AWS Security Blog

Enabling Federation to AWS Using Windows Active Directory, ADFS, and SAML 2.0

by Jeff Wierer | on 10 DEC 2013 | in How-To | Permalink | Comments | Share

Update from January 17, 2018: The techniques demonstrated in this blog post relate to traditional SAML federation for AWS. These techniques are still valid and useful. However, AWS Single Sign-On (AWS SSO) provides analogous capabilities by way of a managed service. If you are just getting started with federating access to your AWS accounts, we recommend that you evaluate AWS SSO for this purpose.

At this year's re:Invent I had the opportunity to present on the topic of delegating access to your AWS environment. One use case I demonstrated was enterprise federation to AWS using Windows Active Directory (AD), Active Directory Federation Services (ADFS) 2.0, and SAML (Security Assertion Markup Language) 2.0. The presentation must have struck a nerve, because a number of folks approached me afterwards and asked me if I could publish my configuration—hence the inspiration for this post.

In this post I describe the use case for enterprise federation, describe how the integration between ADFS and AWS works, and then provide the setup details that I used for my re:Invent demo. If you missed my session and you're interested in hearing my talk, you can catch the recording or view my slides.

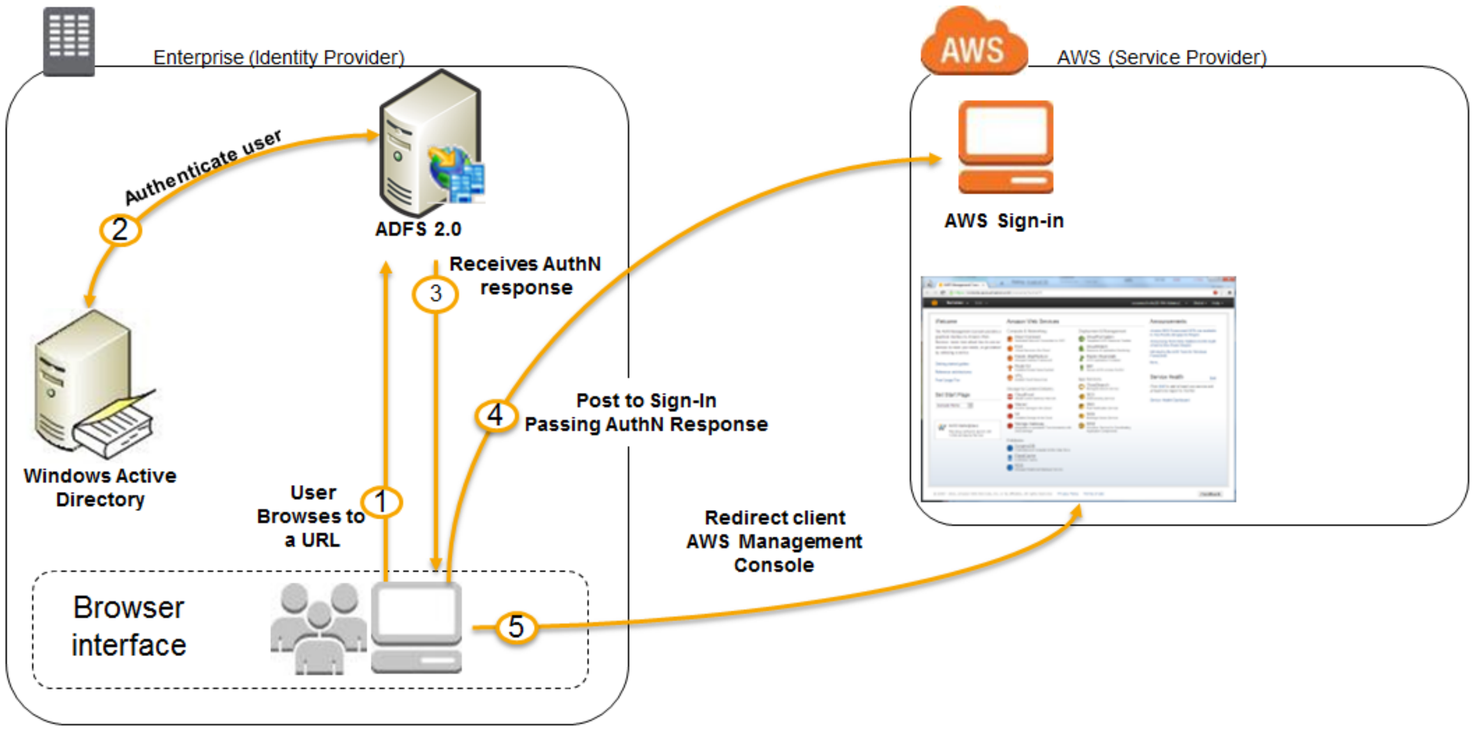
Background

AWS recently added support for SAML, an open standard used by many identity providers. This new feature enables federated single sign-on (SSO), which lets users sign into the AWS Management Console or make programmatic calls to AWS APIs by using assertions from a SAML-compliant identity provider (IdP) like ADFS. Many of you are using Windows AD for your corporate directory. And since Windows Server includes ADFS, it makes sense that you might use ADFS as your IdP. That's one reason I used Windows AD with ADFS as one of my re:Invent demos.

To set up my domain, I used Amazon EC2 because that made it easy to access the domain from anywhere. My EC2 instance used Windows Server 2008 R2 running Internet Information Server (IIS), AD, and ADFS. If you want to do the same, I encourage you to use a nifty CloudFormation template that creates a Windows instance and sets up a domain for you.

How Integration Between AD FS and AWS Works

Before we get too far into the configuration details, let's walk through how this all works.



1. The flow is initiated when a user (let's call him Bob) browses to the ADFS sample site (https://*Fully.Qualified.Domain.Name.Here*/adfs/ls/IdpInitiatedSignOn.aspx) inside his domain. When you install ADFS, you get a new virtual directory named adfs for your default website, which includes this page
2. The sign-on page authenticates Bob against AD. Depending on the browser Bob is using, he might be prompted for his AD username and password.
3. Bob's browser receives a SAML assertion in the form of an authentication response from ADFS.
4. Bob's browser posts the SAML assertion to the AWS sign-in endpoint for SAML (https://signin.aws.amazon.com/saml). Behind the scenes, sign-in uses the AssumeRoleWithSAML API to request temporary security credentials and then constructs a sign-in URL for the AWS Management Console.
5. Bob's browser receives the sign-in URL and is redirected to the console.

From Bob's perspective, the process happens transparently. He starts at an internal web site and ends up at the AWS Management Console, without ever having to supply any AWS credentials.

Now that we understand how it works, let's take a look at setting it all up.

By the way, this post is fairly long. The next couple sections cover installing and configuring ADFS. If you already have ADFS in your environment, you may want to skip ahead to the Configuring AWS section.

Configuring Active Directory

If you want follow along with my description, you're going to need a Windows domain. If you don't already have one, I recommend that you take advantage of the CloudFormation template I mentioned earlier to quickly launch an Amazon EC2 Windows instance as a Windows AD domain controller. For demonstration purposes, I used a single user (Bob) who is a member of two AD groups (AWS-Production and AWS-Dev) and a service account (ADFSSVC) used by ADFS. Note that the names of the AD groups both start with AWS-. This is significant, because Bob's permission to sign in to AWS will be based on a match of group names that start with AWS-, as I'll explain later.

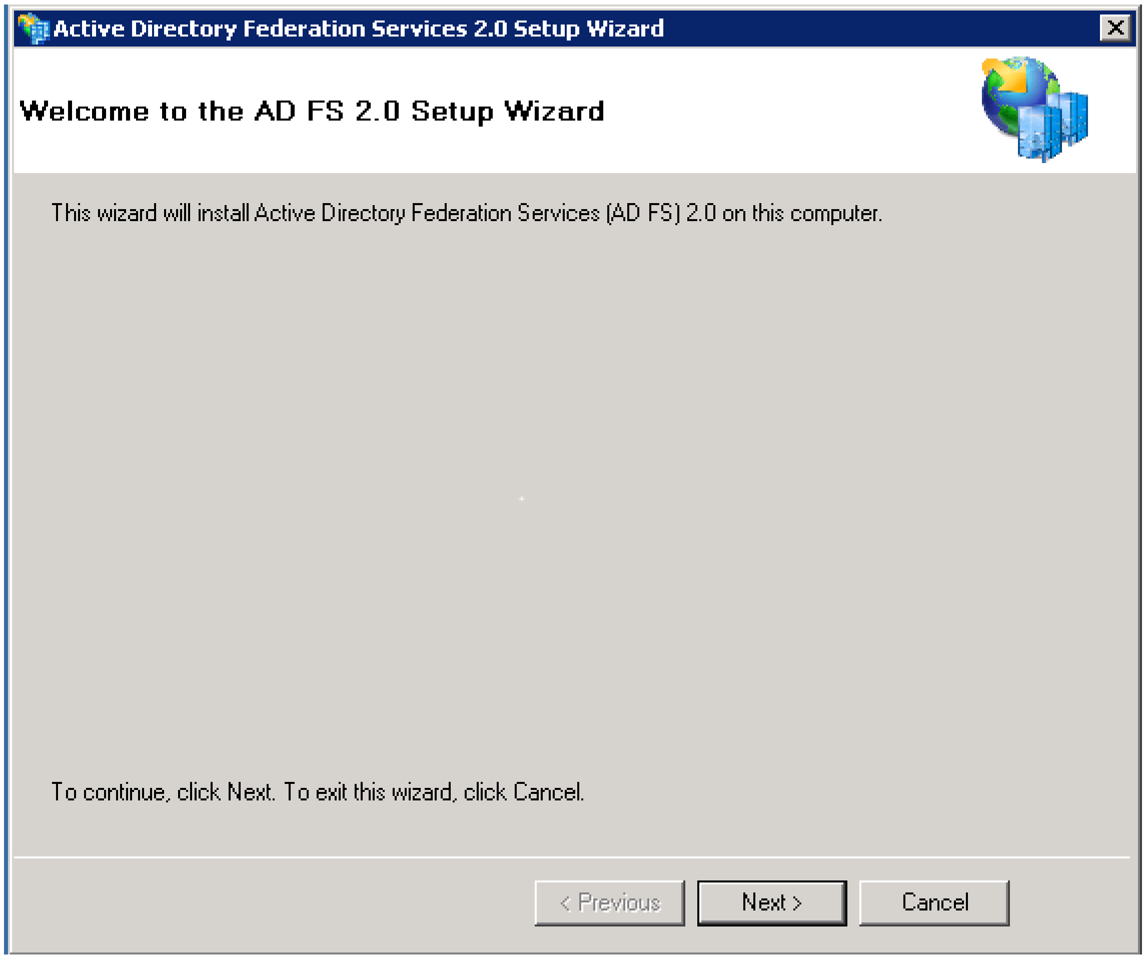
Note If you follow along with the instructions, make sure you use exactly the same names we do for users, AD groups, and IAM roles, including uppercase and lowercase letters.

Perform the following in your domain:

1. Create two AD Groups named AWS-Production and AWS-Dev.
2. Create a user named Bob
3. Give Bob an email address (e.g., bob@example.com).
4. Add Bob to the AWS-Production and AWS-Dev groups.
5. Create another user named ADFSSVC. This account will be used as the ADFS service account later on.

Installing ADFS

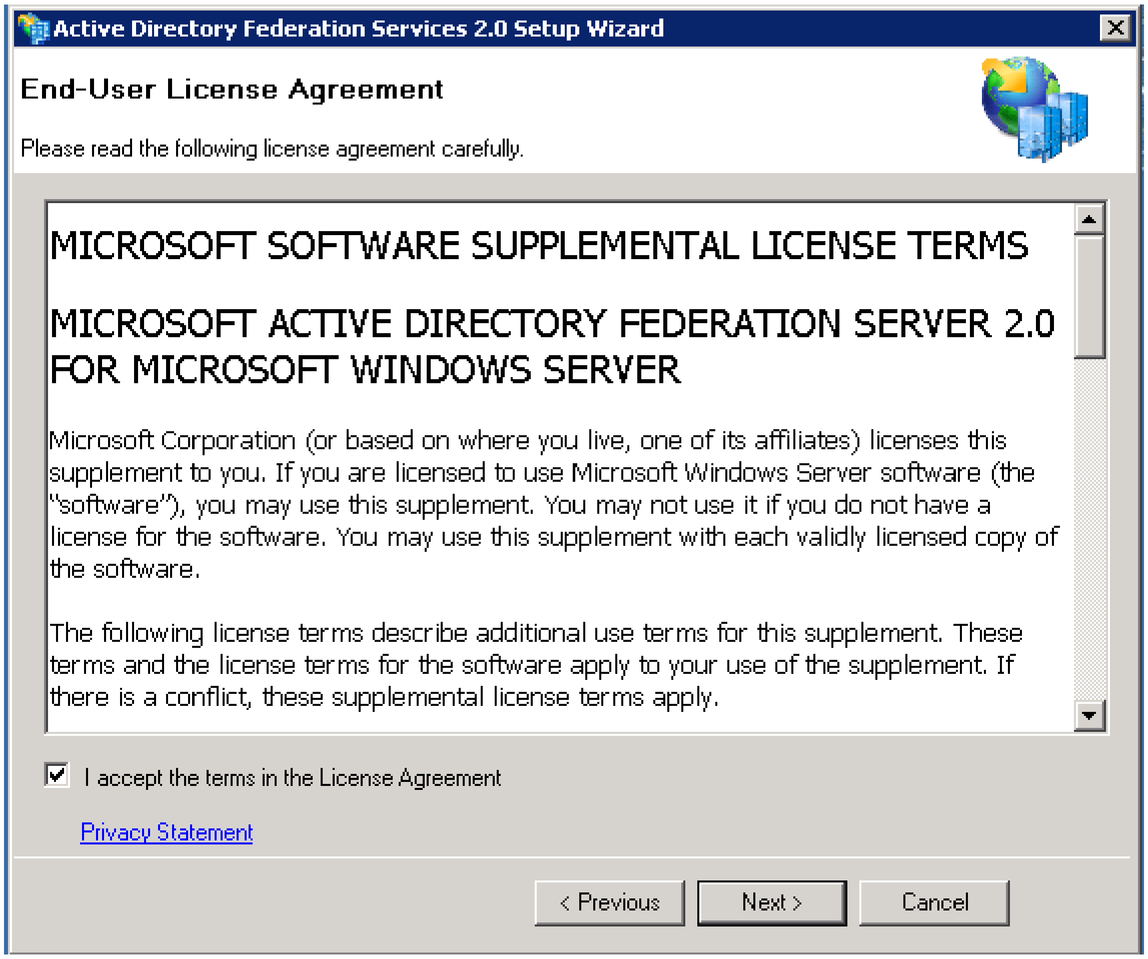
With my accounts and groups set up, I moved on to installing ADFS. The Windows Server 2008 R2 I used came with an older version of ADFS. I skipped installing that version and instead downloaded ADFS 2.0. After downloading the package, you launch the ADFS setup wizard by double-clicking AdfsSetup.exe. I set up my environment as a federation server using the default settings. In other words, I made no special settings. The screenshots show the process.


Active Directory Federation Services 2.0 Setup Wizard

Welcome to the AD FS 2.0 Setup Wizard

This wizard will install Active Directory Federation Services (AD FS) 2.0 on this computer.

To continue, click Next. To exit this wizard, click Cancel.



Active Directory Federation Services 2.0 Setup Wizard

End-User License Agreement

Please read the following license agreement carefully.

MICROSOFT SOFTWARE SUPPLEMENTAL LICENSE TERMS

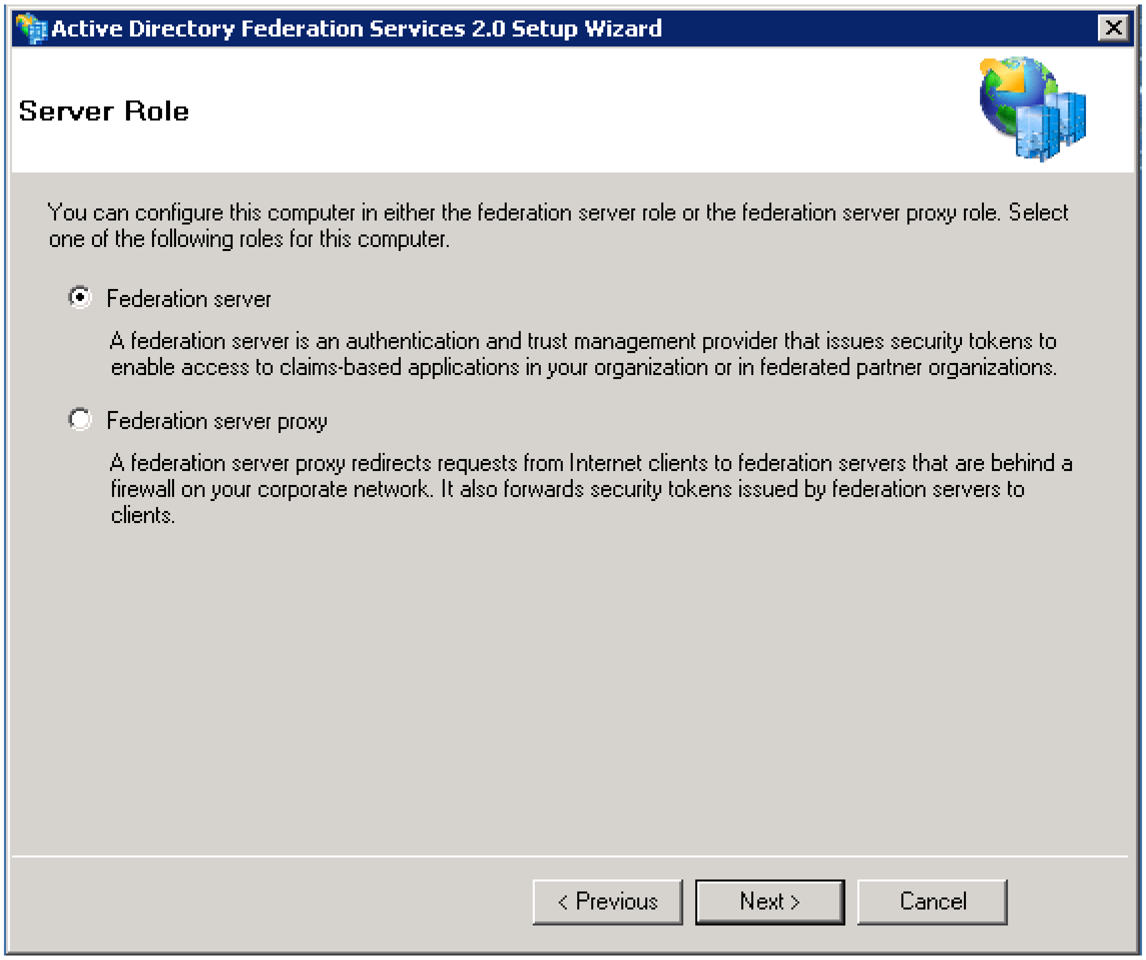
MICROSOFT ACTIVE DIRECTORY FEDERATION SERVER 2.0 FOR MICROSOFT WINDOWS SERVER

Microsoft Corporation (or based on where you live, one of its affiliates) licenses this supplement to you. If you are licensed to use Microsoft Windows Server software (the "software"), you may use this supplement. You may not use it if you do not have a license for the software. You may use this supplement with each validly licensed copy of the software.

