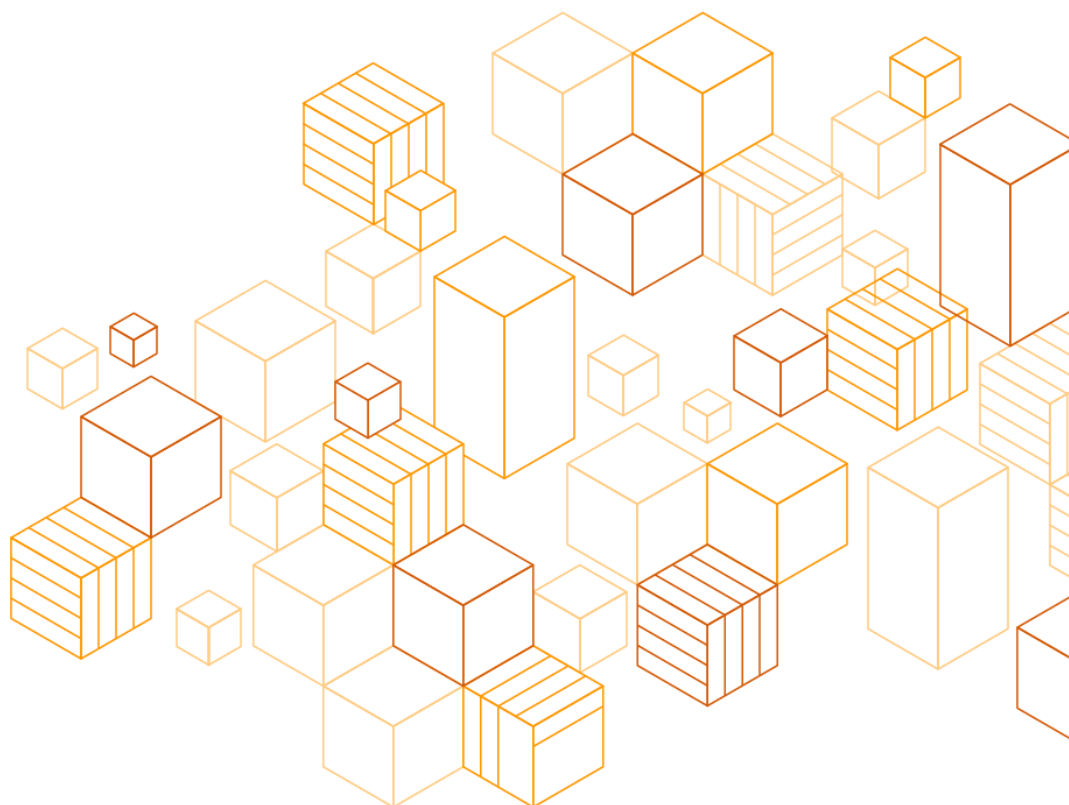


# Securing Remote Access with Multi-Factor Authentication

**Using AWS Systems Manager Session Manager and AWS Single Sign-On (AWS SSO)**

Published March 1, 2021



## Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

© 2021 Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Contents

Overview .....	1
About AWS Systems Manager Session Manager .....	2
About AWS Single Sign-On (AWS SSO) .....	3
Before you begin .....	3
Architectural overview .....	4
Costs .....	6
Walkthrough.....	6
Set up AWS Single Sign-on.....	6
Set up AWS Systems Manager Session Manager .....	32
Test your configuration.....	45
(Optional) Configure Session Manager to manage on-premises servers .....	53
Clean up .....	58
Conclusion .....	59
Contributors .....	59
Additional Resources .....	59
Document Revisions.....	60

## About this Guide

Many customers rely on a bastion host in a public-facing subnet to access and manage servers inside an Amazon Virtual Private Cloud (Amazon VPC) network. With this approach, customers are required to open the management port to the internet which introduces security risks.

This guide provides instructions to set up secured server remote access sessions with multi-factor authentication (MFA) using AWS Systems Manager Session Manager and AWS Single Sign-On. This architecture helps to eliminate the risks and reduce attack surfaces associated with the bastion host approach.

The intended audiences for this guide are system administrators, security architects, and solution architects.

## Overview

Typically, customers use a bastion host in a public-facing subnet to access and manage servers inside an Amazon Virtual Private Cloud (Amazon VPC) network. This case is especially used when production servers are located in the private subnet without direct access to the internet.

