5138, 5139, 5141
Logon-Logoff security	4624, 4625, 4634, 4647, 4648, 4672, 4675, 4964
Object Access security	None

EVENT CATEGORY NAME	EVENT IDS
Policy Change security	4670, 4703, 4704, 4705, 4706, 4707, 4713, 4715, 4716, 4717, 4718, 4719, 4739, 4864, 4865, 4866, 4867, 4904, 4906, 4911, 4912
Privilege Use security	4985
System security	4612, 4621

Next steps

For specific information on Kusto, see the following articles:

- Overview of the Kusto query language.
- Kusto tutorial to familiarize you with query basics.
- Sample queries that help you learn new ways to see your data.
- Kusto best practices to optimize your queries for success.

Review security audit events in Azure Active Directory Domain Services using Azure Monitor Workbooks

7/20/2020 • 4 minutes to read • Edit Online

To help you understand the state of your Azure Active Directory Domain Services (Azure AD DS) managed domain, you can enable security audit events. These security audit events can then be reviewed using Azure Monitor Workbooks that combine text,analytics queries, and parameters into rich interactive reports. Azure AD DS includes workbook templates for security overview and account activity that let you dig into audit events and manage your environment.

This article shows you how to use Azure Monitor Workbooks to review security audit events in Azure AD DS.

Before you begin

To complete this article, you need the following resources and privileges:

- An active Azure subscription.
  - If you don't have an Azure subscription, create an account.
- An Azure Active Directory tenant associated with your subscription, either synchronized with an on-premises directory or a cloud-only directory.
  - If needed, create an Azure Active Directory tenant or associate an Azure subscription with your account.
- An Azure Active Directory Domain Services managed domain enabled and configured in your Azure AD tenant.
  - If needed, complete the tutorial to create and configure an Azure Active Directory Domain Services managed domain.
- Security audit events enabled for your managed domain that stream data to a Log Analytics workspace.
  - If needed, enable security audits for Azure AD DS.

Azure Monitor Workbooks overview

When security audit events are turned on in Azure AD DS, it can be hard to analyze and identify issues in the managed domain. Azure Monitor lets you aggregate these security audit events and query the data. With Azure Monitor Workbooks, you can visualize this data to make it quicker and easier to identify issues.

Workbook templates are curated reports that are designed for flexible reuse by multiple users and teams. When you open a workbook template, the data from your Azure Monitor environment is loaded. You can use templates without an impact on other users in your organization, and can save your own workbooks based on the template.

Azure AD DS includes the following two workbook templates:

- Security overview report
- Account activity report

For more information about how to edit and manage workbooks, see Azure Monitor Workbooks overview.

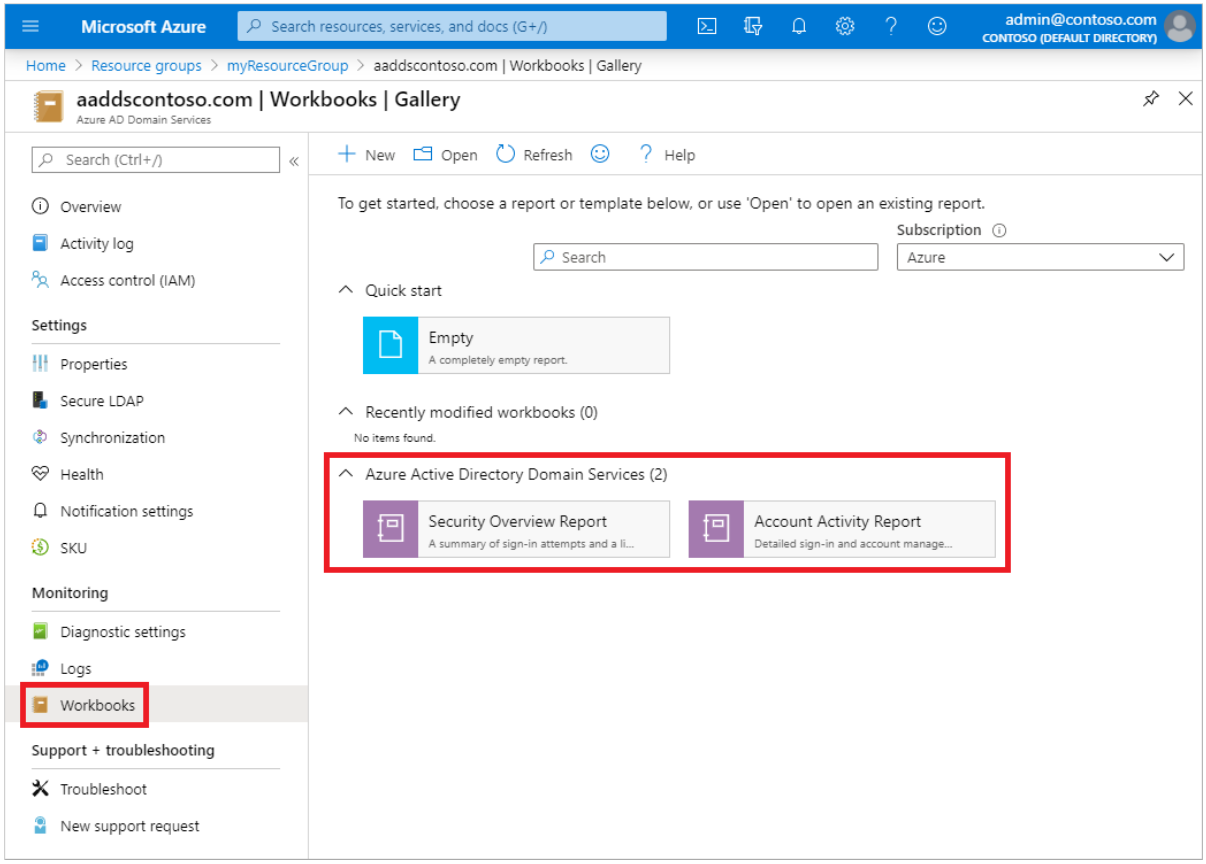
Use the security overview report workbook

To help you better understand usage and identify potential security threats, the security overview report summarizes sign-in data and identifies accounts you might want to check on. You can view events in a particular

date range, and drill down into specific sign-in events, such as bad password attempts or where the account was disabled.

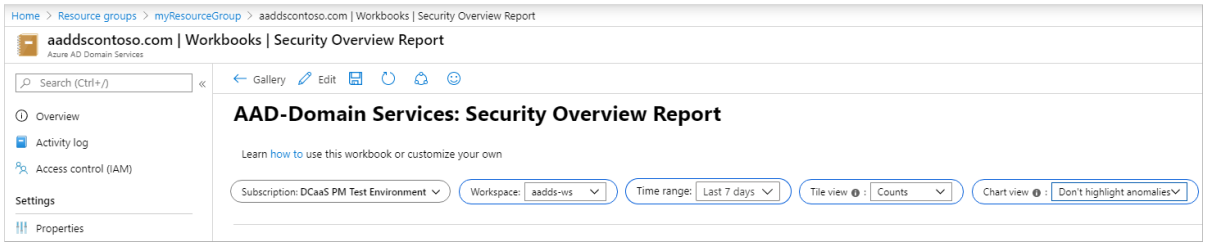
To access the workbook template for the security overview report, complete the following steps:

1. Search for and select **Azure Active Directory Domain Services** in the Azure portal.
2. Select your managed domain, such as *aaddscontoso.com*
3. From the menu on the left-hand side, choose **Monitoring > Workbooks**



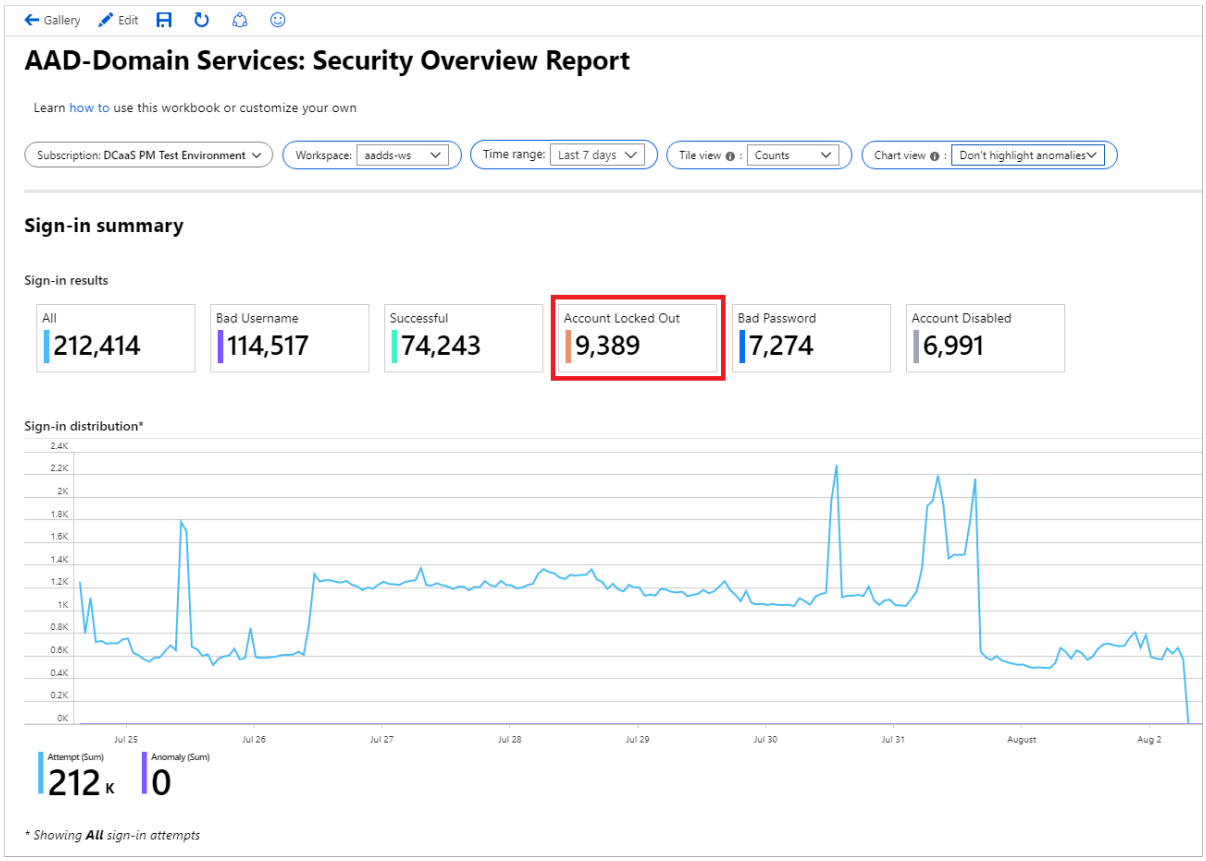
4. Choose the **Security Overview Report**.
5. From the drop-down menus at the top of the workbook, select your Azure subscription and then an Azure Monitor workspace.

   Choose a **Time range**, such as *Last 7 days*, as shown in the following example screenshot:

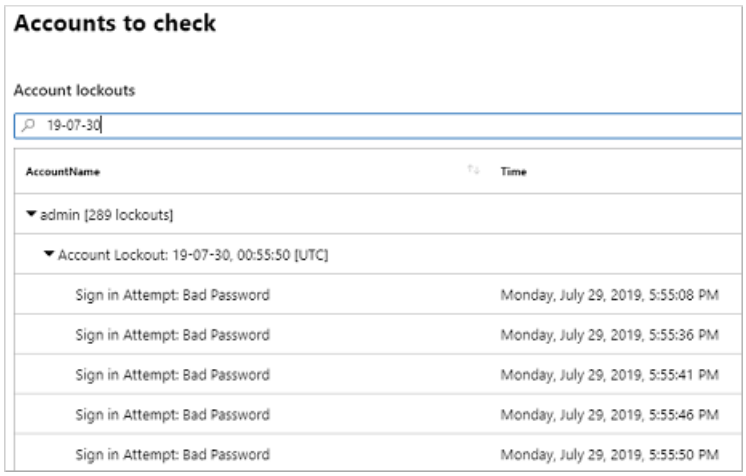


   The **Tile view** and **Chart view** options can also be changed to analyze and visualize the data as desired.

6. To drill down into a specific event type, select the one of the **Sign-in result** cards such as *Account Locked Out*, as shown in the following example:



7. The lower part of the security overview report below the chart then breaks down the activity type selected. You can filter by usernames involved on the right-hand side, as shown in the following example report:



Use the account activity report workbook

To help you troubleshoot issues for a specific user account, the account activity report breaks down detailed audit event log information. You can review when a bad username or password was provided during sign-in, and the source of the sign-in attempt.

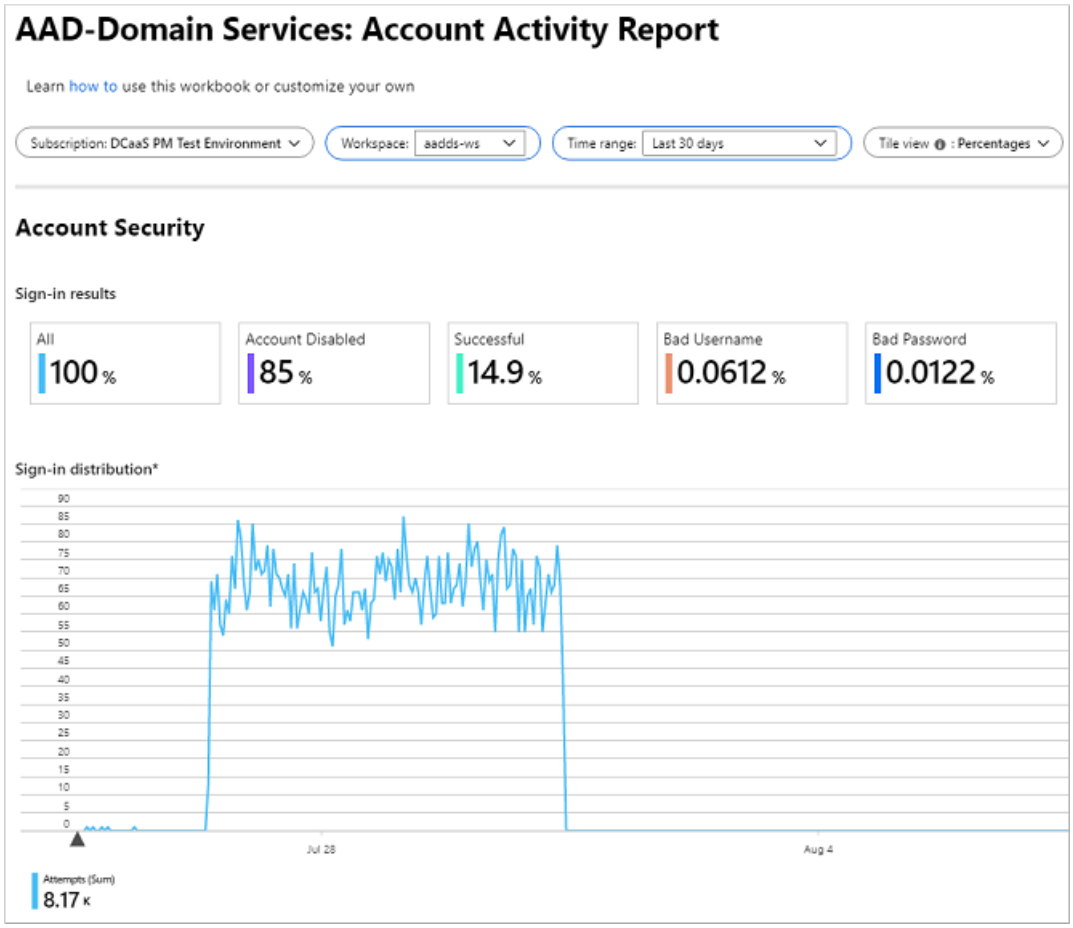
To access the workbook template for the account activity report, complete the following steps:

1. Search for and select **Azure Active Directory Domain Services** in the Azure portal.
2. Select your managed domain, such as *aaddscontoso.com*
3. From the menu on the left-hand side, choose **Monitoring > Workbooks**
4. Choose the **Account Activity Report**.
5. From the drop-down menus at the top of the workbook, select your Azure subscription and then an Azure

Monitor workspace.

Choose a **Time range**, such as *Last 30 days*, then how you want the **Tile view** to represent the data.

You can filter by **Account username**, such as *felix*, as shown in the following example report:



The area below the chart shows individual sign-in events along with information such as the activity result and source workstation. This information can help determine repeated sources of sign-in events that may cause account lockouts or indicate a potential attack.

As with the security overview report, you can drill down into the different tiles at the top of the report to visualize and analyze the data as needed.

Save and edit workbooks

The two template workbooks provided by Azure AD DS are a good place to start with your own data analysis. If you need to get more granular in the data queries and investigations, you can save your own workbooks and edit the queries.

1. To save a copy of one of the workbook templates, select **Edit** > **Save as** > **Shared reports**, then provide a name and save it.
2. From your own copy of the template, select **Edit** to enter the edit mode. You can choose the blue **Edit** button next to any part of the report and change it.

All of the charts and tables in Azure Monitor Workbooks are generated using Kusto queries. For more information on creating your own queries, see Azure Monitor log queries and Kusto queries tutorial.

Next steps

If you need to adjust password and lockout policies, see Password and account lockout policies on managed domains.

For problems with users, learn how to troubleshoot account sign-in problems or account lockout problems.

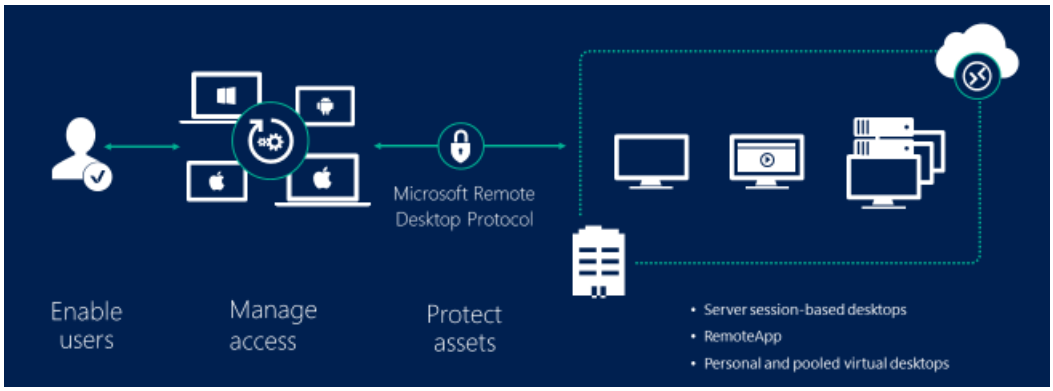
Secure remote access to virtual machines in Azure Active Directory Domain Services

7/20/2020 • 5 minutes to read • Edit Online

To secure remote access to virtual machines (VMs) that run in an Azure Active Directory Domain Services (Azure AD DS) managed domain, you can use Remote Desktop Services (RDS) and Network Policy Server (NPS). Azure AD DS authenticates users as they request access through the RDS environment. For enhanced security, you can integrate Azure Multi-Factor Authentication to provide an additional authentication prompt during sign-in events. Azure Multi-Factor Authentication uses an extension for NPS to provide this feature.

> **IMPORTANT**
>
> The recommended way to securely connect to your VMs in an Azure AD DS managed domain is using Azure Bastion, a fully platform-managed PaaS service that you provision inside your virtual network. A bastion host provides secure and seamless Remote Desktop Protocol (RDP) connectivity to your VMs directly in the Azure portal over SSL. When you connect via a bastion host, your VMs don't need a public IP address, and you don't need to use network security groups to expose access to RDP on TCP port 3389.
>
> We strongly recommend that you use Azure Bastion in all regions where it's supported. In regions without Azure Bastion availability, follow the steps detailed in this article until Azure Bastion is available. Take care with assigning public IP addresses to VMs joined to Azure AD DS where all incoming RDP traffic is allowed.
>
> For more information, see What is Azure Bastion?.

This article shows you how to configure RDS in Azure AD DS and optionally use the Azure Multi-Factor Authentication NPS extension.



Prerequisites

To complete this article, you need the following resources:

- An active Azure subscription.
  - If you don't have an Azure subscription, create an account.
- An Azure Active Directory tenant associated with your subscription, either synchronized with an on-premises directory or a cloud-only directory.
  - If needed, create an Azure Active Directory tenant or associate an Azure subscription with your account.
- An Azure Active Directory Domain Services managed domain enabled and configured in your Azure AD tenant.
  - If needed, create and configure an Azure Active Directory Domain Services managed domain.
- A *workloads* subnet created in your Azure Active Directory Domain Services virtual network.

    - If needed, Configure virtual networking for an Azure Active Directory Domain Services managed domain.
- A user account that's a member of the *Azure AD DC administrators* group in your Azure AD tenant.