The following license terms describe additional use terms for this supplement. These terms and the license terms for the software apply to your use of the supplement. If there is a conflict, these supplemental license terms apply.

☑ I accept the terms in the License Agreement

Privacy Statement



Active Directory Federation Services 2.0 Setup Wizard

Server Role

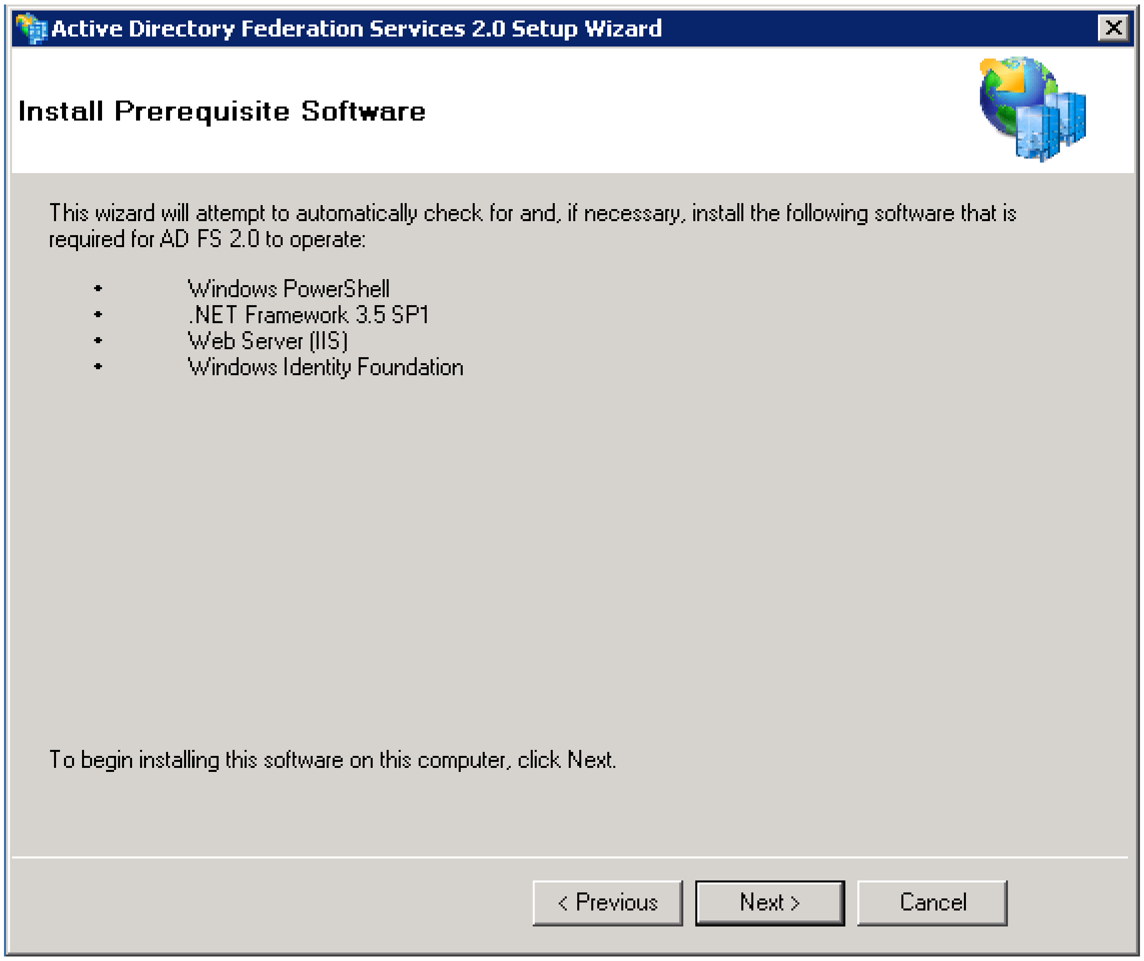
You can configure this computer in either the federation server role or the federation server proxy role. Select one of the following roles for this computer.

● Federation server

A federation server is an authentication and trust management provider that issues security tokens to enable access to claims-based applications in your organization or in federated partner organizations.

○ Federation server proxy

A federation server proxy redirects requests from Internet clients to federation servers that are behind a firewall on your corporate network. It also forwards security tokens issued by federation servers to clients.

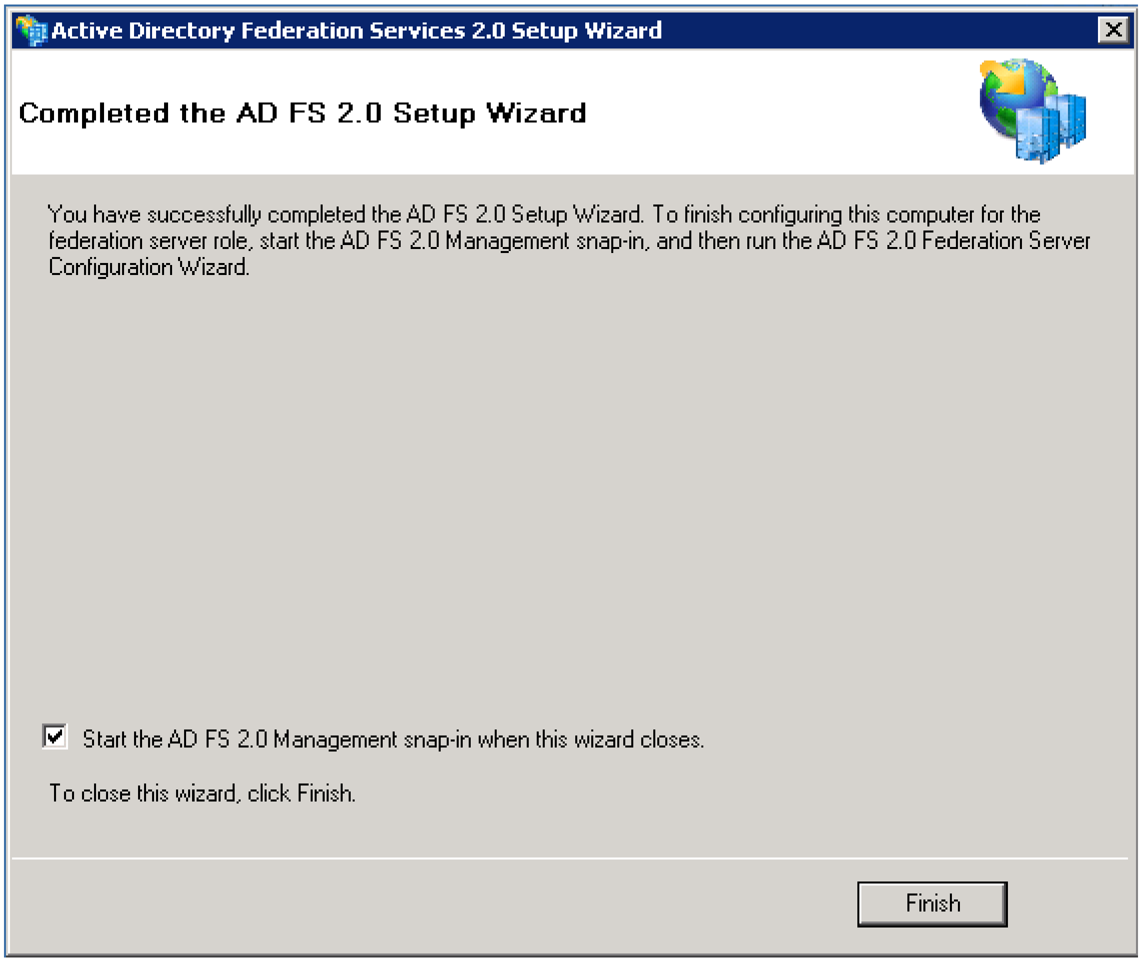


Install Prerequisite Software

This wizard will attempt to automatically check for and, if necessary, install the following software that is required for AD FS 2.0 to operate:

- Windows PowerShell
- .NET Framework 3.5 SP1
- Web Server (IIS)
- Windows Identity Foundation

To begin installing this software on this computer, click Next.



Completed the AD FS 2.0 Setup Wizard

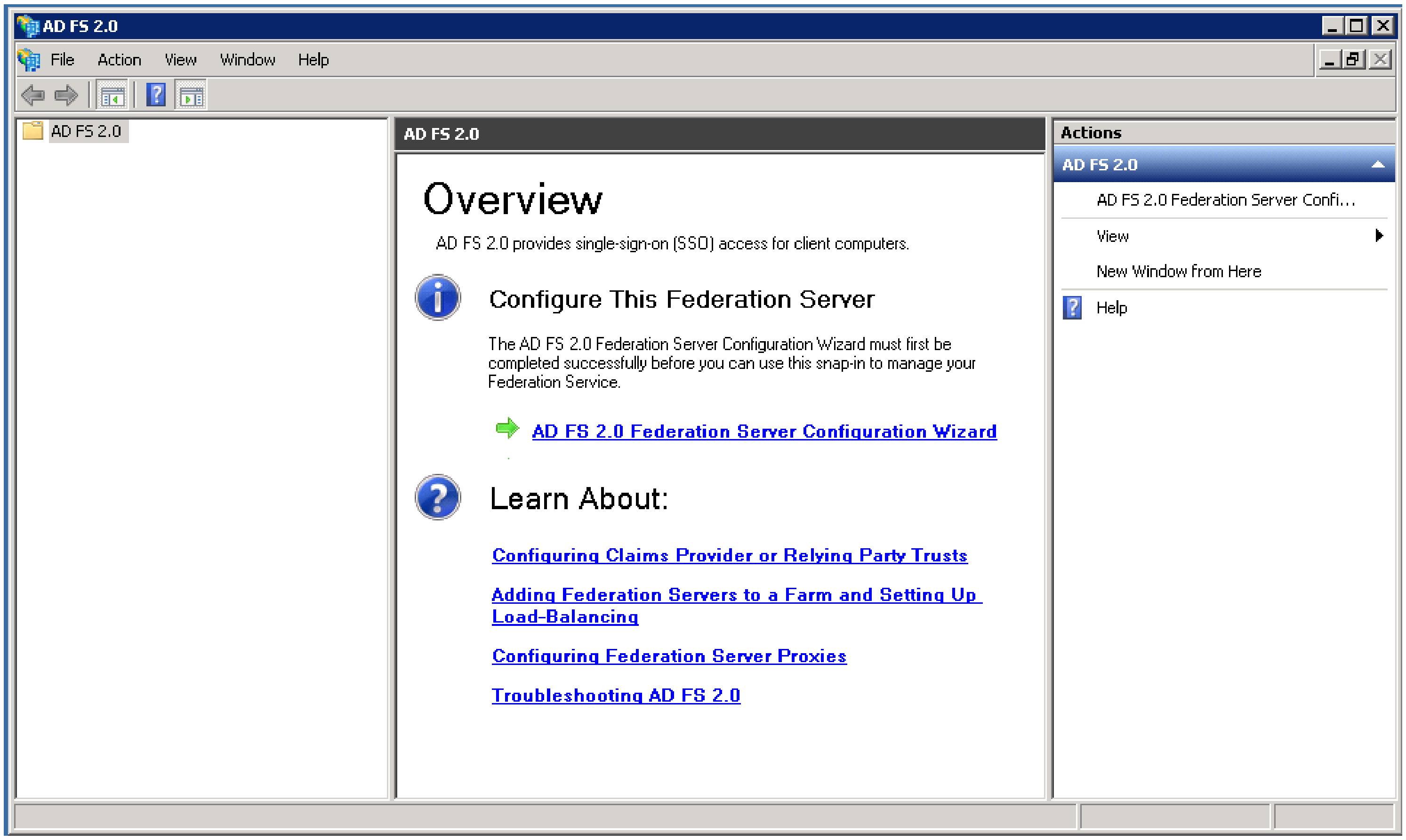
You have successfully completed the AD FS 2.0 Setup Wizard. To finish configuring this computer for the federation server role, start the AD FS 2.0 Management snap-in, and then run the AD FS 2.0 Federation Server Configuration Wizard.

[✓] Start the AD FS 2.0 Management snap-in when this wizard closes.

To close this wizard, click Finish.

Configuring ADFS

The next step is to configure ADFS. During setup, I checked the **Start the AD FS 2.0 Management snap-in when this wizard closes** box, so the window loaded after I clicked **Finish**. If you don't check that box during setup, you can get to the window from **Start > All Programs > Administration Tools > AD FS 2.0 Management**. When ADFS is launched, it looks like this:



Overview

AD FS 2.0 provides single-sign-on (SSO) access for client computers.

Configure This Federation Server

The AD FS 2.0 Federation Server Configuration Wizard must first be completed successfully before you can use this snap-in to manage your Federation Service.

AD FS 2.0 Federation Server Configuration Wizard

Learn About:

Configuring Claims Provider or Relying Party Trusts

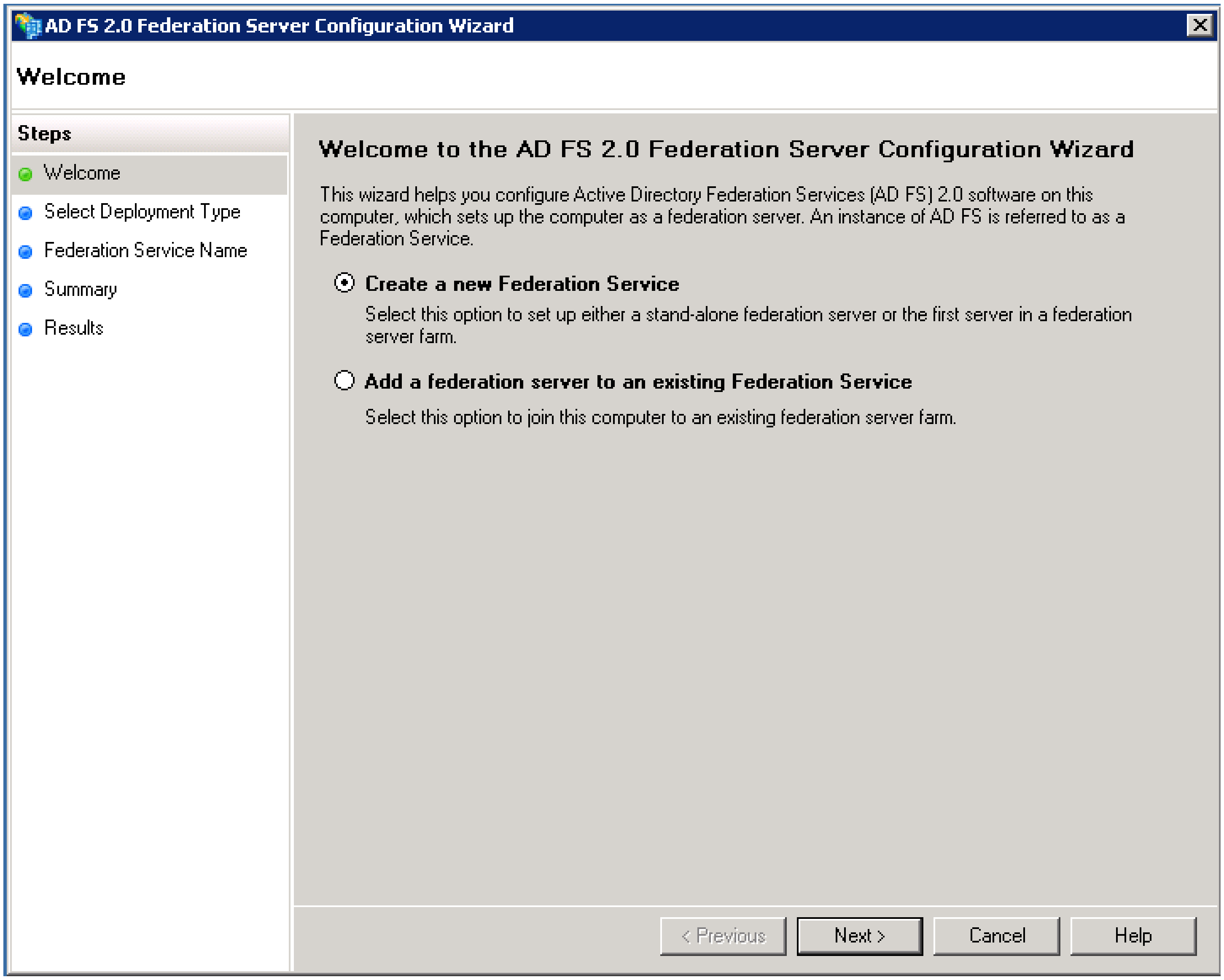
Adding Federation Servers to a Farm and Setting Up Load-Balancing