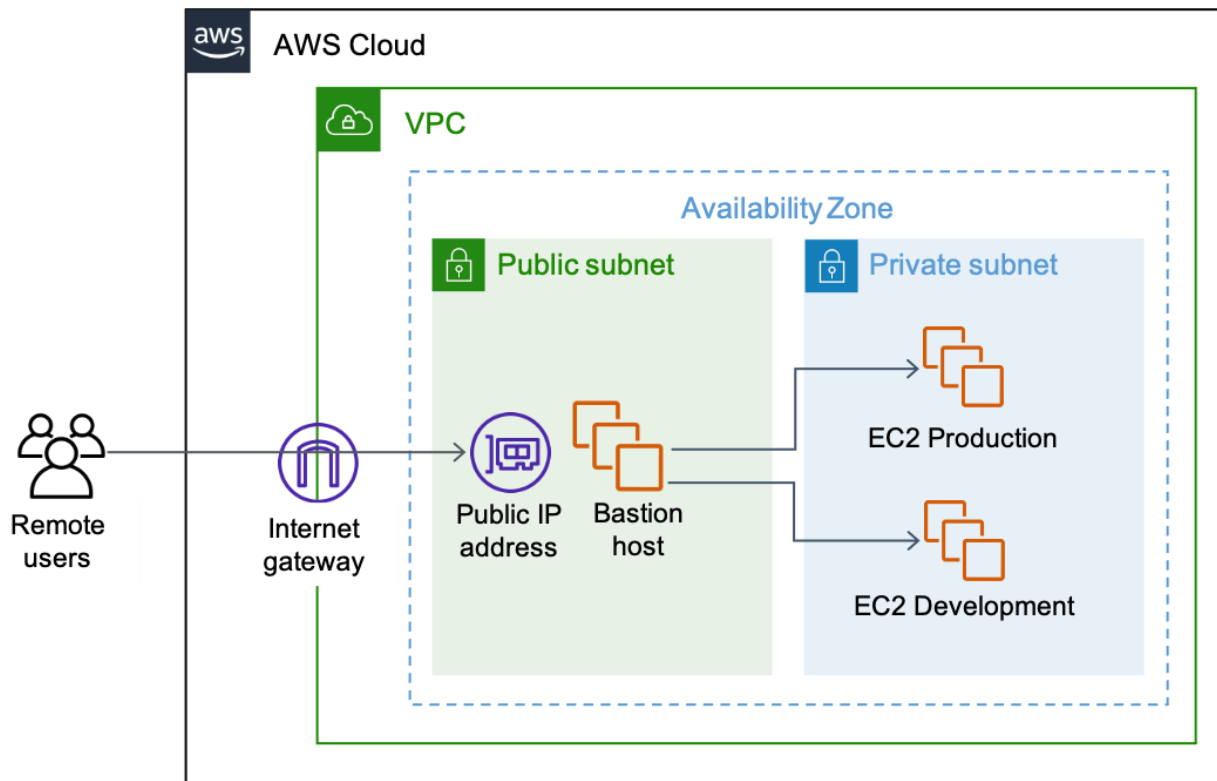


Figure 1: Typical Bastion host setup to access and manage servers inside an Amazon VPC

With the Bastion host approach, there are several challenges:

- A public security group allows access from 0.0.0.0. (internet) to bastion host public IP port 22 (SSH). You are required to open this management port to the internet (such as port 22 for SSH or 3389 for RDP).
- You need to manage the availability of the bastion server itself.
- You need to manage your users' access and permission to access the bastion server itself.
- No multi-factor authentication (MFA) protection exists for the bastion host in the public subnet.

This guide shows how AWS helps you isolate server remote access sessions through [AWS Systems Manager Session Manager](#). AWS Systems Manager Session Manager can provide secure remote access for the user to manage their servers. You can eliminate opening the management port to the internet and preparing and managing the Bastion host.

By integrating Session Manager with [AWS Single Sign-On \(AWS SSO\)](#), you can provide centralized user management to segregate the user permission into groups of users. With AWS SSO, you can also control the authorization based on tagging and user group (e.g., Dev and Prod server access separation). AWS SSO also enables you to increase security by enabling multi-factor authentication (MFA) with authenticator applications.