Deploy and configure the Remote Desktop environment

To get started, create a minimum of two Azure VMs that run Windows Server 2016 or Windows Server 2019. For redundancy and high availability of your Remote Desktop (RD) environment, you can add and load balance additional hosts later.

A suggested RDS deployment includes the following two VMs:

- *RDGVM01* - Runs the RD Connection Broker server, RD Web Access server, and RD Gateway server.
- *RDSHVM01* - Runs the RD Session Host server.

Make sure that VMs are deployed into a *workloads* subnet of your Azure AD DS virtual network, then join the VMs to managed domain. For more information, see how to create and join a Windows Server VM to a managed domain.

The RD environment deployment contains a number of steps. The existing RD deployment guide can be used without any specific changes to use in a managed domain:

1. Sign in to VMs created for the RD environment with an account that's part of the *Azure AD DC Administrators* group, such as *contosoadmin*.
2. To create and configure RDS, use the existing Remote Desktop environment deployment guide. Distribute the RD server components across your Azure VMs as desired.
    - Specific to Azure AD DS - when you configure RD licensing, set it to **Per Device** mode, not **Per User** as noted in the deployment guide.
3. If you want to provide access using a web browser, set up the Remote Desktop web client for your users.

With RD deployed into the managed domain, you can manage and use the service as you would with an on-premises AD DS domain.

Deploy and configure NPS and the Azure MFA NPS extension

If you want to increase the security of the user sign-in experience, you can optionally integrate the RD environment with Azure Multi-Factor Authentication. With this configuration, users receive an additional prompt during sign-in to confirm their identity.

To provide this capability, an additional Network Policy Server (NPS) is installed in your environment along with the Azure Multi-Factor Authentication NPS extension. This extension integrates with Azure AD to request and return the status of multi-factor authentication prompts.

Users must be registered to use Azure Multi-Factor Authentication, which may require additional Azure AD licenses.

To integrate Azure Multi-Factor Authentication in to your Azure AD DS Remote Desktop environment, create an NPS Server and install the extension:

1. Create an additional Windows Server 2016 or 2019 VM, such as *NPSVM01*, that's connected to a *workloads* subnet in your Azure AD DS virtual network. Join the VM to the managed domain.
2. Sign in to NPS VM as account that's part of the *Azure AD DC Administrators* group, such as *contosoadmin*.
3. From **Server Manager**, select **Add Roles and Features**, then install the *Network Policy and Access Services* role.
4. Use the existing how-to article to install and configure the Azure MFA NPS extension.

With the NPS server and Azure Multi-Factor Authentication NPS extension installed, complete the next section to configure it for use with the RD environment.

Integrate Remote Desktop Gateway and Azure Multi-Factor Authentication

To integrate the Azure Multi-Factor Authentication NPS extension, use the existing how-to article to integrate your Remote Desktop Gateway infrastructure using the Network Policy Server (NPS) extension and Azure AD.

The following additional configuration options are needed to integrate with a managed domain:

1. Don't register the NPS server in Active Directory. This step fails in a managed domain.
2. In step 4 to configure network policy, also check the box to **Ignore user account dial-in properties**.
3. If you use Windows Server 2019 for the NPS server and Azure Multi-Factor Authentication NPS extension, run the following command to update the secure channel to allow the NPS server to communicate correctly:

   ```
   sc sidtype IAS unrestricted
   ```

Users are now prompted for an additional authentication factor when they sign in, such as a text message or prompt in the Microsoft Authenticator app.

Next steps

For more information on improving resiliency of your deployment, see Remote Desktop Services - High availability.

For more information about securing user sign-in, see How it works: Azure Multi-Factor Authentication.

Join a Windows Server virtual machine to an Azure Active Directory Domain Services managed domain using a Resource Manager template

7/20/2020 • 6 minutes to read • Edit Online

To automate the deployment and configuration of Azure virtual machines (VMs), you can use a Resource Manager template. These templates let you create consistent deployments each time. Extensions can also be included in templates to automatically configure a VM as part of the deployment. One useful extension joins VMs to a domain, which can be used with Azure Active Directory Domain Services (Azure AD DS) managed domains.

This article shows you how to create and join a Windows Server VM to an Azure AD DS managed domain using Resource Manager templates. You also learn how to join an existing Windows Server VM to an Azure AD DS domain.

Prerequisites

To complete this tutorial, you need the following resources and privileges:

- An active Azure subscription.
  - If you don't have an Azure subscription, create an account.
- An Azure Active Directory tenant associated with your subscription, either synchronized with an on-premises directory or a cloud-only directory.
  - If needed, create an Azure Active Directory tenant or associate an Azure subscription with your account.
- An Azure Active Directory Domain Services managed domain enabled and configured in your Azure AD tenant.
  - If needed, the first tutorial creates and configures an Azure Active Directory Domain Services managed domain.
- A user account that's a part of the managed domain.

Azure Resource Manager template overview

Resource Manager templates let you define Azure infrastructure in code. The required resources, network connections, or configuration of VMs can all be defined in a template. These templates create consistent, reproducible deployments each time, and can be versioned as you make changes. For more information, see Azure Resource Manager templates overview.

Each resource is defined in a template using JavaScript Object Notation (JSON). The following JSON example uses the *Microsoft.Compute/virtualMachines/extensions* resource type to install the Active Directory domain join extension. Parameters are used that you specify at deployment time. When the extension is deployed, the VM is joined to the specified managed domain.

```
{
     "apiVersion": "2015-06-15",
     "type": "Microsoft.Compute/virtualMachines/extensions",
     "name": "[concat(parameters('dnsLabelPrefix'),'/joindomain')]",
     "location": "[parameters('location')]",
     "dependsOn": [
       "[concat('Microsoft.Compute/virtualMachines/', parameters('dnsLabelPrefix'))]"
     ],
     "properties": {
       "publisher": "Microsoft.Compute",
       "type": "JsonADDomainExtension",
       "typeHandlerVersion": "1.3",
       "autoUpgradeMinorVersion": true,
       "settings": {
         "Name": "[parameters('domainToJoin')]",
         "OUPath": "[parameters('ouPath')]",
         "User": "[concat(parameters('domainToJoin'), '\\', parameters('domainUsername'))]",
         "Restart": "true",
         "Options": "[parameters('domainJoinOptions')]"
       },
       "protectedSettings": {
         "Password": "[parameters('domainPassword')]"
       }
     }
   }
```

This VM extension can be deployed even if you don't create a VM in the same template. The examples in this article show both of the following approaches:

- Create a Windows Server VM and join to a managed domain
- Join an existing Windows Server VM to a managed domain

Create a Windows Server VM and join to a managed domain

If you need a Windows Server VM, you can create and configure one using a Resource Manager template. When the VM is deployed, an extension is then installed to join the VM to a managed domain. If you already have a VM you wish to join to a managed domain, skip to Join an existing Windows Server VM to a managed domain.

To create a Windows Server VM then join it to a managed domain, complete the following steps:

1. Browse to the quickstart template. Select the option to **Deploy to Azure**.
2. On the **Custom deployment** page, enter the following information to create and join a Windows Server VM to the managed domain:

SETTING	VALUE
Subscription	Pick the same Azure subscription in which you have enabled Azure AD Domain Services.
Resource group	Choose the resource group for your VM.
Location	Select the location of for your VM.
Existing VNET Name	The name of the existing virtual network to connect the VM to, such as *myVnet*.
Existing Subnet Name	The name of the existing virtual network subnet, such as *Workloads*.

SETTING	VALUE
DNS Label Prefix	Enter a DNS name to use for the VM, such as *myvm*.
VM size	Specify a VM size, such as *Standard_DS2_v2*.
Domain To Join	The managed domain DNS name, such as *aaddscontoso.com*.
Domain Username	The user account in the managed domain that should be used to join the VM to the managed domain, such as `contosoadmin@aaddscontoso.com`. This account must be a part of the managed domain.
Domain Password	The password for the user account specified in the previous setting.
Optional OU Path	The custom OU in which to add the VM. If you don't specify a value for this parameter, the VM is added to the default *AAD DC Computers* OU.
VM Admin Username	Specify a local administrator account to create on the VM.
VM Admin Password	Specify a local administrator password for the VM. Create a strong local administrator password to protect against password brute-force attacks.

3. Review the terms and conditions, then check the box for **I agree to the terms and conditions stated above**. When ready, select **Purchase** to create and join the VM to the managed domain.

> **WARNING**
>
> **Handle passwords with caution.** The template parameter file requests the password for a user account that's a part of the managed domain. Don't manually enter values into this file and leave it accessible on file shares or other shared locations.

It takes a few minutes for the deployment to complete successfully. When finished, the Windows VM is created and joined to the managed domain. The VM can be managed or signed into using domain accounts.

Join an existing Windows Server VM to a managed domain

If you have an existing VM, or group of VMs, that you wish to join to a managed domain, you can use a Resource Manager template to just deploy the VM extension.

To join an existing Windows Server VM to a managed domain, complete the following steps:

1. Browse to the quickstart template. Select the option to **Deploy to Azure**.
2. On the **Custom deployment** page, enter the following information to join the VM to the managed domain:

SETTING	VALUE
Subscription	Pick the same Azure subscription in which you have enabled Azure AD Domain Services.
Resource group	Choose the resource group with your existing VM.

SETTING	VALUE
Location	Select the location of your existing VM.
VM list	Enter the comma-separated list of the existing VM(s) to join to the managed domain, such as *myVM1,myVM2*.
Domain Join User Name	The user account in the managed domain that should be used to join the VM to the managed domain, such as `contosoadmin@aaddscontoso.com`. This account must be a part of the managed domain.
Domain Join User Password	The password for the user account specified in the previous setting.
Optional OU Path	The custom OU in which to add the VM. If you don't specify a value for this parameter, the VM is added to the default *AAD DC Computers* OU.

3. Review the terms and conditions, then check the box for **I agree to the terms and conditions stated above**. When ready, select **Purchase** to join the VM to the managed domain.

> **WARNING**
>
> **Handle passwords with caution.** The template parameter file requests the password for a user account that's a part of the managed domain. Don't manually enter values into this file and leave it accessible on file shares or other shared locations.

It takes a few moments for the deployment to complete successfully. When finished, the specified Windows VMs are joined to the managed domain and can be managed or signed into using domain accounts.

Next steps

In this article, you used the Azure portal to configure and deploy resources using templates. You can also deploy resources with Resource Manager templates using Azure PowerShell or the Azure CLI.

Join a CentOS Linux virtual machine to an Azure Active Directory Domain Services managed domain

7/20/2020 • 6 minutes to read • Edit Online

To let users sign in to virtual machines (VMs) in Azure using a single set of credentials, you can join VMs to an Azure Active Directory Domain Services (Azure AD DS) managed domain. When you join a VM to an Azure AD DS managed domain, user accounts and credentials from the domain can be used to sign in and manage servers. Group memberships from the managed domain are also applied to let you control access to files or services on the VM.

This article shows you how to join a CentOS Linux VM to a managed domain.

Prerequisites

To complete this tutorial, you need the following resources and privileges:

- An active Azure subscription.
  - If you don't have an Azure subscription, create an account.
- An Azure Active Directory tenant associated with your subscription, either synchronized with an on-premises directory or a cloud-only directory.
  - If needed, create an Azure Active Directory tenant or associate an Azure subscription with your account.
- An Azure Active Directory Domain Services managed domain enabled and configured in your Azure AD tenant.
  - If needed, the first tutorial creates and configures an Azure Active Directory Domain Services managed domain.
- A user account that's part of the managed domain.

Create and connect to a CentOS Linux VM

If you have an existing CentOS Linux VM in Azure, connect to it using SSH, then continue on to the next step to start configuring the VM.

If you need to create a CentOS Linux VM, or want to create a test VM for use with this article, you can use one of the following methods:

- Azure portal
- Azure CLI
- Azure PowerShell

When you create the VM, pay attention to the virtual network settings to make sure that the VM can communicate with the managed domain:

- Deploy the VM into the same, or a peered, virtual network in which you have enabled Azure AD Domain Services.
- Deploy the VM into a different subnet than your managed domain.

Once the VM is deployed, follow the steps to connect to the VM using SSH.

Configure the hosts file

To make sure that the VM host name is correctly configured for the managed domain, edit the */etc/hosts* file and

set the hostname:

```
sudo vi /etc/hosts
```

In the *hosts* file, update the *localhost* address. In the following example:

- *aaddscontoso.com* is the DNS domain name of your managed domain.
- *centos* is the hostname of your CentOS VM that you're joining to the managed domain.

Update these names with your own values:

```
127.0.0.1 centos.aaddscontoso.com centos
```

When done, save and exit the *hosts* file using the `:wq` command of the editor.

Install required packages

The VM needs some additional packages to join the VM to the managed domain. To install and configure these packages, update and install the domain-join tools using `yum`:

```
sudo yum install realmd sssd krb5-workstation krb5-libs oddjob oddjob-mkhomedir samba-common-tools
```

Join VM to the managed domain

Now that the required packages are installed on the VM, join the VM to the managed domain.

1. Use the `realm discover` command to discover the managed domain. The following example discovers the realm *AADDSCONTOSO.COM*. Specify your own managed domain name in ALL UPPERCASE:

   ```
   sudo realm discover AADDSCONTOSO.COM
   ```

   If the `realm discover` command can't find your managed domain, review the following troubleshooting steps:

   - Make sure that the domain is reachable from the VM. Try `ping aaddscontoso.com` to see if a positive reply is returned.
   - Check that the VM is deployed to the same, or a peered, virtual network in which the managed domain is available.
   - Confirm that the DNS server settings for the virtual network have been updated to point to the domain controllers of the managed domain.
2. Now initialize Kerberos using the `kinit` command. Specify a user that's a part of the managed domain. If needed, add a user account to a group in Azure AD.

   Again, the managed domain name must be entered in ALL UPPERCASE. In the following example, the account named `contosoadmin@aaddscontoso.com` is used to initialize Kerberos. Enter your own user account that's a part of the managed domain:

   ```
   kinit contosoadmin@AADDSCONTOSO.COM
   ```
3. Finally, join the VM to the managed domain using the `realm join` command. Use the same user account that's a part of the managed domain that you specified in the previous `kinit` command, such as

`contosoadmin@AADDSCONTOSO.COM`:

```
sudo realm join --verbose AADDSCONTOSO.COM -U 'contosoadmin@AADDSCONTOSO.COM'
```

It takes a few moments to join the VM to the managed domain. The following example output shows the VM has successfully joined to the managed domain:

```
Successfully enrolled machine in realm
```

If your VM can't successfully complete the domain-join process, make sure that the VM's network security group allows outbound Kerberos traffic on TCP + UDP port 464 to the virtual network subnet for your managed domain.

Allow password authentication for SSH

By default, users can only sign in to a VM using SSH public key-based authentication. Password-based authentication fails. When you join the VM to a managed domain, those domain accounts need to use password-based authentication. Update the SSH configuration to allow password-based authentication as follows.

1. Open the *sshd_conf* file with an editor:

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

2. Update the line for *PasswordAuthentication* to *yes*:

   ```
   PasswordAuthentication yes
   ```

   When done, save and exit the *sshd_conf* file using the `:wq` command of the editor.

3. To apply the changes and let users sign in using a password, restart the SSH service:

   ```
   sudo systemctl restart sshd
   ```

Grant the 'AAD DC Administrators' group sudo privileges

To grant members of the *AAD DC Administrators* group administrative privileges on the CentOS VM, you add an entry to the */etc/sudoers*. Once added, members of the *AAD DC Administrators* group can use the `sudo` command on the CentOS VM.

1. Open the *sudoers* file for editing:

   ```
   sudo visudo
   ```

2. Add the following entry to the end of */etc/sudoers* file. The *AAD DC Administrators* group contains whitespace in the name, so include the backslash escape character in the group name. Add your own domain name, such as *aaddscontoso.com*:

   ```
   # Add 'AAD DC Administrators' group members as admins.
   %AAD\ DC\ Administrators@aaddscontoso.com ALL=(ALL) NOPASSWD:ALL
   ```

   When done, save and exit the editor using the `:wq` command of the editor.

Sign in to the VM using a domain account

To verify that the VM has been successfully joined to the managed domain, start a new SSH connection using a domain user account. Confirm that a home directory has been created, and that group membership from the domain is applied.

1. Create a new SSH connection from your console. Use a domain account that belongs to the managed domain using the `ssh -l` command, such as `contosoadmin@aaddscontoso.com` and then enter the address of your VM, such as *centos.aaddscontoso.com*. If you use the Azure Cloud Shell, use the public IP address of the VM rather than the internal DNS name.

   ```
   ssh -l contosoadmin@AADDSCONTOSO.com centos.aaddscontoso.com
   ```

2. When you've successfully connected to the VM, verify that the home directory was initialized correctly:

   ```
   pwd
   ```

   You should be in the */home* directory with your own directory that matches the user account.

3. Now check that the group memberships are being resolved correctly:

   ```
   id
   ```

   You should see your group memberships from the managed domain.

4. If you signed in to the VM as a member of the *AAD DC Administrators* group, check that you can correctly use the `sudo` command:

   ```
   sudo yum update
   ```

Next steps

If you have problems connecting the VM to the managed domain or signing in with a domain account, see Troubleshooting domain join issues.

Join a CoreOS virtual machine to an Azure Active Directory Domain Services managed domain

7/20/2020 • 4 minutes to read • Edit Online

To let users sign in to virtual machines (VMs) in Azure using a single set of credentials, you can join VMs to an Azure Active Directory Domain Services (Azure AD DS) managed domain. When you join a VM to an Azure AD DS managed domain, user accounts and credentials from the domain can be used to sign in and manage servers. Group memberships from the managed domain are also applied to let you control access to files or services on the VM.

This article shows you how to join a CoreOS VM to a managed domain.

Prerequisites

To complete this tutorial, you need the following resources and privileges:

- An active Azure subscription.
  - If you don't have an Azure subscription, create an account.
- An Azure Active Directory tenant associated with your subscription, either synchronized with an on-premises directory or a cloud-only directory.
  - If needed, create an Azure Active Directory tenant or associate an Azure subscription with your account.
- An Azure Active Directory Domain Services managed domain enabled and configured in your Azure AD tenant.
  - If needed, the first tutorial creates and configures an Azure Active Directory Domain Services managed domain.
- A user account that's a part of the managed domain.

Create and connect to a CoreOS Linux VM

If you have an existing CoreOS Linux VM in Azure, connect to it using SSH, then continue on to the next step to start configuring the VM.

If you need to create a CoreOS Linux VM, or want to create a test VM for use with this article, you can use one of the following methods:

- Azure portal
- Azure CLI
- Azure PowerShell

When you create the VM, pay attention to the virtual network settings to make sure that the VM can communicate with the managed domain:

- Deploy the VM into the same, or a peered, virtual network in which you have enabled Azure AD Domain Services.
- Deploy the VM into a different subnet than your Azure AD Domain Services managed domain.

Once the VM is deployed, follow the steps to connect to the VM using SSH.

Configure the hosts file

To make sure that the VM host name is correctly configured for the managed domain, edit the */etc/hosts* file and

set the hostname:

```
sudo vi /etc/hosts
```

In the *hosts* file, update the *localhost* address. In the following example:

- *aaddscontoso.com* is the DNS domain name of your managed domain.
- *coreos* is the hostname of your CoreOS VM that you're joining to the managed domain.

Update these names with your own values:

```
127.0.0.1 coreos coreos.aaddscontoso.com
```

When done, save and exit the *hosts* file using the `:wq` command of the editor.

Configure the SSSD service

Update the */etc/sssd/sssd.conf* SSSD configuration.

```
sudo vi /etc/sssd/sssd.conf
```

Specify your own managed domain name for the following parameters:

- *domains* in ALL UPPER CASE
- *[domain/AADDSCONTOSO]* where AADDSCONTOSO is in ALL UPPER CASE
- *ldap_uri*
- *ldap_search_base*
- *krb5_server*
- *krb5_realm* in ALL UPPER CASE

```
[sssd]
config_file_version = 2
services = nss, pam
domains = AADDSCONTOSO.COM

[domain/AADDSCONTOSO]
id_provider = ad
auth_provider = ad
chpass_provider = ad

ldap_uri = ldap://aaddscontoso.com
ldap_search_base = dc=aaddscontoso,dc=com
ldap_schema = rfc2307bis
ldap_sasl_mech = GSSAPI
ldap_user_object_class = user
ldap_group_object_class = group
ldap_user_home_directory = unixHomeDirectory
ldap_user_principal = userPrincipalName
ldap_account_expire_policy = ad
ldap_force_upper_case_realm = true
fallback_homedir = /home/%d/%u

krb5_server = aaddscontoso.com
krb5_realm = AADDSCONTOSO.COM
```

Join the VM to the managed domain

With the SSSD configuration file updated, now join the virtual machine to the managed domain.