Configuring Federation Server Proxies

Troubleshooting AD FS 2.0

To launch the configuration wizard, you click **AD FS 2.0 Federation Server Configuration Wizard**. If you want to follow along with my configuration, do this:

1. Select **Create a new Federation Service**.


AD FS 2.0 Federation Server Configuration Wizard

Welcome

Steps
- Welcome
- Select Deployment Type
- Federation Service Name
- Summary
- Results

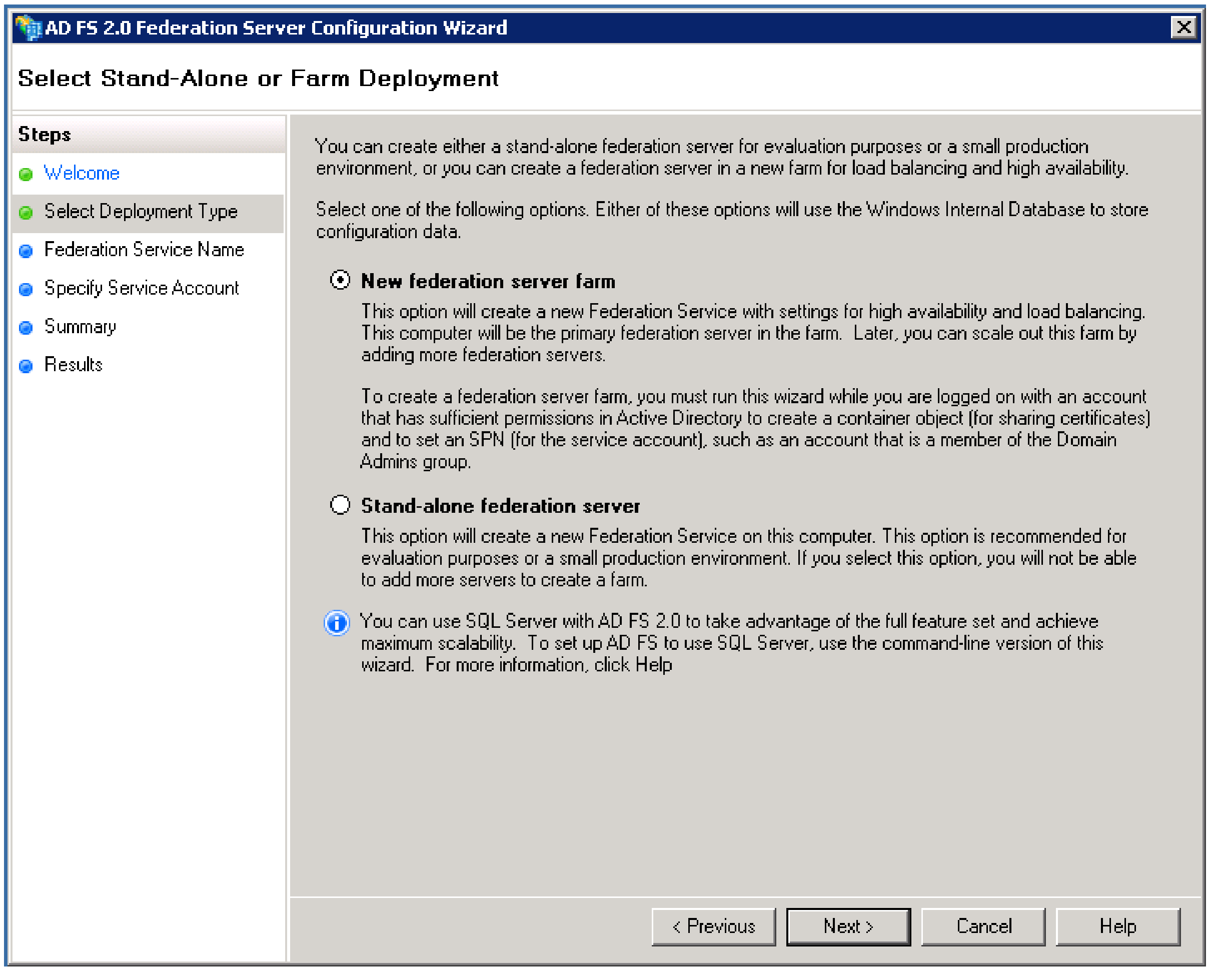
Welcome to the AD FS 2.0 Federation Server Configuration Wizard

This wizard helps you configure Active Directory Federation Services (AD FS) 2.0 software on this computer, which sets up the computer as a federation server. An instance of AD FS is referred to as a Federation Service.

[✓ Create a new Federation Service]
Select this option to set up either a stand-alone federation server or the first server in a federation server farm.

Add a federation server to an existing Federation Service
Select this option to join this computer to an existing federation server farm.

2. Select **New federation server farm**.



Select Stand-Alone or Farm Deployment

You can create either a stand-alone federation server for evaluation purposes or a small production environment, or you can create a federation server in a new farm for load balancing and high availability.

Select one of the following options. Either of these options will use the Windows Internal Database to store configuration data.

[✓ New federation server farm]

This option will create a new Federation Service with settings for high availability and load balancing. This computer will be the primary federation server in the farm. Later, you can scale out this farm by adding more federation servers.

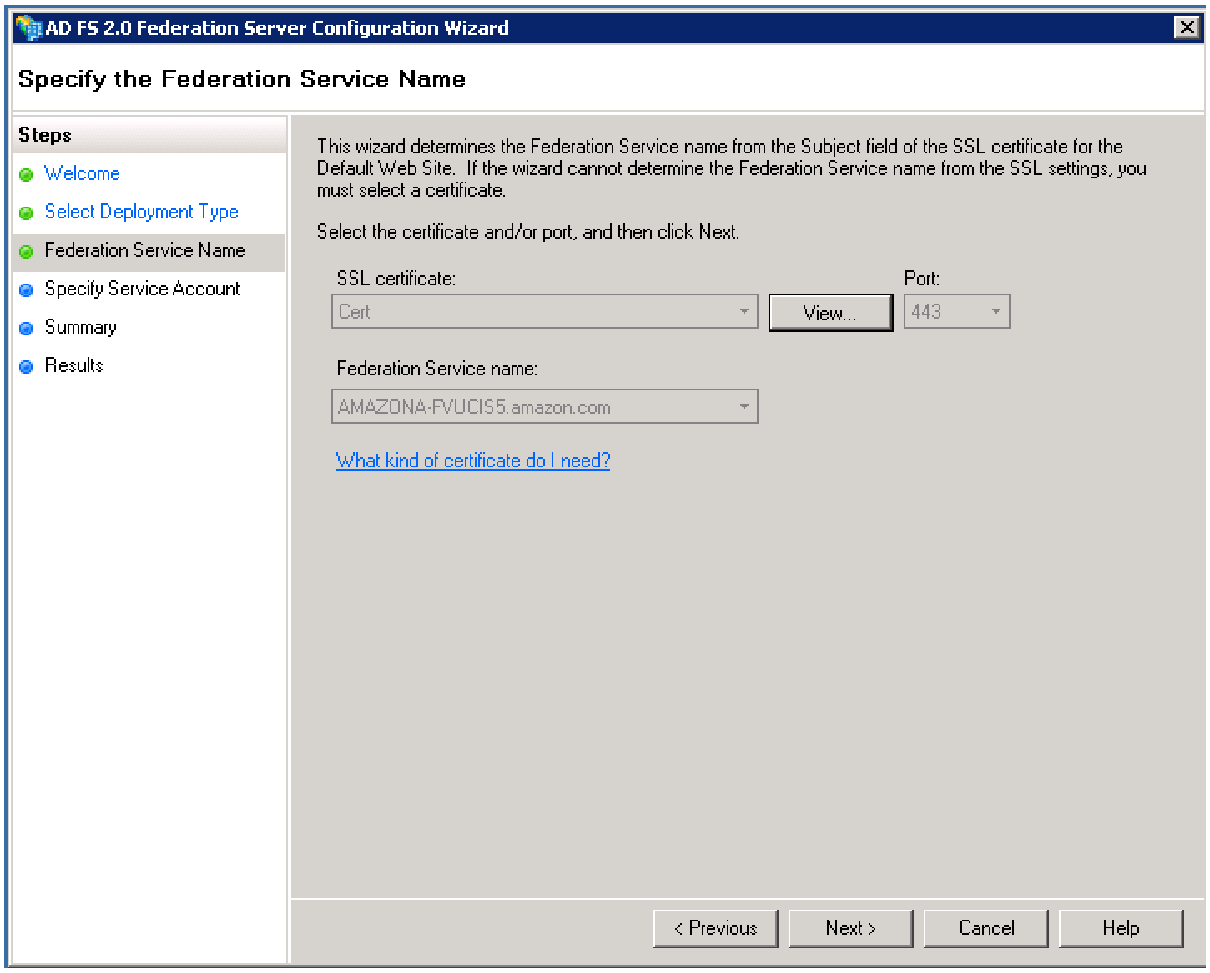
To create a federation server farm, you must run this wizard while you are logged on with an account that has sufficient permissions in Active Directory to create a container object (for sharing certificates) and to set an SPN (for the service account), such as an account that is a member of the Domain Admins group.

Stand-alone federation server

This option will create a new Federation Service on this computer. This option is recommended for evaluation purposes or a small production environment. If you select this option, you will not be able to add more servers to create a farm.

You can use SQL Server with AD FS 2.0 to take advantage of the full feature set and achieve maximum scalability. To set up AD FS to use SQL Server, use the command-line version of this wizard. For more information, click Help

3. Select an SSL certificate. On my instance, I had an existing certificate I could use. If you don't have a certificate, you can create a self-signed certificate using IIS. Self-signed certificates are convenient for testing and development. For production use, you'll want to use a certificate from a trusted certificate authority (CA).



Specify the Federation Service Name

This wizard determines the Federation Service name from the Subject field of the SSL certificate for the Default Web Site. If the wizard cannot determine the Federation Service name from the SSL settings, you must select a certificate.

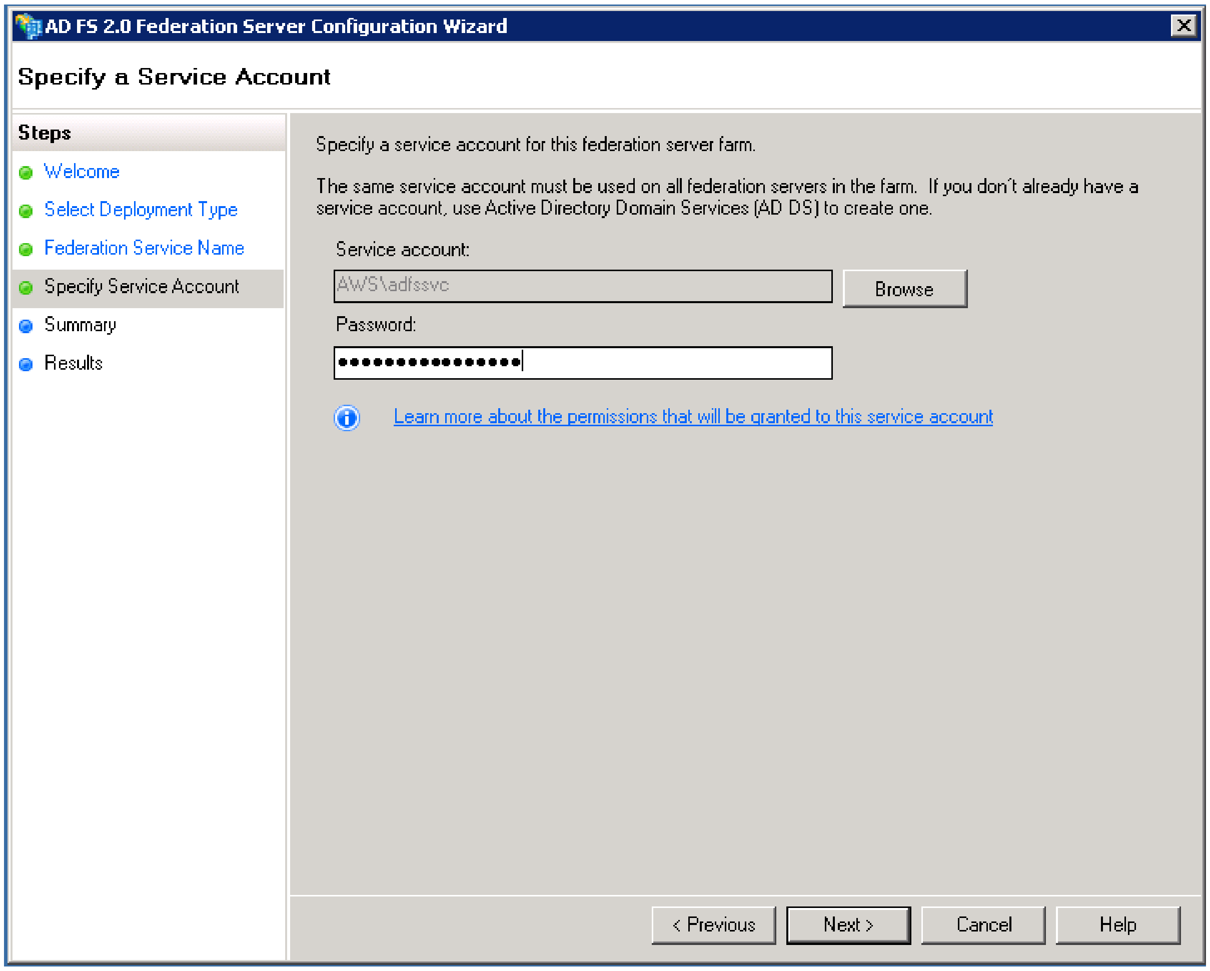
Select the certificate and/or port, and then click Next.

SSL certificate: Cert

Port: 443

Federation Service name: AMAZONA-FVUCIS5.amazon.com

4. Remember the service account I mentioned earlier? This is where you use it.


AD FS 2.0 Federation Server Configuration Wizard
Specify a Service Account
Steps
Welcome
Select Deployment Type
Federation Service Name
Specify Service Account
Summary
Results
Browse
Learn more about the permissions that will be granted to this service account
< Previous
Next >
Cancel
Help

Specify a service account for this federation server farm.

The same service account must be used on all federation servers in the farm. If you don't already have a service account, use Active Directory Domain Services (AD DS) to create one.

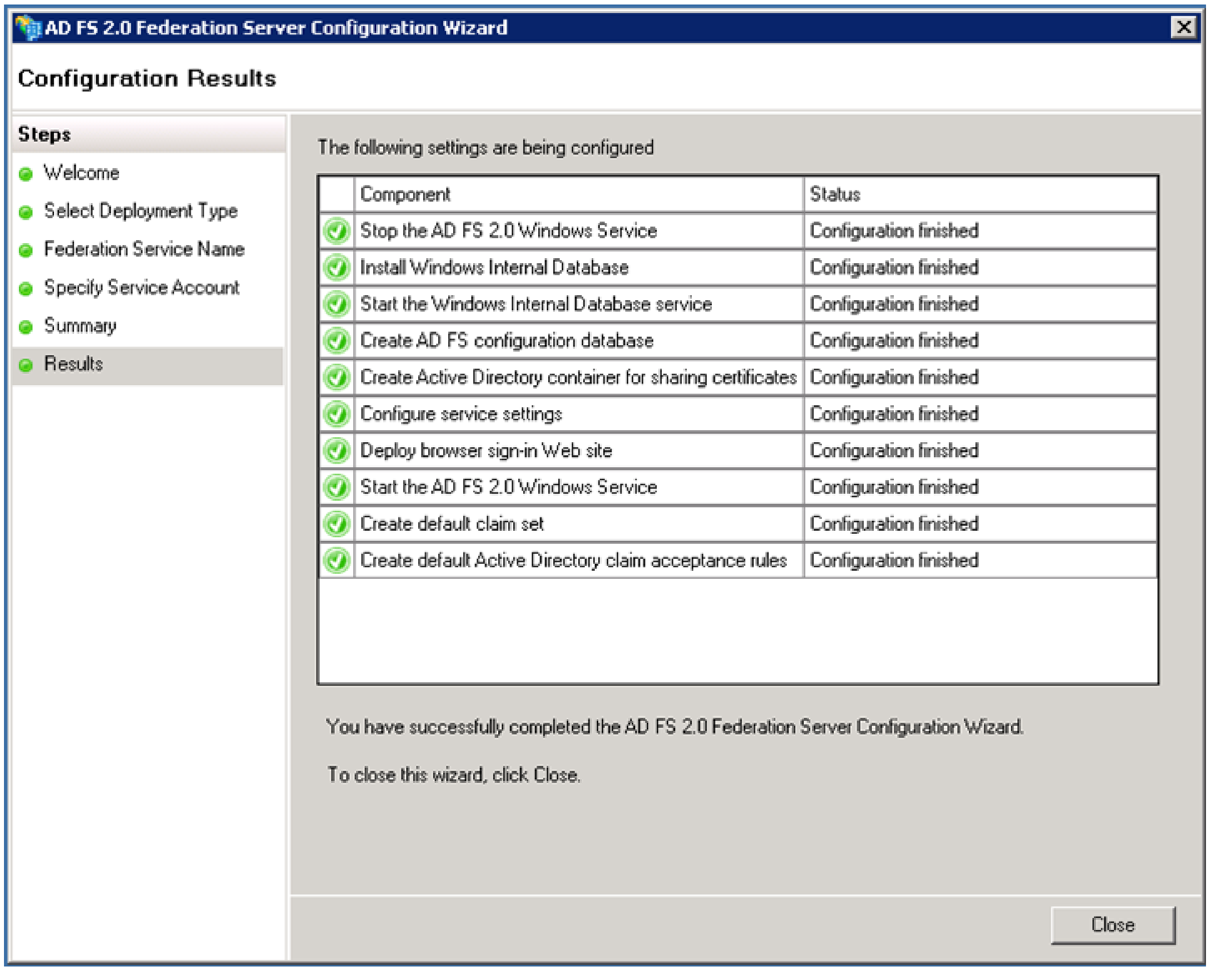
Service account: AWS\adfssvc

Password: ●●●●●●●●●●●●●●●●

5. Almost there – just need to confirm your settings and click **Next**.

6. Nothing left but to click **Close** to finish.

If all goes well you get a report with all successful configurations. If so, skip ahead to the **Configuring AWS** section.


