

# SECURING MULTI-VENDOR CLOUDS

*LABS 1 - Deploying Azure AD for single sign-on to an individual AWS account*



## ABSTRACT

This lab document contains a step-by-step guide to configuring Azure AD as the identity provider for an AWS Account. It will provide attendees with instructions on how to deploy Azure AD for single sign-on to an individual AWS account.

## Luciana Blanchard

Written for “Securing Multi-vendor Clouds” a series of events for **Microsoft Partners** created by Luciana Blanchard.

[Luciana.blanchard@microsoft.com](mailto:Luciana.blanchard@microsoft.com)

## Contents

1. Getting started .....	3
2. Pre-requisites .....	3
3. Add AWS Single-Account Access from the gallery .....	3
4. Configure Azure AD SSO.....	6
5. Configure AWS Single-Account Access SSO .....	12
6. Create AWS roles .....	14
7. Create AWS Policy for fetching the roles .....	17
8. Configure role provisioning in AWS Single-Account Access .....	22
9. Create Azure AD test users .....	27
10. Create Azure AD test groups.....	28
11. Assign the Azure AD test groups.....	29
12. Test SSO .....	32

## 1. Getting started

Azure AD supports single sign-on integration with AWS SSO. With AWS SSO you can connect Azure AD to AWS in one place and centrally govern access across hundreds of accounts and AWS SSO integrated applications. This enables seamless Azure AD sign-in experience for users to use the AWS Console.

The following Microsoft security solution procedure implements SSO for the example roles **AWS Administrators** and **AWS Developers**. Repeat this process for any additional roles you need.

This lab covers the following steps:

1. Create a new Azure AD enterprise application.
2. Configure Azure AD SSO for AWS.
3. Enable Azure AD to provision AWS IAM roles.
4. Update role mapping.
5. Test Azure AD SSO into AWS Management Console.

## 2. Pre-requisites

To get started, you need the following items:

- An Azure AD subscription. If you don't have a subscription, you can get a free account.
- An AWS single sign-on (SSO) enabled subscription.

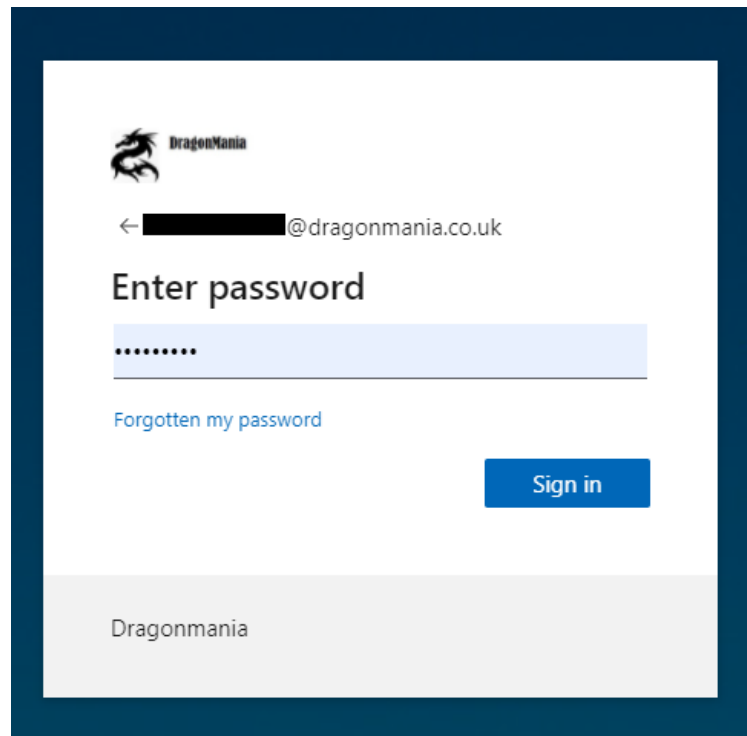
If you haven't done so yet, please read the "Part 0 - Getting Started - Lab Guide - Securing Multi-vendor Clouds" document. It will go through the steps required to setup the pre-requisites listed above.

## 3. Add AWS Single-Account Access from the gallery

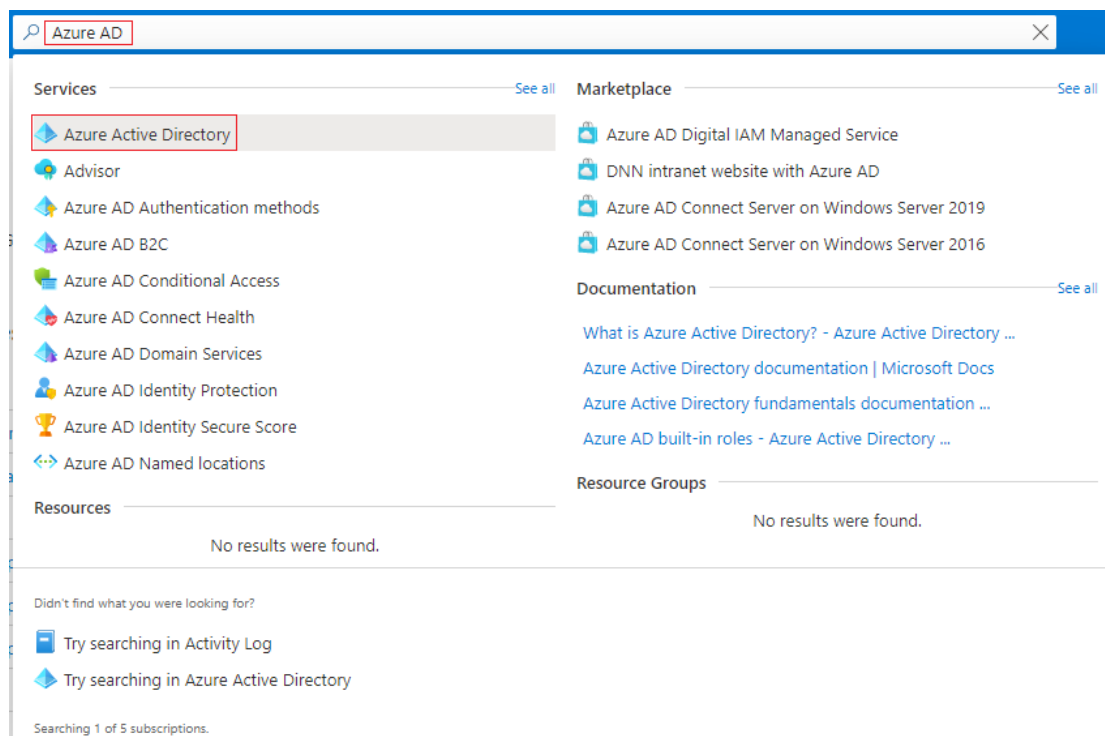
AWS administrators and developers use an enterprise application to sign in to Azure AD for authentication, then redirect to AWS for authorization and access to AWS resources. The simplest method to see the application is by signing in to <https://myapps.microsoft.com>.

To configure the integration of AWS Single-Account Access into Azure AD, you need to add AWS Single-Account Access from the gallery to your list of managed SaaS apps.

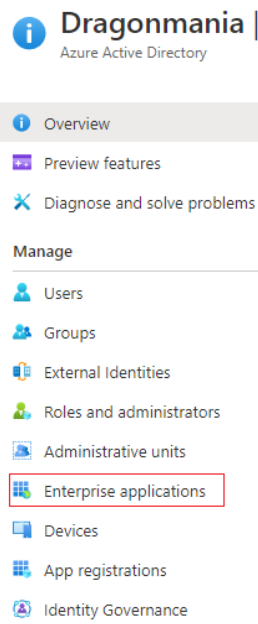
- 3.1 Sign in to the Azure portal using the admin username and password obtained during the creation of the tenant in "Part 0 - Getting Started - Lab Guide - Securing Multi-vendor Clouds", "How to create a new M365 Demo Tenant" section.