1. First, use the `adcli info` command to verify you can see information about the managed domain. The following example gets information for the domain *AADDSCONTOSO.COM*. Specify your own managed domain name in ALL UPPERCASE:

   ```
   sudo adcli info AADDSCONTOSO.COM
   ```

   If the `adcli info` command can't find your managed domain, review the following troubleshooting steps:

   - Make sure that the domain is reachable from the VM. Try `ping aaddscontoso.com` to see if a positive reply is returned.
   - Check that the VM is deployed to the same, or a peered, virtual network in which the managed domain is available.
   - Confirm that the DNS server settings for the virtual network have been updated to point to the domain controllers of the managed domain.

2. Now join the VM to the managed domain using the `adcli join` command. Specify a user that's a part of the managed domain. If needed, add a user account to a group in Azure AD.

   Again, the managed domain name must be entered in ALL UPPERCASE. In the following example, the account named `contosoadmin@aaddscontoso.com` is used to initialize Kerberos. Enter your own user account that's a part of the managed domain.

   ```
   sudo adcli join -D AADDSCONTOSO.COM -U contosoadmin@AADDSCONTOSO.COM -K /etc/krb5.keytab -H
   coreos.aaddscontoso.com -N coreos
   ```

   The `adcli join` command doesn't return any information when the VM has successfully joined to the managed domain.

3. To apply the domain-join configuration, start the SSSD service:

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

Sign in to the VM using a domain account

To verify that the VM has been successfully joined to the managed domain, start a new SSH connection using a domain user account. Confirm that a home directory has been created, and that group membership from the domain is applied.

1. Create a new SSH connection from your console. Use a domain account that belongs to the managed domain using the `ssh -l` command, such as `contosoadmin@aaddscontoso.com` and then enter the address of your VM, such as *coreos.aaddscontoso.com*. If you use the Azure Cloud Shell, use the public IP address of the VM rather than the internal DNS name.

   ```
   ssh -l contosoadmin@AADDSCONTOSO.com coreos.aaddscontoso.com
   ```

2. Now check that the group memberships are being resolved correctly:

   ```
   id
   ```

   You should see your group memberships from the managed domain.

Next steps

If you have problems connecting the VM to the managed domain or signing in with a domain account, see Troubleshooting domain join issues.

Join a Red Hat Enterprise Linux virtual machine to an Azure Active Directory Domain Services managed domain

7/20/2020 • 8 minutes to read • Edit Online

To let users sign in to virtual machines (VMs) in Azure using a single set of credentials, you can join VMs to an Azure Active Directory Domain Services (Azure AD DS) managed domain. When you join a VM to an Azure AD DS managed domain, user accounts and credentials from the domain can be used to sign in and manage servers. Group memberships from the managed domain are also applied to let you control access to files or services on the VM.

This article shows you how to join a Red Hat Enterprise Linux (RHEL) VM to a managed domain.

Prerequisites

To complete this tutorial, you need the following resources and privileges:

- An active Azure subscription.
  - If you don't have an Azure subscription, create an account.
- An Azure Active Directory tenant associated with your subscription, either synchronized with an on-premises directory or a cloud-only directory.
  - If needed, create an Azure Active Directory tenant or associate an Azure subscription with your account.
- An Azure Active Directory Domain Services managed domain enabled and configured in your Azure AD tenant.
  - If needed, the first tutorial creates and configures an Azure Active Directory Domain Services managed domain.
- A user account that's a part of the managed domain.

Create and connect to a RHEL Linux VM

If you have an existing RHEL Linux VM in Azure, connect to it using SSH, then continue on to the next step to start configuring the VM.

If you need to create a RHEL Linux VM, or want to create a test VM for use with this article, you can use one of the following methods:

- Azure portal
- Azure CLI
- Azure PowerShell

When you create the VM, pay attention to the virtual network settings to make sure that the VM can communicate with the managed domain:

- Deploy the VM into the same, or a peered, virtual network in which you have enabled Azure AD Domain Services.
- Deploy the VM into a different subnet than your Azure AD Domain Services managed domain.

Once the VM is deployed, follow the steps to connect to the VM using SSH.

Configure the hosts file

To make sure that the VM host name is correctly configured for the managed domain, edit the */etc/hosts* file and set the hostname:

```
sudo vi /etc/hosts
```

In the *hosts* file, update the *localhost* address. In the following example:

- *aaddscontoso.com* is the DNS domain name of your managed domain.
- *rhel* is the hostname of your RHEL VM that you're joining to the managed domain.

Update these names with your own values:

```
127.0.0.1 rhel rhel.aaddscontoso.com
```

When done, save and exit the *hosts* file using the `:wq` command of the editor.

Install required packages

The VM needs some additional packages to join the VM to the managed domain. To install and configure these packages, update and install the domain-join tools using `yum`. There are some differences between RHEL 7.x and RHEL 6.x, so use the appropriate commands for your distro version in the remaining sections of this article.

RHEL 7

```
sudo yum install realmd sssd krb5-workstation krb5-libs oddjob oddjob-mkhomedir samba-common-tools
```

RHEL 6

```
sudo yum install adcli sssd authconfig krb5-workstation
```

Join VM to the managed domain

Now that the required packages are installed on the VM, join the VM to the managed domain. Again, use the appropriate steps for your RHEL distro version.

RHEL 7

1. Use the `realm discover` command to discover the managed domain. The following example discovers the realm *AADDSCONTOSO.COM*. Specify your own managed domain name in ALL UPPERCASE:

   ```
   sudo realm discover AADDSCONTOSO.COM
   ```

   If the `realm discover` command can't find your managed domain, review the following troubleshooting steps:

   - Make sure that the domain is reachable from the VM. Try `ping aaddscontoso.com` to see if a positive reply is returned.
   - Check that the VM is deployed to the same, or a peered, virtual network in which the managed domain is available.
   - Confirm that the DNS server settings for the virtual network have been updated to point to the domain controllers of the managed domain.
2. Now initialize Kerberos using the `kinit` command. Specify a user that's a part of the managed domain. If

needed, add a user account to a group in Azure AD.

Again, the managed domain name must be entered in ALL UPPERCASE. In the following example, the account named `contosoadmin@aaddscontoso.com` is used to initialize Kerberos. Enter your own user account that's a part of the managed domain:

```
kinit contosoadmin@AADDSCONTOSO.COM
```

3. Finally, join the VM to the managed domain using the `realm join` command. Use the same user account that's a part of the managed domain that you specified in the previous `kinit` command, such as `contosoadmin@AADDSCONTOSO.COM`:

```
sudo realm join --verbose AADDSCONTOSO.COM -U 'contosoadmin@AADDSCONTOSO.COM'
```

It takes a few moments to join the VM to the managed domain. The following example output shows the VM has successfully joined to the managed domain:

```
Successfully enrolled machine in realm
```

RHEL 6

1. Use the `adcli info` command to discover the managed domain. The following example discovers the realm *ADDDSCONTOSO.COM*. Specify your own managed domain name in ALL UPPERCASE:

```
sudo adcli info aaddscontoso.com
```

If the `adcli info` command can't find your managed domain, review the following troubleshooting steps:

- Make sure that the domain is reachable from the VM. Try `ping aaddscontoso.com` to see if a positive reply is returned.
- Check that the VM is deployed to the same, or a peered, virtual network in which the managed domain is available.
- Confirm that the DNS server settings for the virtual network have been updated to point to the domain controllers of the managed domain.

2. First, join the domain using the `adcli join` command, this command also creates the keytab to authenticate the machine. Use a user account that's a part of the managed domain.

```
sudo adcli join aaddscontoso.com -U contosoadmin
```

3. Now configure the `/ect/krb5.conf` and create the `/etc/sssd/sssd.conf` files to use the `aaddscontoso.com` Active Directory domain. Make sure that `AADDSCONTOSO.COM` is replaced by your own domain name:

Open the `/ect/krb5.conf` file with an editor:

```
sudo vi /etc/krb5.conf
```

Update the `krb5.conf` file to match the following sample:

```
[logging]
 default = FILE:/var/log/krb5libs.log
 kdc = FILE:/var/log/krb5kdc.log
 admin_server = FILE:/var/log/kadmind.log

[libdefaults]
 default_realm = AADDSCONTOSO.COM
 dns_lookup_realm = true
 dns_lookup_kdc = true
 ticket_lifetime = 24h
 renew_lifetime = 7d
 forwardable = true

[realms]
 AADDSCONTOSO.COM = {
 kdc = AADDSCONTOSO.COM
 admin_server = AADDSCONTOSO.COM
 }

[domain_realm]
 .AADDSCONTOSO.COM = AADDSCONTOSO.COM
 AADDSCONTOSO.COM = AADDSCONTOSO.COM
```

Create the `/etc/sssd/sssd.conf` file:

```
sudo vi /etc/sssd/sssd.conf
```

Update the `sssd.conf` file to match the following sample:

```
[sssd]
 services = nss, pam, ssh, autofs
 config_file_version = 2
 domains = AADDSCONTOSO.COM

[domain/AADDSCONTOSO.COM]

 id_provider = ad
```

4. Make sure `/etc/sssd/sssd.conf` permissions are 600 and is owned by root user:

```
sudo chmod 600 /etc/sssd/sssd.conf
sudo chown root:root /etc/sssd/sssd.conf
```

5. Use `authconfig` to instruct the VM about the AD Linux integration:

```
sudo authconfig --enablesssd --enablesssdauth --update
```

6. Start and enable the sssd service:

```
sudo service sssd start
sudo chkconfig sssd on
```

If your VM can't successfully complete the domain-join process, make sure that the VM's network security group allows outbound Kerberos traffic on TCP + UDP port 464 to the virtual network subnet for your managed domain.

Now check if you can query user AD information using `getent`

```
sudo getent passwd contosoadmin
```

Allow password authentication for SSH

By default, users can only sign in to a VM using SSH public key-based authentication. Password-based authentication fails. When you join the VM to a managed domain, those domain accounts need to use password-based authentication. Update the SSH configuration to allow password-based authentication as follows.

1. Open the *sshd_conf* file with an editor:

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

2. Update the line for *PasswordAuthentication* to *yes*:

   ```
   PasswordAuthentication yes
   ```

   When done, save and exit the *sshd_conf* file using the `:wq` command of the editor.

3. To apply the changes and let users sign in using a password, restart the SSH service for your RHEL distro version:

   RHEL 7

   ```
   sudo systemctl restart sshd
   ```

   RHEL 6

   ```
   sudo service sshd restart
   ```

Grant the 'AAD DC Administrators' group sudo privileges

To grant members of the *AAD DC Administrators* group administrative privileges on the RHEL VM, you add an entry to the */etc/sudoers*. Once added, members of the *AAD DC Administrators* group can use the `sudo` command on the RHEL VM.

1. Open the *sudoers* file for editing:

   ```
   sudo visudo
   ```

2. Add the following entry to the end of */etc/sudoers* file. The *AAD DC Administrators* group contains whitespace in the name, so include the backslash escape character in the group name. Add your own domain name, such as *aaddscontoso.com*:

   ```
   # Add 'AAD DC Administrators' group members as admins.
   %AAD\ DC\ Administrators@aaddscontoso.com ALL=(ALL) NOPASSWD:ALL
   ```

   When done, save and exit the editor using the `:wq` command of the editor.

Sign in to the VM using a domain account

To verify that the VM has been successfully joined to the managed domain, start a new SSH connection using a domain user account. Confirm that a home directory has been created, and that group membership from the domain is applied.

1. Create a new SSH connection from your console. Use a domain account that belongs to the managed domain using the `ssh -l` command, such as `contosoadmin@aaddscontoso.com` and then enter the address of your VM, such as *rhel.aaddscontoso.com*. If you use the Azure Cloud Shell, use the public IP address of the VM rather than the internal DNS name.

   ```
   ssh -l contosoadmin@AADDSCONTOSO.com rhel.aaddscontoso.com
   ```

2. When you've successfully connected to the VM, verify that the home directory was initialized correctly:

   ```
   pwd
   ```

   You should be in the */home* directory with your own directory that matches the user account.

3. Now check that the group memberships are being resolved correctly:

   ```
   id
   ```

   You should see your group memberships from the managed domain.

4. If you signed in to the VM as a member of the *AAD DC Administrators* group, check that you can correctly use the `sudo` command:

   ```
   sudo yum update
   ```

Next steps

If you have problems connecting the VM to the managed domain or signing in with a domain account, see Troubleshooting domain join issues.

Join an Ubuntu Linux virtual machine to an Azure Active Directory Domain Services managed domain

7/20/2020 • 8 minutes to read • Edit Online

To let users sign in to virtual machines (VMs) in Azure using a single set of credentials, you can join VMs to an Azure Active Directory Domain Services (Azure AD DS) managed domain. When you join a VM to an Azure AD DS managed domain, user accounts and credentials from the domain can be used to sign in and manage servers. Group memberships from the managed domain are also applied to let you control access to files or services on the VM.

This article shows you how to join an Ubuntu Linux VM to a managed domain.

Prerequisites

To complete this tutorial, you need the following resources and privileges:

- An active Azure subscription.
  - If you don't have an Azure subscription, create an account.
- An Azure Active Directory tenant associated with your subscription, either synchronized with an on-premises directory or a cloud-only directory.
  - If needed, create an Azure Active Directory tenant or associate an Azure subscription with your account.
- An Azure Active Directory Domain Services managed domain enabled and configured in your Azure AD tenant.
  - If needed, the first tutorial creates and configures an Azure Active Directory Domain Services managed domain.
- A user account that's a part of the managed domain.

Create and connect to an Ubuntu Linux VM

If you have an existing Ubuntu Linux VM in Azure, connect to it using SSH, then continue on to the next step to start configuring the VM.

If you need to create an Ubuntu Linux VM, or want to create a test VM for use with this article, you can use one of the following methods:

- Azure portal
- Azure CLI
- Azure PowerShell

When you create the VM, pay attention to the virtual network settings to make sure that the VM can communicate with the managed domain:

- Deploy the VM into the same, or a peered, virtual network in which you have enabled Azure AD Domain Services.
- Deploy the VM into a different subnet than your Azure AD Domain Services managed domain.

Once the VM is deployed, follow the steps to connect to the VM using SSH.

Configure the hosts file

To make sure that the VM host name is correctly configured for the managed domain, edit the */etc/hosts* file and

set the hostname:

```
sudo vi /etc/hosts
```

In the *hosts* file, update the *localhost* address. In the following example:

- *aaddscontoso.com* is the DNS domain name of your managed domain.
- *ubuntu* is the hostname of your Ubuntu VM that you're joining to the managed domain.

Update these names with your own values:

```
127.0.0.1 ubuntu.aaddscontoso.com ubuntu
```

When done, save and exit the *hosts* file using the `:wq` command of the editor.

Install required packages

The VM needs some additional packages to join the VM to the managed domain. To install and configure these packages, update and install the domain-join tools using `apt-get`

During the Kerberos installation, the *krb5-user* package prompts for the realm name in ALL UPPERCASE. For example, if the name of your managed domain is *aaddscontoso.com*, enter *AADDSCONTOSO.COM* as the realm. The installation writes the `[realm]` and `[domain_realm]` sections in */etc/krb5.conf* configuration file. Make sure that you specify the realm an ALL UPPERCASE:

```
sudo apt-get update
sudo apt-get install krb5-user samba sssd sssd-tools libnss-sss libpam-sss ntp ntpdate realmd adcli
```

Configure Network Time Protocol (NTP)

For domain communication to work correctly, the date and time of your Ubuntu VM must synchronize with the managed domain. Add your managed domain's NTP hostname to the */etc/ntp.conf* file.

1. Open the *ntp.conf* file with an editor:

   ```
   sudo vi /etc/ntp.conf
   ```

2. In the *ntp.conf* file, create a line to add your managed domain's DNS name. In the following example, an entry for *aaddscontoso.com* is added. Use your own DNS name:

   ```
   server aaddscontoso.com
   ```

   When done, save and exit the *ntp.conf* file using the `:wq` command of the editor.

3. To make sure that the VM is synchronized with the managed domain, the following steps are needed:

   - Stop the NTP server
   - Update the date and time from the managed domain
   - Start the NTP service

   Run the following commands to complete these steps. Use your own DNS name with the `ntpdate` command:

```
sudo systemctl stop ntp
sudo ntpdate aaddscontoso.com
sudo systemctl start ntp
```

Join VM to the managed domain

Now that the required packages are installed on the VM and NTP is configured, join the VM to the managed domain.

1. Use the `realm discover` command to discover the managed domain. The following example discovers the realm *AADDSCONTOSO.COM*. Specify your own managed domain name in ALL UPPERCASE:

   ```
   sudo realm discover AADDSCONTOSO.COM
   ```

   If the `realm discover` command can't find your managed domain, review the following troubleshooting steps:

   - Make sure that the domain is reachable from the VM. Try `ping aaddscontoso.com` to see if a positive reply is returned.
   - Check that the VM is deployed to the same, or a peered, virtual network in which the managed domain is available.
   - Confirm that the DNS server settings for the virtual network have been updated to point to the domain controllers of the managed domain.

2. Now initialize Kerberos using the `kinit` command. Specify a user that's a part of the managed domain. If needed, add a user account to a group in Azure AD.

   Again, the managed domain name must be entered in ALL UPPERCASE. In the following example, the account named `contosoadmin@aaddscontoso.com` is used to initialize Kerberos. Enter your own user account that's a part of the managed domain:

   ```
   kinit contosoadmin@AADDSCONTOSO.COM
   ```

3. Finally, join the VM to the managed domain using the `realm join` command. Use the same user account that's a part of the managed domain that you specified in the previous `kinit` command, such as `contosoadmin@AADDSCONTOSO.COM`:

   ```
   sudo realm join --verbose AADDSCONTOSO.COM -U 'contosoadmin@AADDSCONTOSO.COM' --install=/
   ```

It takes a few moments to join the VM to the managed domain. The following example output shows the VM has successfully joined to the managed domain:

```
Successfully enrolled machine in realm
```

If your VM can't successfully complete the domain-join process, make sure that the VM's network security group allows outbound Kerberos traffic on TCP + UDP port 464 to the virtual network subnet for your managed domain.

If you received the error *Unspecified GSS failure. Minor code may provide more information (Server not found in Kerberos database)*, open the file */etc/krb5.conf* and add the following code in `[libdefaults]` section and try again:

```
rdns=false
```

Update the SSSD configuration

One of the packages installed in a previous step was for System Security Services Daemon (SSSD). When a user tries to sign in to a VM using domain credentials, SSSD relays the request to an authentication provider. In this scenario, SSSD uses Azure AD DS to authenticate the request.

1. Open the *sssd.conf* file with an editor:

   ```
   sudo vi /etc/sssd/sssd.conf
   ```

2. Comment out the line for *use_fully_qualified_names* as follows:

   ```
   # use_fully_qualified_names = True
   ```

   When done, save and exit the *sssd.conf* file using the `:wq` command of the editor.

3. To apply the change, restart the SSSD service:

   ```
   sudo service sssd restart
   ```

Configure user account and group settings

With the VM joined to the managed domain and configured for authentication, there are a few user configuration options to complete. These configuration changes include allowing password-based authentication, and automatically creating home directories on the local VM when domain users first sign in.

Allow password authentication for SSH

By default, users can only sign in to a VM using SSH public key-based authentication. Password-based authentication fails. When you join the VM to a managed domain, those domain accounts need to use password-based authentication. Update the SSH configuration to allow password-based authentication as follows.

1. Open the *sshd_conf* file with an editor:

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

2. Update the line for *PasswordAuthentication* to *yes*:

   ```
   PasswordAuthentication yes
   ```

   When done, save and exit the *sshd_conf* file using the `:wq` command of the editor.

3. To apply the changes and let users sign in using a password, restart the SSH service:

   ```
   sudo systemctl restart ssh
   ```

Configure automatic home directory creation

To enable automatic creation of the home directory when a user first signs in, complete the following steps:

1. Open the */etc/pam.d/common-session* file in an editor:

```
sudo vi /etc/pam.d/common-session
```

2. Add the following line in this file below the line `session optional pam_sss.so`:

```
session required pam_mkhomedir.so skel=/etc/skel/ umask=0077
```

When done, save and exit the *common-session* file using the `:wq` command of the editor.

Grant the 'AAD DC Administrators' group sudo privileges

To grant members of the *AAD DC Administrators* group administrative privileges on the Ubuntu VM, you add an entry to the */etc/sudoers*. Once added, members of the *AAD DC Administrators* group can use the `sudo` command on the Ubuntu VM.

1. Open the *sudoers* file for editing:

```
sudo visudo
```

2. Add the following entry to the end of */etc/sudoers* file:

```
# Add 'AAD DC Administrators' group members as admins.
%AAD\ DC\ Administrators ALL=(ALL) NOPASSWD:ALL
```

When done, save and exit the editor using the `Ctrl-X` command.

Sign in to the VM using a domain account

To verify that the VM has been successfully joined to the managed domain, start a new SSH connection using a domain user account. Confirm that a home directory has been created, and that group membership from the domain is applied.