AD FS 2.0 Federation Server Configuration Wizard

Configuration Results

Steps
- Welcome
- Select Deployment Type
- Federation Service Name
- Specify Service Account
- Summary
- Results

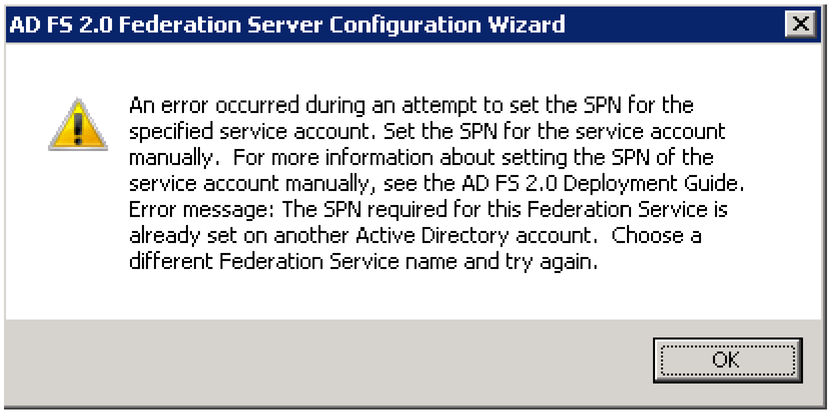
The following settings are being configured

	Component	Status
✓	Stop the AD FS 2.0 Windows Service	Configuration finished
✓	Install Windows Internal Database	Configuration finished
✓	Start the Windows Internal Database service	Configuration finished
✓	Create AD FS configuration database	Configuration finished
✓	Create Active Directory container for sharing certificates	Configuration finished
✓	Configure service settings	Configuration finished
✓	Deploy browser sign-in Web site	Configuration finished
✓	Start the AD FS 2.0 Windows Service	Configuration finished
✓	Create default claim set	Configuration finished
✓	Create default Active Directory claim acceptance rules	Configuration finished

You have successfully completed the AD FS 2.0 Federation Server Configuration Wizard.

To close this wizard, click Close.

During my testing, I went through this wizard on several different Windows servers and didn't always have 100% success. In some cases I encountered the following error message:


AD FS 2.0 Federation Server Configuration Wizard

An error occurred during an attempt to set the SPN for the specified service account. Set the SPN for the service account manually. For more information about setting the SPN of the service account manually, see the AD FS 2.0 Deployment Guide. Error message: The SPN required for this Federation Service is already set on another Active Directory account. Choose a different Federation Service name and try again.

It turns out this is a known issue that can be fixed by running the following at the command line. (Make sure you run the command window as an administrator.)

```
setspn -a host/localhost adfssvc
```

Note that is the name of the service account I used.

If the command is successful, you see output like this:

```
Registering ServicePrincipalNames for
CN=ADFSSVC,CN=Users,DC=mydomain,DC=aws,DC=amazon,DC=com
host/localhost
```

Configuring AWS

You've finished configuring AD FS. The next step is to configure the AWS end of things. To do this, I used the AWS Management Console.

The first step is to create a SAML provider. If you've never done this, I recommend taking a look at the IAM user guide. Before you create a SAML provider, you need to download the SAML metadata document for your ADFS federation server. By default, you can download it from following address:

https://<yourservername>/FederationMetadata/2007-06/FederationMetadata.xml

I named my SAML provider **ADFS**. When you have the SAML metadata document, you can create the SAML provider in AWS. As part of that process, you upload the metadata document.

When I finished creating the SAML provider, I created two IAM roles. Once again the IAM documentation has a great walkthrough of these steps, so I won't repeat them here. I created two roles using the **Grant Web Single Sign-On (WebSSO) access to SAML providers** role wizard template and specified the ADFS SAML provider that I just created. I named the two roles ADFS-Production and ADFS-Dev. Do these names look familiar? They should. They are the complement to the AD groups created earlier. During the SAML authentication process in AWS, these IAM roles will be matched by name to the AD groups (AWS-Production and AWS-Dev) via ADFS claim rules.

> **Note**: Remember that if you're following along with this description, you need to use exactly the same names that we use. Make sure that you name the IAM roles ADFS-Production and ADFS-Dev.

Find the ARNs for the SAML provider and for the roles that you created and record them. You'll need the ARNs later when you configure claims in the IdP.

That's it for the AWS configuration steps.

Configuring AWS as a Trusted Relying Party

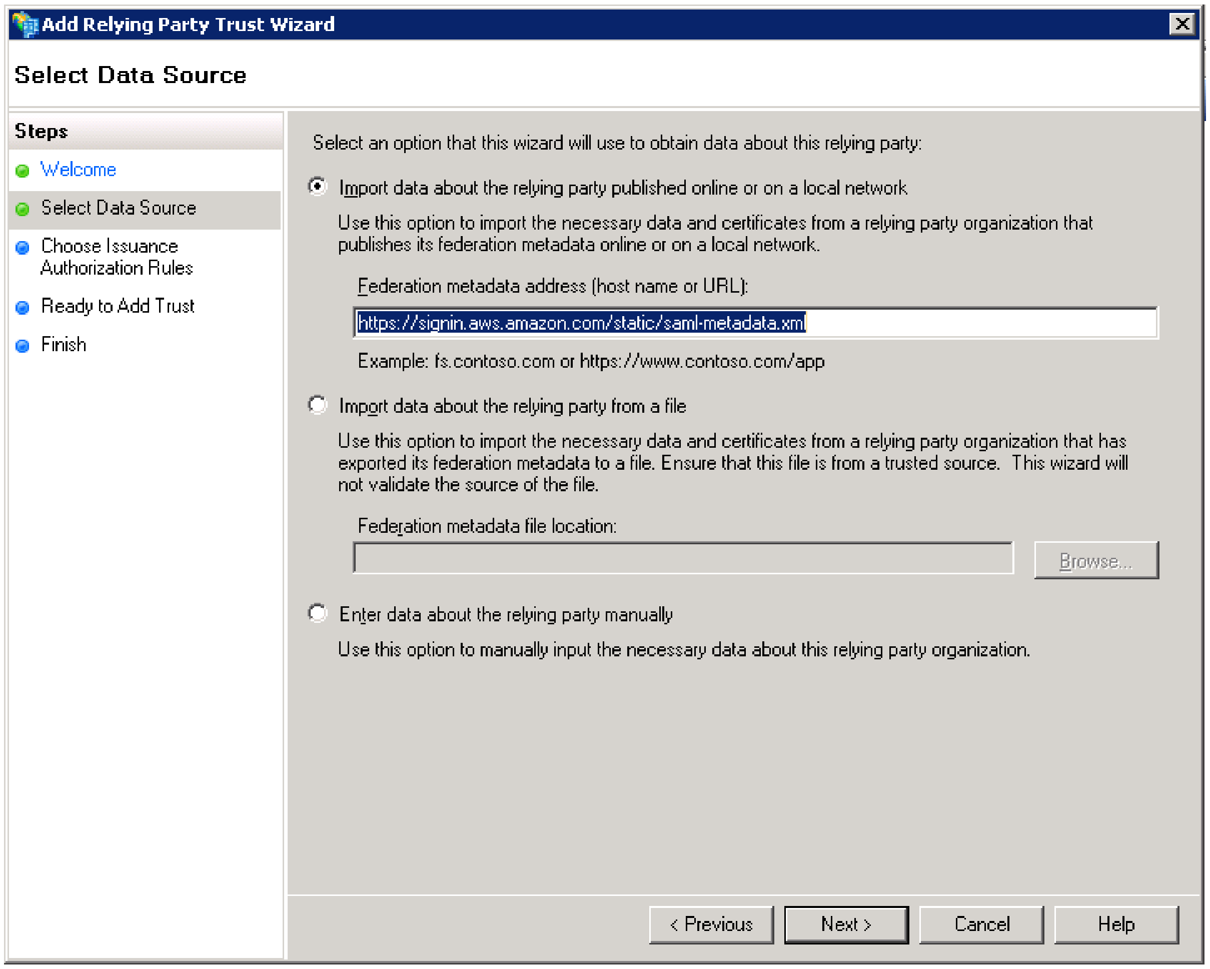
Federation using SAML requires setting up two-way trust. In the preceding section I created a SAML provider and some IAM roles. This is one half of the trust relationship, where the ADFS server is trusted as an identity provider. Similarly, ADFS has to be configured to trust AWS as a relying party. I configured this by returning to the AD FS Management Console. To recreate my setup, perform the following:

1. From the ADFS Management Console, right-click **ADFS 2.0** and select **Add Relying Party Trust**.

2. In the **Add Relying Party Trust Wizard**, click **Start**.

3. Check **Import data about the relying party published online or on a local network**, type https://signin.aws.amazon.com/static/saml-metadata.xml, and then click **Next**. The metadata XML file is a

standard SAML metadata document that describes AWS as a relying party.


Add Relying Party Trust Wizard

Select Data Source

Select an option that this wizard will use to obtain data about this relying party:

(●) Import data about the relying party published online or on a local network

Use this option to import the necessary data and certificates from a relying party organization that publishes its federation metadata online or on a local network.

Federation metadata address (host name or URL):

https://signin.aws.amazon.com/static/saml-metadata.xml

Example: fs.contoso.com or https://www.contoso.com/app

( ) Import data about the relying party from a file

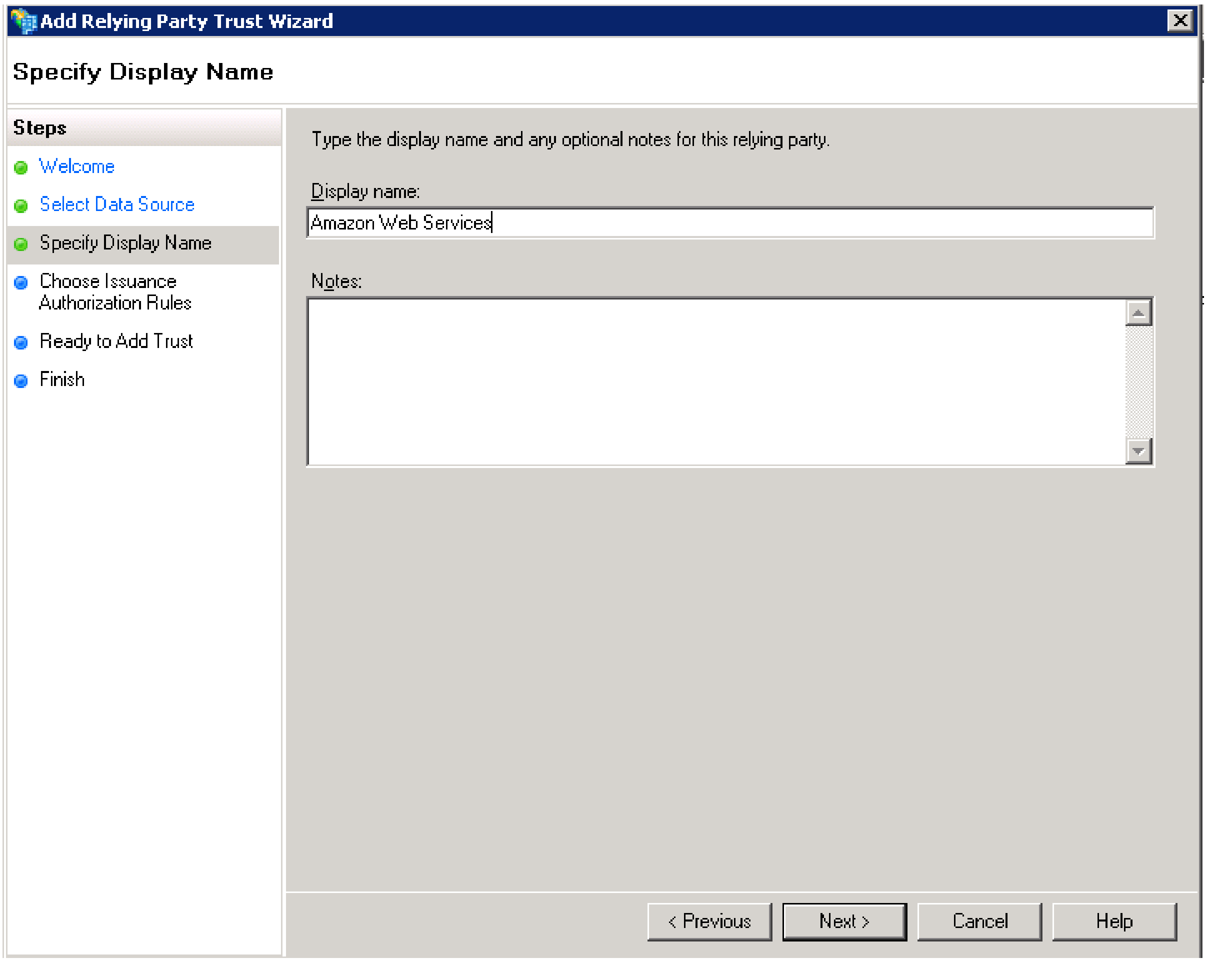
Use this option to import the necessary data and certificates from a relying party organization that has exported its federation metadata to a file. Ensure that this file is from a trusted source. This wizard will not validate the source of the file.

Federation metadata file location:

( ) Enter data about the relying party manually

Use this option to manually input the necessary data about this relying party organization.

4. Set the display name for the relying party and then click **Next**.

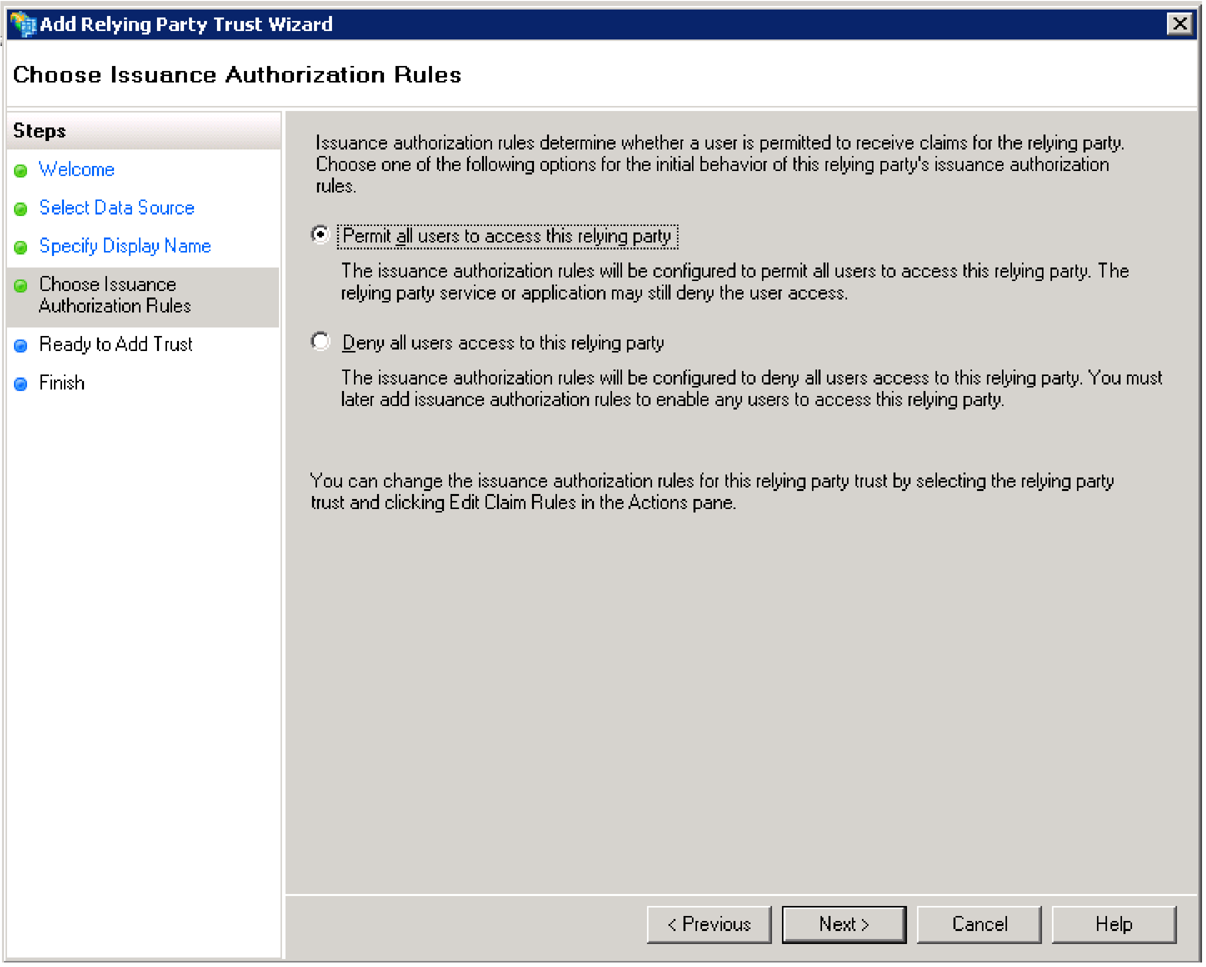


Type the display name and any optional notes for this relying party.

