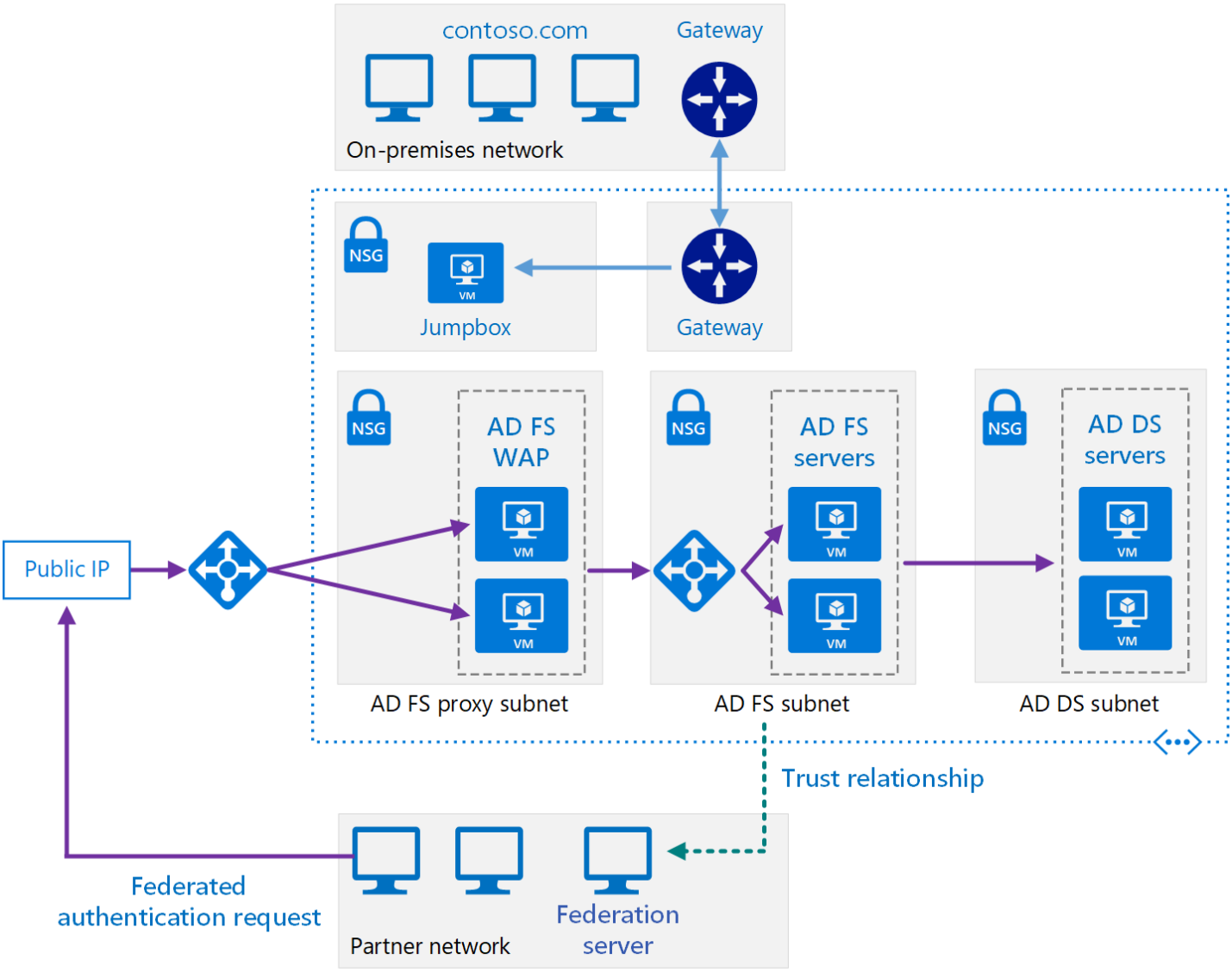
Extend Active Directory Federation Services (AD FS) to Azure

12/18/2018 • 15 minutes to read • Contributors all

In this article

This reference architecture implements a secure hybrid network that extends your on-premises network to Azure and uses Active Directory Federation Services (AD FS) to perform federated authentication and authorization for components running in Azure. **Deploy this solution**.



Download a Visio file of this architecture.

AD FS can be hosted on-premises, but if your application is a hybrid in which some parts are implemented in Azure, it may be more efficient to replicate AD FS in the cloud.