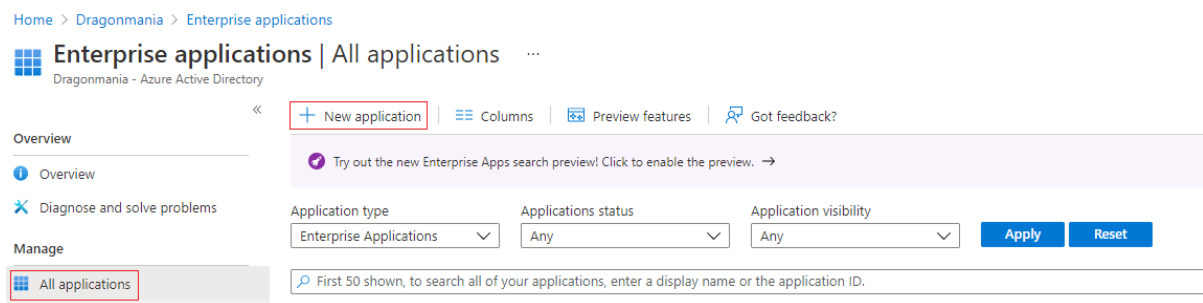
3.2 In the **Azure portal**, search for and select **Azure Active Directory**.



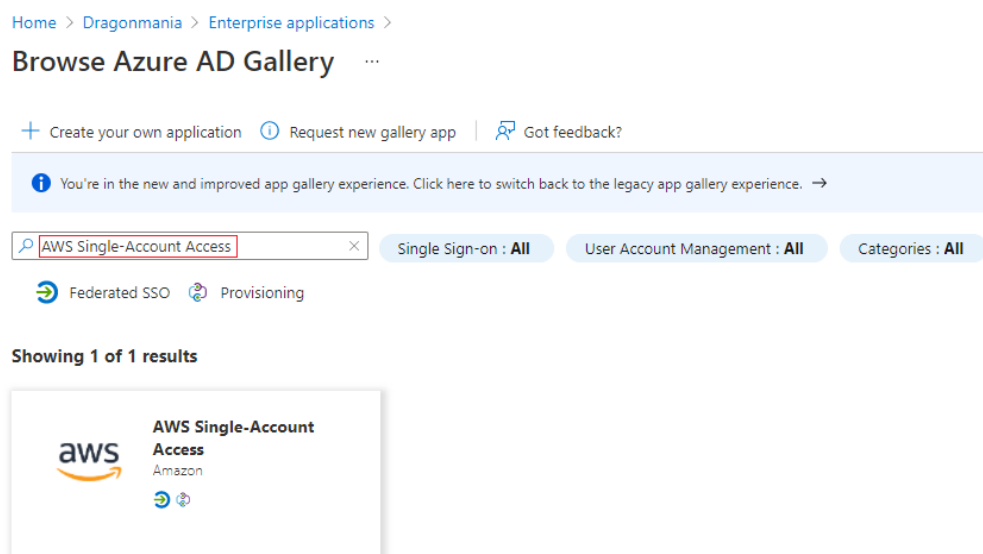
3.3 Within the **Azure Active Directory** overview menu, choose **Enterprise Applications**.



### 3.4 Select **New application** to add an application.



### 3.5 In the **Browse Azure AD Gallery** section, type **AWS Single-Account Access** in the search box.



### 3.6 Select **AWS Single-Account Access** from results panel.

[Home](#) > [Dragonmania](#) > [Enterprise applications](#) >

## Browse Azure AD Gallery ...



[+ Create your own application](#) [🕒 Request new gallery app](#) | [🗨️ Got feedback?](#)

**i** You're in the new and improved app gallery experience. [Click here to switch back to the legacy app gallery experience.](#) →

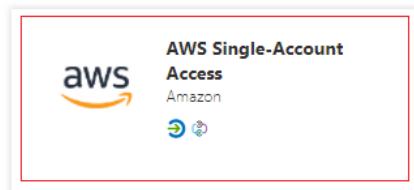
Single Sign-on : **All**

User Account Management : **All**

Categories : **All**

 Federated SSO  Provisioning

Showing 1 of 1 results



3.7 Leave the default name as is or give it a name that you can easily identify, then click **Create**.

### AWS Single-Account Access

×

[🗨️ Got feedback?](#)

Logo ⓘ



Name \* ⓘ

Publisher ⓘ

Amazon

Provisioning ⓘ

Automatic provisioning supported

Single Sign-On Mode ⓘ

Password-based Sign-on  
SAML-based Sign-on  
Linked Sign-on

URL ⓘ

<http://aws.amazon.com/>

[Read our step-by-step AWS Single-Account Access integration tutorial](#)

Federate to a single AWS account and use SAML claims to authorize access to AWS IAM roles. If you have many AWS accounts, consider using the AWS Single Sign-On gallery application instead.

**Create**

## 4. Configure Azure AD SSO

Follow these steps to enable Azure AD SSO in the Azure portal.

4.1 In the Azure portal, on the **AWS Single-Account Access** application integration page (the application you have created in the steps above), find the **Manage** section and select **single sign-on**.

**AWS Single-Account Access | Overview** ...

Enterprise Application

«

**Overview**

Deployment Plan

**Manage**

Properties

Owners

Roles and administrators (Preview)

Users and groups

**Single sign-on**

Provisioning

Self-service

**Security**

Conditional Access

Permissions

Token encryption

**Activity**

Sign-in logs

Usage & insights

Audit logs

Provisioning logs

Access reviews

**Properties**

aws

Name ⓘ

AWS Single-Account Access

Application ID ⓘ

Object ID ⓘ

**Getting Started**

**1. Assign users and groups**

Provide specific users and groups access to the applications

[Assign users and groups](#)

**What's New**

**Sign in charts have moved!**

The new Insights view shows sign in info along with otl

**Delete Application has moved to Properties**

You can now delete your application from the Properti

**Getting started has moved to Overview**

The Getting Started page has been replaced by the ste

\*If prompted with the dialog box below, select **Yes** to save the fixed settings and skip to section 4.12. Otherwise, follow the steps below.

### Save single sign-on setting

LAB1-AWS Single-Account Access uses fixed identifier and reply URLs. Do you want to save the following single sign-on settings?

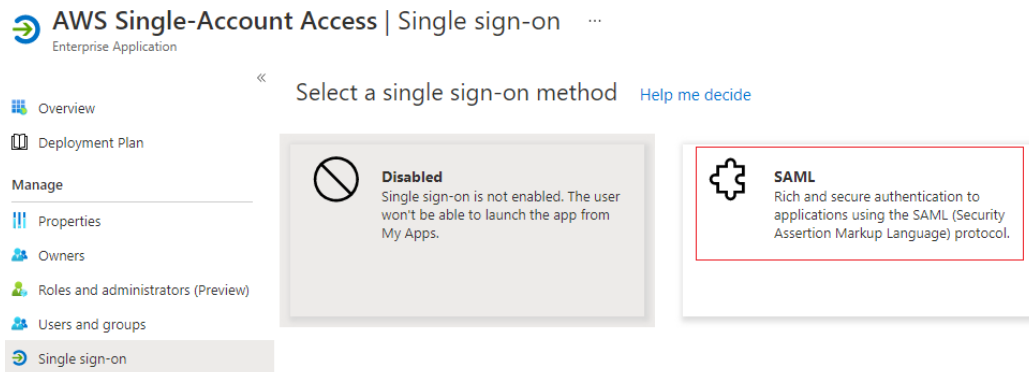
Identifier: <https://signin.aws.amazon.com/saml>

Reply URL: <https://signin.aws.amazon.com/saml>

Yes

No, I'll save later

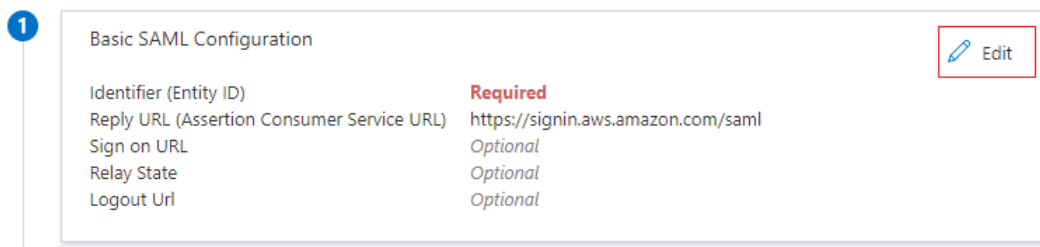
4.2 On the Select a single sign-on method page, select SAML.



#### 4.3 On the Set up single sign-on with SAML page, click the edit/pen icon for Basic SAML Configuration to edit the settings.