In this guide, you complete the following steps:

1. Set up AWS Single Sign-on
1. Set up AWS Systems Manager
2. Test your configuration
3. (Optional) Configure AWS Systems Manager Session Manager to manage on-premises servers
4. Clean up

## About AWS Systems Manager Session Manager

As a fully managed service, Systems Manager Session Manager can help you access and manage EC2 instances and/or on-premises servers through a browser-based shell or [AWS Command Line Interface \(AWS CLI\)](#). Session Manager eliminates the need to maintain bastion hosts open inbound ports or SSH keys. Additional benefits of Session Manager include:

- Centralized access control using AWS Identity and Access Management (IAM) policies
- One-click access to the instance through the [AWS Management Console](#) or AWS CLI
- Port forwarding capability
- Cross-platform support for both Windows and Linux

- Logging and auditing session activity. Session Manager can be integrated with [AWS CloudTrail](#), [AWS CloudWatch Logs](#), [Amazon Simple Storage Service \(Amazon S3\)](#), and the combination of [Amazon CloudWatch Events](#) and [Amazon Simple Notification Service](#) (Amazon SNS).

This document walks you through integrating AWS Systems Manager Session Manager with AWS SSO to enable centralized sign-in with multi-factor (MFA) authentication capability.

## About AWS Single Sign-On (AWS SSO)

AWS Single Sign-On can manage centralized access to multiple AWS accounts or numerous business applications; while providing a seamless single sign-on experience for all the users. AWS SSO can be integrated with Identity Providers such as Microsoft Active Directory (both [AWS Managed Microsoft AD Directory](#) or on-premises AD server), Okta Universal Directory, Azure Active Directory (Azure AD), or [another supported IdP](#).

AWS SSO can be enabled in the [AWS Organization](#), which can consolidate multiple AWS accounts under the same management domain. AWS SSO has built Multi-Factor Authentication capability. AWS SSO is available at no additional cost.

## Before you begin

To follow along with this guide, make sure you have the following prerequisites in place:

- Active AWS Account (a stand-alone account that has not been managed under the existing AWS Organization). See [How do I create and activate a new AWS account?](#) for detail steps.
- IAM user with **AdministratorAccess** policy. All of the configuration steps demonstrated in this guide are completed using this IAM user unless stated otherwise. See [Creating an administrator IAM user and group \(console\)](#) for detail steps.
- Two Amazon EC2 instances (Amazon Linux 2) in a private subnet. Ensure both instances have internet connectivity (NAT gateway, located in the public subnet). For instructions, see [Creating a VPC with Public and Private Subnets for Your Clusters](#) and [Get started with Amazon EC2 Linux instances](#).

- If you use a different OS version other than Amazon Linux, Amazon Linux 2, or Ubuntu Linux (version 16.04 or 18.04), make sure that you have SSM Agent installed and running. For details, see [Installing and configuring SSM Agent on EC2 instances for Linux](#).

For this guide, add the following tags to your Amazon EC2 instances:

*Table 1 - EC2 Required Tags*

Instance	Key	Value
Instance 1	Name	dev-server01
	project	Development
Instance 2	Name	prod-server01
	project	Production

- Create an Amazon CloudWatch Log Group to store Session Manager logging. In this guide, the Log Group is named **SessionManagerLogGroup**.
- Create an Amazon S3 bucket to store Session Manager logging. In this guide, the S3 bucket is named **session-manager-log-bucket123**. (Your S3 bucket must be unique. For more information about S3 bucket naming, see [Bucket restrictions and limitations](#) documentation.)
- A mobile phone with an authenticator application, such as Authy, as MFA device to generate authentication codes.

## Architectural overview

The following diagram illustrates the architecture you set up in this guide. For descriptions, see the following table.