Display name: Amazon Web Services

Notes:

5. Choose your authorization rules. For my scenario, I chose **Permit all users to access this relying party**. When you're done, click **Next**.


Add Relying Party Trust Wizard
Choose Issuance Authorization Rules
Steps
Welcome
Select Data Source
Specify Display Name
Choose Issuance Authorization Rules
Ready to Add Trust
Finish
< Previous
Next >
Cancel
Help

Issuance authorization rules determine whether a user is permitted to receive claims for the relying party. Choose one of the following options for the initial behavior of this relying party's issuance authorization rules.

[✓ Permit all users to access this relying party]

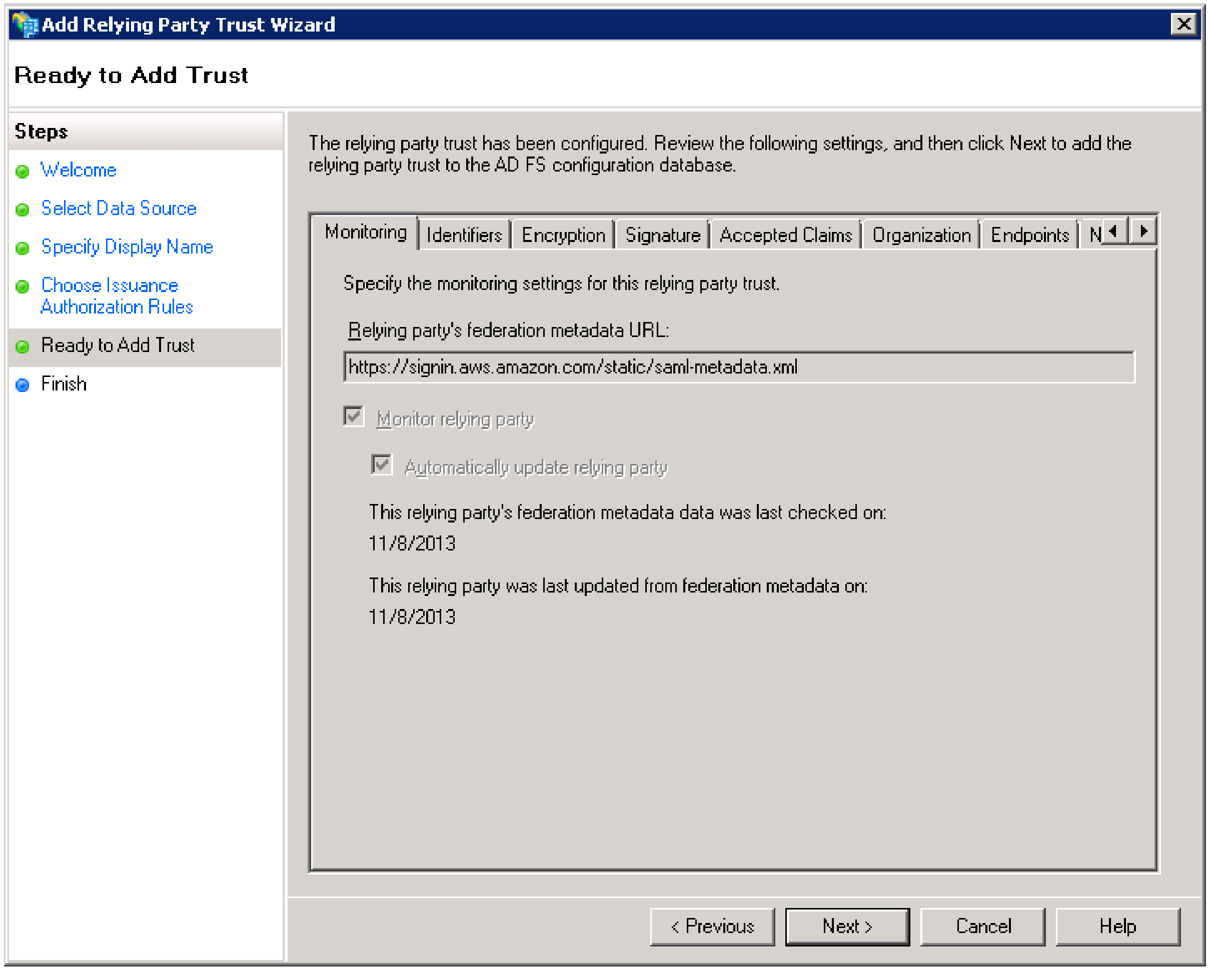
The issuance authorization rules will be configured to permit all users to access this relying party. The relying party service or application may still deny the user access.

Deny all users access to this relying party

The issuance authorization rules will be configured to deny all users access to this relying party. You must later add issuance authorization rules to enable any users to access this relying party.

You can change the issuance authorization rules for this relying party trust by selecting the relying party trust and clicking Edit Claim Rules in the Actions pane.

6. Review your settings and then click **Next**.


Add Relying Party Trust Wizard
Ready to Add Trust
Steps
Welcome
Select Data Source
Specify Display Name
Choose Issuance Authorization Rules
Ready to Add Trust
Finish

The relying party trust has been configured. Review the following settings, and then click Next to add the relying party trust to the AD FS configuration database.

Monitoring | Identifiers | Encryption | Signature | Accepted Claims | Organization | Endpoints | N

Specify the monitoring settings for this relying party trust.

Relying party's federation metadata URL:

https://signin.aws.amazon.com/static/saml-metadata.xml

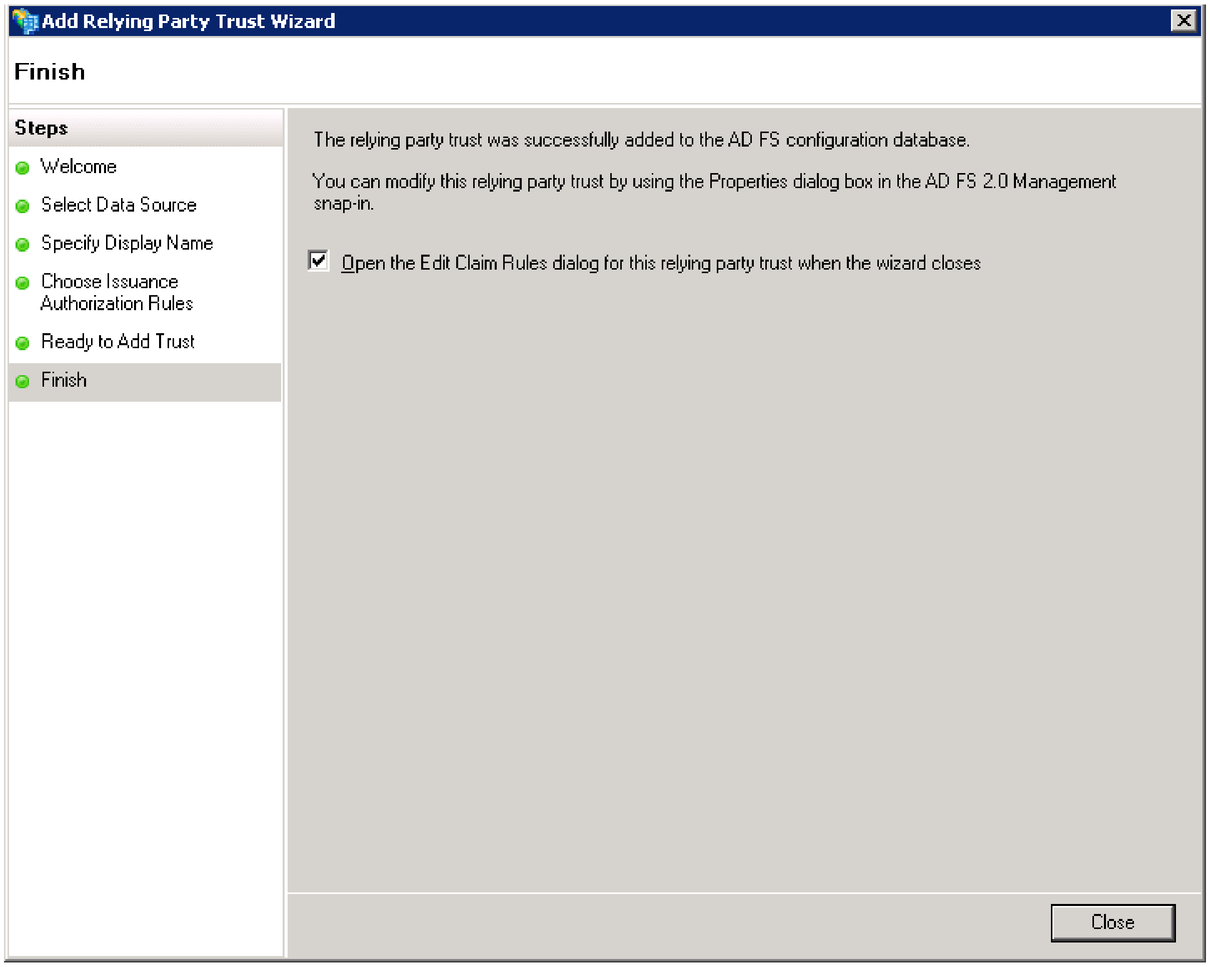
☑ Monitor relying party

☑ Automatically update relying party

This relying party's federation metadata data was last checked on:
11/8/2013

This relying party was last updated from federation metadata on:
11/8/2013

7. Check **Open the Edit Claim Rules dialog for this relying part trust when the wizard closes** and then click **Close**.



The relying party trust was successfully added to the AD FS configuration database.

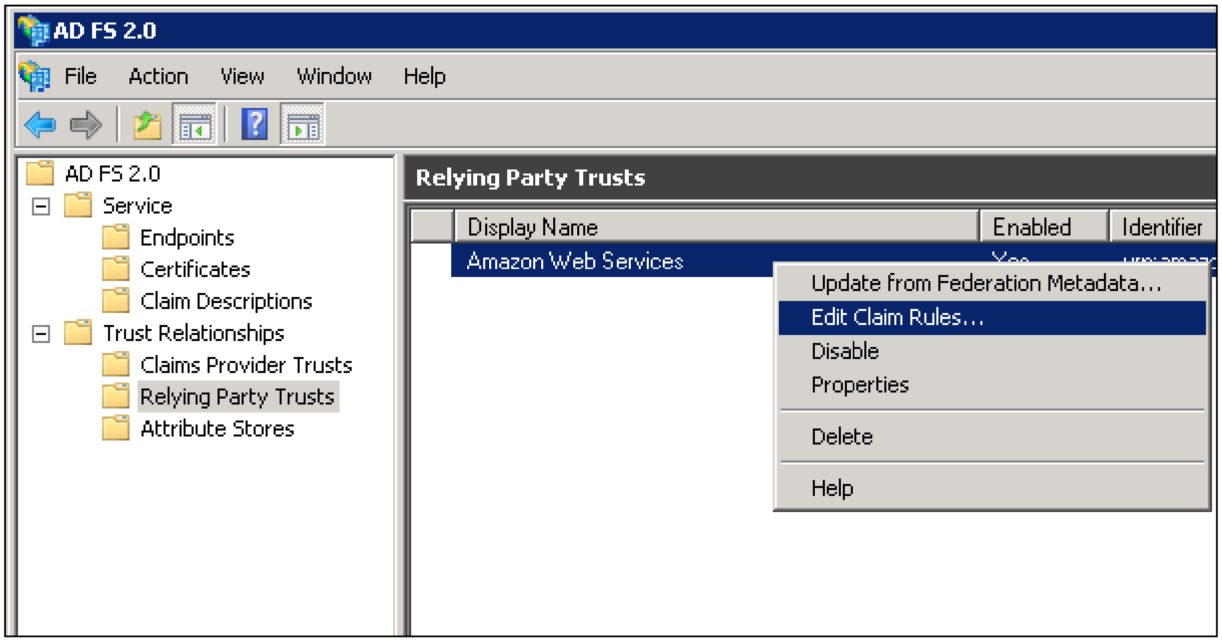
You can modify this relying party trust by using the Properties dialog box in the AD FS 2.0 Management snap-in.

[✓] Open the Edit Claim Rules dialog for this relying party trust when the wizard closes

You’re done configuring AWS as a relying party.

Configuring Claim Rules for the AWS Relying Party

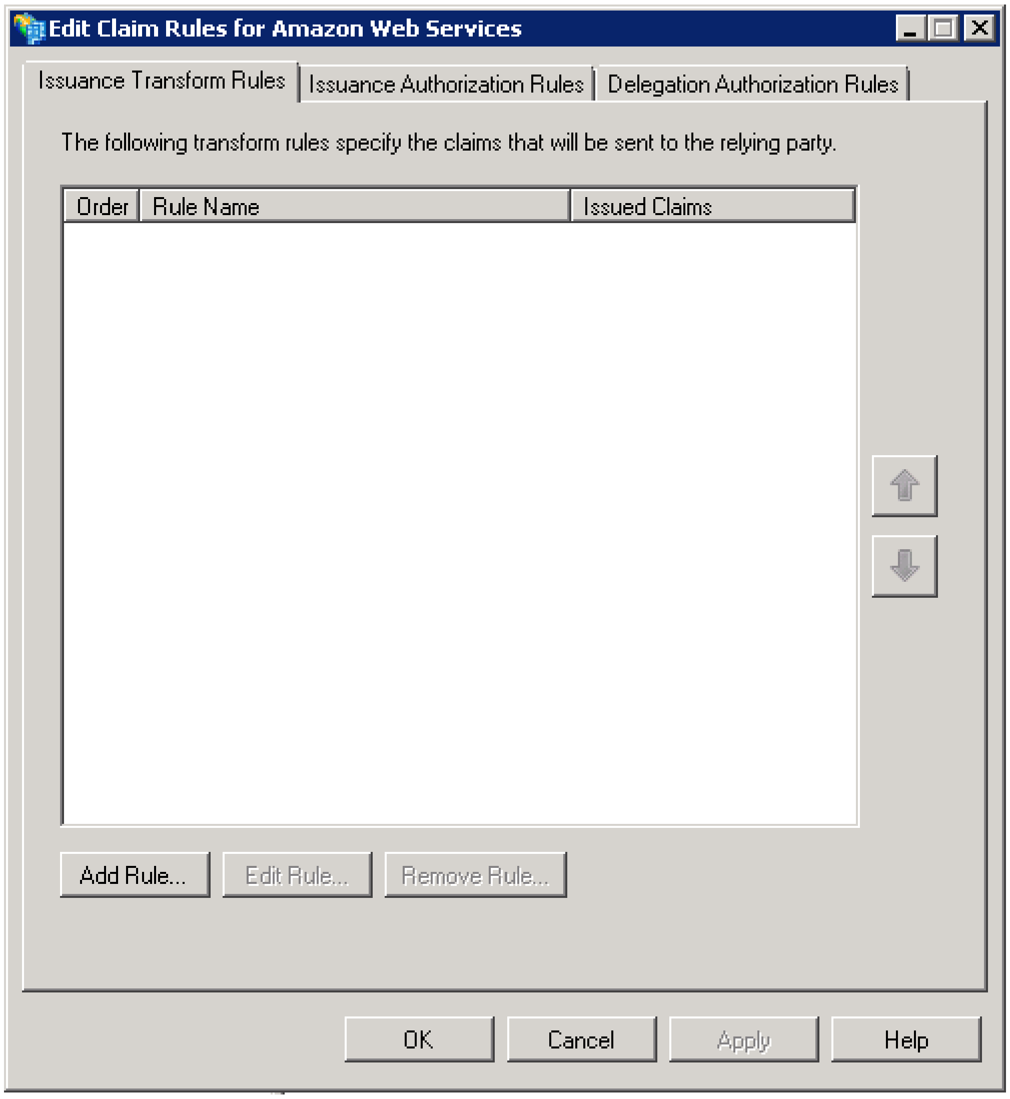
In these steps we’re going to add the claim rules so that the elements AWS requires and ADFS doesn’t provide by default (NameId, RoleSessionName, and Roles) are added to the SAML authentication response. If you forgot to check the box to launch the claim rule dialog, right-click on the relying party (in this case Amazon Web Services) and then click **Edit Claim Rules**.



Here are the steps I used to create the claim rules for NameId, RoleSessionName, and Roles.

