

The 3 steps to building a sample AWS Management Console federation site with Active Directory Federation Services

December 15th, 2014
AWS Professional Services
Yuki, Chiba

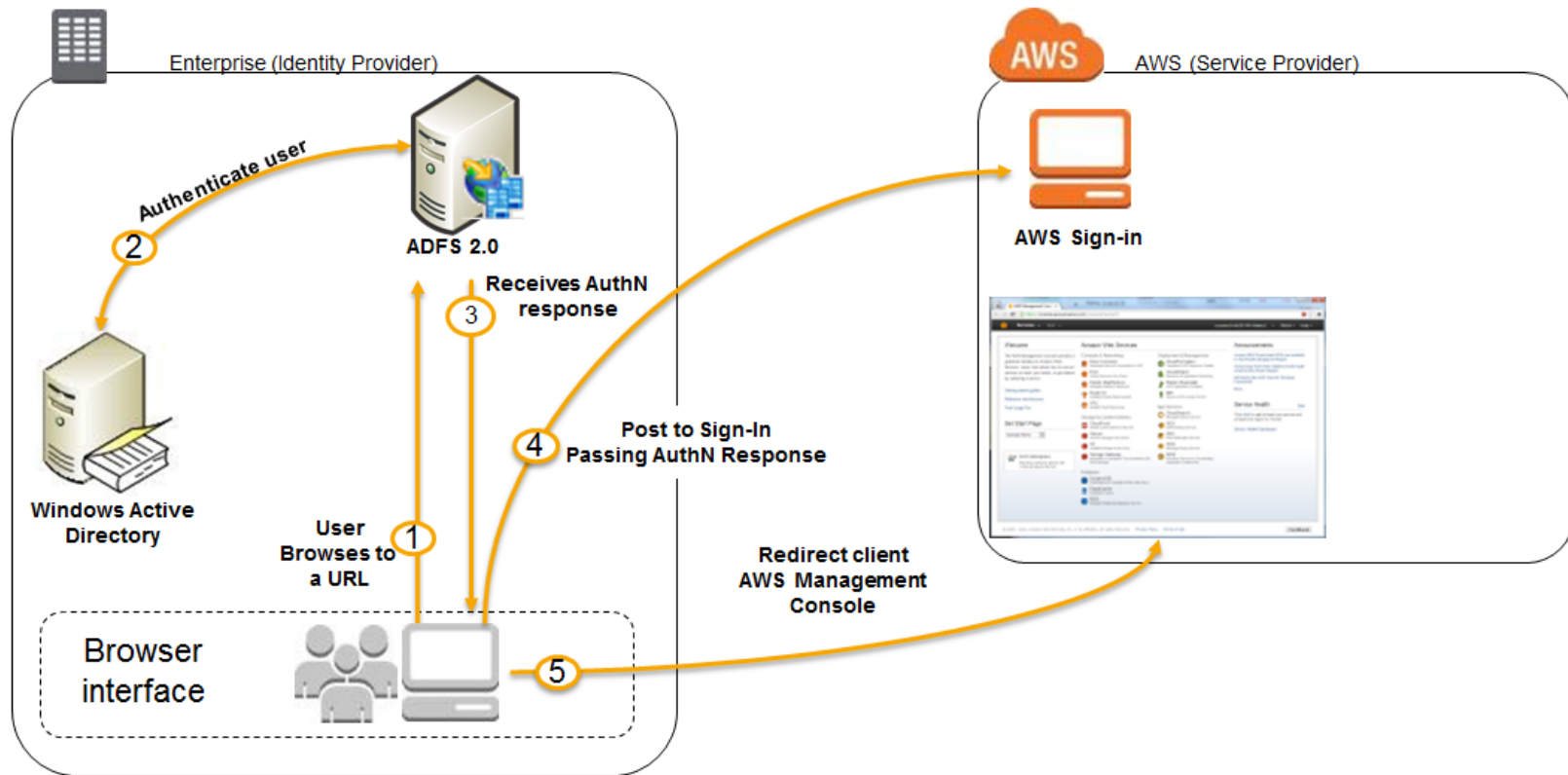
Introduction

- Some enterprise customers prefer leveraging their existing corporate identities to access AWS services and resources via identity federation
- This deck intends to illustrate how to build a sample federation site using [Active Directory Federation Services](#) and [SAML 2.0](#)
- I also provides the powershell script, `Configure-AWSFederation.ps1`, to setup federation with AWS management console
- Basically this procedure is along with [the AWS security blog by Jeff Wierer](#)