1. Create a new SSH connection from your console. Use a domain account that belongs to the managed domain using the `ssh -l` command, such as `contosoadmin@aaddscontoso.com` and then enter the address of your VM, such as *ubuntu.aaddscontoso.com*. If you use the Azure Cloud Shell, use the public IP address of the VM rather than the internal DNS name.

```
ssh -l contosoadmin@AADDSCONTOSO.com ubuntu.aaddscontoso.com
```

2. When you've successfully connected to the VM, verify that the home directory was initialized correctly:

```
pwd
```

You should be in the */home* directory with your own directory that matches the user account.

3. Now check that the group memberships are being resolved correctly:

```
id
```

You should see your group memberships from the managed domain.

4. If you signed in to the VM as a member of the *AAD DC Administrators* group, check that you can correctly use the `sudo` command:

```
sudo apt-get update
```

Next steps

If you have problems connecting the VM to the managed domain or signing in with a domain account, see Troubleshooting domain join issues.

Deploy Azure AD Application Proxy for secure access to internal applications in an Azure Active Directory Domain Services managed domain

7/20/2020 • 5 minutes to read • Edit Online

With Azure AD Domain Services (Azure AD DS), you can lift-and-shift legacy applications running on-premises into Azure. Azure Active Directory (AD) Application Proxy then helps you support remote workers by securely publishing those internal applications part of an Azure AD DS managed domain so they can be accessed over the internet.

If you're new to the Azure AD Application Proxy and want to learn more, see How to provide secure remote access to internal applications.

This article shows you how to create and configure an Azure AD Application Proxy connector to provide secure access to applications in a managed domain.

Before you begin

To complete this article, you need the following resources and privileges:

- An active Azure subscription.
  - If you don't have an Azure subscription, create an account.
- An Azure Active Directory tenant associated with your subscription, either synchronized with an on-premises directory or a cloud-only directory.
  - If needed, create an Azure Active Directory tenant or associate an Azure subscription with your account.
  - An **Azure AD Premium license** is required to use the Azure AD Application Proxy.
- An Azure Active Directory Domain Services managed domain enabled and configured in your Azure AD tenant.
  - If needed, create and configure an Azure Active Directory Domain Services managed domain.

Create a domain-joined Windows VM

To route traffic to applications running in your environment, you install the Azure AD Application Proxy connector component. This Azure AD Application Proxy connector must be installed on a Windows Server virtual machine (VM) that's joined to the managed domain. For some applications, you can deploy multiple servers that each have the connector installed. This deployment option gives you greater availability and helps handle heavier authentication loads.

The VM that runs the Azure AD Application Proxy connector must be on the same, or a peered, virtual network as your managed domain. The VMs that then host the applications you publish using the Application Proxy must also be deployed on the same Azure virtual network.

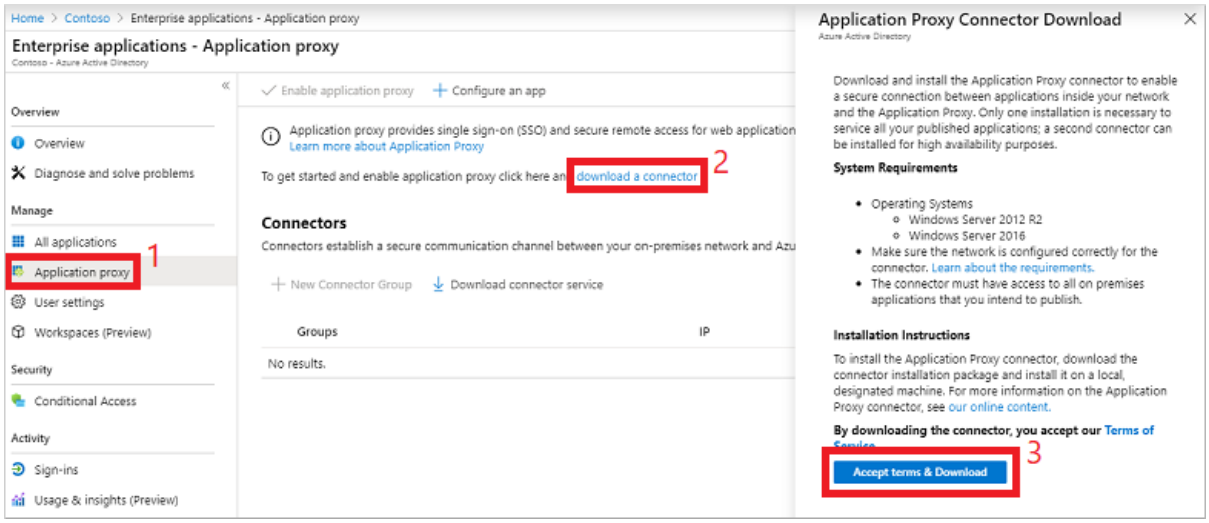
To create a VM for the Azure AD Application Proxy connector, complete the following steps:

1. Create a custom OU. You can delegate permissions to manage this custom OU to users within the managed domain. The VMs for Azure AD Application Proxy and that run your applications must be a part of the custom OU, not the default *AAD DC Computers* OU.
2. Domain-join the virtual machines, both the one that runs the Azure AD Application Proxy connector, and the ones that run your applications, to the managed domain. Create these computer accounts in the custom OU from the previous step.

Download the Azure AD Application Proxy connector

Perform the following steps to download the Azure AD Application Proxy connector. The setup file you download is copied to your App Proxy VM in the next section.

1. Sign in to the Azure portal with a user account that has *Enterprise administrator* permissions in Azure AD.
2. Search for and select **Azure Active Directory** at the top of the portal, then choose **Enterprise applications**.
3. Select **Application proxy** from the menu on the left-hand side. To create your first connector and enable App Proxy, select the link to **download a connector**.
4. On the download page, accept the license terms and privacy agreement, then select **Accept terms & Download**.



Install and register the Azure AD Application Proxy connector

With a VM ready to be used as the Azure AD Application Proxy connector, now copy and run the setup file downloaded from the Azure portal.

1. Copy the Azure AD Application Proxy connector setup file to your VM.
2. Run the setup file, such as *AADApplicationProxyConnectorInstaller.exe*. Accept the software license terms.
3. During the install, you're prompted to register the connector with the Application Proxy in your Azure AD directory.

   - Provide the credentials for a global administrator in your Azure AD directory. The Azure AD global administrator credentials may be different from your Azure credentials in the portal

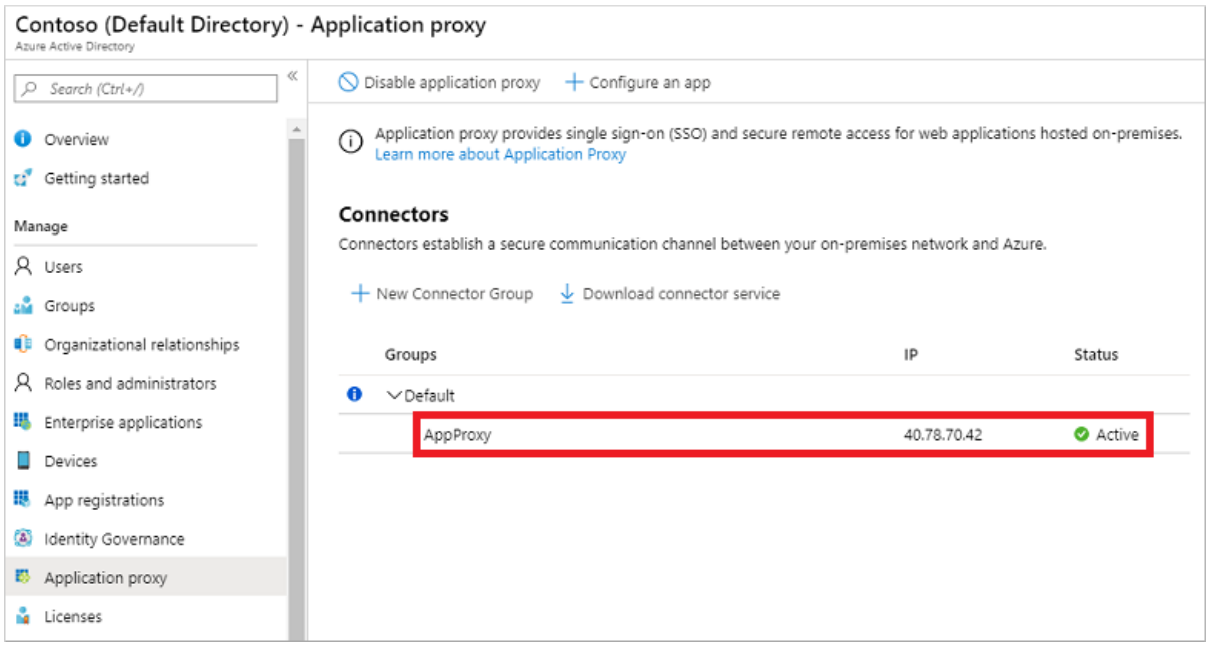
     > **NOTE**
     >
     > The global administrator account used to register the connector must belong to the same directory where you enable the Application Proxy service.
     >
     > For example, if the Azure AD domain is *contoso.com*, the global administrator should be `admin@contoso.com` or another valid alias on that domain.

   - If Internet Explorer Enhanced Security Configuration is turned on for the VM where you install the connector, the registration screen might be blocked. To allow access, follow the instructions in the error message, or turn off Internet Explorer Enhanced Security during the install process.

- If connector registration fails, see Troubleshoot Application Proxy.

4. At the end of the setup, a note is shown for environments with an outbound proxy. To configure the Azure AD Application Proxy connector to work through the outbound proxy, run the provided script, such as `C:\Program Files\Microsoft AAD App Proxy connector\ConfigureOutBoundProxy.ps1`.

5. On the Application proxy page in the Azure portal, the new connector is listed with a status of *Active*, as shown in the following example:



> **NOTE**
> To provide high availability for applications authenticating through the Azure AD Application Proxy, you can install connectors on multiple VMs. Repeat the same steps listed in the previous section to install the connector on other servers joined to the managed domain.

Enable resource-based Kerberos constrained delegation

If you want to use single sign-on to your applications using Integrated Windows Authentication (IWA), grant the Azure AD Application Proxy connectors permission to impersonate users and send and receive tokens on their behalf. To grant these permissions, you configure Kerberos constrained delegation (KCD) for the connector to access resources on the managed domain. As you don't have domain administrator privileges in a managed domain, traditional account-level KCD cannot be configured on a managed domain. Instead, use resource-based KCD.

For more information, see Configure Kerberos constrained delegation (KCD) in Azure Active Directory Domain Services.

> **NOTE**
> You must be signed in to a user account that's a member of the *Azure AD DC administrators* group in your Azure AD tenant to run the following PowerShell cmdlets.
>
> The computer accounts for your App Proxy connector VM and application VMs must be in a custom OU where you have permissions to configure resource-based KCD. You can't configure resource-based KCD for a computer account in the built-in *AAD DC Computers* container.

Use the Get-ADComputer to retrieve the settings for the computer on which the Azure AD Application Proxy connector is installed. From your domain-joined management VM and logged in as user account that's a member

of the *Azure AD DC administrators* group, run the following cmdlets.

The following example gets information about the computer account named *appproxy.aaddscontoso.com*. Provide your own computer name for the Azure AD Application Proxy VM configured in the previous steps.

```
$ImpersonatingAccount = Get-ADComputer -Identity appproxy.aaddscontoso.com
```

For each application server that runs the apps behind Azure AD Application Proxy use the Set-ADComputer PowerShell cmdlet to configure resource-based KCD. In the following example, the Azure AD Application Proxy connector is granted permissions to use the *appserver.aaddscontoso.com* computer:

```
Set-ADComputer appserver.aaddscontoso.com -PrincipalsAllowedToDelegateToAccount $ImpersonatingAccount
```

If you deploy multiple Azure AD Application Proxy connectors, you must configure resource-based KCD for each connector instance.

Next steps

With the Azure AD Application Proxy integrated with Azure AD DS, publish applications for users to access. For more information, see publish applications using Azure AD Application Proxy.

Configure Azure Active Directory Domain Services to support user profile synchronization for SharePoint Server

7/20/2020 • 2 minutes to read • Edit Online

SharePoint Server includes a service to synchronize user profiles. This feature allows user profiles to be stored in a central location and accessible across multiple SharePoint sites and farms. To configure the SharePoint Server user profile service, the appropriate permissions must be granted in an Azure Active Directory Domain Services (Azure AD DS) managed domain. For more information, see user profile synchronization in SharePoint Server.

This article shows you how to configure Azure AD DS to allow the SharePoint Server user profile sync service.

Before you begin

To complete this article, you need the following resources and privileges:

- An active Azure subscription.
  - If you don't have an Azure subscription, create an account.
- An Azure Active Directory tenant associated with your subscription, either synchronized with an on-premises directory or a cloud-only directory.
  - If needed, create an Azure Active Directory tenant or associate an Azure subscription with your account.
- An Azure Active Directory Domain Services managed domain enabled and configured in your Azure AD tenant.
  - If needed, complete the tutorial to create and configure an Azure Active Directory Domain Services managed domain.
- A Windows Server management VM that is joined to the Azure AD DS managed domain.
  - If needed, complete the tutorial to create a management VM.
- A user account that's a member of the *Azure AD DC administrators* group in your Azure AD tenant.
- A SharePoint service account for the user profile synchronization service.
  - If needed, see Plan for administrative and service accounts in SharePoint Server.

Service accounts overview

In a managed domain, a security group named *AAD DC Service Accounts* exists as part of the *Users* organizational unit (OU). Members of this security group are delegated the following privileges:

- **Replicate Directory Changes** privilege on the root DSE.
- **Replicate Directory Changes** privilege on the *Configuration* naming context (`cn=configuration` container).

The *AAD DC Service Accounts* security group is also a member of the built-in group *Pre-Windows 2000 Compatible Access*.

When added to this security group, the service account for SharePoint Server user profile synchronization service is granted the required privileges to work correctly.

Enable support for SharePoint Server user profile sync

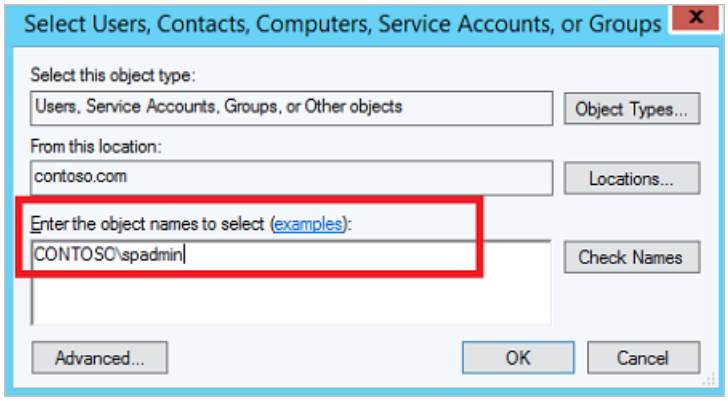
The service account for SharePoint Server needs adequate privileges to replicate changes to the directory and let SharePoint Server user profile sync work correctly. To provide these privileges, add the service account used for

SharePoint user profile synchronization to the *AAD DC Service Accounts* group.

From your Azure AD DS management VM, complete the following steps:

> **NOTE**
> To edit group membership in a managed domain, you must be signed in to a user account that's a member of the *AAD DC Administrators* group.

1. From the Start screen, select **Administrative Tools**. A list of available management tools is shown that were installed in the tutorial to create a management VM.
2. To manage group membership, select **Active Directory Administrative Center** from the list of administrative tools.
3. In the left pane, choose your managed domain, such as *aaddscontoso.com*. A list of existing OUs and resources is shown.
4. Select the **Users** OU, then choose the *AAD DC Service Accounts* security group.
5. Select **Members**, then choose **Add...**.
6. Enter the name of the SharePoint service account, then select **OK**. In the following example, the SharePoint service account is named *spadmin*:



Next steps

For more information, see Grant Active Directory Domain Services permissions for profile synchronization in SharePoint Server

Common errors and troubleshooting steps for Azure Active Directory Domain Services

7/20/2020 • 8 minutes to read • Edit Online

As a central part of identity and authentication for applications, Azure Active Directory Domain Services (Azure AD DS) sometimes has problems. If you run into issues, there are some common error messages and associated troubleshooting steps to help you get things running again. At any time, you can also open an Azure support request for additional troubleshooting assistance.

This article provides troubleshooting steps for common issues in Azure AD DS.

You cannot enable Azure AD Domain Services for your Azure AD directory

If you have problems enabling Azure AD DS, review the following common errors and steps to resolve them:

SAMPLE ERROR MESSAGE	RESOLUTION
The name aaddscontoso.com is already in use on this network. Specify a name that is not in use.	Domain name conflict in the virtual network
Domain Services could not be enabled in this Azure AD tenant. The service does not have adequate permissions to the application called 'Azure AD Domain Services Sync'. Delete the application called 'Azure AD Domain Services Sync' and then try to enable Domain Services for your Azure AD tenant.	Domain Services doesn't have adequate permissions to the Azure AD Domain Services Sync application
Domain Services could not be enabled in this Azure AD tenant. The Domain Services application in your Azure AD tenant does not have the required permissions to enable Domain Services. Delete the application with the application identifier d87dcbc6-a371-462e-88e3-28ad15ec4e64 and then try to enable Domain Services for your Azure AD tenant.	The Domain Services application isn't configured properly in your Azure AD tenant
Domain Services could not be enabled in this Azure AD tenant. The Microsoft Azure AD application is disabled in your Azure AD tenant. Enable the application with the application identifier 00000002-0000-0000-c000-000000000000 and then try to enable Domain Services for your Azure AD tenant.	The Microsoft Graph application is disabled in your Azure AD tenant

Domain Name conflict

Error message

The name aaddscontoso.com is already in use on this network. Specify a name that is not in use.

Resolution

Check that you don't have an existing AD DS environment with the same domain name on the same, or a peered, virtual network. For example, you may have an AD DS domain named *aaddscontoso.com* that runs on Azure VMs. When you try to enable an Azure AD DS managed domain with the same domain name of *aaddscontoso.com* on the virtual network, the requested operation fails.

This failure is due to name conflicts for the domain name on the virtual network. A DNS lookup checks if an

existing AD DS environment responds on the requested domain name. To resolve this failure, use a different name to set up your managed domain, or de-provision the existing AD DS domain and then try again to enable Azure AD DS.

Inadequate permissions

Error message

Domain Services could not be enabled in this Azure AD tenant. The service does not have adequate permissions to the application called 'Azure AD Domain Services Sync'. Delete the application called 'Azure AD Domain Services Sync' and then try to enable Domain Services for your Azure AD tenant.

Resolution

Check if there's an application named *Azure AD Domain Services Sync* in your Azure AD directory. If this application exists, delete it, then try again to enable Azure AD DS. To check for an existing application and delete it if needed, complete the following steps:

1. In the Azure portal, select **Azure Active Directory** from the left-hand navigation menu.
2. Select **Enterprise applications**. Choose *All applications* from the **Application Type** drop-down menu, then select **Apply**.
3. In the search box, enter *Azure AD Domain Services Sync*. If the application exists, select it and choose **Delete**.
4. Once you've deleted the application, try to enable Azure AD DS again.

Invalid configuration

Error message

Domain Services could not be enabled in this Azure AD tenant. The Domain Services application in your Azure AD tenant does not have the required permissions to enable Domain Services. Delete the application with the application identifier d87dcbc6-a371-462e-88e3-28ad15ec4e64 and then try to enable Domain Services for your Azure AD tenant.

Resolution

Check if you have an existing application named *AzureActiveDirectoryDomainControllerServices* with an application identifier of *d87dcbc6-a371-462e-88e3-28ad15ec4e64* in your Azure AD directory. If this application exists, delete it and then try again to enable Azure AD DS.

Use the following PowerShell script to search for an existing application instance and delete it if needed:

```
$InformationPreference = "Continue"
$WarningPreference = "Continue"

$aadDsSp = Get-AzureADServicePrincipal -Filter "AppId eq 'd87dcbc6-a371-462e-88e3-28ad15ec4e64'" -ErrorAction
Ignore
if ($aadDsSp -ne $null)
{
    Write-Information "Found Azure AD Domain Services application. Deleting it ..."
    Remove-AzureADServicePrincipal -ObjectId $aadDsSp.ObjectId
    Write-Information "Deleted the Azure AD Domain Services application."
}

$identifierUri = "https://sync.aaddc.activedirectory.windowsazure.com"
$appFilter = "IdentifierUris eq '" + $identifierUri + "'"
$app = Get-AzureADApplication -Filter $appFilter
if ($app -ne $null)
{
    Write-Information "Found Azure AD Domain Services Sync application. Deleting it ..."
    Remove-AzureADApplication -ObjectId $app.ObjectId
    Write-Information "Deleted the Azure AD Domain Services Sync application."
}

$spFilter = "ServicePrincipalNames eq '" + $identifierUri + "'"
$sp = Get-AzureADServicePrincipal -Filter $spFilter
if ($sp -ne $null)
{
    Write-Information "Found Azure AD Domain Services Sync service principal. Deleting it ..."
    Remove-AzureADServicePrincipal -ObjectId $sp.ObjectId
    Write-Information "Deleted the Azure AD Domain Services Sync service principal."
}
```

Microsoft Graph disabled

Error message

Domain Services could not be enabled in this Azure AD tenant. The Microsoft Azure AD application is disabled in your Azure AD tenant. Enable the application with the application identifier 00000002-0000-0000-c000-000000000000 and then try to enable Domain Services for your Azure AD tenant.