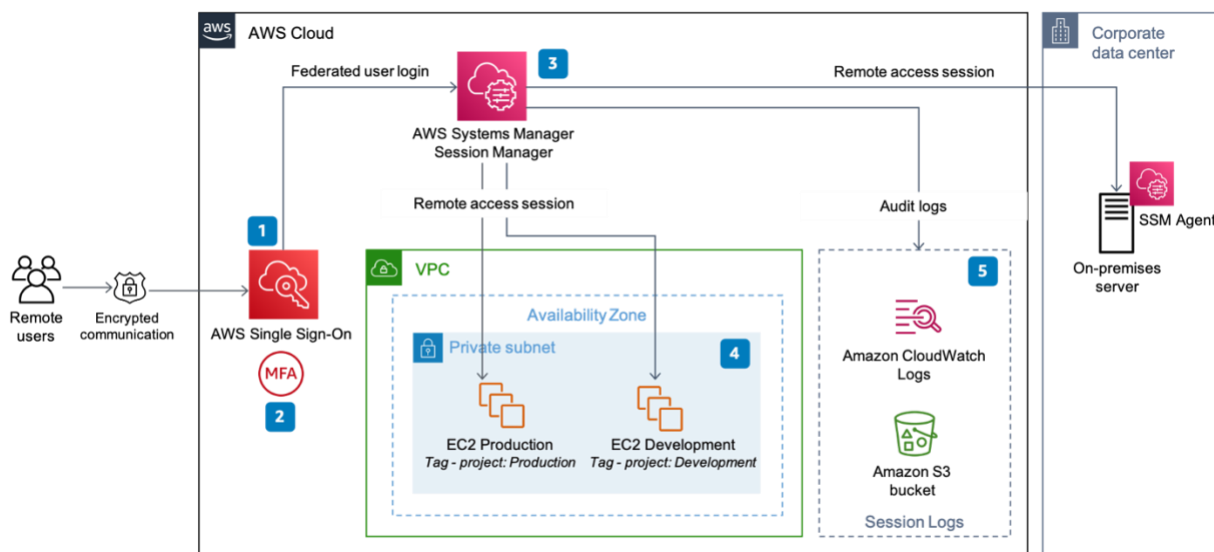


Figure 2 - Architecture diagram for integrating AWS SSO with AWS SSM Session Manager

See following callouts describe this architecture:

Callout	Description
1	AWS SSO is used as the Identity Provider (IdP) for centralized management of users/groups. You can use this architecture with other IdPs, such as Microsoft Active Directory.
2	User must supply the MFA token upon signing in. Users can manage MFA devices by themselves and they can use third-party MFA tokens such as Authy.
3	The users' permission is limited to access Session Manager only; no other AWS services are accessible from within the console (exception only to list all EC2 instances). In a real-life scenario, you can modify the configuration to give access to more AWS services.
4	Some users have privileges to manage Production servers, whereas others only have access Development servers. Access segregation is based on resource tags. This guide uses project tags attached to all EC2 instances with possible Development or Production values.
5	Session Manager stores all the session logs in Amazon CloudWatch Logs and a designated Amazon S3 bucket.

## Costs

The total cost of setting up remote access with AWS Systems Manager and AWS SSO varies depending on your needs and configuration, in particular these factors:

- Size of logs generated from each of your user sessions stored in S3 bucket and CloudWatch Logs.
- Number of on-premises servers you manage as explained in [\(Optional\) Configure Session Manager to manage on premises servers](#).
- Number and duration of configuration tests

If you use the resources described in the implementation guide for 2 hours, your cost will be less than \$1. This excludes the cost of configuring an on-premises server. This estimate assumes that you generate less than 10 KB of session logs in an Amazon S3 bucket and send less than 10 KB of data to Amazon CloudWatch Logs through Session Manager testing.

## Walkthrough

In the following sections, you set up AWS SSO and AWS Systems Manager Session Manager, then test your configuration.

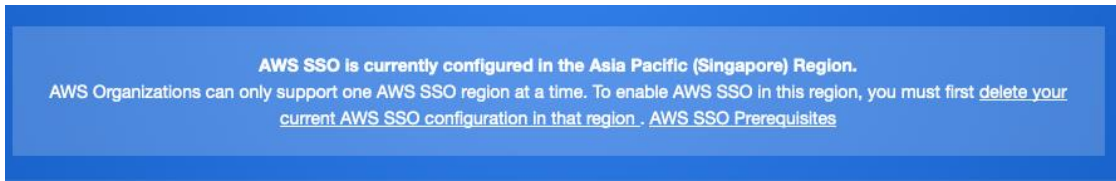
### Set up AWS Single Sign-on

#### Enable AWS Single Sign-On

**Note:** AWS SSO is available in select AWS Regions. To check Region availability, see the [AWS Regional Services List](#).

If your AWS account is a member of an existing AWS Organization, you cannot enable AWS Single Sign-On. This scenario is only applicable for the stand-alone account or the management account of an existing AWS Organization.