How Integration Between ADFS and AWS Works

1. The flow is initiated when a browses to the ADFS sample site inside his domain
2. The sign-on page authenticates the user against AD
3. The browser receives a SAML assertion in the form of an authentication response from ADFS
4. The browser posts the SAML assertion to [the AWS sign-in endpoint for SAML](#)
 - Sign-in uses the [AssumeRoleWithSAML](#) API to request temporary security credentials and then constructs a sign-in URL for the AWS Management Console
5. The browser receives the sign-in URL and is redirected to the console

How Integration Between ADFS and AWS Works



Building steps

- Step1: Build Active Directory Domain
 - Launch an instance
 - Install features and create the Domain forest on the instance
- Step2: Configure ADFS farm and claims on the instance
 - Create certificate and import it to the machine
 - Install ADFS farm
 - Create the relyingPartyTrust and claims
- Step3: Configuring IAM
 - [Create SAML Provider in IAM](#)
 - [Create two IAM Roles for SAML-Based Federation](#)

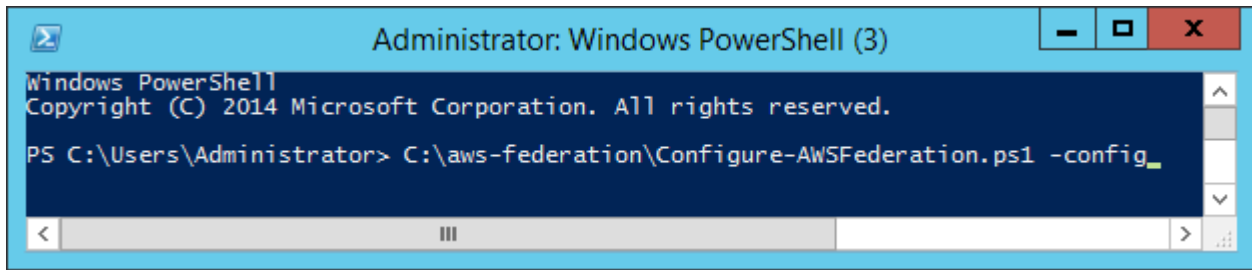
Step1 : Build Active Directory Domain

- Download the Configure-AWSFederation.ps1 script on Windows Server 2012 R2 on EC2
 - Script repository
 - <https://github.com/number13/Configure-AWSFederation>
 - You must allow following access from the client to the instance
 - Inbound : RDP, HTTPS
- Run the script with “-Init” option
- Or, you can use the user-data when launching the instance (The instance will be rebooted automatically once after initialization completed)

```
<powershell>
mkdir C:\aws-federation
$ScriptFile="C:\aws-federation\Configure-AWSFederation.ps1"
Invoke-WebRequest "https://raw.githubusercontent.com/number13/Configure-AWSFederation/master/Configure-AWSFederation.ps1" -
OutFile $ScriptFile
Set-ExecutionPolicy Unrestricted
& $ScriptFile -init
</powershell>
```

Step2 : Configure ADFS farm and craims

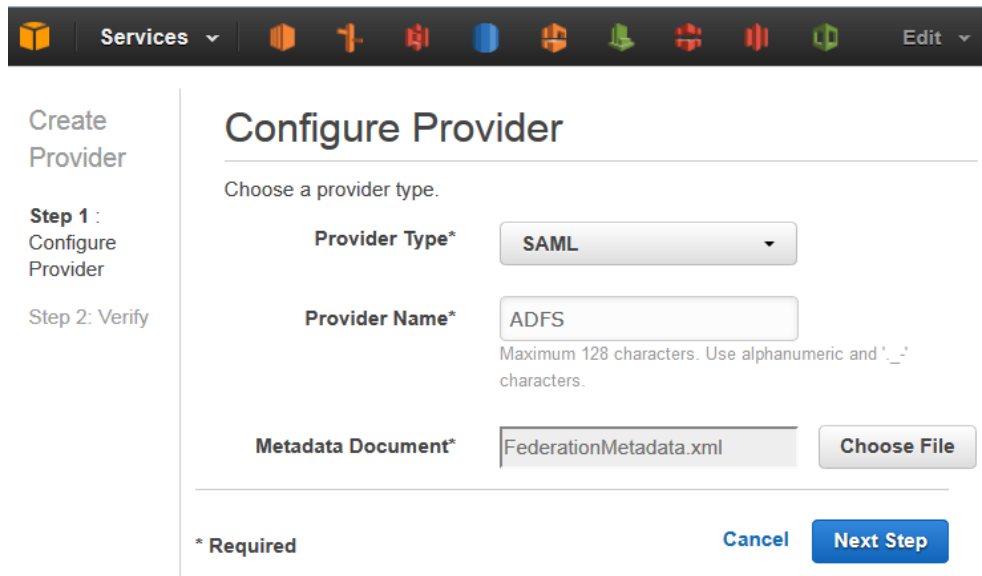
- Log on to the instance as “example\Administrator” through remote desktop
- Run the Configure-AWSFederation.ps1 script with “-Config” option