The diagram shows the following scenarios:

- Application code from a partner organization accesses a web application hosted inside your Azure VNet.
- An external, registered user with credentials stored inside Active Directory Domain Services (DS) accesses a web application hosted inside your Azure VNet.
- A user connected to your VNet using an authorized device executes a web application hosted inside your Azure VNet.

Typical uses for this architecture include:

- Hybrid applications where workloads run partly on-premises and partly in Azure.
- Solutions that use federated authorization to expose web applications to partner organizations.
- Systems that support access from web browsers running outside of the organizational firewall.
- Systems that enable users to access to web applications by connecting from authorized external devices such as remote computers, notebooks, and other mobile devices.

This reference architecture focuses on *passive federation*, in which the federation servers decide how and when to authenticate a user. The user provides sign in information when the application is started. This mechanism is most commonly used by web browsers and involves a protocol that redirects the browser to a site where the user authenticates. AD FS also supports *active federation*, where an application takes on responsibility for supplying credentials without further user interaction, but that scenario is outside the scope of this architecture.

For additional considerations, see Choose a solution for integrating on-premises Active Directory with Azure.

Architecture

This architecture extends the implementation described in Extending AD DS to Azure. It contains the following components.

- **AD DS subnet**. The AD DS servers are contained in their own subnet with network security group (NSG) rules acting as a firewall.
- **AD DS servers**. Domain controllers running as VMs in Azure. These servers provide authentication of local identities within the domain.
- **AD FS subnet**. The AD FS servers are located within their own subnet with NSG rules acting as a firewall.
- **AD FS servers**. The AD FS servers provide federated authorization and authentication. In this architecture, they perform the following tasks:
  - Receiving security tokens containing claims made by a partner federation server on behalf of a partner user. AD FS verifies that the tokens are valid before passing the claims to the web application running in Azure to authorize requests.

    The application running in Azure is the *relying party*. The partner federation server must issue claims that are understood by the web application. The partner federation servers are referred to as *account partners*, because they submit access requests on behalf of authenticated accounts in the partner organization. The AD FS servers are called *resource partners* because they provide access to resources (the web application).
  - Authenticating and authorizing incoming requests from external users running a web browser or device that needs access to web applications, by using AD DS and the Active Directory Device Registration Service.

  The AD FS servers are configured as a farm accessed through an Azure load balancer. This implementation improves availability and scalability. The AD FS servers are not exposed directly to the Internet. All Internet traffic is filtered through AD FS web application proxy servers and a DMZ (also referred to as a perimeter network).

For more information about how AD FS works, see Active Directory Federation Services Overview. Also, the article AD FS deployment in Azure contains a detailed step-by-step introduction to implementation.

- **AD FS proxy subnet**. The AD FS proxy servers can be contained within their own subnet, with NSG rules providing protection. The servers in this subnet are exposed to the Internet through a set of network virtual appliances that provide a firewall between your Azure virtual network and the Internet.
- **AD FS web application proxy (WAP) servers**. These VMs act as AD FS servers for incoming requests from partner organizations and external devices. The WAP servers act as a filter, shielding the AD FS servers from direct access from the Internet. As with the AD FS servers, deploying the WAP servers in a farm with load balancing gives you greater availability and scalability than deploying a collection of stand-alone servers.

  > **Note**
  >
  > For detailed information about installing WAP servers, see **Install and Configure the Web Application Proxy Server**

- **Partner organization**. A partner organization running a web application that requests access to a web application running in Azure. The federation server at the partner organization authenticates requests locally, and submits security tokens containing claims to AD FS running in Azure. AD FS in Azure validates the security tokens, and if valid can pass the claims to the web application running in Azure to authorize them.

  > **Note**
  >
  > You can also configure a VPN tunnel using Azure gateway to provide direct access to AD FS for trusted partners. Requests received from these partners do not pass through the WAP servers.

Recommendations

The following recommendations apply for most scenarios. Follow these recommendations unless you have a specific requirement that overrides them.

Networking recommendations

Configure the network interface for each of the VMs hosting AD FS and WAP servers with static private IP addresses.

Do not give the AD FS VMs public IP addresses. For more information, see the Security considerations section.

Set the IP address of the preferred and secondary domain name service (DNS) servers for the network interfaces for each AD FS and WAP VM to reference the Active Directory DS VMs. The Active Directory DS VMs should be running DNS. This step is necessary to enable each VM to join the domain.