#### Set up Single Sign-On with SAML

Read the [configuration guide](#) for help integrating AWS Single-Account Access.



#### 4.4 In the **Basic SAML Configuration** section, update both **Identifier** (Entity ID) and **Reply URL** with the same default value: `https://signin.aws.amazon.com/saml`. You must select Save to save the configuration changes.

\* When you are configuring more than one instance (for example, you have multiple AWS accounts you want to integrate with Azure AD), provide an identifier value. From second instance onwards, use the following format, including a # sign to specify a unique SPN value. For example:  
`https://signin.aws.amazon.com/saml#2`.



## Basic SAML Configuration

 Save |  Got feedback?

### Identifier (Entity ID) \*

The default identifier will be the audience of the SAML response for IDP-initiated SSO

Default

**Patterns:** https://signin.aws.amazon.com/saml

### Reply URL (Assertion Consumer Service URL) \*

The default reply URL will be the destination in the SAML response for IDP-initiated SSO

Default

**Patterns:** https://signin.aws.amazon.com/saml

### Sign on URL

### Relay State

### Logout URL

- 4.5 AWS application expects the SAML assertions in a specific format, which requires you to add custom attribute mappings to your SAML token attributes configuration. The following screenshot shows the list of default attributes.

2

#### User Attributes & Claims

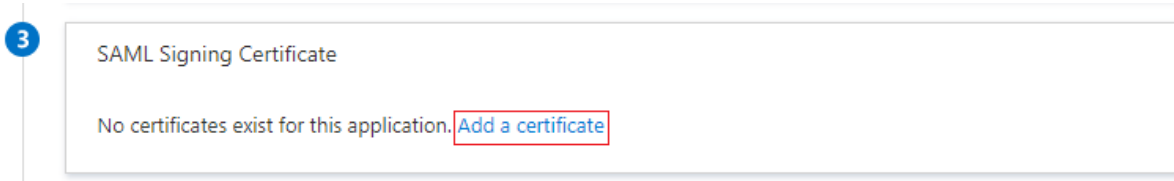
 Edit

givenname	user.givenname
surname	user.surname
emailaddress	user.mail
name	user.userprincipalname
Unique User Identifier	user.userprincipalname

- 4.6 In addition to above, AWS application expects few more attributes to be passed back in SAML response which are shown below. These attributes are also pre populated but you can review them as per your requirements.

Name	Source attribute	Namespace
RoleSessionName	user.userprincipalname	https://aws.amazon.com/SAML/Attributes
Role	user.assignedroles	https://aws.amazon.com/SAML/Attributes
SessionDuration	"provide a value between 900 seconds (15 minutes) to 43200 seconds (12 hours)"	https://aws.amazon.com/SAML/Attributes

4.7 On the **Set up single sign-on with SAML** page, in the **SAML Signing Certificate** (Step 3) dialog box, select **Add a certificate**.



4.8 Generate a new SAML signing certificate by selecting **New Certificate**. Enter an email address for certificate notifications.

## SAML Signing Certificate

×

Manage the certificate used by Azure AD to sign SAML tokens issued to your app





Save
**+ New Certificate**
↑ Import Certificate
Got feedback?

Status	Expiration Date	Thumbprint
Signing Option	Sign SAML assertion	
Signing Algorithm	SHA-256	
Notification Email Addresses		
<div> <div></div> <div>@dragonmania.co.uk</div> <div>✕</div> </div>		
<input type="text"/>		

4.9 In the **SAML Signing Certificate**, click **Save**.

## SAML Signing Certificate

Manage the certificate used by Azure AD to sign SAML tokens issued to your app

 Save  New Certificate  Import Certificate  Got feedback?

Status	Expiration Date	Thumbprint
n/a	08/16/2024	Will be displayed on save

Signing Option Sign SAML assertion

Signing Algorithm SHA-256





Notification Email Addresses

luciblanchard@dragonmania.co.uk

4.10 In the **SAML Signing Certificate**, click on the “...” and select **Mark certificate active**.

## SAML Signing Certificate

Manage the certificate used by Azure AD to sign SAML tokens issued to your app

 Save  New Certificate  Import Certificate  Got feedback?







Status	Expiration Date	Thumbprint
Inactive	8/16/2024, 3:46:11 PM	E0704FD15E553BED6AEEED5E03DF73158EDC00BD

Signing Option Sign SAML assertion

Signing Algorithm SHA-256

Notification Email Addresses





luciblanchard@dragonmania.co.uk

-  Make certificate active
-  Base64 certificate download
-  PEM certificate download
-  Raw certificate download
-  Download federated certificate XML
-  Delete Certificate

4.11 In the **Activating your certificate** prompt, select **Yes** to activate the certificate.

## SAML Signing Certificate

Manage the certificate used by Azure AD to sign SAML tokens issued to your app

 Save  New Certificate  Import Certificate  Got feedback?

### Activating your certificate

You are about to activate an inactive certificate. To prevent application downtime, ensure that this certificate has been successfully onboarded to your application on the application's site.

4.12 In the **SAML Signing Certificate** section, select **Download** to download the **Federated Metadata XML** and save it on your device.

3

**SAML Signing Certificate** Edit

Status	Active
Thumbprint	9CEA37643ACE0D710AD63296857B251D1FCA5C48
Expiration	12/20/2025, 8:50:17 PM
Notification Email	luciblanchard@dragonmania.co.uk
App Federation Metadata Url	<a href="https://login.microsoftonline.com/69e13e16-254d...">https://login.microsoftonline.com/69e13e16-254d...</a>
Certificate (Base64)	<a href="#">Download</a>
Certificate (Raw)	<a href="#">Download</a>
Federation Metadata XML	<a href="#">Download</a>

4.13 In the **Set up AWS Single-Account Access** section, copy the following URL(s):

4

**Set up AWS Single-Account Access**

You'll need to configure the application to link with Azure AD.

Login URL	<a href="https://login.microsoftonline.com/...">https://login.microsoftonline.com/...</a>
Azure AD Identifier	<a href="https://sts.windows.net/...">https://sts.windows.net/...</a>
Logout URL	<a href="https://login.microsoftonline.com/...">https://login.microsoftonline.com/...</a>

[View step-by-step instructions](#)

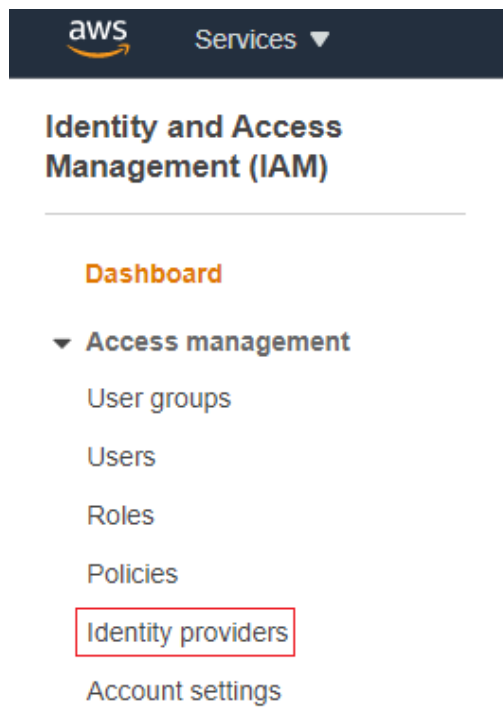
## 5. Configure AWS Single-Account Access SSO

5.1 In a different browser window, sign-on to your AWS company site as an administrator.

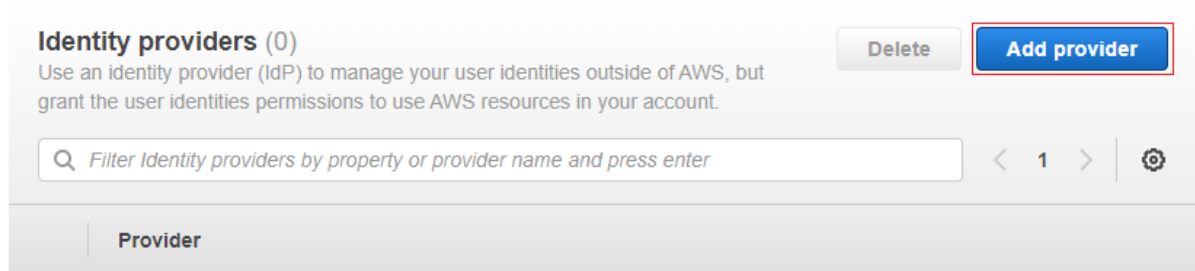
5.2 In the search box, type “IAM”, select “IAM – Manage access to AWS resources”.

