AWS Federated Authentication with ADFS

With AWS IAM, you can create AWS users and use permissions to allow and deny their access to AWS resources. Howerver, you can also grante federated access to users in your existing identity system, thus leveraging all your passwords, policies, roles and groups.

This guide will walk you through the creation a federation setup using Microsoft Active Directory.

AD FS Federated Authentication Process

Here is the process a user will follow to authenticate to AWS using Active Directory and AD FS:

- A corporate user access the corporate Active Directory Federation Services portal sign-in page and provides Active Directory authentication credentials.
- AD FS authenticates the user against Active Directory.
- Active Directory returns the user's information, including AD group membership information.
- AD FS dynamically builds ARNs by using Active Diretory group membership for the IAM roles andu ser atributes for the AWS account IDs, and sends a signed assertion to the users browser with a redirect to post the assertion to AWS STS.
- Temporary credentials are returned using STS AssumeRoleWithSAML.
- The use is authenticated and provided access to the AWS management console.

Architecture

Objectives

After completing this lab, you will be able to:

- Create new users and groups in Microsoft Active Directory Federation Services
- Enable federated access to the AWS Management Console using na existing Microsoft AD server
- Create new roles in IAM and map those to your federated users

Task 1: Connect to your Windows Instance

1. In the AWS Management Console, under the **Services** menu, click **EC2.**
2. In the navigation pane, click **Instances.**
3. Select **LabADFS.**
4. Copy the **Public DNS** shown in the lower pane.
5. Establish a Remote Desktop Connection to LabADFS:
    - **Connection name:** LAB ADFS
    - **PC Name:** Enter the LabADFS public IP
    - **User Name:** mydomain\Stackadmin
    - **Password:** %TGBnhy67ujm

Task 2: Configure IIS

Windows IIS will host a login page to simulate the typical enterprise experience when logging into the AWS Management Console in a federated environment.

Taske 2.1: Create a Self-signed certificate
In order to create a secure login page, you first need to create a certificate.

6. Click on the **Start button.**
7. Click on the search icon (top right corner).
8. Start typing inet
9. Click on the **IIS Manager** icon that appears on the list.
10. Under **Connections** in the left-hand side, click on the hostname. It should look similar to **WIN-xxxxx.**
11. You may get a pop-up asking to get started with "Microsoft Web Platform".If so, check **Do not show this message and** click **No.**
12. In the **Features View** section, double-click on **Server Certificates.**
13. In the **Actions panel** on the right-hand side, click on **Create Self-Signed Certificate...**
14. For **Specify a friendly name for the certificate**, enter adfscert
15. Click **OK.**
    You should now see the certificate listed under **Server Certificates.**
    Note: Do not close the IIS Manager. You will need it for the next step.

Taske 2.2: Update binding to enable HTTPS
Now that you have created a certificate, you can enable HTTPS to make your login webpage secure.

16. In the **Connections** panel on the left-hand side, expand the level under the Hostname. It should look similar to **WIN-xxxxx**.
17. Expan the **Sites** folder.

18. Click on **Default Web Site.**
19. In the **Actions** panel on the right, click on **Bindings**.
20. In the **Site Bindings** Windows, click on the **Add...** button.
21. For **Type** select **https.**
22. Under **SSL Certificate**, select the **adfscert** certificate from the drop-down.
23. Click the **OK** button.
24. Click the **Close** button.
25. Close IIS Manager windows.

Task 3: Set up Active Directory

Na Active Directory has already been set up for you. However, it is currently empty. In order to log in, you will need to create some users and groups.

Task 3.1: Create the AD groups

First, create the two groups.

26. Open the **Windows PowerShell** by clicking the icon on the task bar at the borrom of the screen.
27. Run the following PowerShell commands:
    New-ADGroup AWS-Production -GroupScope Global -GroupCategory Security

    New-ADGroup AWS-Dev -GroupScope Global -GroupCategory Security

    The **AWS-Production** group will eventually grant its members permission to make changes on the EC2 console. The **AWS-Dev** group will eventually grant its members permission to make changes on the CloudBuild Screen.

Task 3.2: Create the AD users

Next, create some users.

28. Create the user **Bob** by running the following command:
    *New-ADUser -Name Bob -PasswordNeverExpires $true -AccountPassword (ConvertTo-SecureString '**$RFVbgt56yhn**' -AsPlainText -Force) -EmailAddress 'bob@mydomain.local' -Enabled $true -UserPrincipalName bob@mydomain.local*

29. Add the user **Bob** to ADFS-Dev:
    Add-ADGroupMember -Identity AWS-Dev -Members bob

30. Add the user **Bob** to ADFS-Production:
    Add-ADGroupMember -Identity AWS-Production -Members bob

You now have a new user “Bob” who is a member of both the **AWS-Dev** and **AWS-Production** groups.

31. Create a **ADFSVC service account:**
    New-ADUser -Name ADFSSVC -PasswordNeverExpires $true -AccountPassword (ConvertTo-SecureString '&UJMnhy65tgb' -AsPlainText -Force) -Enabled $true -UserPrincipalName adfssvc@mydomain.local -Description 'created AD FS service account'

This is the service account that will be used by ADFS to authenticate against the domain.