AD FS installation

The article Deploying a Federation Server Farm provides detailed instructions for installing and configuring AD FS. Perform the following tasks before configuring the first AD FS server in the farm:

1. Obtain a publicly trusted certificate for performing server authentication. The *subject name* must contain the name clients use to access the federation service. This can be the DNS name registered for the load balancer, for example, *adfs.contoso.com* (avoid using wildcard names such as **.contoso.com*, for security reasons). Use the same certificate on all AD FS server VMs. You can purchase a certificate from a trusted certification authority, but if your organization uses Active Directory Certificate Services you can create your own.

The *subject alternative name* is used by the device registration service (DRS) to enable access from external devices. This should be of the form *enterpriseregistration.contoso.com*.

For more information, see Obtain and Configure a Secure Sockets Layer (SSL) Certificate for AD FS.

2. On the domain controller, generate a new root key for the Key Distribution Service. Set the effective time to the current time minus 10 hours (this configuration reduces the delay that can occur in distributing and synchronizing keys across the domain). This step is necessary to support creating the group service account that is used to run the AD FS service. The following PowerShell command shows an example of how to do this:

```
Add-KdsRootKey -EffectiveTime (Get-Date).AddHours(-10)
```

3. Add each AD FS server VM to the domain.

> **Note**
>
> To install AD FS, the domain controller running the primary domain controller (PDC) emulator flexible single master operation (FSMO) role for the domain must be running and accessible from the AD FS VMs. <<RBC: Is there a way to make this less repetitive?>>

AD FS trust

Establish federation trust between your AD FS installation, and the federation servers of any partner organizations. Configure any claims filtering and mapping required.

- DevOps staff at each partner organization must add a relying party trust for the web applications accessible through your AD FS servers.
- DevOps staff in your organization must configure claims-provider trust to enable your AD FS servers to trust the claims that partner organizations provide.
- DevOps staff in your organization must also configure AD FS to pass claims on to your organization's web applications.

For more information, see Establishing Federation Trust.

Publish your organization's web applications and make them available to external partners by using preauthentication through the WAP servers. For more information, see Publish Applications using AD FS Preauthentication

AD FS supports token transformation and augmentation. Azure Active Directory does not provide this feature. With AD FS, when you set up the trust relationships, you can:

- Configure claim transformations for authorization rules. For example, you can map group security from a representation used by a non-Microsoft partner organization to something that Active Directory DS can authorize in your organization.
- Transform claims from one format to another. For example, you can map from SAML 2.0 to SAML 1.1 if your application only supports SAML 1.1 claims.

AD FS monitoring

The Microsoft System Center Management Pack for Active Directory Federation Services 2012 R2 provides both proactive and reactive monitoring of your AD FS deployment for the federation server. This management pack monitors:

- Events that the AD FS service records in its event logs.

- The performance data that the AD FS performance counters collect.
- The overall health of the AD FS system and web applications (relying parties), and provides alerts for critical issues and warnings.

Scalability considerations

The following considerations, summarized from the article Plan your AD FS deployment, give a starting point for sizing AD FS farms:

- If you have fewer than 1000 users, do not create dedicated servers, but instead install AD FS on each of the Active Directory DS servers in the cloud. Make sure that you have at least two Active Directory DS servers to maintain availability. Create a single WAP server.
- If you have between 1000 and 15000 users, create two dedicated AD FS servers and two dedicated WAP servers.
- If you have between 15000 and 60000 users, create between three and five dedicated AD FS servers and at least two dedicated WAP servers.

These considerations assume that you are using dual quad-core VM (Standard D4_v2, or better) sizes in Azure.

If you are using the Windows Internal Database to store AD FS configuration data, you are limited to eight AD FS servers in the farm. If you anticipate that you will need more in the future, use SQL Server. For more information, see The Role of the AD FS Configuration Database.

Availability considerations

Create an AD FS farm with at least two servers to increase availability of the service. Use different storage accounts for each AD FS VM in the farm. This approach helps to ensure that a failure in a single storage account does not make the entire farm inaccessible.

Create separate Azure availability sets for the AD FS and WAP VMs. Ensure that there are at least two VMs in each set. Each availability set must have at least two update domains and two fault domains.