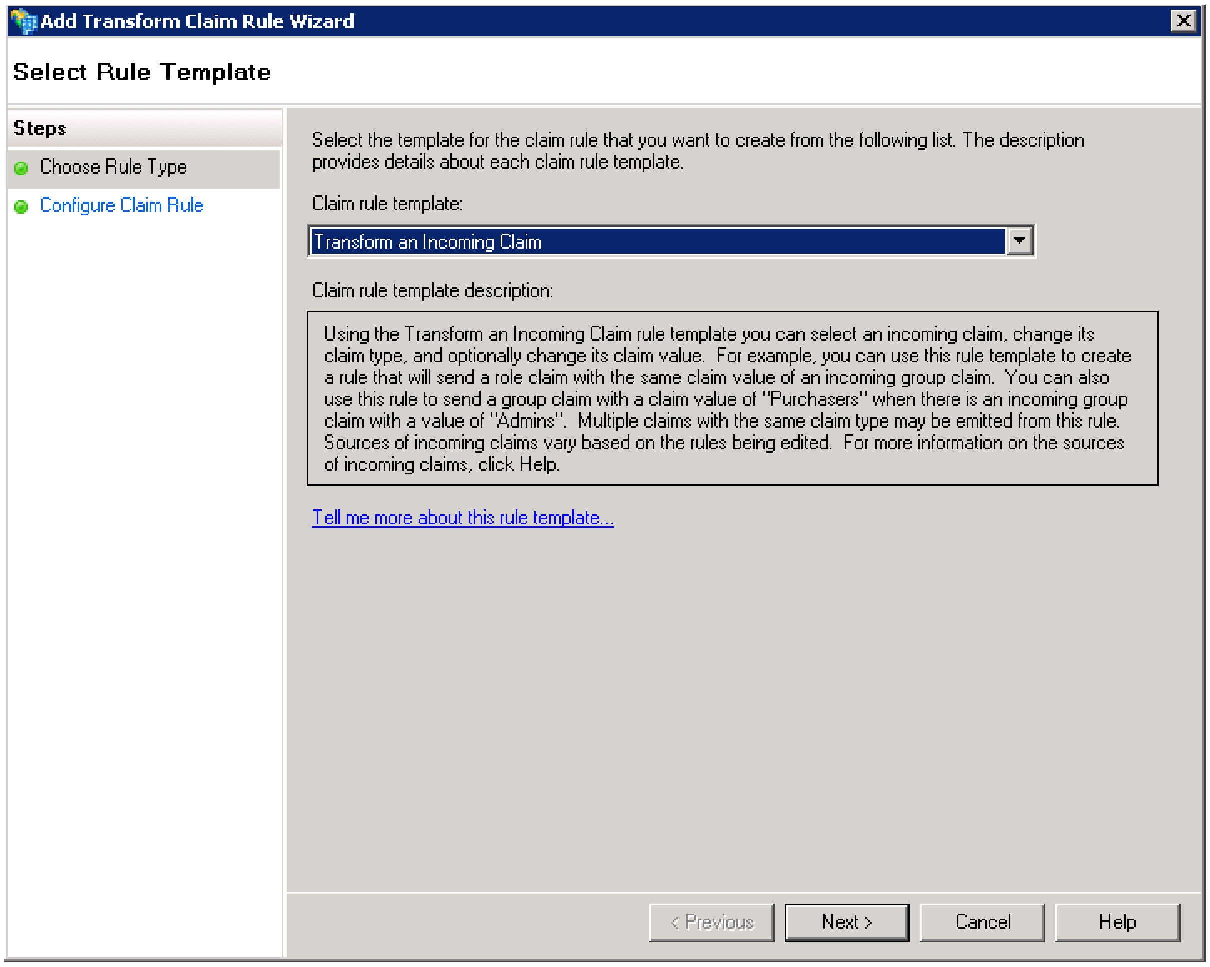
Adding NameId

1. In the **Edit Claim Rules for <relying party>** dialog box, click **Add Rule**.



The following transform rules specify the claims that will be sent to the relying party.

2. Select **Transform an Incoming Claim** and then click **Next**.



Select the template for the claim rule that you want to create from the following list. The description provides details about each claim rule template.

Claim rule template:

Transform an Incoming Claim

Claim rule template description:

Using the Transform an Incoming Claim rule template you can select an incoming claim, change its claim type, and optionally change its claim value. For example, you can use this rule template to create a rule that will send a role claim with the same claim value of an incoming group claim. You can also use this rule to send a group claim with a claim value of "Purchasers" when there is an incoming group claim with a value of "Admins". Multiple claims with the same claim type may be emitted from this rule. Sources of incoming claims vary based on the rules being edited. For more information on the sources of incoming claims, click Help.

Tell me more about this rule template...

3. Use the following settings:

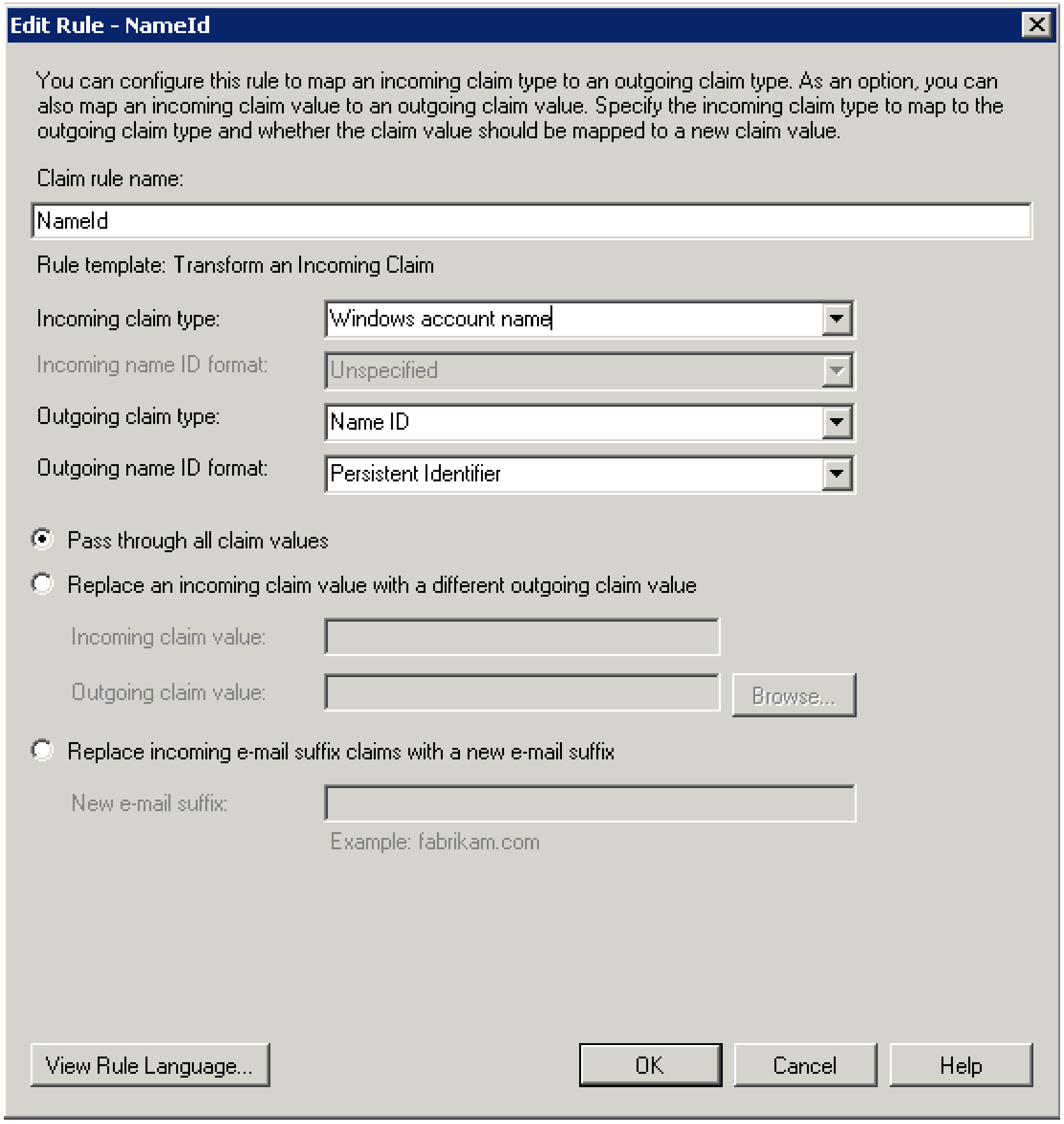
a. **Claim rule name**: NameId

b. **Incoming claim type**: Windows Account Name

c. **Outgoing claim type**: Name ID

d. **Outgoing name ID format**: Persistent Identifier

e. **Pass through all claim values:** checked


Edit Rule - NameId

You can configure this rule to map an incoming claim type to an outgoing claim type. As an option, you can also map an incoming claim value to an outgoing claim value. Specify the incoming claim type to map to the outgoing claim type and whether the claim value should be mapped to a new claim value.

Claim rule name: NameId

Rule template: Transform an Incoming Claim

Incoming claim type: Windows account name

Incoming name ID format: Unspecified

Outgoing claim type: Name ID

Outgoing name ID format: Persistent Identifier

[✓ Pass through all claim values]

Replace an incoming claim value with a different outgoing claim value

Incoming claim value:

Outgoing claim value:

Replace incoming e-mail suffix claims with a new e-mail suffix

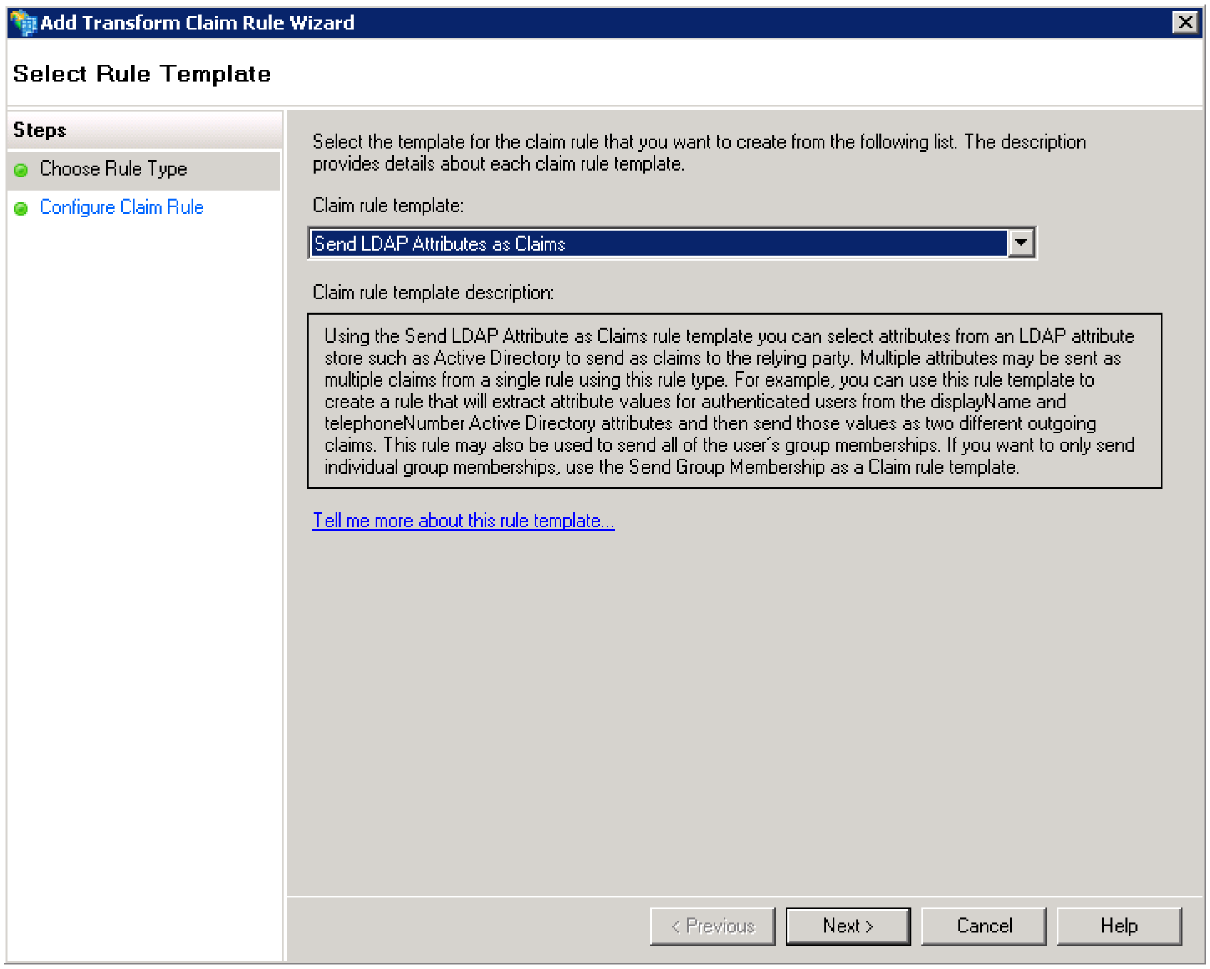
New e-mail suffix:

Example: fabrikam.com

4. Click **Finish.**

Adding a RoleSessionName

1. Click **Add Rule**
2. In the **Claim rule template** list, select **Send LDAP Attributes as Claims**.


Add Transform Claim Rule Wizard

Select Rule Template

Select the template for the claim rule that you want to create from the following list. The description provides details about each claim rule template.

Claim rule template:

Send LDAP Attributes as Claims

Claim rule template description:

Using the Send LDAP Attribute as Claims rule template you can select attributes from an LDAP attribute store such as Active Directory to send as claims to the relying party. Multiple attributes may be sent as multiple claims from a single rule using this rule type. For example, you can use this rule template to create a rule that will extract attribute values for authenticated users from the displayName and telephoneNumber Active Directory attributes and then send those values as two different outgoing claims. This rule may also be used to send all of the user's group memberships. If you want to only send individual group memberships, use the Send Group Membership as a Claim rule template.

Tell me more about this rule template...

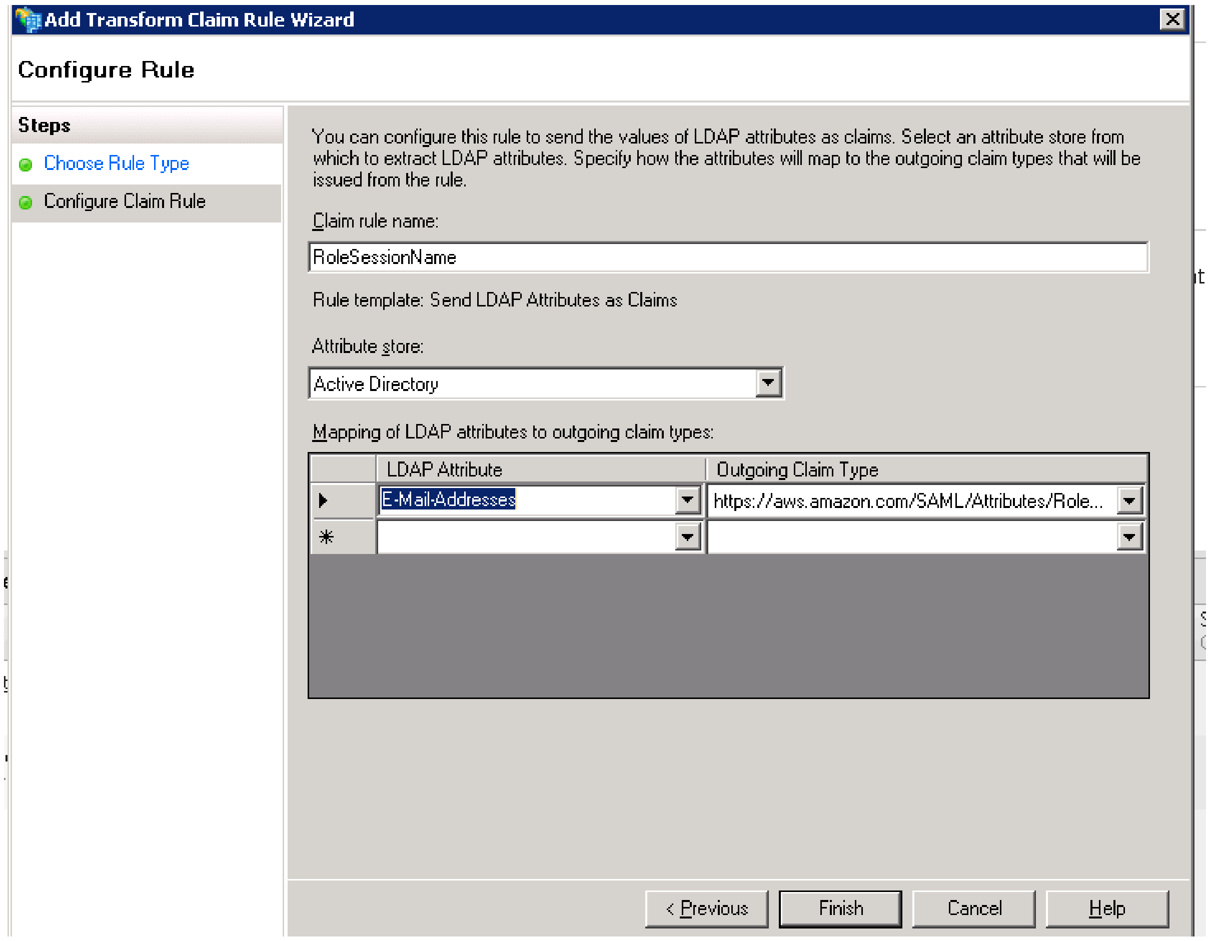
3. Use the following settings:

a. **Claim rule name**: RoleSessionName

b. **Attribute store**: Active Directory

c. **LDAP Attribute**: E-Mail-Addresses

d. **Outgoing Claim Type** : https://aws.amazon.com/SAML/Attributes/RoleSessionName



Add Transform Claim Rule Wizard

Configure Rule

You can configure this rule to send the values of LDAP attributes as claims. Select an attribute store from which to extract LDAP attributes. Specify how the attributes will map to the outgoing claim types that will be issued from the rule.