- Download the federation meta-data xml file
 - Download URL
 - https://<your_instance_public_dns>/FederationMetadata/2007-06/FederationMetadata.xml

Step3 : Configure IAM

- [Log in to the IAM Management console](#)
- [Create SAML Provider in IAM](#)
 - Provider Type : SAML
 - Provider Name : ADFS (must be this name)
 - Metadata Document : Specify the xml file downloaded in setp2



The screenshot shows the AWS IAM 'Configure Provider' console. The left sidebar contains a 'Create Provider' section with 'Step 1 : Configure Provider' selected. The main area is titled 'Configure Provider' and includes the instruction 'Choose a provider type.' Below this, there are three required fields: 'Provider Type*' with a dropdown menu set to 'SAML', 'Provider Name*' with a text box containing 'ADFS' and a note about character limits, and 'Metadata Document*' with a text box containing 'FederationMetadata.xml' and a 'Choose File' button. At the bottom, there are 'Cancel' and 'Next Step' buttons, along with a '* Required' label.

Services ▾

Create Provider

Step 1 : Configure Provider

Step 2: Verify

Configure Provider

Choose a provider type.

Provider Type* SAML ▾

Provider Name* ADFS
Maximum 128 characters. Use alphanumeric and '._-' characters.

Metadata Document* FederationMetadata.xml **Choose File**

* Required Cancel Next Step

Step3 : Configure IAM

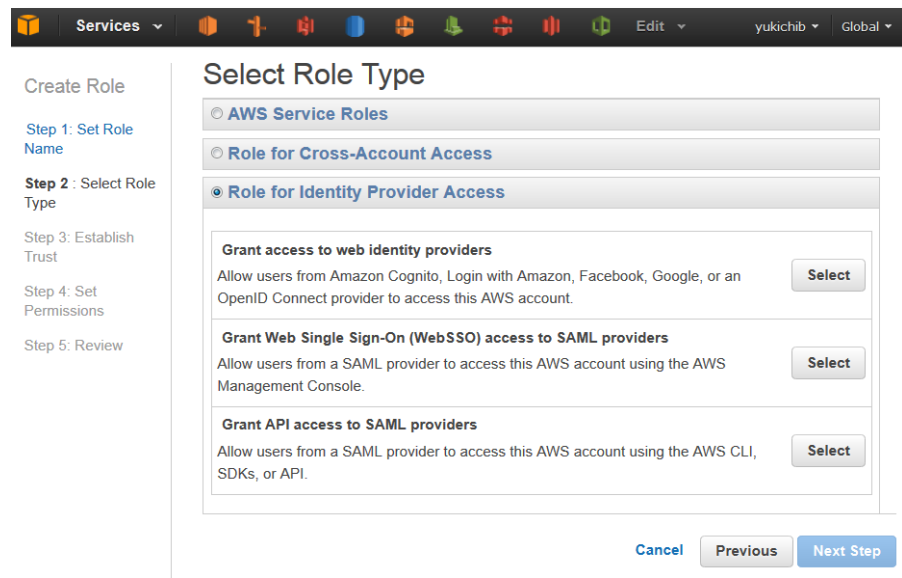
- Create two IAM Roles for SAML-Based Federation