Configure the load balancers for the AD FS VMs and WAP VMs as follows:

- Use an Azure load balancer to provide external access to the WAP VMs, and an internal load balancer to distribute the load across the AD FS servers in the farm.
- Only pass traffic appearing on port 443 (HTTPS) to the AD FS/WAP servers.
- Give the load balancer a static IP address.
- Create a health probe using HTTP against `/adfs/probe`. For more information, see Hardware Load Balancer Health Checks and Web Application Proxy / AD FS 2012 R2.

  > **Note**
  >
  > AD FS servers use the Server Name Indication (SNI) protocol, so attempting to probe using an HTTPS endpoint from the load balancer fails.

- Add a DNS *A* record to the domain for the AD FS load balancer. Specify the IP address of the load balancer, and give it a name in the domain (such as adfs.contoso.com). This is the name clients and the WAP servers use to access the AD FS server farm.

You can use either SQL Server or the Windows Internal Database to hold AD FS configuration information. The Windows Internal Database provides basic redundancy. Changes are written directly to only one of the AD FS

databases in the AD FS cluster, while the other servers use pull replication to keep their databases up to date. Using SQL Server can provide full database redundancy and high availability using failover clustering or mirroring.

Manageability considerations

DevOps staff should be prepared to perform the following tasks:

- Managing the federation servers, including managing the AD FS farm, managing trust policy on the federation servers, and managing the certificates used by the federation services.
- Managing the WAP servers including managing the WAP farm and certificates.
- Managing web applications including configuring relying parties, authentication methods, and claims mappings.
- Backing up AD FS components.

Security considerations

AD FS uses HTTPS, so make sure that the NSG rules for the subnet containing the web tier VMs permit HTTPS requests. These requests can originate from the on-premises network, the subnets containing the web tier, business tier, data tier, private DMZ, public DMZ, and the subnet containing the AD FS servers.

Prevent direct exposure of the AD FS servers to the Internet. AD FS servers are domain-joined computers that have full authorization to grant security tokens. If a server is compromised, a malicious user can issue full access tokens to all web applications and to all federation servers that are protected by AD FS. If your system must handle requests from external users not connecting from trusted partner sites, use WAP servers to handle these requests. For more information, see Where to Place a Federation Server Proxy.

Place AD FS servers and WAP servers in separate subnets with their own firewalls. You can use NSG rules to define firewall rules. All firewalls should allow traffic on port 443 (HTTPS).

Restrict direct sign in access to the AD FS and WAP servers. Only DevOps staff should be able to connect. Do not join the WAP servers to the domain.

Consider using a set of network virtual appliances that logs detailed information on traffic traversing the edge of your virtual network for auditing purposes.

Deploy the solution

A deployment for this architecture is available on GitHub. Note that the entire deployment can take up to two hours, which includes creating the VPN gateway and running the scripts that configure Active Directory and AD FS.

Prerequisites

1. Clone, fork, or download the zip file for the GitHub repository.
2. Install Azure CLI 2.0.
3. Install the Azure building blocks npm package.

   ```
   npm install -g @mspnp/azure-building-blocks
   ```

4. From a command prompt, bash prompt, or PowerShell prompt, sign into your Azure account as follows:

   ```
   
   ```

```
az login
```

Deploy the simulated on-premises datacenter

1. Navigate to the `adfs` folder of the GitHub repository.
2. Open the `onprem.json` file. Search for instances of `adminPassword`, `Password`, and `SafeModeAdminPassword` and update the passwords.
3. Run the following command and wait for the deployment to finish:

```
azbb –s <subscription_id> –g <resource group> –l <location> –p onprem.json ––deploy
```

Deploy the Azure infrastructure

1. Open the `azure.json` file. Search for instances of `adminPassword` and `Password` and add values for the passwords.
2. Run the following command and wait for the deployment to finish:

```
azbb –s <subscription_id> –g <resource group> –l <location> –p azure.json ––deploy
```

Set up the AD FS farm

1. Open the `adfs-farm-first.json` file. Search for `AdminPassword` and replace the default password.
2. Run the following command:

```
azbb –s <subscription_id> –g <resource group> –l <location> –p adfs-farm-first.json ––
deploy
```

3. Open the `adfs-farm-rest.json` file. Search for `AdminPassword` and replace the default password.
4. Run the following command and wait for the deployment to finish:

```
azbb –s <subscription_id> –g <resource group> –l <location> –p adfs-farm-rest.json ––deploy
```

Configure AD FS (part 1)

1. Open a remote desktop session to the VM named `ra-adfs-jb-vm1`, which is the jumpbox VM. The user name is `testuser`.
2. From the jumpbox, open a remote desktop session to the VM named `ra-adfs-proxy-vm1`. The private IP address is 10.0.6.4.
3. From this remote desktop session, run the PowerShell ISE.