Claim rule name: RoleSessionName

Rule template: Send LDAP Attributes as Claims

Attribute store: Active Directory

Mapping of LDAP attributes to outgoing claim types:

LDAP Attribute	Outgoing Claim Type
E-Mail-Addresses	https://aws.amazon.com/SAML/Attributes/Role...

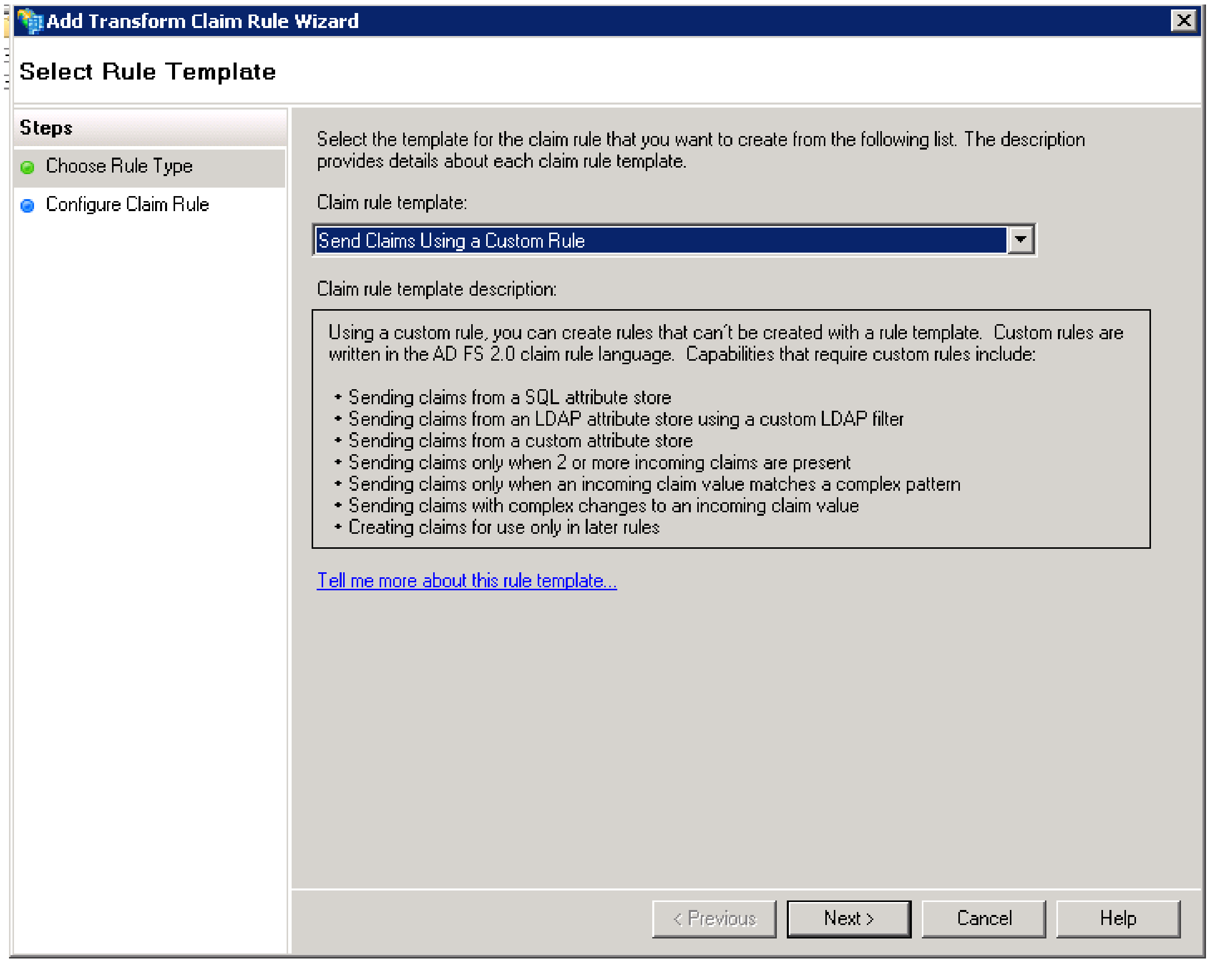
4. Click **Finish**

Adding Role Attributes

I'll pause here to provide a little more context because for these steps it might not be as obvious what's going on. Unlike the two previous claims, here I used custom rules to send role attributes. This is done by retrieving all the authenticated user's AD groups and then matching the groups that start with to IAM roles of a similar name. I used the names of these groups to create Amazon Resource Names (ARNs) of IAM roles in my AWS account (i.e., those that start with `AWS-`).

Sending role attributes required two custom rules. The first rule retrieves all the authenticated user's AD group memberships and the second rule performs the transformation to the roles claim. Here's how I did it.

1. Click **Add Rule**.
2. In the **Claim rule template** list, select **Send Claims Using a Custom Rule** and then click **Next**.



Select the template for the claim rule that you want to create from the following list. The description provides details about each claim rule template.

Claim rule template:

Send Claims Using a Custom Rule

Claim rule template description:

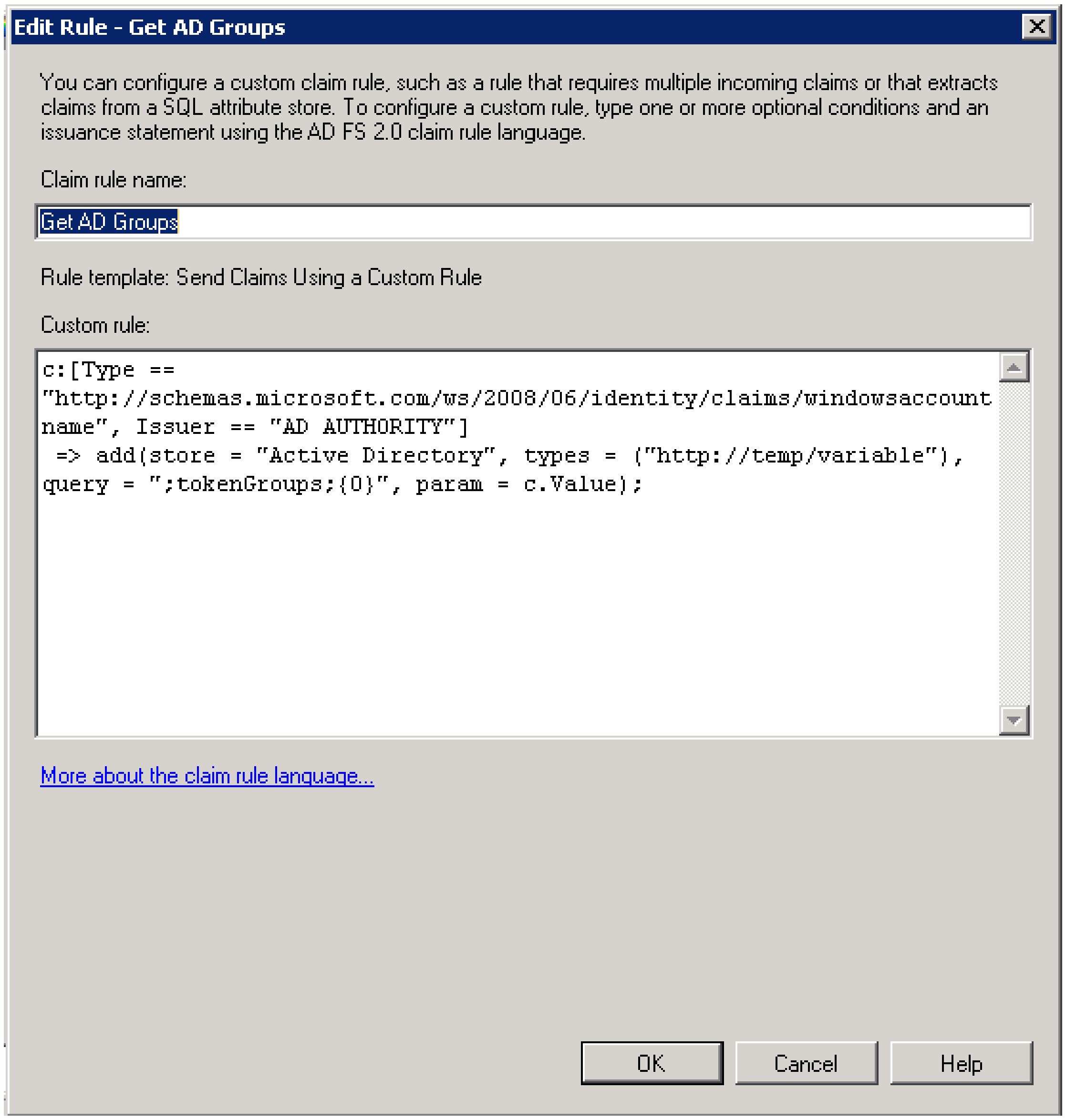
Using a custom rule, you can create rules that can't be created with a rule template. Custom rules are written in the AD FS 2.0 claim rule language. Capabilities that require custom rules include:

- Sending claims from a SQL attribute store
- Sending claims from an LDAP attribute store using a custom LDAP filter
- Sending claims from a custom attribute store
- Sending claims only when 2 or more incoming claims are present
- Sending claims only when an incoming claim value matches a complex pattern
- Sending claims with complex changes to an incoming claim value
- Creating claims for use only in later rules

Tell me more about this rule template...

3. For **Claim Rule Name**, select **Get AD Groups**, and then in **Custom rule**, enter the following:

```
c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname"
```


Edit Rule - Get AD Groups

You can configure a custom claim rule, such as a rule that requires multiple incoming claims or that extracts claims from a SQL attribute store. To configure a custom rule, type one or more optional conditions and an issuance statement using the AD FS 2.0 claim rule language.

Claim rule name: Get AD Groups

Rule template: Send Claims Using a Custom Rule

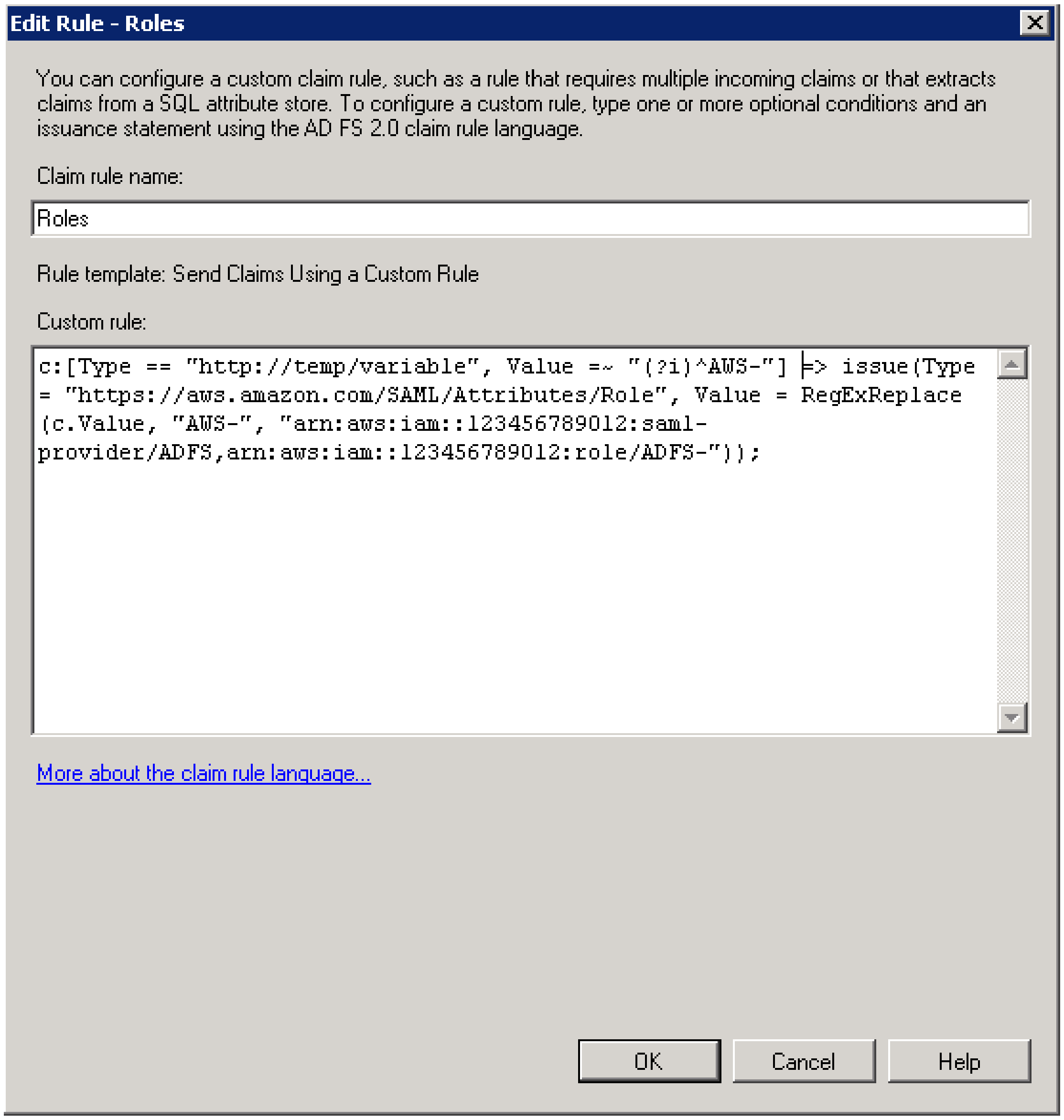
Custom rule:

```
c:[Type ==
"http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccount
name", Issuer == "AD AUTHORITY"]
 => add(store = "Active Directory", types = ("http://temp/variable"),
query = ";tokenGroups;{0}", param = c.Value);
```

This custom rule uses a script in the claim rule language that retrieves all the groups the authenticated user is a member of and places them into a temporary claim named `http://temp/variable`. (Think of this as a variable you can access later.) I use this in the next rule to transform the groups into IAM role ARNs.

4. Click **OK**.
5. Click **Add Rule.**
6. Repeat the preceding steps, but this time, type **Roles** for **Claim rule name** and use the following script:

```
c:[Type == "http://temp/variable", Value =~ "(?i)^AWS-"] => issue(Type = "https://aws.a
```



You can configure a custom claim rule, such as a rule that requires multiple incoming claims or that extracts claims from a SQL attribute store. To configure a custom rule, type one or more optional conditions and an issuance statement using the AD FS 2.0 claim rule language.

Claim rule name: Roles

Rule template: Send Claims Using a Custom Rule

Custom rule:

```
c:[Type == "http://temp/variable", Value =~ "(?i)^AWS-"] => issue(Type
= "https://aws.amazon.com/SAML/Attributes/Role", Value = RegExReplace
(c.Value, "AWS-", "arn:aws:iam::123456789012:saml-
provider/ADFS,arn:aws:iam::123456789012:role/ADFS-"));
```

This rule uses a custom script to get all the groups from the temporary claim () and then uses the name of the group to create the principal/role pair, which has this format:

ARN of SAML provider,ARN of role to assume

In my example, it comes out this way:

arn:aws:iam:123456789012:saml-provider/ADFS,arn:aws:iam:123456789012:role/ADFS-

In the example, I used an account number of 123456789012. Make sure you change this to your own AWS account.

7. Click **OK.**

Testing the configuration

Setup is complete. If you're using any browser except Chrome, you're ready to test—skip ahead to the testing steps.

If you're using Chrome as your browser, you need to configure the browser to work with AD FS. The default AD FS site uses a feature called Extended Protection that by default isn't compatible with Chrome. However, it's easy to turn off extended protection for the **ADFS->LS website:**

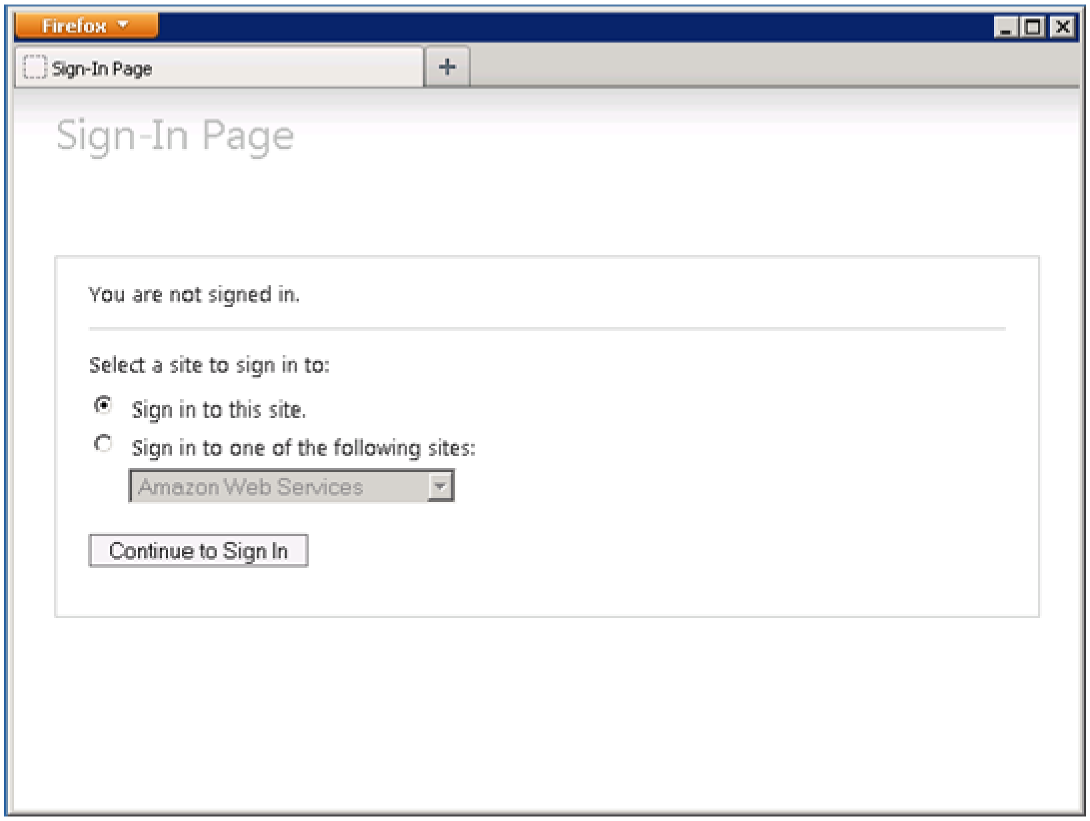
1. In Windows Server, select **Start** > **Administrative Tools** > **IIS Manager**.
2. Expand: **<server-name>, Sites, Default Web Site,** and **adfs.**
3. Select the **ls** application and double-click **Authentication**.
4. Select **Windows Authentication** and select **Advanced Settings**.
5. Set **Extended Protection** to **Off** and then click **OK**.
6. Restart ADFS and IIS by running the following as an administrator at the command line:
    a. IISReset
    b. Net Stop "AD FS 2.0 Windows Service"
    c. Net Start "AD FS 2.0 Windows Service"

Testing steps

1. In your domain, browse to the following address: https://localhost/adfs/ls/IdpInitiatedSignOn.aspx

If you're using a locally signed certificate from IIS, you might get a certificate warning.