1. Admin Role

- Role Name : ADFS-Admin (must be this name)
- Role Type : Grant Web Single Sign-On (WebSSO) access to SAML providers
- SAML Provider : ADFS
- Permission : Administrator Access

2. Read only Role

- Role Name : ADFS-RO (must be this name)
- Role Type : Grant Web Single Sign-On (WebSSO) access to SAML providers
- SAML Provider : ADFS
- Permission : Read Only Access



The screenshot shows the AWS IAM console interface for creating a new role. The top navigation bar includes the 'Services' dropdown and user information 'yukichib' and 'Global'. The left sidebar shows the 'Create Role' process with five steps: Step 1: Set Role Name, Step 2: Select Role Type (current), Step 3: Establish Trust, Step 4: Set Permissions, and Step 5: Review. The main content area is titled 'Select Role Type' and features three radio button options: 'AWS Service Roles', 'Role for Cross-Account Access', and 'Role for Identity Provider Access' (which is selected). Below these options are three expandable sections, each with a 'Select' button: 'Grant access to web identity providers' (describing access for Amazon Cognito, Login with Amazon, Facebook, Google, or OpenID Connect), 'Grant Web Single Sign-On (WebSSO) access to SAML providers' (describing access for SAML providers to the AWS Management Console), and 'Grant API access to SAML providers' (describing access for SAML providers to the AWS CLI, SDKs, or API). At the bottom right, there are 'Cancel', 'Previous', and 'Next Step' buttons.

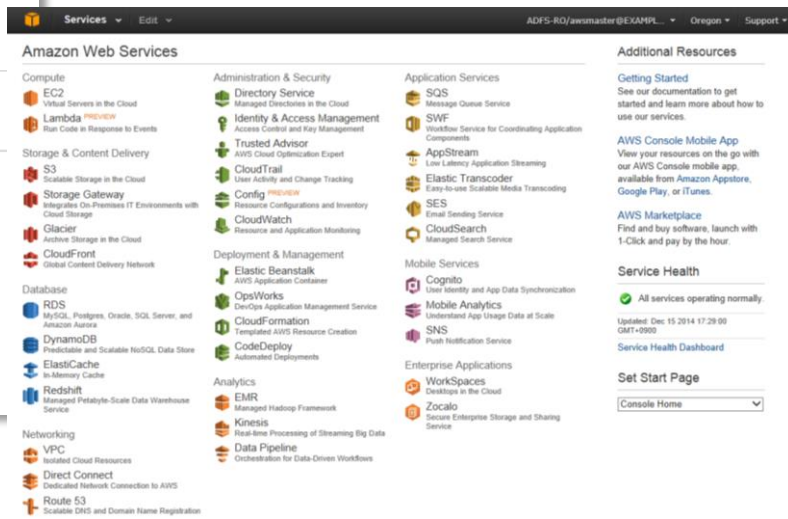
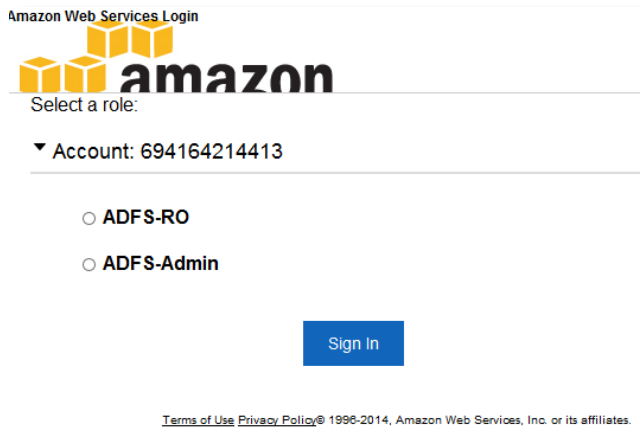
Now you can try SSO!

- Access to the ADFS federation site
 - https://<your_instance_public_dns>/adfs/ls/IdpInitiatedSignOn.aspx
 - You can sign in the site as following users (Password : P@ssW0rd), These users was created by the script in Step2
 - awsmaster@example.com
 - awsadmin@example.com
 - awsreadonly@example.com



Now you can try SSO!

- If you sign in as awsmaster, you can see the role selection view
- Select a role and then click **Sign In**. Then you can sign into the Management Console



- If you sign in as awsadmin or awsreadonly, you skip the role selection step and automatically sign into the AWS Management Console