4. In PowerShell, navigate to the following directory:

```
C:\Packages\Plugins\Microsoft.Powershell.DSC\2.77.0.0\DSCWork\adfs-v2.0
```

5. Paste the following code into a script pane and run it:

```
. .\adfs-webproxy.ps1
$cd = @{
    AllNodes = @(
        @{
            NodeName = 'localhost'
            PSDscAllowPlainTextPassword = $true
            PSDscAllowDomainUser = $true
        }
    )
}

$c1 = Get-Credential -UserName testuser -Message "Enter password"
InstallWebProxyApp -DomainName contoso.com -FederationName adfs.contoso.com -WebApplica-
tionProxyName "Contoso App" -AdminCreds $c1 -ConfigurationData $cd
Start-DscConfiguration .\InstallWebProxyApp
```

At the `Get-Credential` prompt, enter the password that you specified in the deployment parameter file.

6. Run the following command to monitor the progress of the DSC configuration:

```
Get-DscConfigurationStatus
```

It can take several minutes to reach consistency. During this time, you may see errors from the command. When the configuration succeeds, the output should look similar to the following:

```
PS C:\Packages\Plugins\Microsoft.Powershell.DSC\2.77.0.0\DSCWork\adfs-v2.0> Get-DscConfigu-
rationStatus

Status     StartDate                 Type            Mode  RebootRequested      NumberOfRe-
sources
------     ---------                 ----            ----  ---------------      ----------
------
Success    12/17/2018 8:21:09 PM     Consistency     PUSH  True                 4
```

Configure AD FS (part 2)

1. From the jumpbox, open a remote desktop session to the VM named `ra-adfs-proxy-vm2`. The private IP address is 10.0.6.5.

2. From this remote desktop session, run the PowerShell ISE.

3. Navigate to the following directory:

```
C:\Packages\Plugins\Microsoft.Powershell.DSC\2.77.0.0\DSCWork\adfs-v2.0
```

4. Past the following in a script pane and run the script:

```
. .\adfs-webproxy-rest.ps1
$cd = @{
    AllNodes = @(
        @{
            NodeName = 'localhost'
            PSDscAllowPlainTextPassword = $true
            PSDscAllowDomainUser = $true
        }
    )
}

$c1 = Get-Credential -UserName testuser -Message "Enter password"
InstallWebProxy -DomainName contoso.com -FederationName adfs.contoso.com -WebApplication-
ProxyName "Contoso App" -AdminCreds $c1 -ConfigurationData $cd
Start-DscConfiguration .\InstallWebProxy
```

   At the `Get-Credential` prompt, enter the password that you specified in the deployment parameter file.

5. Run the following command to monitor the progress of the DSC configuration:

```
Get-DscConfigurationStatus
```

   It can take several minutes to reach consistency. During this time, you may see errors from the command. When the configuration succeeds, the output should look similar to the following:

```
PS C:\Packages\Plugins\Microsoft.Powershell.DSC\2.77.0.0\DSCWork\adfs-v2.0> Get-DscConfigu-
rationStatus

Status     StartDate                 Type            Mode  RebootRequested      NumberOfRe-
sources
------     ---------                 ----            ----  ---------------      ----------
------
Success    12/17/2018 8:21:09 PM     Consistency     PUSH  True                 4
```

   Sometimes this DSC fails. If the status check shows `Status=Failure` and `Type=Consistency`, try re-running step 4.

Sign into AD FS

1. From the jumpbox, open a remote desktop session to the VM named `ra-adfs-adfs-vm1`. The private IP address is 10.0.5.4.

2. Follow the steps in Enable the Idp-Initiated Sign on page to enable the sign-on page.

3. From the jump box, browse to `https://adfs.contoso.com/adfs/ls/idpinitiatedsignon.htm`. You may receive a certificate warning that you can ignore for this test.

4. Verify that the Contoso Corporation sign-in page appears. Sign in as **contoso\testuser**.