Resolution

Check if you've disabled an application with the identifier *00000002-0000-0000-c000-000000000000*. This application is the Microsoft Azure AD application and provides Graph API access to your Azure AD tenant. To synchronize your Azure AD tenant, this application must be enabled.

To check the status of this application and enable it if needed, complete the following steps:

1. In the Azure portal, select **Azure Active Directory** from the left-hand navigation menu.
2. Select **Enterprise applications**. Choose *All applications* from the **Application Type** drop-down menu, then select **Apply**.
3. In the search box, enter *00000002-0000-0000-c000-00000000000*. Select the application, then choose **Properties**.
4. If **Enabled for users to sign-in** is set to *No*, set the value to *Yes*, then select **Save**.
5. Once you've enabled the application, try to enable Azure AD DS again.

Users are unable to sign in to the Azure AD Domain Services managed domain

If one or more users in your Azure AD tenant can't sign in to the managed domain, complete the following troubleshooting steps:

- **Credentials format** - Try using the UPN format to specify credentials, such as `dee@aaddscontoso.onmicrosoft.com`. The UPN format is the recommended way to specify credentials in Azure AD DS. Make sure this UPN is configured correctly in Azure AD.

  The *SAMAccountName* for your account, such as *AADDSCONTOSO\driley* may be autogenerated if there are multiple users with the same UPN prefix in your tenant or if your UPN prefix is overly long. Therefore, the *SAMAccountName* format for your account may be different from what you expect or use in your on-premises domain.

- **Password synchronization** - Make sure that you've enabled password synchronization for cloud-only users or for hybrid environments using Azure AD Connect.

  - **Hybrid synchronized accounts:** If the affected user accounts are synchronized from an on-premises directory, verify the following areas:

    - You've deployed, or updated to, the latest recommended release of Azure AD Connect.
    - You've configured Azure AD Connect to perform a full synchronization.
    - Depending on the size of your directory, it may take a while for user accounts and credential hashes to be available in the managed domain. Make sure you wait long enough before trying to authenticate against the managed domain.
    - If the issue persists after verifying the previous steps, try restarting the *Microsoft Azure AD Sync Service*. From your Azure AD Connect server, open a command prompt, then run the following commands:

      ```
      net stop 'Microsoft Azure AD Sync'
      net start 'Microsoft Azure AD Sync'
      ```

  - **Cloud-only accounts**: If the affected user account is a cloud-only user account, make sure that the user has changed their password after you enabled Azure AD DS. This password reset causes the required credential hashes for the managed domain to be generated.

- **Verify the user account is active**: By default, five invalid password attempts within 2 minutes on the managed domain cause a user account to be locked out for 30 minutes. The user can't sign in while the account is locked out. After 30 minutes, the user account is automatically unlocked.

  - Invalid password attempts on the managed domain don't lock out the user account in Azure AD. The user account is locked out only within the managed domain. Check the user account status in the *Active Directory Administrative Console (ADAC)* using the management VM, not in Azure AD.
  - You can also configure fine grained password policies to change the default lockout threshold and duration.

- **External accounts** - Check that the affected user account isn't an external account in the Azure AD tenant. Examples of external accounts include Microsoft accounts like `dee@live.com` or user accounts from an external Azure AD directory. Azure AD DS doesn't store credentials for external user accounts so they can't sign in to the managed domain.

There are one or more alerts on your managed domain

If there are active alerts on the managed domain, it may prevent the authentication process from working correctly.

To see if there are any active alerts, check the health status of a managed domain. If any alerts are shown, troubleshoot and resolve them.

Users removed from your Azure AD tenant are not removed from your

managed domain

Azure AD protects against accidental deletion of user objects. When you delete a user account from an Azure AD tenant, the corresponding user object is moved to the recycle bin. When this delete operation is synchronized to your managed domain, the corresponding user account is marked as disabled. This feature helps you recover, or undelete, the user account.

The user account remains in the disabled state in the managed domain, even if you re-create a user account with the same UPN in the Azure AD directory. To remove the user account from the managed domain, you need to forcibly delete it from the Azure AD tenant.

To fully remove a user account from a managed domain, delete the user permanently from your Azure AD tenant using the Remove-MsolUser PowerShell cmdlet with the `-RemoveFromRecycleBin` parameter.

Next steps

If you continue to have issues, open an Azure support request for additional troubleshooting assistance.

Troubleshoot domain-join problems with an Azure Active Directory Domain Services managed domain

7/20/2020 • 3 minutes to read • Edit Online

When you try to join a virtual machine (VM) or connect an application to an Azure Active Directory Domain Services (Azure AD DS) managed domain, you may get an error that you're unable to do so. To troubleshoot domain-join problems, review at which of the following points you have an issue:

- If you don't receive an authentication prompt, the VM or application can't connect to the Azure AD DS managed domain.
  - Start to troubleshoot connectivity issues for domain-join.
- If you receive an error during authentication, the connection to the managed domain is successful.
  - Start to troubleshoot credentials-related issues during domain-join.

Connectivity issues for domain-join

If the VM can't find the managed domain, there's usually a network connection or configuration issue. Review the following troubleshooting steps to locate and resolve the issue:

1. Ensure the VM is connected to the same, or a peered, virtual network as the managed domain. If not, the VM can't find and connect to the domain in order to join.
   - If the VM isn't connected to the same virtual network, confirm that the virtual networking peering or VPN connection is *Active* or *Connected* to allow the traffic to flow correctly.
2. Try to ping the domain using the domain name of the managed domain, such as `ping aaddscontoso.com`.
   - If the ping response fails, try to ping the IP addresses for the domain displayed on the overview page in the portal for your managed domain, such as `ping 10.0.0.4`.
   - If you can successfully ping the IP address but not the domain, DNS may be incorrectly configured. Make sure that you've configured the managed domain DNS servers for the virtual network.
3. Try flushing the DNS resolver cache on the virtual machine, such as `ipconfig /flushdns`.

Network Security Group (NSG) configuration

When you create a managed domain, a network security group is also created with the appropriate rules for successful domain operation. If you edit or create additional network security group rules, you may unintentionally block ports required for Azure AD DS to provide connection and authentication services. These network security group rules can cause issues such as password sync not completing, users not being able to sign in, or domain-join issues.

If you continue to have connection issues, review the following troubleshooting steps:

1. Check the health status of your managed domain in the Azure portal. If you have an alert for *AADDS001*, a network security group rule is blocking access.
2. Review the required ports and network security group rules. Make sure that no network security group rules applied to the VM or virtual network you're connecting from block these network ports.
3. Once any network security group configuration issues are resolved, the *AADDS001* alert disappears from the health page in about 2 hours. With network connectivity now available, try to domain-join the VM again.

Credentials-related issues during domain-join

If you get a dialog box that asks for credentials to join the managed domain, the VM is able to connect to the

domain using the Azure virtual network. The domain-join process fails on authenticating to the domain or authorization to complete the domain-join process using the credentials provides.

To troubleshoot credentials-related issues, review the following troubleshooting steps:

1. Try using the UPN format to specify credentials, such as `dee@contoso.onmicrosoft.com`. Make sure that this UPN is configured correctly in Azure AD.
   - The *SAMAccountName* for your account may be autogenerated if there are multiple users with the same UPN prefix in your tenant or if your UPN prefix is overly long. Therefore, the *SAMAccountName* format for your account may be different from what you expect or use in your on-premises domain.
2. Try to use the credentials for a user account that's a part of the managed domain to join VMs to the managed domain.
3. Make sure that you've enabled password synchronization and waited long enough for the initial password sync to complete.

Next steps

For a deeper understanding of the Active Directory processes as part of the domain-join operation, see Join and authentication issues.

If you still have problems joining your VM to the managed domain, find help and open a support ticket for Azure Active Directory.

Troubleshoot account lockout problems with an Azure Active Directory Domain Services managed domain

7/20/2020 • 3 minutes to read • Edit Online

To prevent repeated malicious sign-in attempts, an Azure Active Directory Domain Services (Azure AD DS) managed domain locks accounts after a defined threshold. This account lockout can also happen by accident without a sign-in attack incident. For example, if a user repeatedly enters the wrong password or a service attempts to use an old password, the account gets locked out.

This troubleshooting article outlines why account lockouts happen and how you can configure the behavior, and how to review security audits to troubleshoot lockout events.

What is an account lockout?

A user account in an Azure AD DS managed domain is locked out when a defined threshold for unsuccessful sign-in attempts has been met. This account lockout behavior is designed to protect you from repeated brute-force sign-in attempts that may indicate an automated digital attack.

By default, if there are 5 bad password attempts in 2 minutes, the account is locked out for 30 minutes.

The default account lockout thresholds are configured using fine-grained password policy. If you have a specific set of requirements, you can override these default account lockout thresholds. However, it's not recommended to increase the threshold limits to try to reduce the number account lockouts. Troubleshoot the source of the account lockout behavior first.

Fine-grained password policy

Fine-grained password policies (FGPPs) let you apply specific restrictions for password and account lockout policies to different users in a domain. FGPP only affects users within a managed domain. Cloud users and domain users synchronized into the managed domain from Azure AD are only affected by the password policies within the managed domain. Their accounts in Azure AD or an on-premises directory aren't impacted.

Policies are distributed through group association in the managed domain, and any changes you make are applied at the next user sign-in. Changing the policy doesn't unlock a user account that's already locked out.

For more information on fine-grained password policies, and the differences between users created directly in Azure AD DS versus synchronized in from Azure AD, see Configure password and account lockout policies.

Common account lockout reasons

The most common reasons for an account to be locked out, without any malicious intent or factors, include the following scenarios:

- **The user locked themselves out.**
  - After a recent password change, has the user continued to use a previous password? The default account lockout policy of five failed attempts in 2 minutes can be caused by the user inadvertently retrying an old password.
- **There's an application or service that has an old password.**
  - If an account is used by applications or services, those resources may repeatedly try to sign in using an

old password. This behavior causes the account to be locked out.
    - Try to minimize account use across multiple different applications or services, and record where credentials are used. If an account password is changed, update the associated applications or services accordingly.
- **Password has been changed in a different environment and the new password hasn't synchronized yet.**
    - If an account password is changed outside of the managed domain, such as in an on-prem AD DS environment, it can take a few minutes for the password change to synchronize through Azure AD and into the managed domain.
    - A user that tries to sign in to a resource in the managed domain before that password synchronization process has completed causes their account to be locked out.

Troubleshoot account lockouts with security audits

To troubleshoot when account lockout events occur and where they're coming from, enable security audits for Azure AD DS. Audit events are only captured from the time you enable the feature. Ideally, you should enable security audits *before* there's an account lockout issue to troubleshoot. If a user account repeatedly has lockout issues, you can enable security audits ready for the next time the situation occurs.

Once you have enabled security audits, the following sample queries show you how to review *Account Lockout Events*, code *4740*.

View all the account lockout events for the last seven days:

```
AADDomainServicesAccountManagement
| where TimeGenerated >= ago(7d)
| where OperationName has "4740"
```

View all the account lockout events for the last seven days for the account named *driley*.

```
AADDomainServicesAccountLogon
| where TimeGenerated >= ago(7d)
| where OperationName has "4740"
| where "driley" == tolower(extract("Logon Account:\t(.+[0-9A-Za-z])",1,tostring(ResultDescription)))
```

View all the account lockout events between June 26, 2020 at 9 a.m. and July 1, 2020 midnight, sorted ascending by the date and time:

```
AADDomainServicesAccountManagement
| where TimeGenerated >= datetime(2020-06-26 09:00) and TimeGenerated <= datetime(2020-07-01)
| where OperationName has "4740"
| sort by TimeGenerated asc
```

Next steps

For more information on fine-grained password policies to adjust account lockout thresholds, see Configure password and account lockout policies.

If you still have problems joining your VM to the managed domain, find help and open a support ticket for Azure Active Directory.

Troubleshoot account sign-in problems with an Azure Active Directory Domain Services managed domain

7/20/2020 • 3 minutes to read • Edit Online

The most common reasons for a user account that can't sign in to an Azure Active Directory Domain Services (Azure AD DS) managed domain include the following scenarios:

- The account isn't synchronized into Azure AD DS yet.
- Azure AD DS doesn't have the password hashes to let the account sign in.
- The account is locked out.

> **TIP**
> Azure AD DS can't synchronize in credentials for accounts that are external to the Azure AD tenant. External users can't sign in to the Azure AD DS managed domain.

Account isn't synchronized into Azure AD DS yet

Depending on the size of your directory, it may take a while for user accounts and credential hashes to be available in a managed domain. For large directories, this initial one-way sync from Azure AD can take few hours, and up to a day or two. Make sure that you wait long enough before retrying authentication.

For hybrid environments that user Azure AD Connect to synchronize on-premises directory data into Azure AD, make sure that you run the latest version of Azure AD Connect and have configured Azure AD Connect to perform a full synchronization after enabling Azure AD DS. If you disable Azure AD DS and then re-enable, you have to follow these steps again.

If you continue to have issues with accounts not synchronizing through Azure AD Connect, restart the Azure AD Sync Service. From the computer with Azure AD Connect installed, open a command prompt window, then run the following commands:

```
net stop 'Microsoft Azure AD Sync'
net start 'Microsoft Azure AD Sync'
```

Azure AD DS doesn't have the password hashes

Azure AD doesn't generate or store password hashes in the format that's required for NTLM or Kerberos authentication until you enable Azure AD DS for your tenant. For security reasons, Azure AD also doesn't store any password credentials in clear-text form. Therefore, Azure AD can't automatically generate these NTLM or Kerberos password hashes based on users' existing credentials.

Hybrid environments with on-premises synchronization

For hybrid environments using Azure AD Connect to synchronize from an on-premises AD DS environment, you can locally generate and synchronize the required NTLM or Kerberos password hashes into Azure AD. After you create your managed domain, enable password hash synchronization to Azure Active Directory Domain Services. Without completing this password hash synchronization step, you can't sign in to an account using the managed domain. If you disable Azure AD DS and then re-enable, you have to follow those steps again.

For more information, see How password hash synchronization works for Azure AD DS.

Cloud-only environments with no on-premises synchronization

Managed domains with no on-premises synchronization, only accounts in Azure AD, also need to generate the required NTLM or Kerberos password hashes. If a cloud-only account can't sign in, has a password change process successfully completed for the account after enabling Azure AD DS?

- **No, the password has not been changed.**
  - Change the password for the account to generate the required password hashes, then wait for 15 minutes before you try to sign in again.
  - If you disable Azure AD DS and then re-enable, each account must follow the steps again to change their password and generate the required password hashes.
- **Yes, the password has been changed.**
  - Try to sign in using the *UPN* format, such as `driley@aaddscontoso.com`, instead of the *SAMAccountName* format like `AADDSCONTOSO\deeriley`.
  - The *SAMAccountName* may be automatically generated for users whose UPN prefix is overly long or is the same as another user on the managed domain. The *UPN* format is guaranteed to be unique within an Azure AD tenant.

The account is locked out

A user account in a managed domain is locked out when a defined threshold for unsuccessful sign-in attempts has been met. This account lockout behavior is designed to protect you from repeated brute-force sign-in attempts that may indicate an automated digital attack.

By default, if there are 5 bad password attempts in 2 minutes, the account is locked out for 30 minutes.

For more information and how to resolve account lockout issues, see Troubleshoot account lockout problems in Azure AD DS.

Next steps

If you still have problems joining your VM to the managed domain, find help and open a support ticket for Azure Active Directory.

Resolve mismatched directory errors for existing Azure Active Directory Domain Services managed domains

7/20/2020 • 2 minutes to read • Edit Online

If an Azure Active Directory Domain Services (Azure AD DS) managed domain shows a mismatched tenant error, you can't administer the managed domain until resolved. This error occurs if the underlying Azure virtual network is moved to a different Azure AD directory.

This article explains why the error occurs and how to resolve it.

What causes this error?

A mismatched directory error happens when an Azure AD DS managed domain and virtual network belong to two different Azure AD tenants. For example, you may have a managed domain called *aaddscontoso.com* that runs in Contoso's Azure AD tenant. However, the Azure virtual network for managed domain is part of the Fabrikam Azure AD tenant.

Azure uses role-based access control (RBAC) to limit access to resources. When you enable Azure AD DS in an Azure AD tenant, credential hashes are synchronized to the managed domain. This operation requires you to be a tenant admin for the Azure AD directory, and access to the credentials must be controlled.

To deploy resources to an Azure virtual network and control traffic, you must have administrative privileges on the virtual network in which you deploy the managed domain.

For RBAC to work consistently and secure access to all the resources Azure AD DS uses, the managed domain and the virtual network must belong to the same Azure AD tenant.

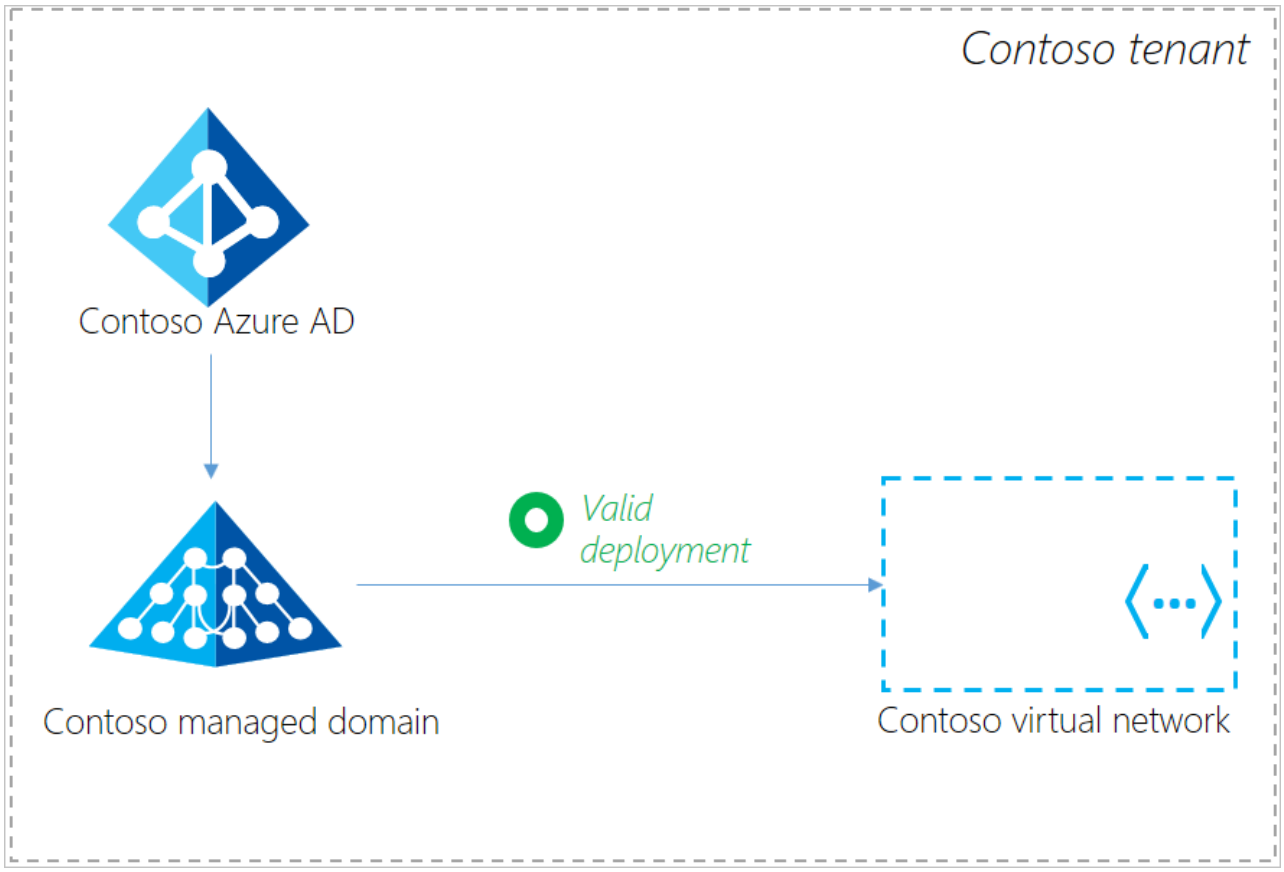
The following rules apply for deployments:

- An Azure AD directory may have multiple Azure subscriptions.
- An Azure subscription may have multiple resources such as virtual networks.
- A single managed domain is enabled for an Azure AD directory.
- A managed domain can be enabled on a virtual network belonging to any of the Azure subscriptions within the same Azure AD tenant.