The screenshot shows the AWS IAM console search results for 'iam'. The search bar at the top contains 'iam'. Below the search bar, the results are categorized into 'Services (1)', 'Features (11)', 'Documentation (78,445)', 'Knowledge Articles (27)', and 'Marketplace (223)'. Under the 'Services' category, the 'IAM' service is highlighted with a red box. The 'IAM' service description is 'Manage access to AWS resources'. A link to 'See all 11 results' is visible at the bottom right.

5.3 Select **Identity Providers > Create Provider**.



5.4 On the **Identity Providers** page, click **Add Provider**.



5.5 On the **Add Identity Provider** page, perform the steps below:

- Select **SAML** for the **Provider Type**
- Enter a name for the **Provider Name** (for example: LabAAD)
- Click **Choose File** and upload the **metadata file** you download from Azure on step 4.12 earlier in this document.
- Click **Add Provider**.

# Add an Identity provider

## Configure provider

### Provider type

☒ SAML

Establish trust between your AWS account and a SAML 2.0 compatible Identity Provider such as Shibboleth or Active Directory Federation Services.

☐ OpenID Connect

Establish trust between your AWS account and Identity Provider services, such as Google or Salesforce.

### Provider name


Enter a meaningful name to identify this provider

LabAAD

Maximum 128 characters. Use alphanumeric or '\_' characters.

### Metadata document

This document is issued by your IdP.

 Choose file

File needs to be a valid UTF-8 XML document.

 AWS Single-Account AccessFederationMetadata.xml

## Add tags (Optional)

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

Add tag

You can add up to 50 more tags

Cancel

Add provider

## 6. Create AWS roles

### 6.1 In AWS IAM, select Roles -> Create Role

Identity and Access Management (IAM)

Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings

IAM > Roles

Roles (2) Info
Refresh
Delete
Create role

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Search

<input type="checkbox"/>	Role name	Trusted entities
<input type="checkbox"/>	AWSServiceRoleForSupport	AWS Service: support
<input type="checkbox"/>	AWSServiceRoleForTrustedAdvisor	AWS Service: trusted





6.2 On the **Create role** page, perform the following steps:

- Under **Select type of trusted entity**, select **SAML 2.0** federation.
- Under **Choose a SAML 2.0 Provider**, select the **SAML** provider you created previously (for example: *LabAAD*).
- Select **Allow programmatic and AWS Management Console access**.
- Select **Next: Permissions**.

## Create role

1 2 3 4

### Select type of trusted entity

 <b>AWS service</b> EC2, Lambda and others	 <b>Another AWS account</b> Belonging to you or 3rd party	 <b>Web identity</b> Cognito or any OpenID provider	 <b>SAML 2.0 federation</b> Your corporate directory
--	---	---	--

Allows users that are federated with SAML 2.0 to assume this role to perform actions in your account. [Learn more](#)

### Choose a SAML 2.0 provider

If you're creating a role for API access, choose an Attribute and then type a Value to include in the role. This restricts access to users with the specified attributes.

**SAML provider**

[Create new provider](#) [Refresh](#)

☐ Allow programmatic access only

☒ Allow programmatic and AWS Management Console access

**Attribute**

**Value\***

**Condition** [+ Add condition \(optional\)](#)

\* Required

[Cancel](#)

[Next: Permissions](#)

6.3 On the **Attach permissions policies** dialog box, select **AdministratorAccess**. Then select **Next: Tags**.

## Create role

1 2 3 4

## ▼ Attach permissions policies

Choose one or more policies to attach to your new role.

Create policy

↺

Filter policies ▼

Q Search

Showing 836 results

	Policy name ▼	Used as
<input type="checkbox"/>	▶ AccessAnalyzerServiceRolePolicy	None
<input checked="" type="checkbox"/>	▶ AdministratorAccess	None
<input type="checkbox"/>	▶ AdministratorAccess-Amplify	None
<input type="checkbox"/>	▶ AdministratorAccess-AWSElasticBeanstalk	None
<input type="checkbox"/>	▶ AlexaForBusinessDeviceSetup	None
<input type="checkbox"/>	▶ AlexaForBusinessFullAccess	None
<input type="checkbox"/>	▶ AlexaForBusinessGatewayExecution	None
<input type="checkbox"/>	▶ AlexaForBusinessLifesizeDelegatedAccessPolicy	None

▶ Set permissions boundary

\* Required

Cancel

Previous

Next: Tags

6.4 On the **Add Tags** dialog box, leave blank and select **Next: Review**.

## Create role

1 2 3 4

## Add tags (optional)

IAM tags are key-value pairs you can add to your role. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this role. [Learn more](#)

Key	Value (optional)	Remove
<input type="text" value="Add new key"/>	<input type="text"/>	

You can add 50 more tags.

Cancel

Previous

Next: Review

6.5 On the **Review** dialog box, perform the following steps:

- In **Role Name**, enter your role name (**Administrator**).
- In **Role Description**, enter the description.
- Select **Create Role**.



## Create role

1 2 3 4

### Review

Provide the required information below and review this role before you create it.

Role name\* Administrator

Use alphanumeric and '+-=, @-\_' characters. Maximum 64 characters.

Role description Full admin access

Maximum 1000 characters. Use alphanumeric and '+-=, @-\_' characters.

\* Required

Cancel

Previous

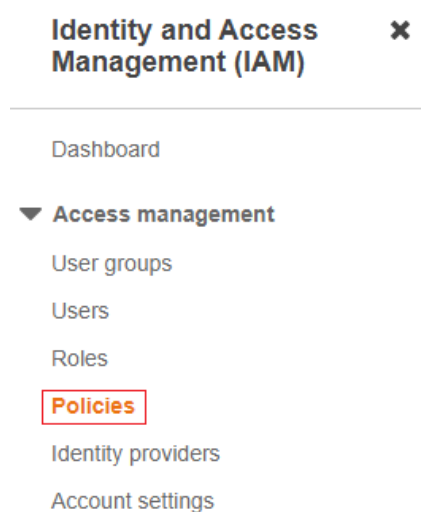
Create role

6.6 Create an additional role following the steps listed above, name it **Developer** and attach to it **AmazonS3FullAccess** permissions.

6.7 You have now successfully create an **Administrator** and a **Developer** role in AWS.

## 7. Create AWS Policy for fetching the roles

7.1 In **AWS IAM**, select **Policies**.



7.2 Create a new policy by selecting **Create policy** for fetching the roles from the AWS account in Azure AD user provisioning.

IAM &gt; Policies

**Policies (837)** [Info](#)  
A policy is an object in AWS that defines permissions.

< 1 2 3 4 5 6 7 ... 42 > [Settings](#)

**Policy Name** ▼

☐ ☒ [AWSDirectConnectReadOnlyAccess](#)

### 7.3 Create your own policy to fetch all the roles from AWS accounts.

- In **Create policy**, select the **JSON** tab.
- In the policy document, add the following JSON:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:ListRoles"
      ],
      "Resource": "*"
    }
  ]
}
```

### Create policy

1 2 3

A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. [Learn more](#)

**Visual editor** **JSON** [Import managed policy](#)

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "iam:ListRoles"
8       ],
9       "Resource": "*"
10    }
11  ]
12 }
```

Security: 0 Errors: 0 Warnings: 0 Suggestions: 0

Character count: 99 of 6,144.

[Cancel](#) [Next: Tags](#)

- Select **Next: Tags** to continue.
- On the **Add Tags** dialog box, leave empty and click **Next: Review**.

## Create policy

1 2 3

## Add tags (Optional)

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

Add tag

You can add up to 50 more tags

Cancel

Previous

Next: Review

## 7.4 Define the new policy.

- For **Name**, enter **AzureAD\_SSOUserRole\_Policy**.
- For **Description**, enter **This policy will allow to fetch the roles from AWS accounts**.
- Select **Create policy**.