2. Select **Sign in to one of the following sites**, select **Amazon Web Services** from the list, and then click **Continue to Sign In**.



Sign-In Page

You are not signed in.

Select a site to sign in to:

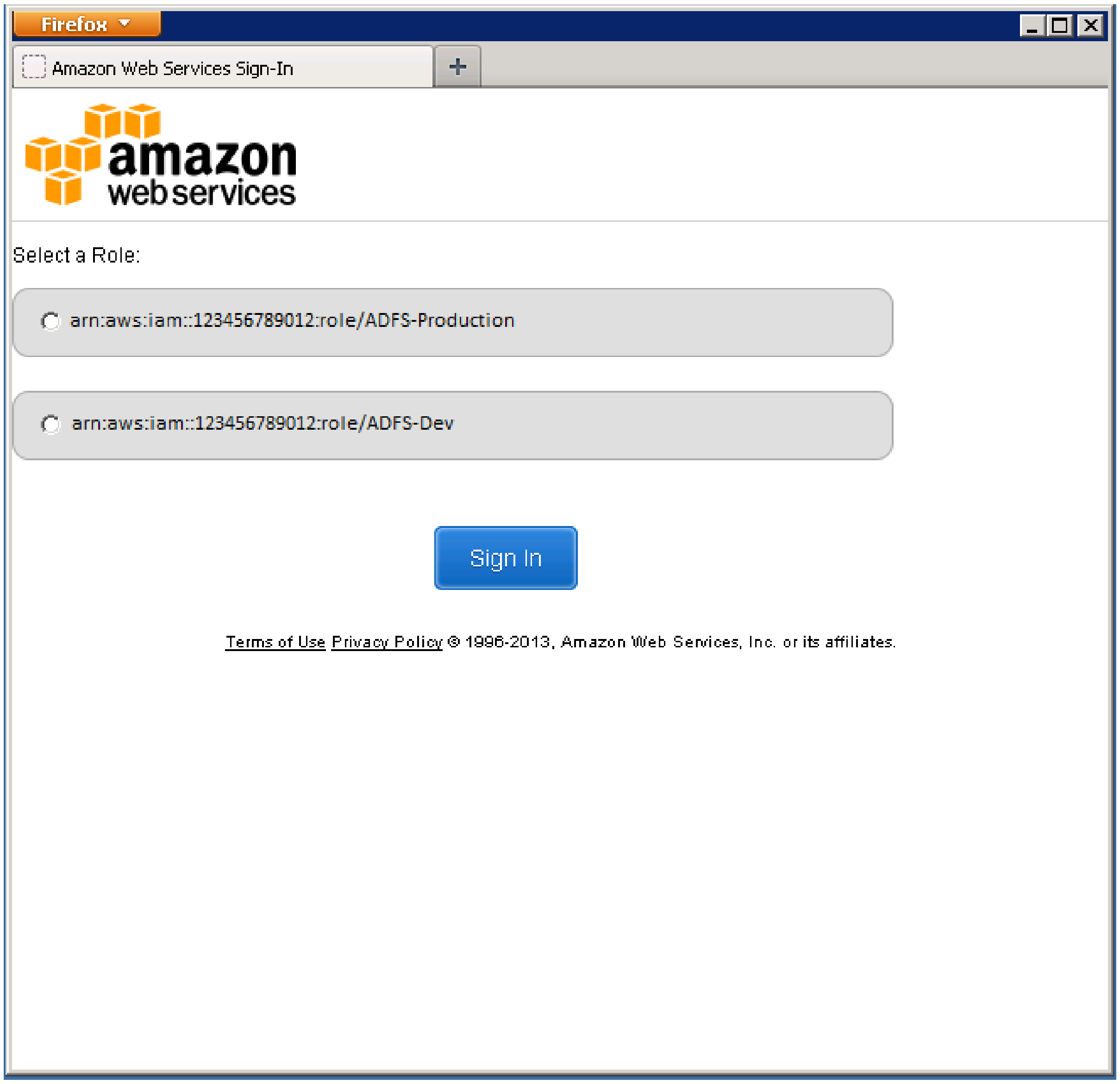
● Sign in to this site.

○ Sign in to one of the following sites:

Amazon Web Services

3. If prompted, enter in a username and password (remember to use Bob's account). You are redirected to the **Amazon Web Services Sign-In** page.

4. Select a role and then click **Sign In**. (If you are mapped to only a single IAM role, you skip the role selection step and are automatically signed into the AWS Management Console.)

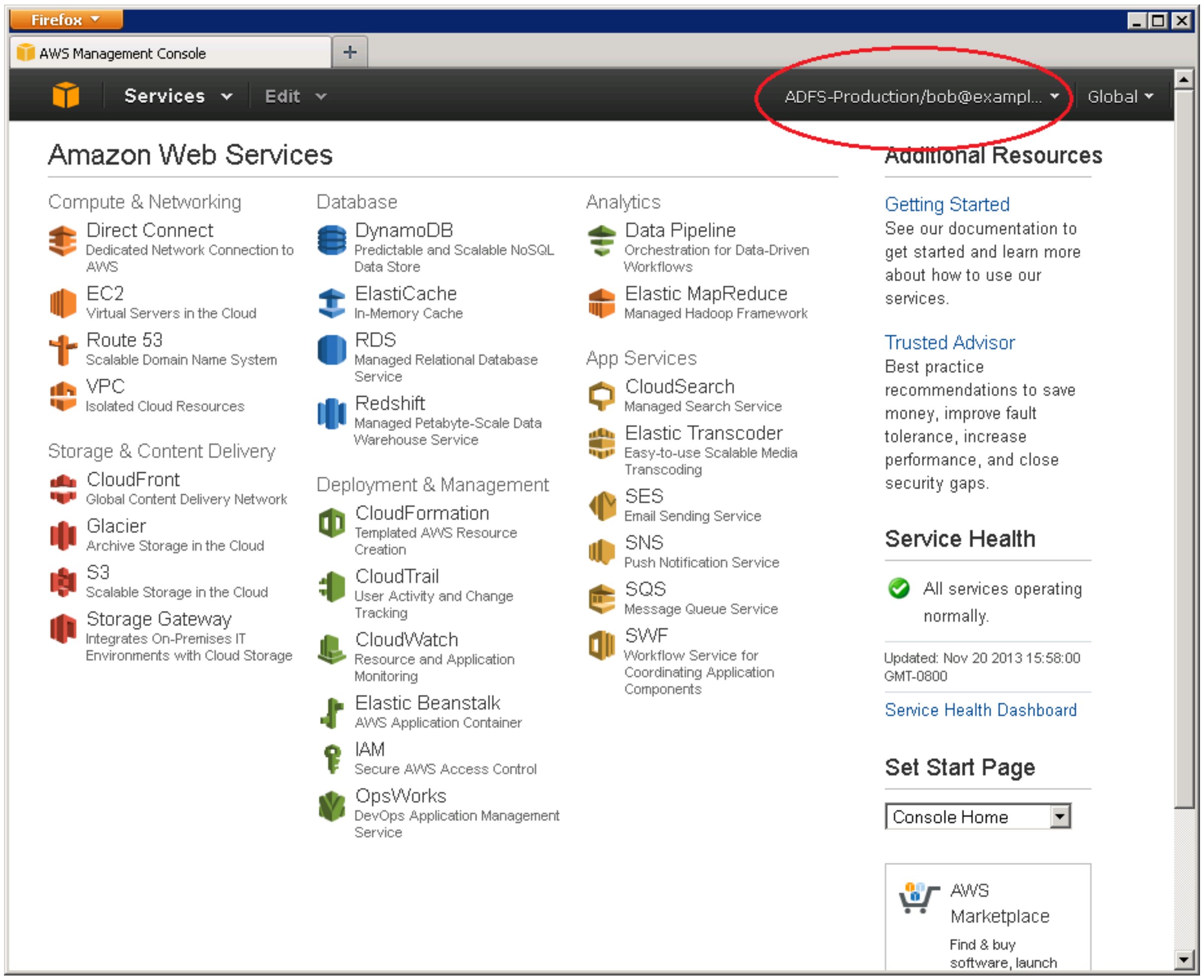

Firefox
Amazon Web Services Sign-In
amazon web services
Sign In
Terms of Use Privacy Policy © 1996-2013, Amazon Web Services, Inc. or its affiliates.

Select a Role:

arn:aws:iam::123456789012:role/ADFS-Production

arn:aws:iam::123456789012:role/ADFS-Dev

You see Bob has signed in.



Amazon Web Services

Compute & Networking
Direct Connect — Dedicated Network Connection to AWS
EC2 — Virtual Servers in the Cloud
Route 53 — Scalable Domain Name System
VPC — Isolated Cloud Resources

Storage & Content Delivery
CloudFront — Global Content Delivery Network
Glacier — Archive Storage in the Cloud
S3 — Scalable Storage in the Cloud
Storage Gateway — Integrates On-Premises IT Environments with Cloud Storage

Database
DynamoDB — Predictable and Scalable NoSQL Data Store
ElastiCache — In-Memory Cache
RDS — Managed Relational Database Service
Redshift — Managed Petabyte-Scale Data Warehouse Service

Deployment & Management
CloudFormation — Templated AWS Resource Creation
CloudTrail — User Activity and Change Tracking
CloudWatch — Resource and Application Monitoring
Elastic Beanstalk — AWS Application Container
IAM — Secure AWS Access Control
OpsWorks — DevOps Application Management Service

Analytics
Data Pipeline — Orchestration for Data-Driven Workflows
Elastic MapReduce — Managed Hadoop Framework

App Services
CloudSearch — Managed Search Service
Elastic Transcoder — Easy-to-use Scalable Media Transcoding
SES — Email Sending Service
SNS — Push Notification Service
SQS — Message Queue Service
SWF — Workflow Service for Coordinating Application Components

Additional Resources

Getting Started
See our documentation to get started and learn more about how to use our services.

Trusted Advisor
Best practice recommendations to save money, improve fault tolerance, increase performance, and close security gaps.

Service Health
All services operating normally.
Updated: Nov 20 2013 15:58:00 GMT-0800
Service Health Dashboard

Set Start Page
Console Home

AWS Marketplace
Find & buy software, launch

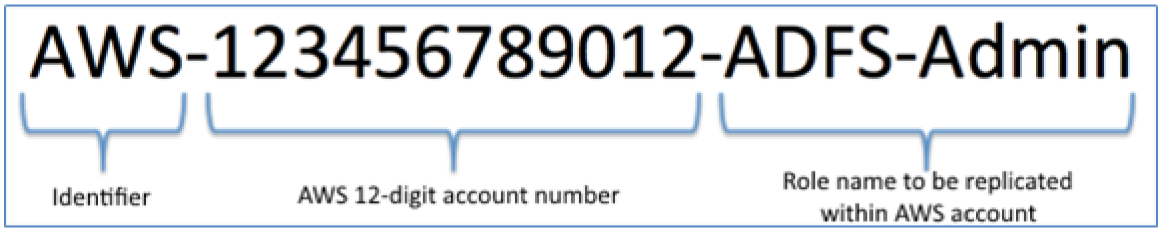
I'm interested in hearing your feedback on this. Feel free to post comments below or start a thread in the Identity and Access Management forum.

-Jeff

Addendum:

Ever since I published this blog post, some readers have asked how to configure the AD FS claims using multiple AWS accounts. Those of you with multiple AWS accounts can leverage AD FS and SSO without adding claim rules for each account. Though there may be other ways to do this, one approach recommended by AWS Senior Solutions Architect Jamie Butler is to use Regex and a common Active Directory security group naming convention. Then, AD FS can provide cross-account authentication for an entire enterprise. Jamie's solution follows.

When using this approach, your security group naming convention must start with an identifier (for example, `AWS-`). This will distinguish your AWS groups from others within the organization. Next, include the 12-digit AWS account number. Finally, add the matching role name within the AWS account. Here is an example.

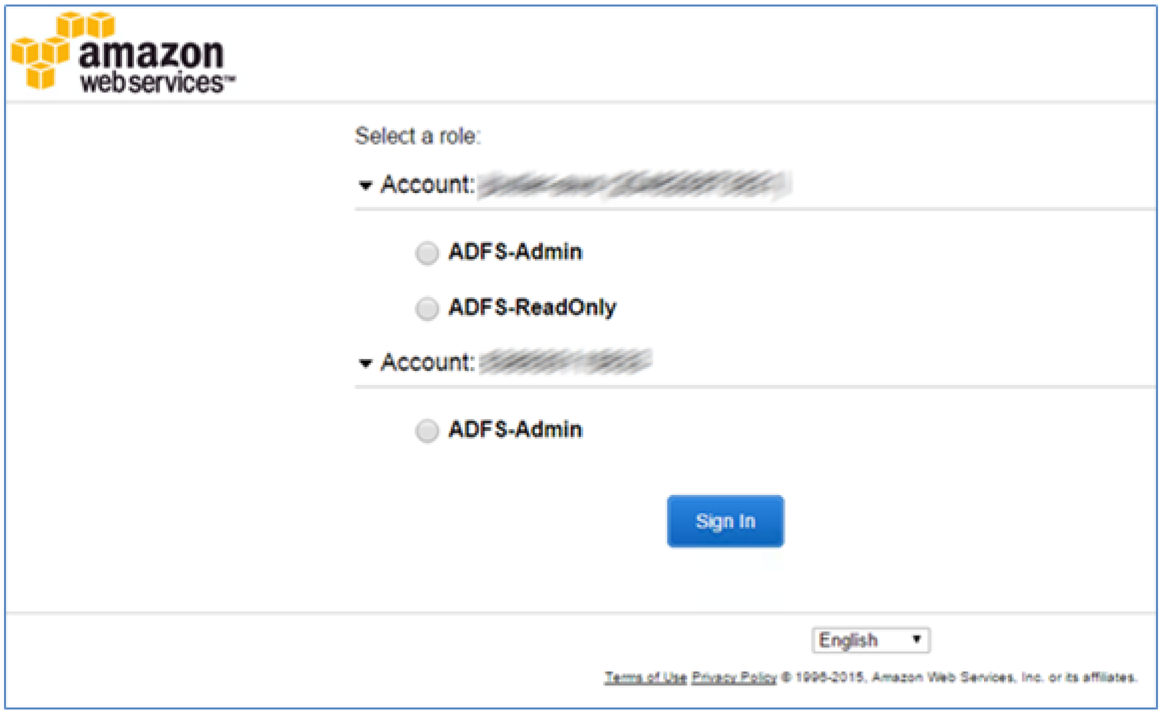

AWS-123456789012-ADFS-Admin
Identifier
AWS 12-digit account number
Role name to be replicated within AWS account

Next, update the `Roles` AD FS claim rule that you created earlier, by using the following code.

```
c:[Type == "http://temp/variable", Value =~ "(?i)^AWS-([\d]{12})"] => issue(Type = "ht
```

This new claim rule limits scope to only Active Directory security groups that begin with `AWS–` and any twelve-digit number. The claim rule then constructs the SAML assertion in the proper format using the AWS account number and the role name from the Active Directory group name. All AWS accounts must be configured with the same IdP name (in this case `ADFS`) as described in the "Configuring AWS" section earlier in this post.

Any users with membership in the Active Directory security group will now be able to authenticate to AWS using their Active Directory credentials and assume the matching AWS role. If a user is associated with multiple Active Directory groups and AWS accounts, they will see a list of roles by AWS account and will have the option to choose which role to assume.


Select a role:

▾ Account: [redacted]

○ ADFS-Admin

○ ADFS-ReadOnly

▾ Account: [redacted]

○ ADFS-Admin

Terms of Use Privacy Policy © 1996-2015, Amazon Web Services, Inc. or its affiliates.

Know of a better way? Please add a comment to this post.

** If you would like to implement federated API and CLI access using SAML 2.0 and ADFS, check out this blog post from AWS Senior IT Transformation Consultant Quint Van Deman.

Want more AWS Security how-to content, news, and feature announcements? Follow us on Twitter.

TAGS: Active Directory, Active Directory Federation Services, AD FS Setup, ADFS, Federation, Microsoft AD FS, reInvent, SAML, Security Blog