You can enable AWS Single Sign-On in one Region only. The following message appears when you try to enable AWS SSO in multiple Regions.



*Figure 3 - Notification when you try to enable AWS SSO in multiple Regions*

Complete the following steps to enable AWS SSO:

1. Open the [AWS SSO console](#) and choose **Enable AWS SSO**.



*Figure 4 - AWS Single Sign-On home page*

2. Choose **Create AWS organization**.

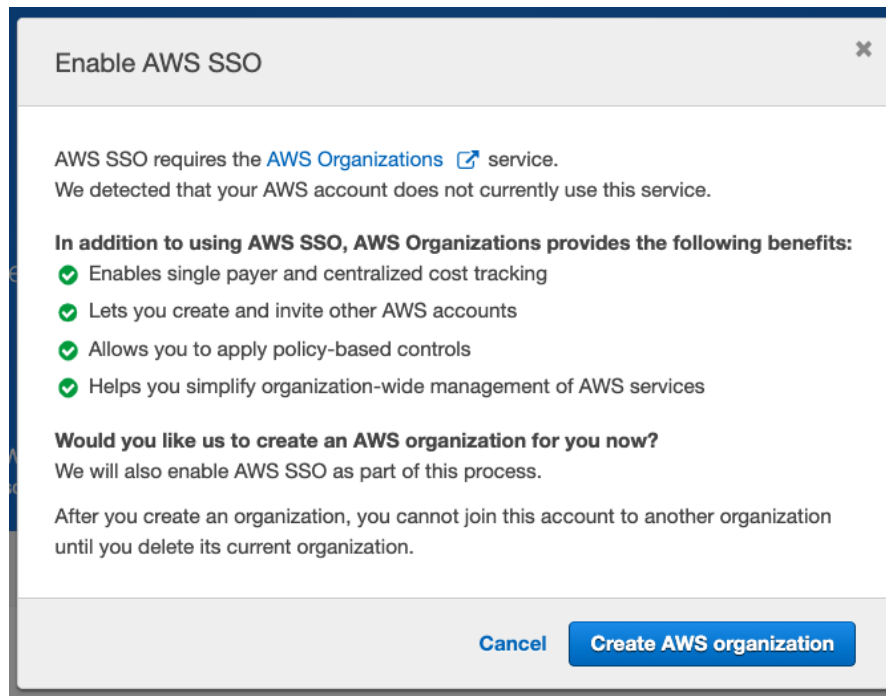


Figure 5 - Menu to create AWS Organization as prerequisite to enable AWS SSO

The **Welcome to AWS Single-Sign On** page appears once AWS SSO is enabled. At this point, you have fully working AWS Single Sign-On with default AWS SSO as an identity source. The **User portal** section of the page displays your user portal URL. Keep note of this URL as you will need it during testing.

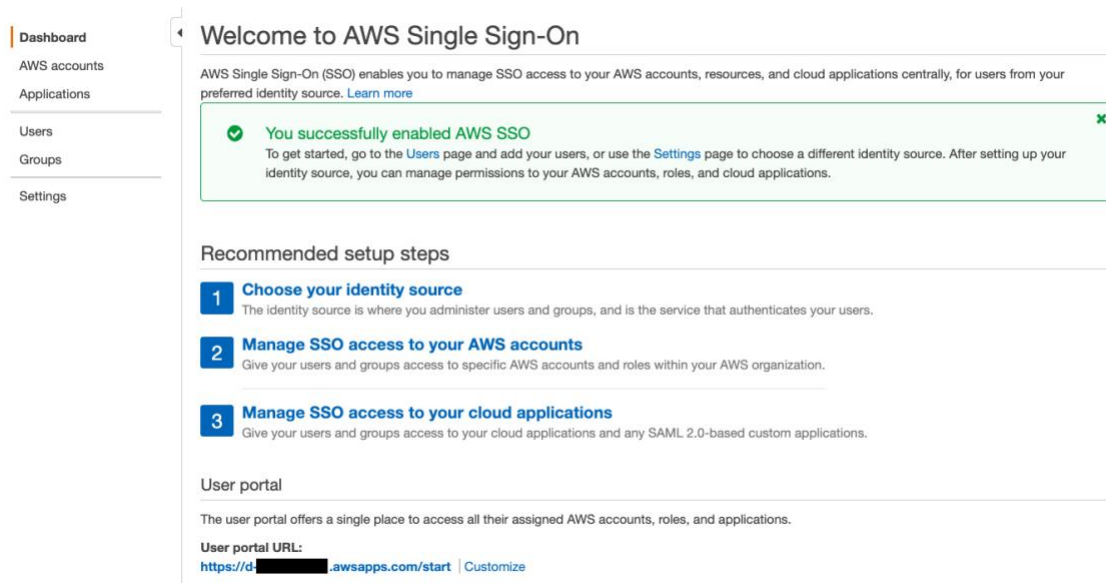


Figure 6 - Notification when AWS SSO has been enabled successfully

Optionally, choose **Customize** to customize the user portal URL.