Valid configuration

In the following example deployment scenario, the Contoso managed domain is enabled in the Contoso Azure AD tenant. The managed domain is deployed in a virtual network that belongs to an Azure subscription owned by the Contoso Azure AD tenant.

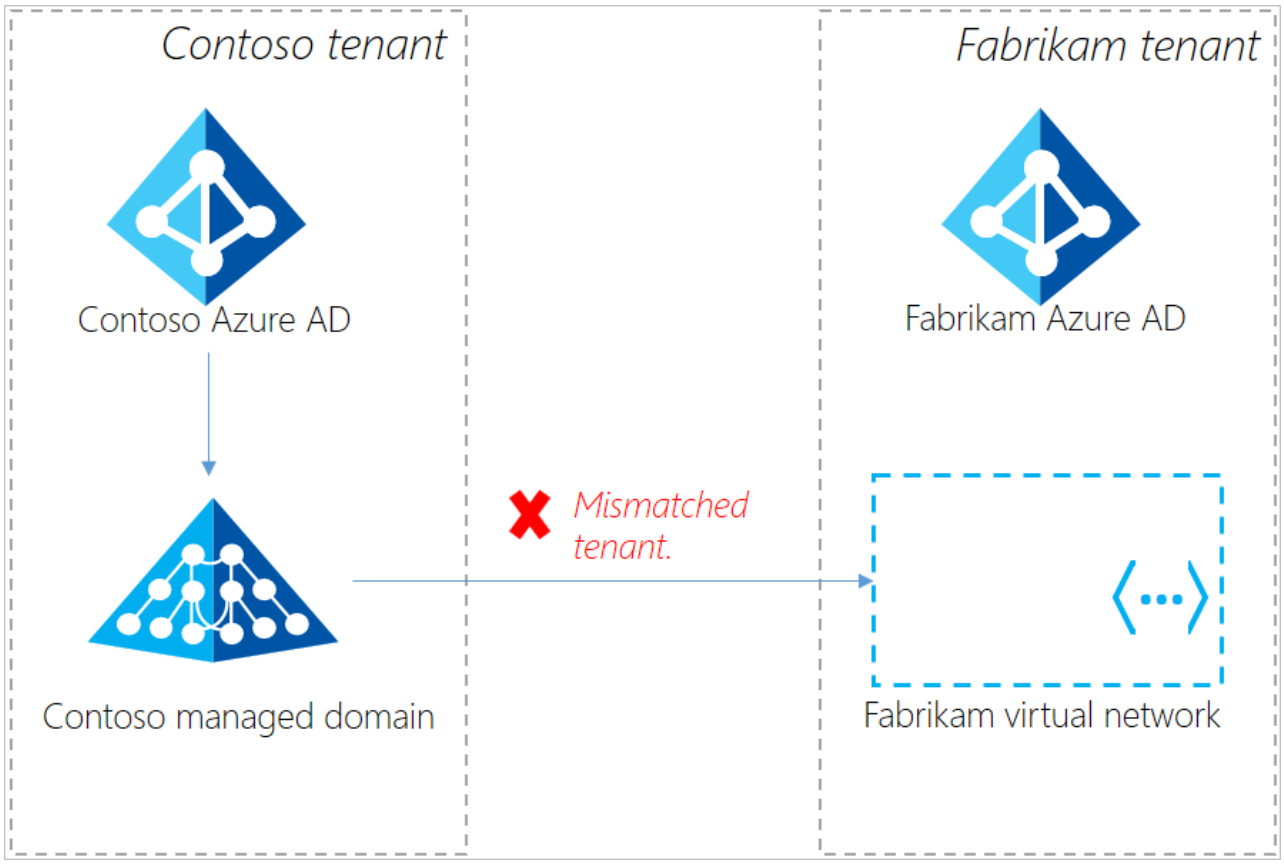
Both the managed domain and the virtual network belong to the same Azure AD tenant. This example configuration is valid and fully supported.



Mismatched tenant configuration

In this example deployment scenario, the Contoso managed domain is enabled in the Contoso Azure AD tenant. However, the managed domain is deployed in a virtual network that belongs to an Azure subscription owned by the Fabrikam Azure AD tenant.

The managed domain and the virtual network belong to two different Azure AD tenants. This example configuration is a mismatched tenant and isn't supported. The virtual network must be moved to the same Azure AD tenant as the managed domain.



Resolve mismatched tenant error

The following two options resolve the mismatched directory error:

- First, delete the managed domain from your existing Azure AD directory. Then, create a replacement managed domain in the same Azure AD directory as the virtual network you wish to use. When ready, join all machines previously joined to the deleted domain to the recreated managed domain.
- Move the Azure subscription containing the virtual network to the same Azure AD directory as the managed domain.

Next steps

For more information on troubleshooting issues with Azure AD DS, see the troubleshooting guide.

Understand the health states and resolve suspended domains in Azure Active Directory Domain Services

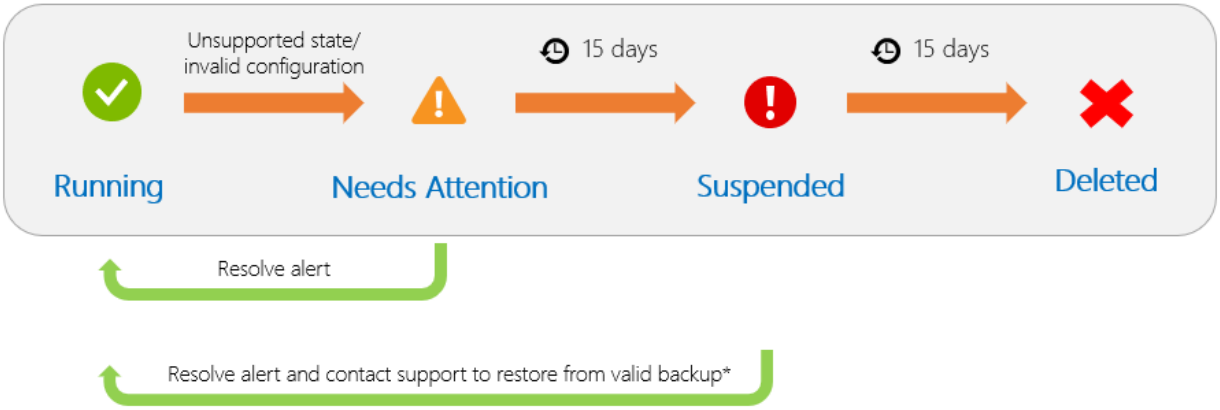
7/20/2020 • 4 minutes to read • Edit Online

When Azure Active Directory Domain Services (Azure AD DS) is unable to service a managed domain for a long period of time, it puts the managed domain into a suspended state. If a managed domain remains in a suspended state, it's automatically deleted. To keep your Azure AD DS managed domain healthy and avoid suspension, resolve any alerts as quickly as you can.

This article explains why managed domains are suspended, and how to recover a suspended domain.

Overview of managed domain states

Through the lifecycle of a managed domain, there are different states that indicate its health. If the managed domain reports an issue, quickly resolve the underlying cause to stop the state from continuing to degrade.



A managed domain can be in one of the following states:

- Running
- Needs attention
- Suspended
- Deleted

Running state

A managed domain that's configured correctly and without problems is in the *Running* state. This is the desired state for a managed domain.

What to expect

- The Azure platform can regularly monitor the health of the managed domain.
- Domain controllers for the managed domain are patched and updated regularly.
- Changes from Azure Active Directory are regularly synchronized to the managed domain.
- Regular backups are taken for the managed domain.

Needs Attention state

A managed domain with one or more issues that need to be fixed is in the *Needs attention* state. The health page

for the managed domain lists the alerts, and indicate where there's a problem.

Some alerts are transient and are automatically resolved by the Azure platform. For other alerts, you can fix the issue by following the resolution steps provided. It there's a critical alert, open an Azure support request for additional troubleshooting assistance.

One example of an alert is when there's a restrictive network security group. In this configuration, the Azure platform may not be able to update and monitor the managed domain. An alert is generated, and the state changes to *Needs attention*.

For more information, see How to troubleshoot alerts for a managed domain.

What to expect

When a managed domain is in the *Needs Attention* state, the Azure platform may not be able to monitor, patch, update, or back-up data on a regular basis. In some cases, like an invalid network configuration, the domain controllers for the managed domain may be unreachable.

- The managed domain is in an unhealthy state and ongoing health monitoring may stop until the alert is resolved.
- Domain controllers for the managed domain can't be patched or updated.
- Changes from Azure Active Directory may not be synchronized to the managed domain.
- Backups for the managed domain may not be taken.
- If you resolve non-critical alerts that are impacting the managed domain, the health should return to the *Running* state.
- Critical alerts are triggered for configuration issues where the Azure platform can't reach the domain controllers. If these critical alerts aren't resolved within 15 days, the managed domain enters the *Suspended* state.

Suspended state

A managed domain enters the **Suspended** state for one of the following reasons:

- One or more critical alerts haven't been resolved in 15 days.
  - Critical alerts can be caused by a misconfiguration that blocks access to resources that are needed by Azure AD DS. For example, the alert AADDS104: Network Error has been unresolved for more than 15 days in the managed domain.
- There's a billing issue with the Azure subscription or the Azure subscription has expired.

Managed domains are suspended when the Azure platform can't manage, monitor, patch, or back up the domain. A managed domain stays in a *Suspended* state for 15 days. To maintain access to the managed domain, resolve critical alerts immediately.

What to expect

The following behavior is experienced when a managed domain is in the *Suspended* state:

- Domain controllers for the managed domain are de-provisioned and aren't reachable within the virtual network.
- Secure LDAP access to the managed domain over the internet, if enabled, stops working.
- There are failures in authenticating to the managed domain, logging on to domain-joined VMs, or connecting over LDAP/LDAPS.
- Backups for the managed domain are no longer taken.
- Synchronization with Azure AD stops.

How do you know if your managed domain is suspended?

You see an alert on the Azure AD DS Health page in the Azure portal that notes the domain is suspended. The state

of the domain also shows *Suspended*.

Restore a suspended domain

To restore the health of a managed domain that's in the *Suspended* state, complete the following steps:

1. In the Azure portal, search for and select **Domain services**.
2. Choose your managed domain from the list, such as *aaddscontoso.com*, then select **Health**.
3. Select the alert, such as *AADDS503* or *AADDS504*, depending on the cause of suspension.
4. Choose the resolution link that's provided in the alert and follow the steps to resolve it.

A managed domain can only be restored to the date of the last backup. The date of your last backup is displayed on the **Health** page of the managed domain. Any changes that occurred after the last backup won't be restored. Backups for a managed domain are stored for up to 30 days. Backups that are older than 30 days are deleted.

After you resolve alerts when the managed domain is in the *Suspended* state, open an Azure support request to return to a healthy state. If there's a backup less than 30 days old, Azure support can restore the managed domain.

Deleted state

If a managed domain stays in the *Suspended* state for 15 days, it's deleted. This process is unrecoverable.

What to expect

When a managed domain enters the *Deleted* state, the following behavior is seen:

- All resources and backups for the managed domain are deleted.
- You can't restore the managed domain. You must create a replacement managed domain to reuse Azure AD DS.
- After it's deleted, you aren't billed for the managed domain.

Next steps

To keep your managed domain healthy and minimize the risk of it becoming suspended, learn how to resolve alerts for your managed domain.

Troubleshoot secure LDAP connectivity issues to an Azure Active Directory Domain Services managed domain

7/20/2020 • 2 minutes to read • Edit Online

Applications and services that use lightweight directory access protocol (LDAP) to communicate with Azure Active Directory Domain Services (Azure AD DS) can be configured to use secure LDAP. An appropriate certificate and required network ports must be open for secure LDAP to work correctly.

This article helps you troubleshoot issues with secure LDAP access in Azure AD DS.

Common connection issues

If you have trouble connecting to an Azure AD DS managed domain using secure LDAP, review the following troubleshooting steps. After each troubleshooting step, try to connect to the managed domain again:

- The issuer chain of the secure LDAP certificate must be trusted on the client. You can add the Root certification authority (CA) to the trusted root certificate store on the client to establish the trust.
  - Make sure you export and apply the certificate to client computers.
- Verify the secure LDAP certificate for your managed domain has the DNS name in the *Subject* or the *Subject Alternative Names* attribute.
  - Review the secure LDAP certificate requirements and create a replacement certificate if needed.
- Verify that the LDAP client, such as *ldp.exe* connects to the secure LDAP endpoint using a DNS name, not the IP address.
  - The certificate applied to the managed domain doesn't include the IP addresses of the service, only the DNS names.
- Check the DNS name the LDAP client connects to. It must resolve to the public IP address for secure LDAP on the managed domain.
  - If the DNS name resolves to the internal IP address, update the DNS record to resolve to the external IP address.
- For external connectivity, the network security group must include a rule that allows the traffic to TCP port 636 from the internet.
  - If you can connect to the managed domain using secure LDAP from resources directly connected to the virtual network but not external connections, make sure you create a network security group rule to allow secure LDAP traffic.

Next steps

If you still have issues, open an Azure support request for additional troubleshooting assistance.

Known issues: Common alerts and resolutions in Azure Active Directory Domain Services

7/20/2020 • 13 minutes to read • Edit Online

As a central part of identity and authentication for applications, Azure Active Directory Domain Services (Azure AD DS) sometimes has problems. If you run into issues, there are some common alerts and associated troubleshooting steps to help you get things running again. At any time, you can also open an Azure support request for additional troubleshooting assistance.

This article provides troubleshooting information for common alerts in Azure AD DS.

AADDS100: Missing directory

Alert message

The Azure AD directory associated with your managed domain may have been deleted. The managed domain is no longer in a supported configuration. Microsoft cannot monitor, manage, patch, and synchronize your managed domain.

Resolution

This error is usually caused when an Azure subscription is moved to a new Azure AD directory and the old Azure AD directory that's associated with Azure AD DS is deleted.

This error is unrecoverable. To resolve the alert, delete your existing managed domain and recreate it in your new directory. If you have trouble deleting the managed domain, open an Azure support request for additional troubleshooting assistance.

AADDS101: Azure AD B2C is running in this directory

Alert message

Azure AD Domain Services cannot be enabled in an Azure AD B2C Directory.

Resolution

Azure AD DS automatically synchronizes with an Azure AD directory. If the Azure AD directory is configured for B2C, Azure AD DS can't be deployed and synchronized.

To use Azure AD DS, you must recreate your managed domain in a non-Azure AD B2C directory using the following steps:

1. Delete the managed domain from your existing Azure AD directory.
2. Create a new Azure AD directory that isn't an Azure AD B2C directory.
3. Create a replacement managed domain.

The managed domain's health automatically updates itself within two hours and removes the alert.

AADDS103: Address is in a public IP range

Alert message

The IP address range for the virtual network in which you have enabled Azure AD Domain Services is in a public IP range. Azure AD Domain Services must be enabled in a virtual network with a private IP address range. This configuration impacts Microsoft's ability to monitor, manage, patch, and synchronize your managed domain.

Resolution

Before you begin, make sure you understand private IP v4 address spaces.

Inside a virtual network, VMs can make requests to Azure resources in the same IP address range as configured for the subnet. If you configure a public IP address range for a subnet, requests routed within a virtual network may not reach the intended web resources. This configuration can lead to unpredictable errors with Azure AD DS.

> **NOTE**
> If you own the IP address range in the internet that is configured in your virtual network, this alert can be ignored. However, Azure AD Domain Services can't commit to the SLA with this configuration since it can lead to unpredictable errors.

To resolve this alert, delete your existing managed domain and recreate it in a virtual network with a private IP address range. This process is disruptive as the managed domain is unavailable and any custom resources you've created like OUs or service accounts are lost.

1. Delete the managed domain from your directory.
2. To update the virtual network IP address range, search for and select *Virtual network* in the Azure portal. Select the virtual network for Azure AD DS that incorrectly has a public IP address range set.
3. Under **Settings**, select *Address Space*.
4. Update the address range by choosing the existing address range and editing it, or adding an additional address range. Make sure the new IP address range is in a private IP range. When ready, **Save** the changes.
5. Select **Subnets** in the left-hand navigation.
6. Choose the subnet you wish to edit, or create an additional subnet.
7. Update or specify a private IP address range then **Save** your changes.
8. Create a replacement managed domain. Make sure you pick the updated virtual network subnet with a private IP address range.

The managed domain's health automatically updates itself within two hours and removes the alert.

AADDS106: Your Azure subscription is not found

Alert message

Your Azure subscription associated with your managed domain has been deleted. Azure AD Domain Services requires an active subscription to continue functioning properly.

Resolution

Azure AD DS requires an active subscription, and can't be moved to a different subscription. If the Azure subscription that the managed domain was associated with is deleted, you must recreate an Azure subscription and managed domain.

1. Create an Azure subscription.
2. Delete the managed domain from your existing Azure AD directory.
3. Create a replacement managed domain.

AADDS107: Your Azure subscription is disabled

Alert message

Your Azure subscription associated with your managed domain is not active. Azure AD Domain Services requires an active subscription to continue functioning properly.

Resolution

Azure AD DS requires an active subscription. If the Azure subscription that the managed domain was associated with isn't active, you must renew it to reactivate the subscription.

1. Renew your Azure subscription.
2. Once the subscription is renewed, an Azure AD DS notification lets you re-enable the managed domain.

When the managed domain is enabled again, the managed domain's health automatically updates itself within two hours and removes the alert.

AADDS108: Subscription moved directories

Alert message

The subscription used by Azure AD Domain Services has been moved to another directory. Azure AD Domain Services needs to have an active subscription in the same directory to function properly.

Resolution

Azure AD DS requires an active subscription, and can't be moved to a different subscription. If the Azure subscription that the managed domain was associated with is moved, move the subscription back to the previous directory, or delete your managed domain from the existing directory and create a replacement managed domain in the chosen subscription.

AADDS109: Resources for your managed domain cannot be found

Alert message

A resource that is used for your managed domain has been deleted. This resource is needed for Azure AD Domain Services to function properly.

Resolution

Azure AD DS creates additional resources to function properly, such as public IP addresses, virtual network interfaces, and a load balancer. If any of these resources are deleted, the managed domain is in an unsupported state and prevents the domain from being managed. For more information on these resources, see Network resources used by Azure AD DS.

This alert is generated when one of these required resources is deleted. If the resource was deleted less than 4 hours ago, there's a chance that the Azure platform can automatically recreate the deleted resource. The following steps outline how to check the health status and timestamp for resource deletion:

1. In the Azure portal, search for and select **Domain Services**. Choose your managed domain, such as *aaddscontoso.com*.
2. In the left-hand navigation, select **Health**.
3. In the health page, select the alert with the ID *AADDS109*.
4. The alert has a timestamp for when it was first found. If that timestamp is less than 4 hours ago, the Azure platform may be able to automatically recreate the resource and resolve the alert by itself.

   If the alert is more than 4 hours old, the managed domain is in an unrecoverable state. Delete the managed domain and then create a replacement managed domain.

AADDS110: The subnet associated with your managed domain is full

Alert message

The subnet selected for deployment of Azure AD Domain Services is full, and does not have space for the additional domain controller that needs to be created.

Resolution

The virtual network subnet for Azure AD DS needs sufficient IP addresses for the automatically created resources. This IP address space includes the need to create replacement resources if there's a maintenance event. To minimize the risk of running out of available IP addresses, don't deploy additional resources, such as your own VMs, into the same virtual network subnet as the managed domain.

This error is unrecoverable. To resolve the alert, delete your existing managed domain and recreate it. If you have trouble deleting the managed domain, open an Azure support request for additional troubleshooting assistance.

AADDS111: Service principal unauthorized

Alert message

A service principal that Azure AD Domain Services uses to service your domain is not authorized to manage resources on the Azure subscription. The service principal needs to gain permissions to service your managed domain.

Resolution

Some automatically generated service principals are used to manage and create resources for a managed domain. If the access permissions for one of these service principals is changed, the domain is unable to correctly manage resources. The following steps show you how to understand and then grant access permissions to a service principal:

1. Read about role-based access control and how to grant access to applications in the Azure portal.
2. Review the access that the service principal with the ID *abba844e-bc0e-44b0-947a-dc74e5d09022* has and grant the access that was denied at an earlier date.

AADDS112: Not enough IP address in the managed domain

Alert message

We have identified that the subnet of the virtual network in this domain may not have enough IP addresses. Azure AD Domain Services needs at-least two available IP addresses within the subnet it is enabled in. We recommend having at-least 3-5 spare IP addresses within the subnet. This may have occurred if other virtual machines are deployed within the subnet, thus exhausting the number of available IP addresses or if there is a restriction on the number of available IP addresses in the subnet.