## Create policy

1 2 3

## Review policy

Name\* AzureAD\_SSOUserRole\_Policy

Use alphanumeric and '+=, @-\_' characters. Maximum 128 characters.

Description This policy will allow to fetch the roles from AWS accounts

Maximum 1000 characters. Use alphanumeric and '+=, @-\_' characters.

## Summary

Filter			
Service	Access level	Resource	Request condition
Allow (1 of 289 services) <a href="#">Show remaining 288</a>			
IAM	Limited: List	All resources	None

## Tags

Key	Value
-----	-------

No tags associated with the resource.

\* Required

Cancel

Previous

Create policy

7.5 Create a new user account in the **AWS IAM** service.

- In the **AWS IAM** console, select **Users**.

## Identity and Access Management (IAM) ✕

Dashboard

### ▼ Access management

User groups

**Users**

Roles

Policies

Identity providers

Account settings

b) To create a new user, select **Add user**.

IAM > Users

**Users (0)** [Info](#)

↻

Delete

**Add users**

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

🔍 Find users by username or access key

< 1 > ⚙️

<input type="checkbox"/>	User name	Groups	Last activity	MFA	Password age	Active key age
--------------------------	-----------	--------	---------------	-----	--------------	----------------

c) In the **Add user** section:

- Enter the user name as **AzureADRoleManager**.
- For the access type, select **Programmatic access**. This way, the user can invoke the APIs and fetch the roles from the AWS account.
- Select **Next Permissions**.

## Add user

1 2 3 4 5

## Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name\*

[+ Add another user](#)

## Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

- Access type\* ☒ **Programmatic access**  
 Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.
- ☐ **AWS Management Console access**  
 Enables a **password** that allows users to sign-in to the AWS Management Console.

\* Required

[Cancel](#)

[Next: Permissions](#)

## 7.6 Create a new policy for this user.

- Select **Attach existing policies directly**.
- Search for the newly created policy in the filter section **AzureAD\_SSOUUserRole\_Policy**.
- Select the policy, and then select **Next: Tags**.

## Add user

1 2 3 4 5

## ▼ Set permissions

Add user to group
 Copy permissions from existing user
 Attach existing policies directly

[Create policy](#)

Filter policies  Showing 1 result

	Policy name ▼	Type	Used as
<input checked="" type="checkbox"/>	AzureAD_SSOUUserRole_Policy	Customer managed	None

## ▼ Set permissions boundary

Set a permissions boundary to control the maximum permissions this user can have. This is an advanced feature used to delegate permission management to others. [Learn more](#)

[Cancel](#)

[Previous](#)

[Next: Tags](#)

- In the **Add Tags** dialog box, leave empty and select **Next: Review**.

## Add user

1 2 3 4 5

## Add tags (optional)

IAM tags are key-value pairs you can add to your user. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this user. [Learn more](#)

Key	Value (optional)	Remove
<input type="text" value="Add new key"/>	<input type="text"/>	

You can add 50 more tags.

Cancel

Previous

Next: Review

7.7 Review the policy to the attached user, select **Create User**.

## Add user

1 2 3 4 5

## Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

## User details

User name	AzureADRoleManager
AWS access type	Programmatic access - with an access key
Permissions boundary	Permissions boundary is not set

Cancel

Previous

Create user

7.8 Download the user credentials by clicking on **download .csv**. Select **Close**.

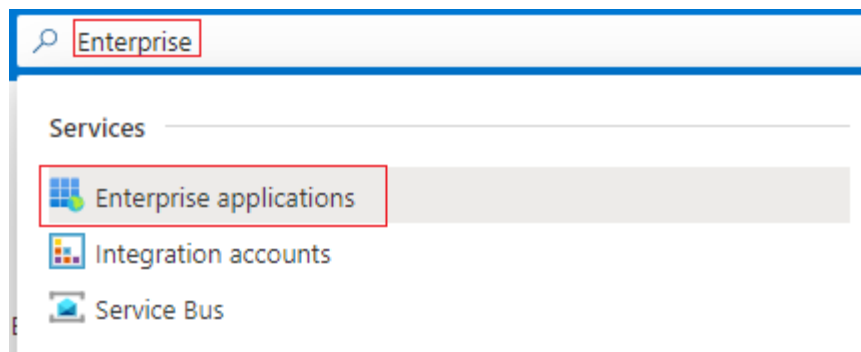
 Download .csv

	User	Access key ID	Secret access key
▶	✓ AzureADRoleManager	AKIA3LHFULBFBTBX676B 	***** <a href="#">Show</a>

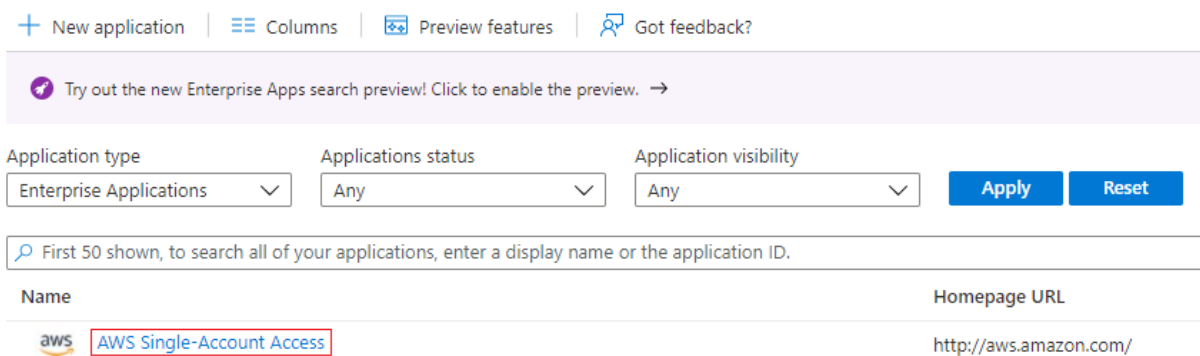
Close

## 8. Configure role provisioning in AWS Single-Account Access

8.1 In **Azure Portal**, search for **Enterprise Applications**.



8.2 In **Enterprise Applications**, select the **AWS Single-Account Access** application you have created in step 3.4 earlier in this document.



8.3 On the application page, select **Provisioning**.

[Home](#) > [Enterprise applications](#) >

## AWS Single-Account Access | Overview ...

Enterprise Application

### Overview

#### Deployment Plan

#### Manage

##### Properties

##### Owners

##### Roles and administrators (Preview)

##### Users and groups

##### Single sign-on

##### Provisioning

##### Self-service

#### Security

##### Conditional Access

##### Permissions

##### Token encryption

#### Activity

##### Sign-in logs


##### Usage & insights

##### Audit logs

##### Provisioning logs

##### Access reviews

### Properties

 Name ⓘ

Application ID ⓘ

Object ID ⓘ

### Getting Started



#### 1. Assign users and groups

Provide specific users and groups access to the applications

[Assign users and groups](#)



#### 2. Set up single sign on

Enable users to sign into their a using their Azure AD credential:

[Get started](#)

### What's New



#### Sign in charts have moved!

The new Insights view shows sign in info along with other useful application data. [View insights](#)



#### Delete Application has moved to Properties

You can now delete your application from the Properties page. [View properties](#)




#### Getting started has moved to Overview

The Getting Started page has been replaced by the steps above

8.4 On the **Provisioning** page, select **Get Started**.



[Home](#) > [Enterprise applications](#) > [AWS Single-Account Access](#)

 **AWS Single-Account Access** | Provisioning  
Enterprise Application

Overview

Deployment Plan

Manage

Properties

Owners

Roles and administrators (Preview)

Users and groups

Single sign-on

**Provisioning**

Self-service

Security


Conditional Access

Permissions

Token encryption

Activity

Got a second? We would love your feedback on user provisioning. →



Automate identity lifecycle management with Azure Active Directory

Automatically create, update, and delete accounts when users join, leave, and move within your organization. [Learn more.](#)

Get started

[What is provisioning?](#)[Plan an application deployment.](#)

[Configure automatic provisioning.](#)

8.5 On the **Provisioning** page, perform the following:

- In **Provisioning Mode**, select **Automatic**.
- In **Admin Credentials**, copy and paste the **Client Secret** (Access Key ID) found in the .csv file you downloaded from AWS in step 4.6.9.
- In **Admin Credentials**, copy and paste the **Secret Token** (Secret Access Key) found in the .csv file you downloaded from AWS in step 7.8 earlier in this document.
- Select **Test Connection** to validate.
- Select **Save**.