32. You can now close the PowerShell.

Task 3.3: Create the AD users

Now that you have created the certificate, the groups, adn the user, you need to add them to the ADFS server.

33. Click the **Start Menu**

34. Click on **Server Manager** icon.

35. At the top, click on the Notifications icon, that has a warming icon.

36. On the **Post-deployment configuration pop-up**, click on **Configure the federation service on this server.**

37. Select: **Create the first federation server in a federate domain.**
38. Click the **Next** button.
39. Ensure that the account listed is **mydomain\StackAdmin** and click the **Next** button.
40. For **SSL Certificate**, select the certificate you created previously. It should look similar to **WIN-XXXX.**
41. For Federation Service Display Name, enter **LAB ADFS**
42. Click the **Next** button.
43. You may see a **Group Managed Service accounts are available**... banner. Dismiss it by clicking on the **X** button.
44. Click on gray **Select** button.
45. In the **Select user or service account**, in the **Enter object name**... field, type: *adfssvc*
46. Click on the **Check names** button.

    This should result in the check returning: ADFSSVC (adfssvc@mydomain.local)

47. Click the **OK** button.
48. Make sure **Use an existing domain user account or group Managed Service Account** is checked. For the **Account Password,** enter: *&UJMnhy65tgb*
49. Click the **Next** button.
50. Make sure the **Create a database on this server using Windows Internal Database** is selected.
51. Click the **Next** button.
52. Review your setting and then click the **Next** button.
53. Click the **Configure** button.

Task 3.4: Resolve the error

At this point, you will likely get an error that says “Na error occured during the attempt to set the SPN for the specific service account... “This is a known issue and can be resolved as follows:

54. Click on the **Close** button.
55. Open the **Windows PowerShell** by clicking the icon on the task bar at the bottom of the screen.
56. Past the following command:
    setspn -a host/localhost adfssvc

    This should result in the following success message:
    Checking domain DC=mydomain,DC=local
    Registering ServicePrincipalNames for CN=ADFSSVC,CN=Users,DC=mydomain,DC=local
    host/localhost
    Updated object

Task 3.5: Obtain the SAML Medatada

In order to establish the Windows Machine as a trusted Identity Provider, you need to get configuration information from it. This information is stored in file called **FederationMetadada.xml**

57. Open a web browser on your **local computer.**

    Note: Do not open a browser in your RDP session.

58. Navigate to:
    https://<LabADFS public ip>/FederationMetadata/2007-06/FederationMetadata.xml

    Note: Replace <*LabADFS public ip*> with the Public IP you previously make a copy of.

59. You will likely get an error message similar to: “Your connection is not secure” or “Your Connection is not private”. This is because you are using a self-signed certificate, and should not be a cause of alarm.

    Depending on the browser, you may need to click on **Advanced** to proced for Chrome, **Add an exception** for Firefox, or **Continue to this website (not recommended)** for IE.

60. Save the FederationMetadata.xml file locally on your machine. You will need this file later to proceed.

Note: Do not attempt to copy and paste the contente manually.

Task 3.6: Create na Identity Provider

Now that you have the **FederationMetadata.xml** file, you can import it to AWS and establish the Windows machine as a trusted Identity Provider.

61. In the **AWS Management Console**, under the **Services** menu, click **IAM.**

62. In the left navigation pane, select **Identity providers.**

63. Click **Create Provider.**
64. For **Provider Type**, click on **Choose a provider type.**
65. Choose **SAML**
66. For the **Provider Name** enter: *LABSAML*
67. Click on **Choose File** and select the **FederationMetadata.xml** you previously saved.
68. Click **Next Step.**
69. Verify the information and click **Create.**
70. Click on the Provider and copy down the ARN. You will use this information later.

Next, create IAM role that match up to your AD groups and assign them the desired level of permissions.

Task 3.7: Create the ADFS-Production Role

71. In the **IAM console,** on the left-hand side, click on **Roles.**

72. Click on **Create role.**

73. Under Select type of trusted entity, select **SAML 2.0 federation**.

74. For SAML provider, choose **LabSAML**.

75. Select the **Allow programmatic and AWS Management Console access** radio button.

76. Click **Next: Permissions**.

77. In the search bar, type *AmazonEC2FullAccess*

This set of permissions will give you access to view and make changes to AWS EC2 console.

78. Select **AmazonEC2FullAccess** from the list.

79. Click **Next: Tags**.

80. Click **Next: Review**.

81. For Role name, enter *ADFS-Production*

82. For Role description, enter *LAB ADFS role Production*

83. Click **Create role**.

Task 3.8: Create the ADFS-Dev Role

84. Click **Create role**.

85. Under **Select Type of Trusted Identity**, select **SAML 2.0 federation**.

86. For SAML provider, choose **LabSAML**.

87. Select the **Allow programmatic and AWS Management Console access** radio button.

88. Click **Next: Permissions**.

89. For Permissions in the search bar, type *AWSCodebuildAdminAccess*.

    This set of permissions will give you access to view and make changes to AWS CodeBuild console.