Resolution

The virtual network subnet for Azure AD DS needs enough IP addresses for the automatically created resources. This IP address space includes the need to create replacement resources if there's a maintenance event. To minimize the risk of running out of available IP addresses, don't deploy additional resources, such as your own VMs, into the same virtual network subnet as the managed domain.

To resolve this alert, delete your existing managed domain and re-create it in a virtual network with a large enough IP address range. This process is disruptive as the managed domain is unavailable and any custom resources you've created like OUs or service accounts are lost.

1. Delete the managed domain from your directory.
2. To update the virtual network IP address range, search for and select *Virtual network* in the Azure portal. Select the virtual network for the managed domain that has the small IP address range.
3. Under **Settings**, select *Address Space*.
4. Update the address range by choosing the existing address range and editing it, or adding an additional address range. Make sure the new IP address range is large enough for the managed domain's subnet range. When ready, **Save** the changes.
5. Select **Subnets** in the left-hand navigation.

6. Choose the subnet you wish to edit, or create an additional subnet.
7. Update or specify a large enough IP address range then **Save** your changes.
8. Create a replacement managed domain. Make sure you pick the updated virtual network subnet with a large enough IP address range.

The managed domain's health automatically updates itself within two hours and removes the alert.

AADDS113: Resources are unrecoverable

Alert message

The resources used by Azure AD Domain Services were detected in an unexpected state and cannot be recovered.

Resolution

This error is unrecoverable. To resolve the alert, delete your existing managed domain and recreate it. If you have trouble deleting the managed domain, open an Azure support request for additional troubleshooting assistance.

AADDS114: Subnet invalid

Alert message

The subnet selected for deployment of Azure AD Domain Services is invalid, and cannot be used.

Resolution

This error is unrecoverable. To resolve the alert, delete your existing managed domain and recreate it. If you have trouble deleting the managed domain, open an Azure support request for additional troubleshooting assistance.

AADDS115: Resources are locked

Alert message

One or more of the network resources used by the managed domain cannot be operated on as the target scope has been locked.

Resolution

Resource locks can be applied to Azure resources to prevent change or deletion. As Azure AD DS is a managed service, the Azure platform needs the ability to make configuration changes. If a resource lock is applied on some of the Azure AD DS components, the Azure platform can't perform its management tasks.

To check for resource locks on the Azure AD DS components and remove them, complete the following steps:

1. For each of the managed domain's network components in your resource group, such as virtual network, network interface, or public IP address, check the operation logs in the Azure portal. These operation logs should indicate why an operation is failing and where a resource lock is applied.
2. Select the resource where a lock is applied, then under **Locks**, select and remove the lock(s).

AADDS116: Resources are unusable

Alert message

One or more of the network resources used by the managed domain cannot be operated on due to policy restriction(s).

Resolution

Policies are applied to Azure resources and resource groups that control what configuration actions are allowed. As Azure AD DS is a managed service, the Azure platform needs the ability to make configuration changes. If a policy is applied on some of the Azure AD DS components, the Azure platform may not be able to perform its

management tasks.

To check for applied policies on the Azure AD DS components and update them, complete the following steps:

1. For each of the managed domain's network components in your resource group, such as virtual network, NIC, or public IP address, check the operation logs in the Azure portal. These operation logs should indicate why an operation is failing and where a restrictive policy is applied.
2. Select the resource where a policy is applied, then under **Policies**, select and edit the policy so it's less restrictive.

AADDS500: Synchronization has not completed in a while

Alert message

The managed domain was last synchronized with Azure AD on [date]. Users may be unable to sign-in on the managed domain or group memberships may not be in sync with Azure AD.

Resolution

Check the Azure AD DS health for any alerts that indicate problems in the configuration of the managed domain. Problems with the network configuration can block the synchronization from Azure AD. If you're able to resolve alerts that indicate a configuration issue, wait two hours and check back to see if the synchronization has successfully completed.

The following common reasons cause synchronization to stop in a managed domain:

- Required network connectivity is blocked. To learn more about how to check the Azure virtual network for problems and what's required, see troubleshoot network security groups and the network requirements for Azure AD DS.
- Password synchronization wasn't set up or successfully completed when the managed domain was deployed. You can set up password synchronization for cloud-only users or hybrid users from on-prem.

AADDS501: A backup has not been taken in a while

Alert message

The managed domain was last backed up on [date].

Resolution

Check the Azure AD DS health for alerts that indicate problems in the configuration of the managed domain. Problems with the network configuration can block the Azure platform from successfully taking backups. If you're able to resolve alerts that indicate a configuration issue, wait two hours and check back to see if the synchronization has successfully completed.

AADDS503: Suspension due to disabled subscription

Alert message

The managed domain is suspended because the Azure subscription associated with the domain is not active.

Resolution

> **WARNING**
> If a managed domain is suspended for an extended period of time, there's a danger of it being deleted. Resolve the reason for suspension as quickly as possible. For more information, see Understand the suspended states for Azure AD DS.

Azure AD DS requires an active subscription. If the Azure subscription that the managed domain was associated

with isn't active, you must renew it to reactivate the subscription.

1. Renew your Azure subscription.
2. Once the subscription is renewed, an Azure AD DS notification lets you re-enable the managed domain.

When the managed domain is enabled again, the managed domain's health automatically updates itself within two hours and removes the alert.

AADDS504: Suspension due to an invalid configuration

Alert message

The managed domain is suspended due to an invalid configuration. The service has been unable to manage, patch, or update the domain controllers for your managed domain for a long time.

Resolution

> **WARNING**
> If a managed domain is suspended for an extended period of time, there's a danger of it being deleted. Resolve the reason for suspension as quickly as possible. For more information, see Understand the suspended states for Azure AD DS.

Check the Azure AD DS health for alerts that indicate problems in the configuration of the managed domain. If you're able to resolve alerts that indicate a configuration issue, wait two hours and check back to see if the synchronization has completed. When ready, open an Azure support request to re-enable the managed domain.

Next steps

If you still have issues, open an Azure support request for additional troubleshooting assistance.

Known issues: Network configuration alerts in Azure Active Directory Domain Services

7/20/2020 • 3 minutes to read • Edit Online

To let applications and services correctly communicate with an Azure Active Directory Domain Services (Azure AD DS) managed domain, specific network ports must be open to allow traffic to flow. In Azure, you control the flow of traffic using network security groups. The health status of an Azure AD DS managed domain shows an alert if the required network security group rules aren't in place.

This article helps you understand and resolve common alerts for network security group configuration issues.

Alert AADDS104: Network error

Alert message

Microsoft is unable to reach the domain controllers for this managed domain. This may happen if a network security group (NSG) configured on your virtual network blocks access to the managed domain. Another possible reason is if there is a user-defined route that blocks incoming traffic from the internet.

Invalid network security group rules are the most common cause of network errors for Azure AD DS. The network security group for the virtual network must allow access to specific ports and protocols. If these ports are blocked, the Azure platform can't monitor or update the managed domain. The synchronization between the Azure AD directory and Azure AD DS is also impacted. Make sure you keep the default ports open to avoid interruption in service.

Default security rules

The following default inbound and outbound security rules are applied to the network security group for a managed domain. These rules keep Azure AD DS secure and allow the Azure platform to monitor, manage, and update the managed domain.

Inbound security rules

PRIORITY	NAME	PORT	PROTOCOL	SOURCE	DESTINATION	ACTION
101	AllowSyncWithAzureAD	443	TCP	AzureActiveDirectoryDomainServices	Any	Allow
201	AllowRD	3389	TCP	CorpNetSaw	Any	Allow
301	AllowPSRemoting	5986	TCP	AzureActiveDirectoryDomainServices	Any	Allow
65000	AllVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow

PRIORITY	NAME	PORT	PROTOCOL	SOURCE	DESTINATION	ACTION
65500	DenyAllInBound	Any	Any	Any	Any	Deny

> **NOTE**
> You may also have an additional rule that allows inbound traffic if you configure secure LDAP. This additional rule is required for the correct LDAPS communication.

Outbound security rules

PRIORITY	NAME	PORT	PROTOCOL	SOURCE	DESTINATION	ACTION
65000	AllVnetOutBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerOutBound	Any	Any	Any	Internet	Allow
65500	DenyAllOutBound	Any	Any	Any	Any	Deny

> **NOTE**
> Azure AD DS needs unrestricted outbound access from the virtual network. We don't recommend that you create any additional rules that restrict outbound access for the virtual network.

Verify and edit existing security rules

To verify the existing security rules and make sure the default ports are open, complete the following steps:

1. In the Azure portal, search for and select **Network security groups**.
2. Choose the network security group associated with your managed domain, such as *AADDS-contoso.com-NSG*.
3. On the **Overview** page, the existing inbound and outbound security rules are shown.

   Review the inbound and outbound rules and compare to the list of required rules in the previous section. If needed, select and then delete any custom rules that block required traffic. If any of the required rules are missing, add a rule in the next section.

   After you add or delete rules to allow the required traffic, the managed domain's health automatically updates itself within two hours and removes the alert.

Add a security rule

To add a missing security rule, complete the following steps:

1. In the Azure portal, search for and select **Network security groups**.
2. Choose the network security group associated with your managed domain, such as *AADDS-contoso.com-NSG*.
3. Under **Settings** in the left-hand panel, click *Inbound security rules* or *Outbound security rules* depending on which rule you need to add.
4. Select **Add**, then create the required rule based on the port, protocol, direction, etc. When ready, select **OK**.

It takes a few moments for the security rule to be added and show in the list.

Next steps

If you still have issues, open an Azure support request for additional troubleshooting assistance.

Known issues: Service principal alerts in Azure Active Directory Domain Services

7/20/2020 • 2 minutes to read • Edit Online

Service principals are applications that the Azure platform uses to manage, update, and maintain an Azure Active Directory Domain Services (Azure AD DS) managed domain. If a service principal is deleted, functionality in the managed domain is impacted.

This article helps you troubleshoot and resolve service principal-related configuration alerts.

Alert AADDS102: Service principal not found

Alert message

A Service Principal required for Azure AD Domain Services to function properly has been deleted from your Azure AD directory. This configuration impacts Microsoft's ability to monitor, manage, patch, and synchronize your managed domain.

If a required service principal is deleted, the Azure platform can't perform automated management tasks. The managed domain may not correctly apply updates or take backups.

Check for missing service principals

To check which service principal is missing and must be recreated, complete the following steps:

1. In the Azure portal, select **Azure Active Directory** from the left-hand navigation menu.
2. Select **Enterprise applications**. Choose *All applications* from the **Application Type** drop-down menu, then select **Apply**.
3. Search for each of the following application IDs. If no existing application is found, follow the *Resolution* steps to create the service principal or re-register the namespace.

APPLICATION ID	RESOLUTION
2565bd9d-da50-47d4-8b85-4c97f669dc36	Recreate a missing service principal
443155a6-77f3-45e3-882b-22b3a8d431fb	Re-register the Microsoft.AAD namespace
abba844e-bc0e-44b0-947a-dc74e5d09022	Re-register the Microsoft.AAD namespace
d87dcbc6-a371-462e-88e3-28ad15ec4e64	Re-register the Microsoft.AAD namespace

Recreate a missing Service Principal

If application ID *2565bd9d-da50-47d4-8b85-4c97f669dc36* is missing from your Azure AD directory, use Azure AD PowerShell to complete the following steps. For more information, see Azure AD PowerShell.

1. If needed, install the Azure AD PowerShell module and import it as follows:

```
Install-Module AzureAD
Import-Module AzureAD
```

2. Now recreate the service principal using the New-AzureAdServicePrincipal cmdlet:

```
New-AzureAdServicePrincipal -AppId "2565bd9d-da50-47d4-8b85-4c97f669dc36"
```

The managed domain's health automatically updates itself within two hours and removes the alert.

Re-register the Microsoft AAD namespace

If application ID *443155a6-77f3-45e3-882b-22b3a8d431fb*, *abba844e-bc0e-44b0-947a-dc74e5d09022*, or *d87dcbc6-a371-462e-88e3-28ad15ec4e64* is missing from your Azure AD directory, complete the following steps to re-register the *Microsoft.AAD* resource provider:

1. In the Azure portal, search for and select **Subscriptions**.
2. Choose the subscription associated with your managed domain.
3. From the left-hand navigation, choose **Resource Providers**.
4. Search for *Microsoft.AAD*, then select **Re-register**.

The managed domain's health automatically updates itself within two hours and removes the alert.

Alert AADDS105: Password synchronization application is out of date

Alert message

The service principal with the application ID "d87dcbc6-a371-462e-88e3-28ad15ec4e64" was deleted and then recreated. The recreation leaves behind inconsistent permissions on Azure AD Domain Services resources needed to service your managed domain. Synchronization of passwords on your managed domain could be affected.

Azure AD DS automatically synchronizes user accounts and credentials from Azure AD. If there's a problem with the Azure AD application used for this process, credential synchronization between Azure AD DS and Azure AD fails.

Resolution

To recreate the Azure AD application used for credential synchronization, use Azure AD PowerShell to complete the following steps. For more information, see install Azure AD PowerShell.

1. If needed, install the Azure AD PowerShell module and import it as follows:

```
Install-Module AzureAD
Import-Module AzureAD
```

2. Now delete the old application and object using the following PowerShell cmdlets:

```
$app = Get-AzureADApplication -Filter "IdentifierUris eq
'https://sync.aaddc.activedirectory.windowsazure.com'"
Remove-AzureADApplication -ObjectId $app.ObjectId
$spObject = Get-AzureADServicePrincipal -Filter "DisplayName eq 'Azure AD Domain Services Sync'"
Remove-AzureADServicePrincipal -ObjectId $spObject
```

After you delete both applications, the Azure platform automatically recreates them and tries to resume password synchronization. The managed domain's health automatically updates itself within two hours and removes the alert.

Next steps

If you still have issues, open an Azure support request for additional troubleshooting assistance.

Known issues: Secure LDAP alerts in Azure Active Directory Domain Services

7/20/2020 • 2 minutes to read • Edit Online

Applications and services that use lightweight directory access protocol (LDAP) to communicate with Azure Active Directory Domain Services (Azure AD DS) can be configured to use secure LDAP. An appropriate certificate and required network ports must be open for secure LDAP to work correctly.

This article helps you understand and resolve common alerts with secure LDAP access in Azure AD DS.

AADDS101: Secure LDAP network configuration

Alert message

Secure LDAP over the internet is enabled for the managed domain. However, access to port 636 is not locked down using a network security group. This may expose user accounts on the managed domain to password brute-force attacks.

Resolution

When you enable secure LDAP, it's recommended to create additional rules that restrict inbound LDAPS access to specific IP addresses. These rules protect the managed domain from brute force attacks. To update the network security group to restrict TCP port 636 access for secure LDAP, complete the following steps:

1. In the Azure portal, search for and select **Network security groups**.
2. Choose the network security group associated with your managed domain, such as *AADDS-contoso.com-NSG*, then select **Inbound security rules**
3. Select **+ Add** to create a rule for TCP port 636. If needed, select **Advanced** in the window to create a rule.
4. For the **Source**, choose *IP Addresses* from the drop-down menu. Enter the source IP addresses that you want to grant access for secure LDAP traffic.
5. Choose *Any* as the **Destination**, then enter *636* for **Destination port ranges**.
6. Set the **Protocol** as *TCP* and the **Action** to *Allow*.
7. Specify the priority for the rule, then enter a name such as *RestrictLDAPS*.
8. When ready, select **Add** to create the rule.

The managed domain's health automatically updates itself within two hours and removes the alert.

> **TIP**
> TCP port 636 isn't the only rule needed for Azure AD DS to run smoothly. To learn more, see the Azure AD DS Network security groups and required ports.

AADDS502: Secure LDAP certificate expiring

Alert message

The secure LDAP certificate for the managed domain will expire on [date]].

Resolution

Create a replacement secure LDAP certificate by following the steps to create a certificate for secure LDAP. Apply the replacement certificate to Azure AD DS, and distribute the certificate to any clients that connect using secure

LDAP.

Next steps

If you still have issues, open an Azure support request for additional troubleshooting assistance.

Frequently asked questions (FAQs) about Azure Active Directory (AD) Domain Services

7/20/2020 • 11 minutes to read • Edit Online

This page answers frequently asked questions about Azure Active Directory Domain Services.

Configuration

Can I create multiple managed domains for a single Azure AD directory?

No. You can only create a single managed domain serviced by Azure AD Domain Services for a single Azure AD directory.

Can I enable Azure AD Domain Services in a Classic virtual network?

Classic virtual networks aren't supported for new deployments. Existing managed domains deployed in Classic virtual networks continue to be supported until they're retired on March 1, 2023. For existing deployments, you can migrate Azure AD Domain Services from the Classic virtual network model to Resource Manager.

For more information, see the official deprecation notice.

Can I enable Azure AD Domain Services in an Azure Resource Manager virtual network?

Yes. Azure AD Domain Services can be enabled in an Azure Resource Manager virtual network. Classic Azure virtual networks are no longer available when you create a managed domain.

Can I migrate my existing managed domain from a Classic virtual network to a Resource Manager virtual network?

Yes. For more information, see Migrate Azure AD Domain Services from the Classic virtual network model to Resource Manager.

Can I enable Azure AD Domain Services in an Azure CSP (Cloud Solution Provider) subscription?

Yes. For more information, see how to enable Azure AD Domain Services in Azure CSP subscriptions.

Can I enable Azure AD Domain Services in a federated Azure AD directory? I do not synchronize password

hashes to Azure AD. Can I enable Azure AD Domain Services for this directory?

No. To authenticate users via NTLM or Kerberos, Azure AD Domain Services needs access to the password hashes of user accounts. In a federated directory, password hashes aren't stored in the Azure AD directory. Therefore, Azure AD Domain Services doesn't work with such Azure AD directories.

However, if you're using Azure AD Connect for password hash synchronization, you can use Azure AD Domain Services because the password hash values are stored in Azure AD.

Can I make Azure AD Domain Services available in multiple virtual networks within my subscription?

The service itself doesn't directly support this scenario. Your managed domain is available in only one virtual network at a time. However, you can configure connectivity between multiple virtual networks to expose Azure AD Domain Services to other virtual networks. For more information, see how to connect virtual networks in Azure using VPN gateways or virtual network peering.

Can I enable Azure AD Domain Services using PowerShell?

Yes. For more information, see how to enable Azure AD Domain Services using PowerShell.

Can I enable Azure AD Domain Services using a Resource Manager Template?

Yes, you can create an Azure AD Domain Services managed domain using a Resource Manager template. A service principal and Azure AD group for administration must be created using the Azure portal or Azure PowerShell before the template is deployed. For more information, see Create an Azure AD DS managed domain using an Azure Resource Manager template. When you create an Azure AD Domain Services managed domain in the Azure portal, there's an also option to export the template for use with additional deployments.

Can I add domain controllers to an Azure AD Domain Services managed domain?

No. The domain provided by Azure AD Domain Services is a managed domain. You don't need to provision, configure, or otherwise manage domain controllers for this domain. These management activities are provided as a service by Microsoft. Therefore, you can't add additional domain controllers (read-write or read-only) for the managed domain.

Can guest users invited to my directory use Azure AD Domain Services?

No. Guest users invited to your Azure AD directory using the Azure AD B2B invite process are synchronized into your Azure AD Domain Services managed domain. However, passwords for these users aren't stored in your Azure AD directory. Therefore, Azure AD Domain Services has no way to synchronize NTLM and Kerberos hashes for these users into your managed domain. Such users can't sign in or join computers to the managed domain.

Can I move an existing Azure AD Domain Services managed domain to a different subscription, resource group, region, or virtual network?

No. After you create an Azure AD Domain Services managed domain, you can't then move the managed domain to a different resource group, virtual network, subscription, etc. Take care to select the most appropriate subscription, resource group, region, and virtual network when you deploy the managed domain.

Does Azure AD Domain Services include high availability options?

Yes. Each Azure AD Domain Services managed domain includes two domain controllers. You don't manage or connect to these domain controllers, they're part of the managed service. If you deploy Azure AD Domain Services into a region that supports Availability Zones, the domain controllers are distributed across zones. In regions that don't support Availability Zones, the domain controllers are distributed across Availability Sets. You have no configuration options or management control over this distribution. For more information, see Availability options for virtual machines in Azure.

Administration and operations

- Can I connect to the domain controller for my managed domain using Remote Desktop?
- I've enabled Azure AD Domain Services. What user account do I use to domain join machines to this domain?

- Do I have domain administrator privileges for the managed domain provided by Azure AD Domain Services?
- Can I modify group memberships using LDAP or other AD administrative tools on managed domains?
- How long does it take for changes I make to my Azure AD directory to be visible in my managed domain?
- Can I extend the schema of the managed domain provided by Azure AD Domain Services?
- Can I modify or add DNS records in my managed domain?
- What is the password lifetime policy on a managed domain?
- Does Azure AD Domain Services provide AD account lockout protection?
- Can I configure Distributed File System (DFS) and replication within Azure AD Domain Services?