Deploying Azure AD for single sign-on to an individual AWS account

Page 25 | 37

# Provisioning ...

 Save  Discard

## Provisioning Mode

Automatic 

Use Azure AD to manage the creation and synchronization of user accounts in AWS Single-Account Access based on user and group assignment.

## Admin Credentials

### Admin Credentials

Azure AD needs the following information to connect to AWS Single-Account Access's API and synchronize user data.

#### clientsecret

.....

#### Secret Token

.....

Test Connection

## 8.1 On the **Provisioning** page, select **Start Provisioning**.

Home > Enterprise applications > AWS Single-Account Access

## AWS Single-Account Access | Provisioning ...

Enterprise Application

Overview

Deployment Plan

Manage

Properties

Owners

Roles and administrators (Preview)

Users and groups

Single sign-on

Provisioning

Self-service

Security

Conditional Access

Permissions

Token encryption

Activity






Sign-in logs


Usage & insights

Audit logs

Provisioning logs

Access reviews

«  Start provisioning ☐ Stop provisioning  Restart provisioning  Edit provisioning  Provision on demand |  Refresh

 Got a second? We would love your feedback on user provisioning. →

### Current cycle status

Initial cycle not run.

0% complete

[View provisioning logs](#)

### Statistics to date

✓ [View provisioning details](#)

✓ [View technical information](#)

### Manage provisioning

[Update credentials](#)

[Edit attribute mappings](#)

[Add scoping filters](#)

[Provision on demand](#)

**Note**

The provisioning service imports roles only from AWS to Azure AD. The service does not provision users and groups from Azure AD to AWS.

**Note**

After you save the provisioning credentials, you must wait for the initial sync cycle to run. Sync usually takes around 40 minutes to finish. You can see the status at the bottom of the **Provisioning** page, under **Current Status**.


## 9. Create Azure AD test users

In this section, you'll create two test users in the Azure portal.

- 9.1 In the **Azure portal**, search for and select **Azure Active Directory**.
- 9.2 Within the **Azure Active Directory** overview menu, choose **Users > All users**.
- 9.3 Select **New user** at the top of the screen.
- 9.4 In the **User properties**, follow these steps:
- 9.5 In the **Name** field, enter **Test-AWSAdmin**.
- 9.6 In the **User name** field, enter the **Test-AWSAdmin@companydomain.extension**. For example, **Test-AWSAdmin@contoso.com**.
- 9.7 Select the **Show password** check box, and then write down the value that's displayed in the Password box.
- 9.8 Click **Create**.
- 9.9 Repeat the steps in this section to create a second user named **Test-AWSDeveloper**.

## New user ...

Dragonmania

 Got feedback?



### Create user

Create a new user in your organization. This user will have a user name like `alice@dragonmania.co.uk`.  
[I want to create users in bulk](#)



### Invite user

Invite a new guest user to collaborate with your organization. The user will be emailed an invitation they can accept in order to begin collaborating.  
[I want to invite guest users in bulk](#)

[Help me decide](#)

## Identity

User name \* ⓘ

 ✓

@



[The domain name I need isn't shown here](#)

Name \* ⓘ

 ✓

First name

Last name

## Password



Auto-generate password



Let me create the password

Initial password



Show Password

Create

## 10. Create Azure AD test groups

In this section, you'll create two test groups in the Azure portal.

- 10.1 In the **Azure portal**, search for and select **Azure Active Directory**.
- 10.2 Within the **Azure Active Directory** overview menu, choose **Groups** > **All groups**.
- 10.3 In the **Group Type**, select **Security**.
- 10.4 In the **Group Name** field, enter **AWS-Account1-Administrators**.
- 10.5 Select **Members**, search for **Test-AWSAdmin** created in the previous section and click **Select**.
- 10.6 Select **Create** to create the new group in Azure AD.
- 10.7 Repeat the steps in this section to create another group called **AWS-Account1-Developers**, add **Test-AWSDeveloper** user to **AWS-Account1-Developers**.

[Home](#) > [Dragonmania](#) > [Groups](#) >

## New Group ...

Group type \* ⓘ

 ▼

Group name \* ⓘ

 ✓

Group description ⓘ

Azure AD roles can be assigned to the group ⓘ

☐ Yes ☒ No

Membership type \* ⓘ

 ▼

Owners

[No owners selected](#)

Members

[1 member selected](#)

## 11. Assign the Azure AD test groups

In this section, you'll enable **Test-AWSDeveloper** and **Test-AWSAdmin** to use Azure single sign-on by granting access to **AWS Single-Account Access**.

11.1 In the **Azure portal**, select **Enterprise Applications**, and then select **All applications**.

[Home](#) > [Enterprise applications](#)



## Enterprise applications

Dragonmania - Azure Active Directory

<<

### Overview

- [Overview](#)
- [Diagnose and solve problems](#)

### Manage

- [All applications](#)
- [Application proxy](#)
- [User settings](#)
- [Collections](#)

11.2 In the applications list, select **AWS Single-Account Access**.

[+ New application](#) | [Columns](#) | [Preview features](#) | [Got feedback?](#)

[Try out the new Enterprise Apps search preview! Click to enable the preview. →](#)

Application type Enterprise Applications	Applications status Any	Application visibility Any	<a href="#">Apply</a>	<a href="#">Reset</a>
---	----------------------------	-------------------------------	-----------------------	-----------------------

First 50 shown, to search all of your applications, enter a display name or the application ID.

Name	Homepage URL
AWS Single-Account Access	http://aws.amazon.com/

11.3 In the app's overview page, find the **Manage** section and select **Users and groups**.

[Home](#) > [Enterprise applications](#) >



## AWS Single-Account Access | Overview

Enterprise Application

<<

- [Overview](#)
- [Deployment Plan](#)
- Manage**
  - [Properties](#)
  - [Owners](#)
  - [Roles and administrators \(Preview\)](#)
  - [Users and groups](#)
  - [Single sign-on](#)

### Properties



Name ⓘ

AWS Single-Account Access

Application ID ⓘ

[Redacted]

Object ID ⓘ

[Redacted]

### Getting Started

11.4 Select **Add user/group**, then select **None Selected** in the **Add Assignment** dialog.

+ Add user/group   Edit   Remove   Update Credentials   Columns   Got feedback?

**i** The application will appear for assigned users within My Apps. Set 'visible to users?' to no in properties to prevent this. →

First 200 shown, to search all users & groups, enter a display name.

Display Name

No application assignments found

11.5 In the **Users and groups** dialog, search for **AWS-Account1-Administrators**, then click the **Select** button at the bottom of the screen.

[Home](#) > [Enterprise applications](#) > [AWS Single-Account Access](#) >

## Add Assignment ...

Dragonmania

**⚠** When you assign a group to an application, only users directly in the group will have access. The assignment does not cascade to nested groups. ×

Users and groups

1 group selected.

Select a role \*

None Selected

11.6 Under **Select a role**, select **None Selected**, then select **Administrator,LabAAD** (the administrator role you have created in AWS) and click **Select** at the bottom of the screen.

Select a role ×

Only a single role can be selected

Enter role name to filter items...

Administrator,LabAAD

Developer,LabAAD

11.7 In the **Add Assignment** dialog, click the **Assign** button.

11.8 Repeat the steps in this section for the **AWS-Account1-Developers** group, assign it to the **Developer,LabAAD** role (the developer role you have created in AWS).

**AWS Single-Account Access** | Users and groups ...

Enterprise Application

Overview  
Deployment Plan  
Manage  
Properties  
Owners  
Roles and administrators (Preview)  
Users and groups

+ Add user/group | Edit | Remove | Update Credentials | Columns | Got feedback?