**Note:** You can customize the URL once. After you have customized the URL, you cannot change it again.

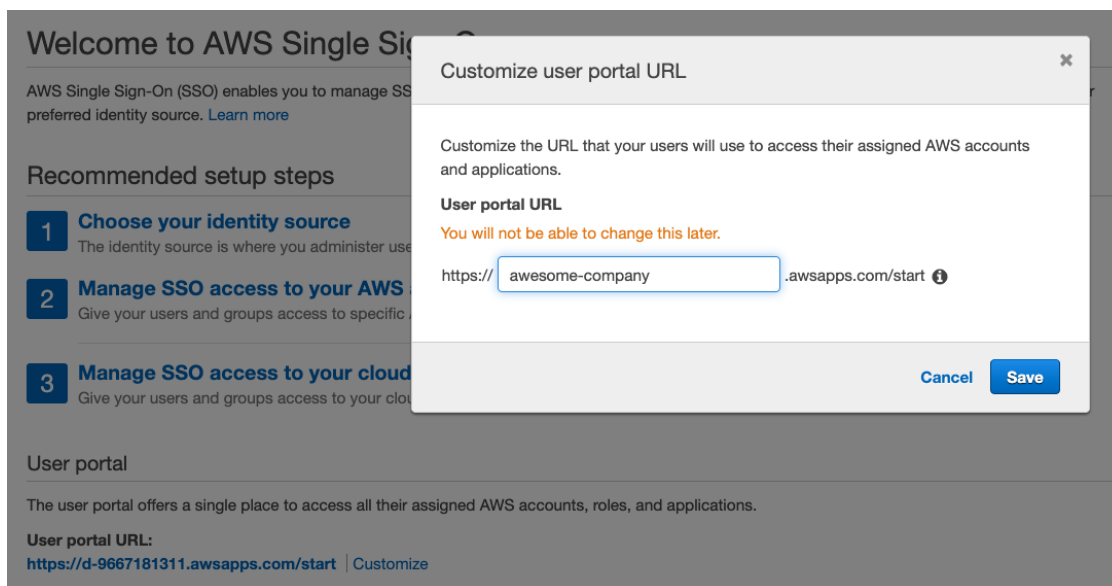


Figure 7 - Option to customize AWS SSO user portal URL

## Create the AWS SSO groups and users

In this step, you create the following two user groups, each with one active user:

Table 2 - AWS SSO groups and user

SSO Group	SSO User
ProductionAccess	testuser01
DevelopmentAccess	testuser02

To create the user group, complete the following steps:

1. In the left navigation pane of the [AWS SSO console](#), choose **Groups**, then choose **Create group**.

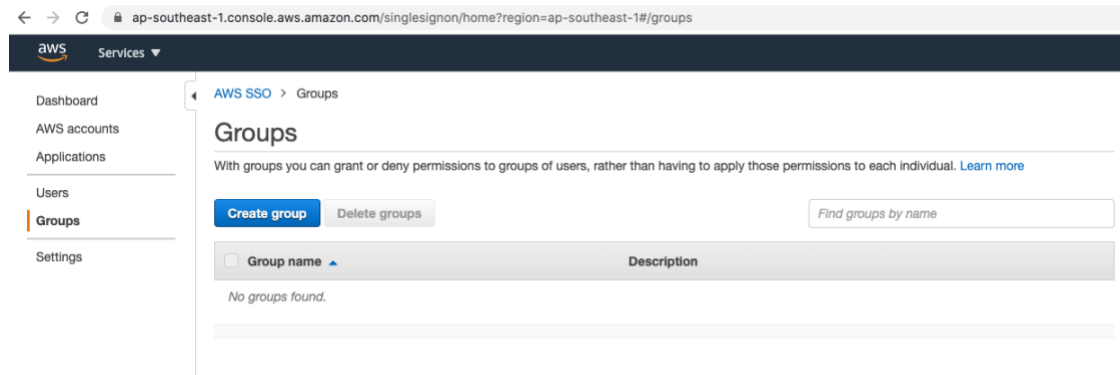


Figure 8 - Groups menu on AWS SSO console

2. In the Create group dialog box, do the following:
  - a. For **Group name**, type *ProductionAccess*.
  - b. For **Description**, type *This user group can manage all production servers*.