Can I connect to the domain controller for my managed domain using Remote Desktop?

No. You don't have permissions to connect to domain controllers for the managed domain using Remote Desktop. Members of the *AAD DC Administrators* group can administer the managed domain using AD administration tools such as the Active Directory Administration Center (ADAC) or AD PowerShell. These tools are installed using the *Remote Server Administration Tools* feature on a Windows server joined to the managed domain. For more information, see Create a management VM to configure and administer an Azure AD Domain Services managed domain.

I've enabled Azure AD Domain Services. What user account do I use to domain join machines to this domain?

Any user account that's part of the managed domain can join a VM. Members of the *AAD DC Administrators* group are granted remote desktop access to machines that have been joined to the managed domain.

Do I have domain administrator privileges for the managed domain provided by Azure AD Domain Services?

No. You aren't granted administrative privileges on the managed domain. *Domain Administrator* and *Enterprise Administrator* privileges aren't available for you to use within the domain. Members of the domain administrator or enterprise administrator groups in your on-premises Active Directory are also not granted domain / enterprise administrator privileges on the managed domain.

Can I modify group memberships using LDAP or other AD administrative tools on managed domains?

Users and groups that are synchronized from Azure Active Directory to Azure AD Domain Services cannot be modified because their source of origin is Azure Active Directory. Any user or group originating in the managed domain may be modified.

How long does it take for changes I make to my Azure AD directory to be visible in my managed domain?

Changes made in your Azure AD directory using either the Azure AD UI or PowerShell are automatically synchronized to your managed domain. This synchronization process runs in the background. There's no defined time period for this synchronization to complete all the object changes.

Can I extend the schema of the managed domain provided by Azure AD Domain Services?

No. The schema is administered by Microsoft for the managed domain. Schema extensions aren't supported by Azure AD Domain Services.

Can I modify or add DNS records in my managed domain?

Yes. Members of the *AAD DC Administrators* group are granted *DNS Administrator* privileges to modify DNS records in the managed domain. Those users can use the DNS Manager console on a machine running Windows Server joined to the managed domain to manage DNS. To use the DNS Manager console, install *DNS Server Tools*, which is part of the *Remote Server Administration Tools* optional feature on the server. For more information, see Administer DNS in an Azure AD Domain Services managed domain.

What is the password lifetime policy on a managed domain?

The default password lifetime on an Azure AD Domain Services managed domain is 90 days. This password lifetime is not synchronized with the password lifetime configured in Azure AD. Therefore, you may have a situation where users' passwords expire in your managed domain, but are still valid in Azure AD. In such scenarios, users need to change their password in Azure AD and the new password will synchronize to your managed domain.

Additionally, the *password-does-not-expire* and *user-must-change-password-at-next-logon* attributes for user accounts aren't synchronized to your managed domain.

Does Azure AD Domain Services provide AD account lockout protection?

Yes. Five invalid password attempts within 2 minutes on the managed domain cause a user account to be locked out for 30 minutes. After 30 minutes, the user account is automatically unlocked. Invalid password attempts on the managed domain don't lock out the user account in Azure AD. The user account is locked out only within your Azure AD Domain Services managed domain. For more information, see Password and account lockout policies on managed domains.

Can I configure Distributed File System and replication within Azure AD Domain Services?

No. Distributed File System (DFS) and replication aren't available when using Azure AD Domain Services.

Billing and availability

- Is Azure AD Domain Services a paid service?
- Is there a free trial for the service?
- Can I pause an Azure AD Domain Services managed domain?
- Can I failover Azure AD Domain Services to another region for a DR event?
- Can I get Azure AD Domain Services as part of Enterprise Mobility Suite (EMS)? Do I need Azure AD Premium to use Azure AD Domain Services?
- What Azure regions is the service available in?

Is Azure AD Domain Services a paid service?

Yes. For more information, see the pricing page.

Is there a free trial for the service?

Azure AD Domain Services is included in the free trial for Azure. You can sign up for a free one-month trial of Azure.

Can I pause an Azure AD Domain Services managed domain?

No. Once you've enabled an Azure AD Domain Services managed domain, the service is available within your selected virtual network until you delete the managed domain. There's no way to pause the service. Billing continues on an hourly basis until you delete the managed domain.

Can I failover Azure AD Domain Services to another region for a DR event?

No. Azure AD Domain Services doesn't currently provide a geo-redundant deployment model. It's limited to a single virtual network in an Azure region. If you want to utilize multiple Azure regions, you need to run your Active Directory Domain Controllers on Azure IaaS VMs. For architecture guidance, see Extend your on-premises Active Directory domain to Azure.

Can I get Azure AD Domain Services as part of Enterprise Mobility Suite (EMS)? Do I need Azure AD Premium to use Azure AD Domain Services?

No. Azure AD Domain Services is a pay-as-you-go Azure service and isn't part of EMS. Azure AD Domain Services can be used with all editions of Azure AD (Free and Premium). You're billed on an hourly basis, depending on usage.

What Azure regions is the service available in?

Refer to the Azure Services by region page to see a list of the Azure regions where Azure AD Domain Services is available.

Troubleshooting

Refer to the Troubleshooting guide for solutions to common issues with configuring or administering Azure AD Domain Services.

Next steps

To learn more about Azure AD Domain Services, see What is Azure Active Directory Domain Services?.

To get started, see Create and configure an Azure Active Directory Domain Services managed domain.

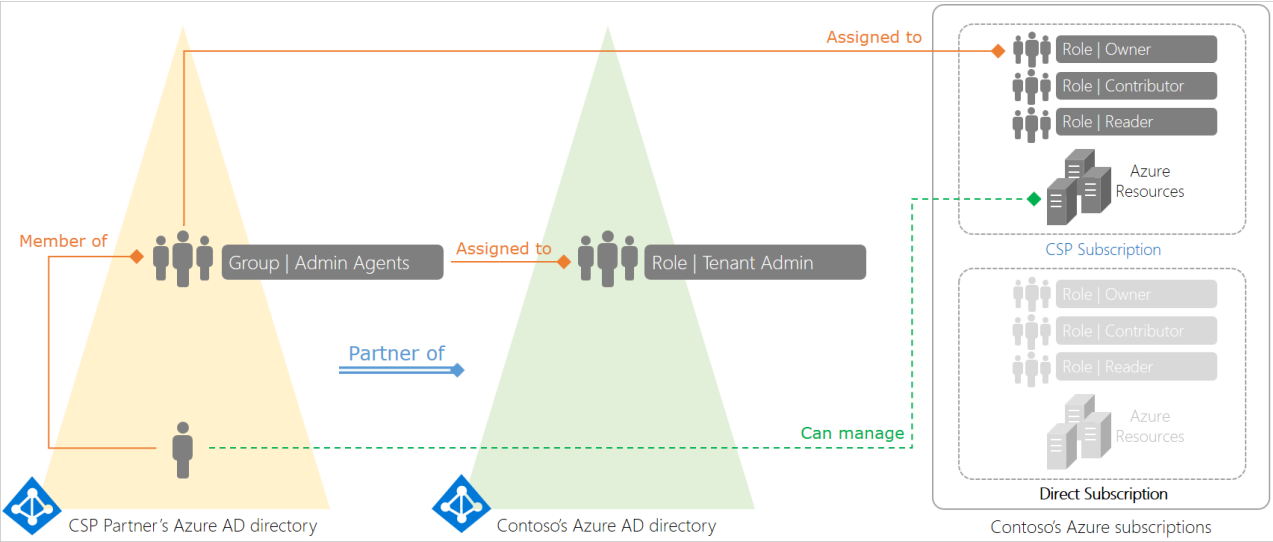
Azure Active Directory Domain Services deployment and management for Azure Cloud Solution Providers

7/20/2020 • 6 minutes to read • Edit Online

Azure Cloud Solution Providers (CSP) is a program for Microsoft Partners and provides a license channel for various Microsoft cloud services. Azure CSP enables partners to manage sales, own the billing relationship, provide technical and billing support, and be the customer's single point of contact. In addition, Azure CSP provides a full set of tools, including a self-service portal and accompanying APIs. These tools enable CSP partners to easily provision and manage Azure resources, and provide billing for customers and their subscriptions.

The Partner Center portal is the entry point for all Azure CSP partners, and provides rich customer management capabilities, automated processing, and more. Azure CSP partners can use Partner Center capabilities by using a web-based UI or by using PowerShell and various API calls.

The following diagram illustrates how the CSP model works at a high level. Here, Contoso has an Azure Active Directory (Azure AD) tenant. They have a partnership with a CSP, who deploys and manages resources in their Azure CSP subscription. Contoso may also have regular (direct) Azure subscriptions, which are billed directly to Contoso.



The CSP partner's tenant has three special agent groups - *Admin* agents, *Helpdesk* agents, and *Sales* agents.

The *Admin* agents group is assigned to the tenant administrator role in Contoso's Azure AD tenant. As a result, a user belonging to the CSP partner's admin agents group has tenant admin privileges in Contoso's Azure AD tenant.

When the CSP partner provisions an Azure CSP subscription for Contoso, their admin agents group is assigned to the owner role for that subscription. As a result, the CSP partner's admin agents have the required privileges to provision Azure resources such as virtual machines, virtual networks, and Azure AD Domain Services on behalf of Contoso.

For more information, see the Azure CSP overview

Benefits of using Azure AD DS in an Azure CSP subscription

Azure Active Directory Domain Services (Azure AD DS) provides managed domain services such as domain join, group policy, LDAP, Kerberos/NTLM authentication that is fully compatible with Windows Server Active Directory Domain Services. Over the decades, many applications have been built to work against AD using these capabilities.

Many independent software vendors (ISVs) have built and deployed applications at their customers' premises. These applications are hard to support since you often require access to the different environments where the applications are deployed. With Azure CSP subscriptions, you have a simpler alternative with the scale and flexibility of Azure.

Azure AD DS supports Azure CSP subscriptions. You can deploy your application in an Azure CSP subscription tied to your customer's Azure AD tenant. As a result, your employees (support staff) can manage, administer, and service the VMs on which your application is deployed using your organization's corporate credentials.

You can also deploy an Azure AD DS managed domain in your customer's Azure AD tenant. Your application is then connected to your customer's managed domain. Capabilities within your application that rely on Kerberos / NTLM, LDAP, or the System.DirectoryServices API work seamlessly against your customer's domain. End customers benefit from consuming your application as a service, without needing to worry about maintaining the infrastructure the application is deployed on.

All billing for Azure resources you consume in that subscription, including Azure AD DS, is charged back to you. You maintain full control over the relationship with the customer when it comes to sales, billing, technical support etc. With the flexibility of the Azure CSP platform, a small team of support agents can service many such customers who have instances of your application deployed.

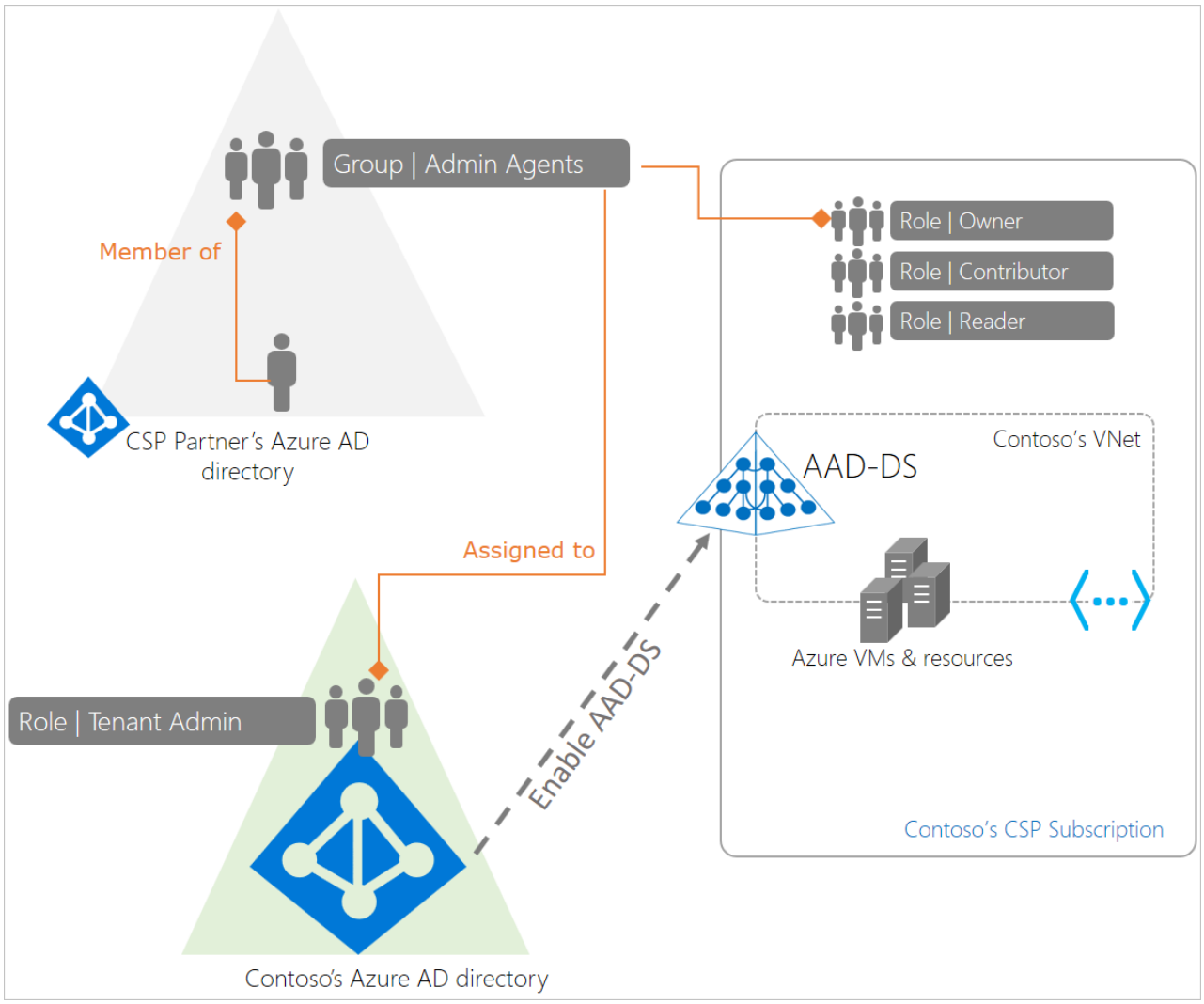
CSP deployment models for Azure AD DS

There are two ways in which you can use Azure AD DS with an Azure CSP subscription. Pick the right one based on the security and simplicity considerations your customers have.

Direct deployment model

In this deployment model, Azure AD DS is enabled within a virtual network that belongs to the Azure CSP subscription. The CSP partner's admin agents have the following privileges:

- *Global administrator* privileges in the customer's Azure AD tenant.
- *Subscription owner* privileges on the Azure CSP subscription.



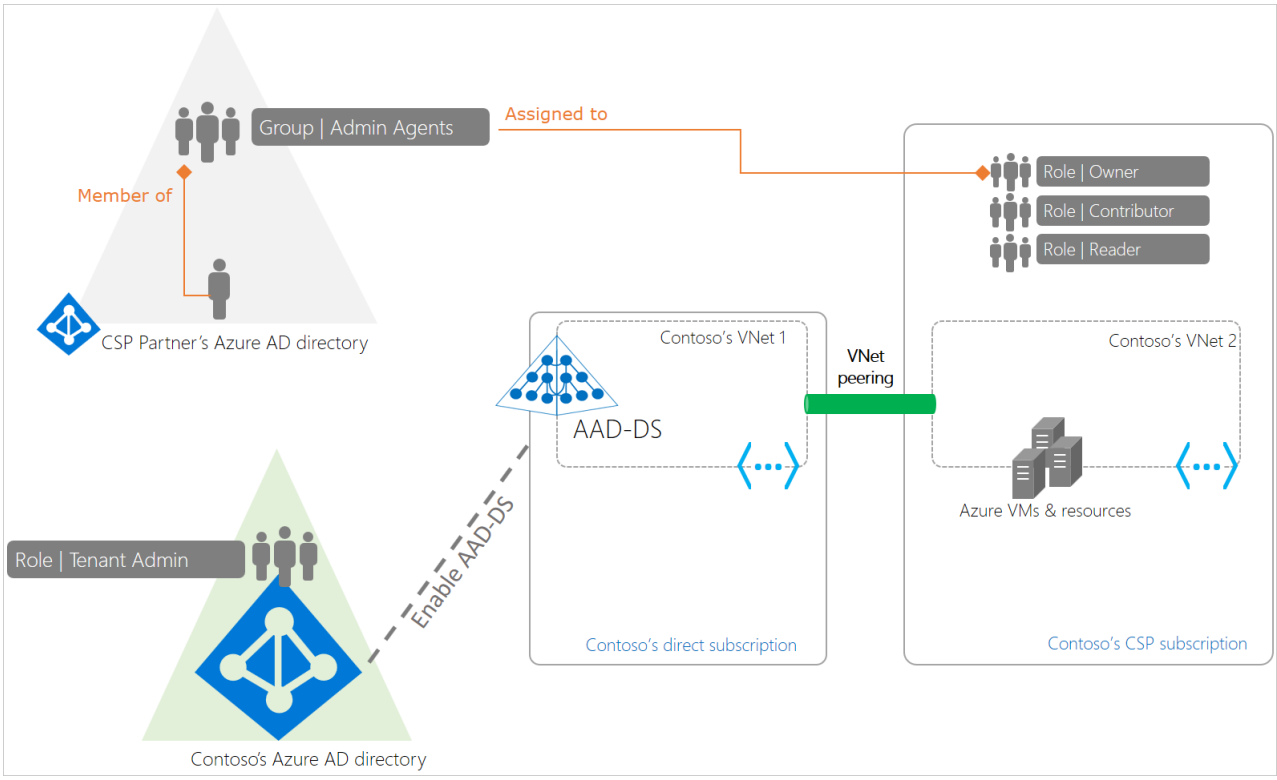
In this deployment model, the CSP provider's admin agents can administer identities for the customer. These admin agents can perform tasks like provision new users or groups, or add applications within the customer's Azure AD tenant.

This deployment model may be suited for smaller organizations that don't have a dedicated identity administrator or prefer for the CSP partner to administer identities on their behalf.

Peered deployment model

In this deployment model, Azure AD DS is enabled within a virtual network belonging to the customer - a direct Azure subscription paid for by the customer. The CSP partner can deploy applications within a virtual network belonging to the customer's CSP subscription. The virtual networks can then be connected using Azure virtual network peering.

With this deployment, the workloads or applications deployed by the CSP partner in the Azure CSP subscription can connect to the customer's managed domain provisioned in the customer's direct Azure subscription.



This deployment model provides a separation of privileges and enables the CSP partner's helpdesk agents to administer the Azure subscription and deploy and manage resources within it. However, the CSP partner's helpdesk agents don't need to have global administrator privileges on the customer's Azure AD directory. The customer's identity administrators can continue to manage identities for their organization.

This deployment model may be suited to scenarios where an ISV provides a hosted version of their on-premises application, which also needs to connect to the customer's Azure AD.

Administer Azure AD DS in CSP subscriptions

The following important considerations apply when administering a managed domain in an Azure CSP subscription:

- **CSP admin agents can provision a managed domain using their credentials:** Azure AD DS supports Azure CSP subscriptions. Users belonging to a CSP partner's admin agents group can provision a new managed domain.
- **CSPs can script creation of new managed domains for their customers using PowerShell:** See how to enable Azure AD DS using PowerShell for details.
- **CSP admin agents can't perform ongoing management tasks on the managed domain using their credentials:** CSP admin users can't perform routine management tasks within the managed domain using their credentials. These users are external to the customer's Azure AD tenant and their credentials aren't available within the customer's Azure AD tenant. Azure AD DS doesn't have access to the Kerberos and NTLM password hashes for these users, so users can't be authenticated on managed domains.

> **WARNING**
>
> You must create a user account within the customer's directory to perform ongoing administration tasks on the managed domain.
>
> You can't sign in to the managed domain using a CSP admin user's credentials. Use the credentials of a user account belonging to the customer's Azure AD tenant to do so. You need these credentials for tasks such as joining VMs to the managed domain, administering DNS, or administering Group Policy.

- **The user account created for ongoing administration must be added to the *AAD DC Administrators* group:** The *AAD DC Administrators* group has privileges to perform certain delegated administration tasks on the managed domain. These tasks include configuring DNS, creating organizational units, and administering group policy.

  For a CSP partner to perform these tasks on a managed domain, a user account must be created within the customer's Azure AD tenant. The credentials for this account must be shared with the CSP partner's admin agents. Also, this user account must be added to the *AAD DC Administrators* group to enable configuration tasks on the managed domain to be performed using this user account.

Next steps

To get started, enroll in the Azure CSP program. You can then enable Azure AD Domain Services using the Azure portal or Azure PowerShell.