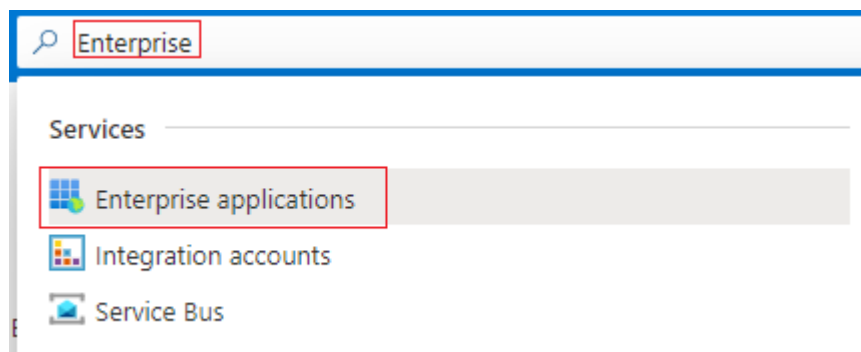
The application will appear for assigned users within My Apps. Set 'visible to users?' to no in properties to prevent this. →

First 200 shown, to search all users & groups, enter a display name.

	Display Name	Object Type	Role assigned
<input type="checkbox"/>	AW AWS-Account1-Developers	Group	Developer,LabAAD
<input type="checkbox"/>	AW AWS-Account1-Administrators	Group	Administrator,LabAAD

## 12. Test SSO

12.1 In Azure Portal, search for **Enterprise Applications**.



12.2 In **Enterprise Applications**, select the **AWS Single-Account Access** application you have created in step 3.4 earlier in this document.

+ New application | Columns | Preview features | Got feedback?

Try out the new Enterprise Apps search preview! Click to enable the preview. →

Application type: Enterprise Applications | Applications status: Any | Application visibility: Any | Apply | Reset

First 50 shown, to search all of your applications, enter a display name or the application ID.

Name	Homepage URL
aws AWS Single-Account Access	http://aws.amazon.com/

12.3 On the application page, select **Single sign-on**.



## AWS Single-Account Access | SAML-based Sign-on

Enterprise Application

« [Upload metadata file](#) [Change single sign-on mode](#) [Test this application](#) | [Got feedback?](#)

Overview

Deployment Plan

Manage

- Properties
- Owners
- Roles and administrators (Preview)
- Users and groups
- Single sign-on**
- Provisioning

### Set up Single Sign-On with SAML

Read the [configuration guide](#) for help integrating AWS Single-Account Access.

1 Basic SAML Configuration [Edit](#)

Identifier (Entity ID)	https://signin.aws.amazon.com/saml#4
Reply URL (Assertion Consumer Service URL)	https://signin.aws.amazon.com/saml#4
Sign on URL	Optional
Relay State	Optional
Logout URL	Optional

12.4 On the **Set up Single Sign-On with SAML** page, click on **Test this application**.

## AWS Single-Account Access | SAML-based Sign-on

Enterprise Application

« [Upload metadata file](#) [Change single sign-on mode](#) [Test this application](#) | [Got feedback?](#)

Overview

Deployment Plan

Manage

- Properties
- Owners
- Roles and administrators (Preview)
- Users and groups
- Single sign-on**
- Provisioning

### Set up Single Sign-On with SAML

Read the [configuration guide](#) for help integrating AWS Single-Account Access.

1 Basic SAML Configuration [Edit](#)

Identifier (Entity ID)	https://signin.aws.amazon.com/saml#4
Reply URL (Assertion Consumer Service URL)	https://signin.aws.amazon.com/saml#4
Sign on URL	Optional
Relay State	Optional
Logout URL	Optional

12.5 On the **Test single sign-on with AWS Single-Account Access** page, select **Sign in as someone else**.

\*If you don't have the **My Apps Secure Sign-in Extension** installed, click on the prompt below to install it.

## Test single sign-on with AWS Single-Account Access



[Got feedback?](#)

**i** Microsoft recommends installing the My Apps Secure Sign-in Extension for automatic error capture and resolution guidance. Make sure you allow third-party cookies if you have installed it but this message still shows up.

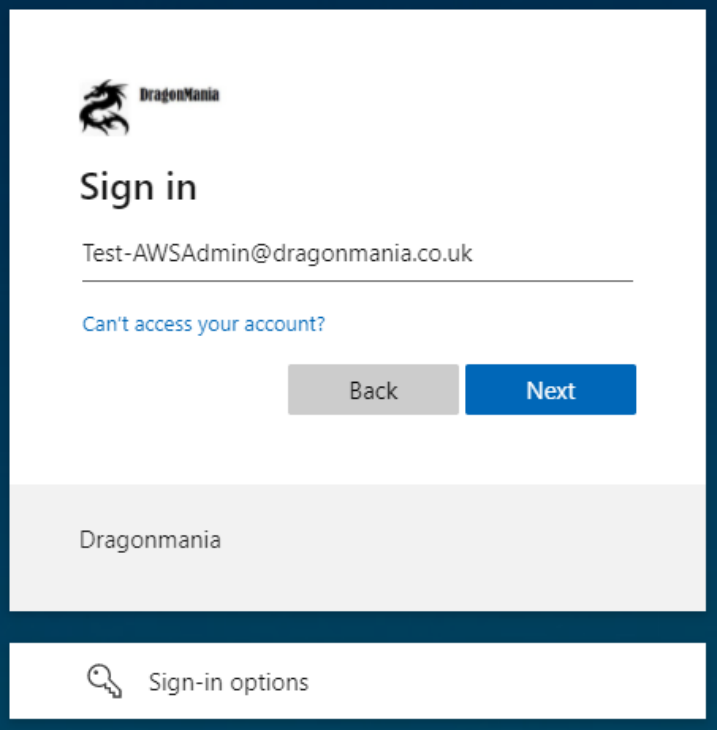
Please make sure you have configured AWS Single-Account Access before testing.

[Sign in as current user](#)

[Sign in as someone else](#)

(requires browser extension)

- 12.6 Sign in with the **Test-AWSAdmin** account you have created in section 9. If it's the first time you are signing in with this account, you will be prompted to change the password (and depending on your security settings to register for MFA). **Make a note of the new password.**



DragonMania


## Sign in

Test-AWSAdmin@dragonmania.co.uk

[Can't access your account?](#)

[Back](#) [Next](#)

Dragonmania

 Sign-in options

- 12.7 You will receive the a response as shown below that your sign in was successful. Any errors will be displayed in this page.

\*If you haven't given the user account a first name, last name and an email address, then you may see a message such as **"Sign in succeed, but 3 claims weren't issued in the token"**. This is because these attributes are listed in **Single Sign-on -> User Attributes & Claims**.

## Test single sign-on with LAB1-AWS Single-Account Access



Got feedback?

Please make sure you have configured LAB1-AWS Single-Account Access before testing.

Sign in as current user

Sign in as someone else

**i** Sign-in succeeded, but 3 claims weren't issued in the token. This can happen when there's no value stored in the directory for the attribute. Please make sure the user Test-AWSAdmin@dragonmania.co.uk has a value stored in the directory for the missing claims.

- [Download the SAML request](#)
- [Download the SAML response](#)

✓ User unique identifier (NameID)

✓ Token Claims

✓ Token signing certificate

For more information see: [I can complete Azure AD sign in, but I'm seeing an error on the application's sign in page](#)

12.8 Alternatively, you can click on **Properties** and copy the **User Access URL** as shown below and paste it in another browser window.

**LAB1-AWS Single-Account Access | Properties** ...

Enterprise Application

« Save Discard Delete Got feedback?