The screenshot shows the 'Create group' dialog box. It has a title bar with 'Create group' and a close button. The main area contains two fields: 'Group name\*' and 'Description'. The 'Group name\*' field has the text 'ProductionAccess' entered. Below it, a note states: 'Can contain only alphanumeric characters, or any of the following: \_ - Maximum of 128 characters'. The 'Description' field has the text 'This is user group that can manage all production servers' entered. Below it, a note states: 'Can contain only alphanumeric characters, or any of the following: \_ - Maximum of 256 characters'. At the bottom, there is a legend '\* Required fields', a 'Cancel' button, and a 'Create' button.

Figure 9 - Group creation dialog box

3. Choose **Create**
4. Choose **Create group** and in the **Create group** dialog box, specify these details to add your second group:
  - a. For **Group name**, type *DevelopmentAccess*.
  - b. For **Description**, type *This user group can manage all development servers*.

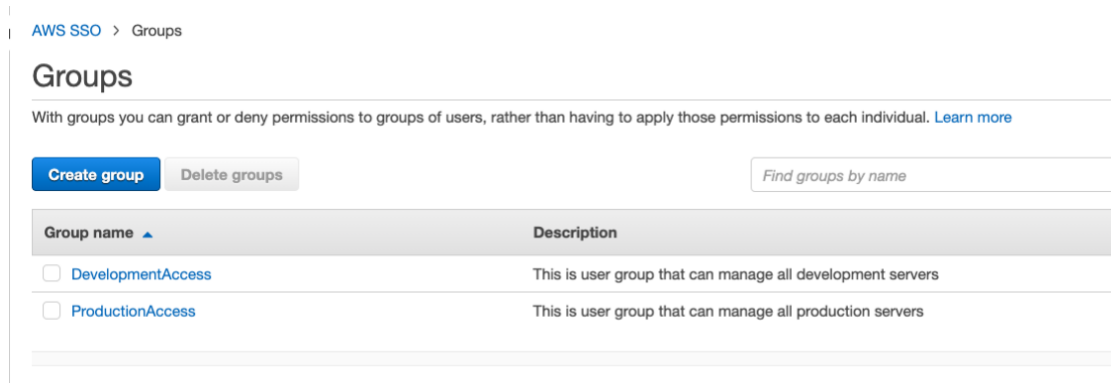


Figure 10 - AWS SSO Groups page showing newly created groups

Next, create the users for your groups.

5. In the left navigation pane of the [AWS SSO console](#), choose **Users**, then choose **Add user**.

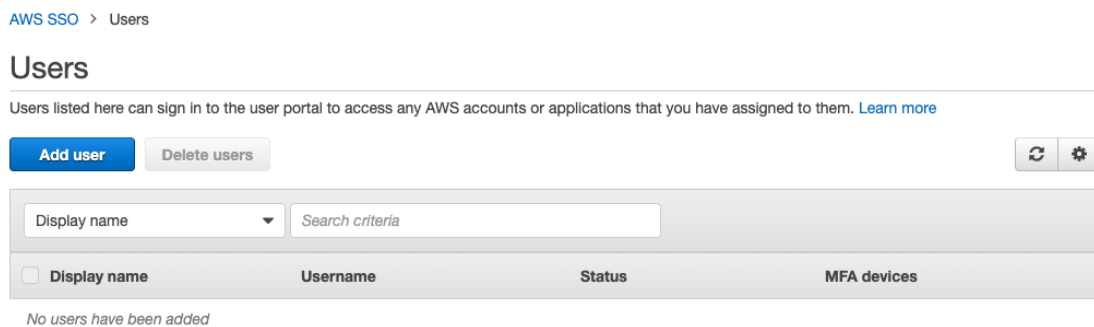


Figure 11 – Add user in AWS SSO console

6. On the **Add user** page, for **User details**, specify these details:
  - a. For **Username**, type *testuser01*.
  - b. For **Password**, choose **Send an email to the user with password setup instructions**.
  - c. For **Email address**, type a valid email address.
  - d. For **First name**, type *Test*.
  - e. For **Last name**, type *User01*.
  - f. For **Display name**, type *Test User01*.