90. Select **AWSCodeBuildAdminAccess**.

91. Click **Next: Tags**.

92. Click **Next: Review**.

93. For the **Role name**, enter *ADFS-Dev*.

94. For the **Role description**, enter *Lab ADFS role Dev*

95. Click the **Create role**.

Task 3.9: Configure AWS as a Trusted Relying party

AWS Management Console creates a **saml-metadata.xml** file that needs to be imported to ADFS to add AWS as a Trusted Relying Party.

96. Open your RDP Console.

97. Click the **Start Button**.

98. Click on **Administrative Tools**.

99. Double click on **AD FS Management**.

    Note: If prompted, in the User Account Control dialog box, click **Yes**.

100. On the right-hand side, click **Add Relying Party Trust...**
    This will open the **Add Relying Trust Wizard**.

101. Click on the **Start** button.

102. Choose the **Import data about the relying party published online radio** button.

103. Copy and paste the following into the field:
    https://signin.aws.amazon.com/static/saml-metadata.xml

104. Click **Next**.

105. For the **Display Name**, enter *Amazon Web Services*.

106. Click **Next**.

107. Ensure that **I do not want to configure multi-factor authentication settings for this relying party trust at this time is selected**, and then click **Next**.

108. Ensure the **Permit all users to access this relying party** radio button is selected.

109. Click **Next**.

110. Click **Next** again.

111. Ensure **Open the Edit Claim Rules dialog for this relying past trust when the wizard closes** checkbox is selected.

112. Click **Close**.

Task 4: Configure the Claim Rules

Claim Rules modify the SAML authentication response to include specific information needed by AWS to determine which role the use ris logging into. In this case, you will be sending the user's name, e-mail address, and their allowed roles.

Task 4.1: Configure Claim Rule 1

113. Ensure the window read **Edit Claim Rules for Amazon Web Services**.

114. Under the **Issuance Transform Rules** tab, click **Add Rule** button.

115. In the **Claim rule template** drop-down list, choose **Transform na Incoming Claim** from the dropdown.

116. Click the **Next** button.

117. Enter the following values:

- Claim rule name: *NameId*
- Incoming claim type: *Windows account name*
- Outgoing claim type: *Name ID*
- Outgoing name ID format: *Persistent Identifier*

118. Select **Pass through all claim values** radio button.

119. Click **Finish**

Task 4.2: Configure Claim Rule 2

120. Under the **Issuance Transform Rules** tab, click **Add Rule** button.

121. Choose **Send LDAP atribute as Claims** and click **Next**.

122. Enter the following values:

- Claim rule name: *RoleSessionName*
- Attribute store *Active Directory*
- LDAP Attribute: *E-mail-Addresses*
- Outgoing Claim type: *https://aws.amazon.com/SAML/Attributes/RoleSessionName*

123. Click **Finish**.

Task 4.3: Configure Claim Rule 3

124. Click **Add Rule**.

125. In the **Claim rule template** drop-down list, select **Send Claims Using a Custom Rule** and click **Next**.

126. For Claim rule name, enter *Get AD Groups*

127. In the Custom rule field, copy and paste the following:

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

128. Click **FInish**.

Task 4.4: Configure Claim Rule 4

129. Click **Add Rule**.

130. Select **Send Claims Using a Custom Rule**.

131. Click **Next**.

132. For **Claim rule name**, enter *Roles*

133. In the following script replace the *<ARN of SAML Provider>* and *<Account ID>* with the corresponding values from the ARN's copied in previous steps:

Replace *<ARN of SAML Provider>* with the value you copied in a previous step.
Replace <Account ID> value provided on the **Connection Details** section in Qwiklabs.

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

134. Click **Finish**.

135. Click **Apply**.

136. Close the **Claim Rules** window.

Task 5: Testing

137. In your web browser on your local machine, navigate to the following URL: *https://<LabADFS public ipp>/adfs/ls/IdpInitiatedSignOn.aspx*

You should see the AD FS signo n page.

138. Select **Sign in to ne of the following sites.**

139. Click **Sign In**.

140. Enter in the following credentials:

Email: ***bob@mydomain.local*** Password: ***$RFVbgt56yhn***

141. Click Sign In.

142. You should see **ADFS-Production** and **ADFS-Dev** roles on the list.

143. Select **ADFS-Production** as the role that user **Bob** will assume.

144. Click **Sign in**.

This will open up the AWS console.

145. In the AWS Management Console, under the **Services** menu, click **CloudFormation**.

This will result in na **Error** message that says it is “Unable to list data...”

146. In the AWS Management Console, under the **Services** menu, click **EC2**.

147. In the navigation pane, click **Instances**.

You should be able to view the running EC2 instances though.

148. Sign out of the AWS Console.

149. In your web browser on your local machine, navigate to the following URL: *https://<LabADFS public ip>/adfs/ls/IdpInitiatedSignOn.aspx*

150. Sign in again, but this time pick the **ADFS-Dev**.

151. Repeat your steps visiting the CodeBuild console and the EC2 console and note the difference in permissions for this role.

152. Sign out of the AWS Console.

That is it!