Overview

Deployment Plan

Manage

**Properties**

Owners

Roles and administrators (Preview)

Users and groups

Single sign-on

Provisioning

Self-service

Security

Conditional Access

Permissions

Token encryption

Activity

Sign-in logs

Usage & insights

Audit logs

Provisioning logs

Access reviews

Enabled for users to sign-in? ☒ Yes ☐ No

Name \*  ✓

Homepage URL

Logo

User access URL

Application ID

Object ID

Terms of Service Url

Privacy Statement Url

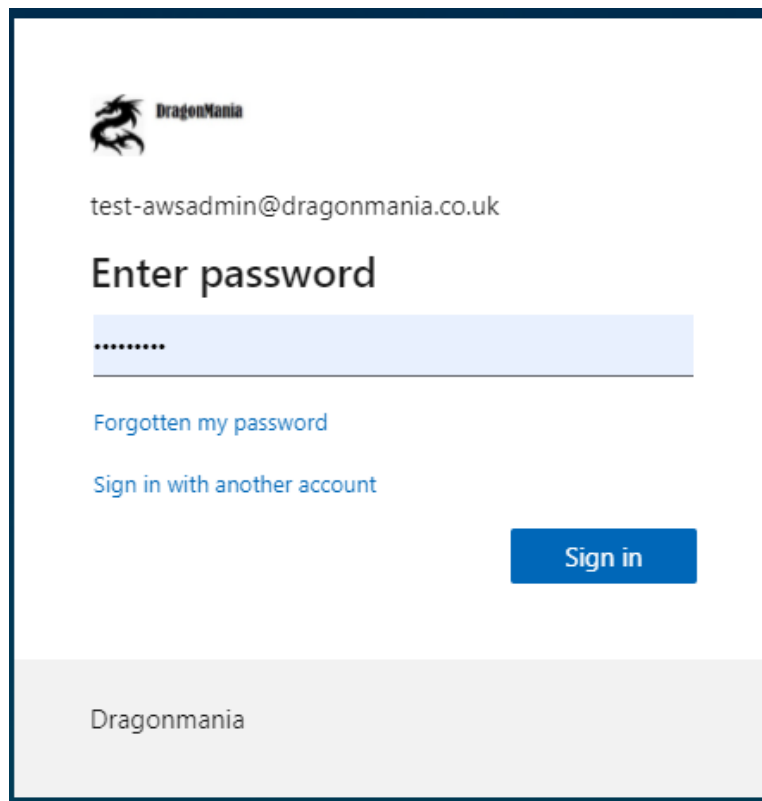
Reply URL

User assignment required? ☒ Yes ☐ No

Visible to users? ☒ Yes ☐ No

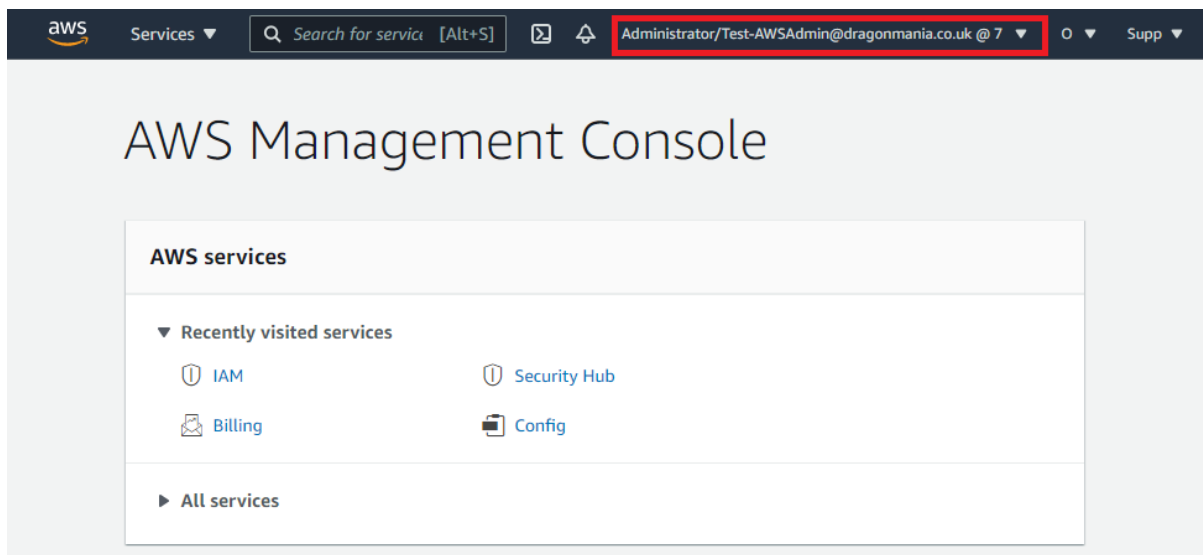
Notes  ✓

12.9 You will be prompted to sign-in, select the **Test-AWSAdmin** account you created earlier.

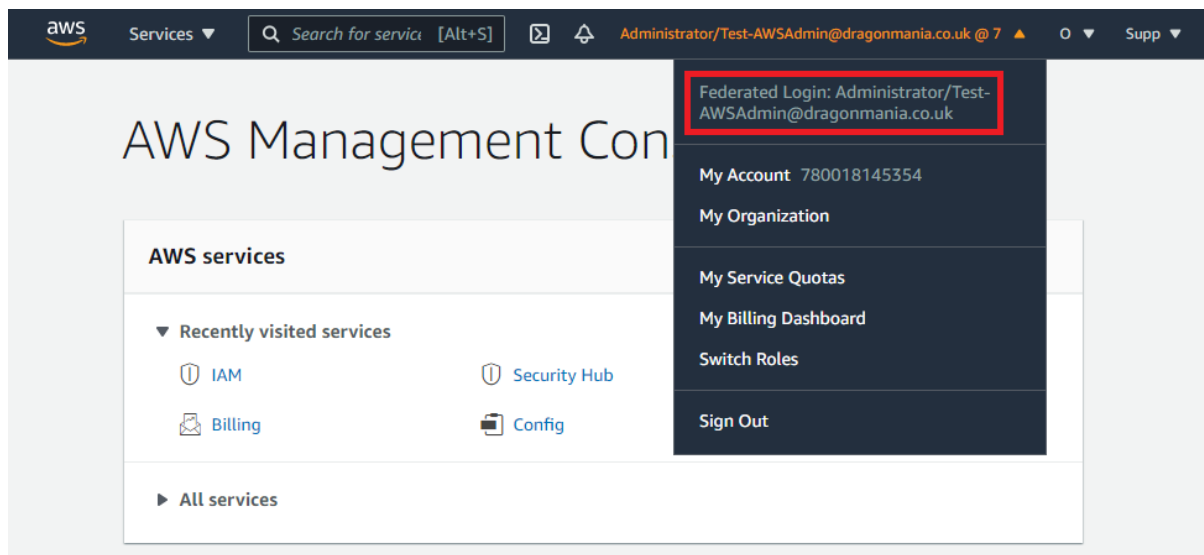


The image shows a login page for DragonMania. At the top left is the DragonMania logo. Below it is the email address 'test-awsadmin@dragonmania.co.uk'. The main heading is 'Enter password'. Below this is a password input field with a masked password '\*\*\*\*\*'. There are two links: 'Forgotten my password' and 'Sign in with another account'. A blue 'Sign in' button is on the right. At the bottom, the word 'Dragonmania' is displayed.

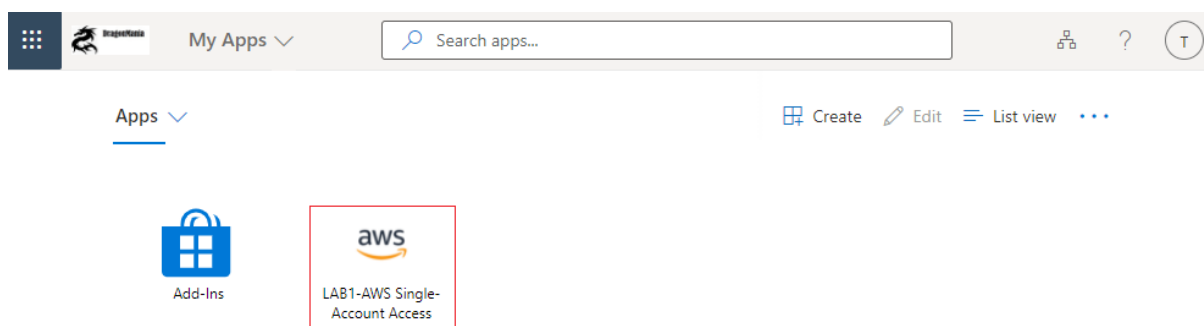
12.10 Once the sign-in is successful, you will be redirected to AWS Console as an Administrator.



12.11 If you select the arrow next to the user name to expand it, you will see the user is a federated user. This account is a full admin in AWS according to the role assignments configured earlier in this document, so play around and see what you can do (for example, create a new user, create a S3 bucket, etc.)



12.12 Another way of access the application is by going to **myapplications.microsoft.com** and signing in with the **Test-AWSAdmin** account, then select your AWS application from the list as shown below.



12.13 Make sure you also test with the **Test-AWSDeveloper** account you created and test the permissions you have assigned to the **Developer** role.

***Congratulations! You have now successfully completed this lab. We hope you found this lab, and the associated lab materials useful. We look forward to seeing what you build as a result of attending this lab!***

Lab Complete.