The screenshot shows the 'Add user' form in the AWS IAM console. At the top, there are two steps: '1 Details' (active) and '2 Groups'. The 'User details' section contains the following fields:

- Username\***: testuser01 (Note: This username will be required to sign in to the user portal. This cannot be changed later.)
- Password**: ☒ Send an email to the user with password setup instructions. [Learn more](#) ☐ Generate a one-time password that you can share with the user. [Learn more](#)
- Email address\***: testuser01@example.com
- Confirm email address\***: testuser01@example.com
- First name\***: Test
- Last name\***: User01
- Display name\***: Test User01

Below these fields are three expandable sections: 'Contact methods (optional)', 'Job-related information (optional)', and 'Address (optional)'. At the bottom right, there are 'Cancel' and 'Next: Groups' buttons.

Figure 12 - User details for first user

7. Choose Next: Groups.
8. Select the **ProductionAccess** group and choose **Add user**.

The screenshot shows the 'Add user to groups' form in the AWS IAM console. At the top, there are two steps: '1 Details' and '2 Groups' (active). The 'Add user to groups' section contains the following elements:

- A 'Create group' button.
- A search bar labeled 'Find by group name'.
- A table with the following data:

Name	Description
<input checked="" type="checkbox"/> ProductionAccess	This is user group that can manage all production servers

At the bottom right, there are 'Cancel', 'Previous', and 'Add user' buttons.

Figure 13 - User to group assignment option



9. On the **Users** page, choose **Add user** again to add your second user.
10. On the **Add user** page, for **User details**, specify these details:
  - a. For **Username**, type *testuser02*.
  - b. For **Password**, choose **Send an email to the user with password setup instructions**.
  - c. For **Email address**, type a valid email address.
  - d. For **First name**, type *Test*
  - e. For **Last name**, type *User02*.
  - f. For **Display name**, type *Test User02*.
11. Choose Next: Groups.
12. Select the **DevelopmentAccess** group and choose **Add user**.

On the **Users** page, you should now see two users listed.

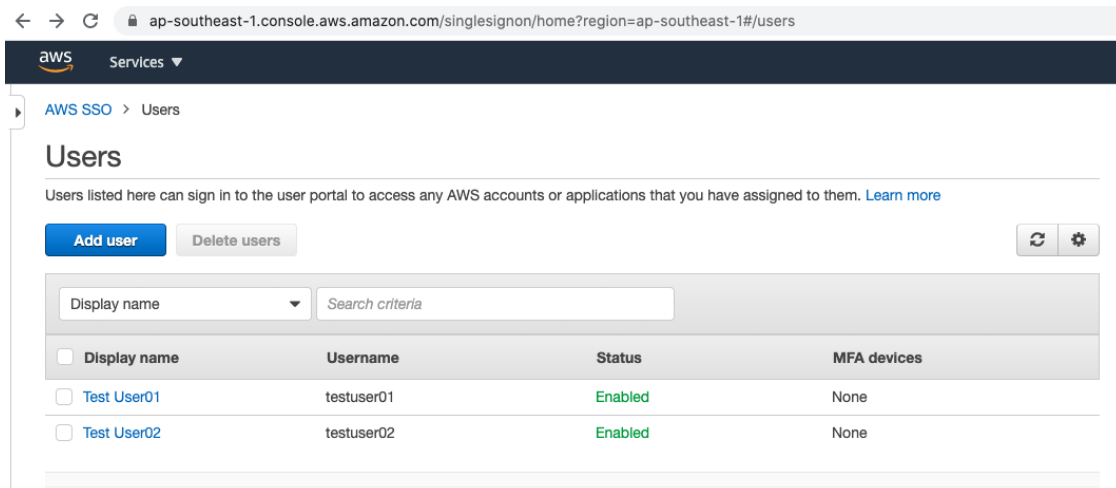


Figure 14 - AWS SSO Users menu with 2 newly created users

Because you choose to send the password setup instruction, each user must complete the user registration sent by AWS Single Sign-On through email.

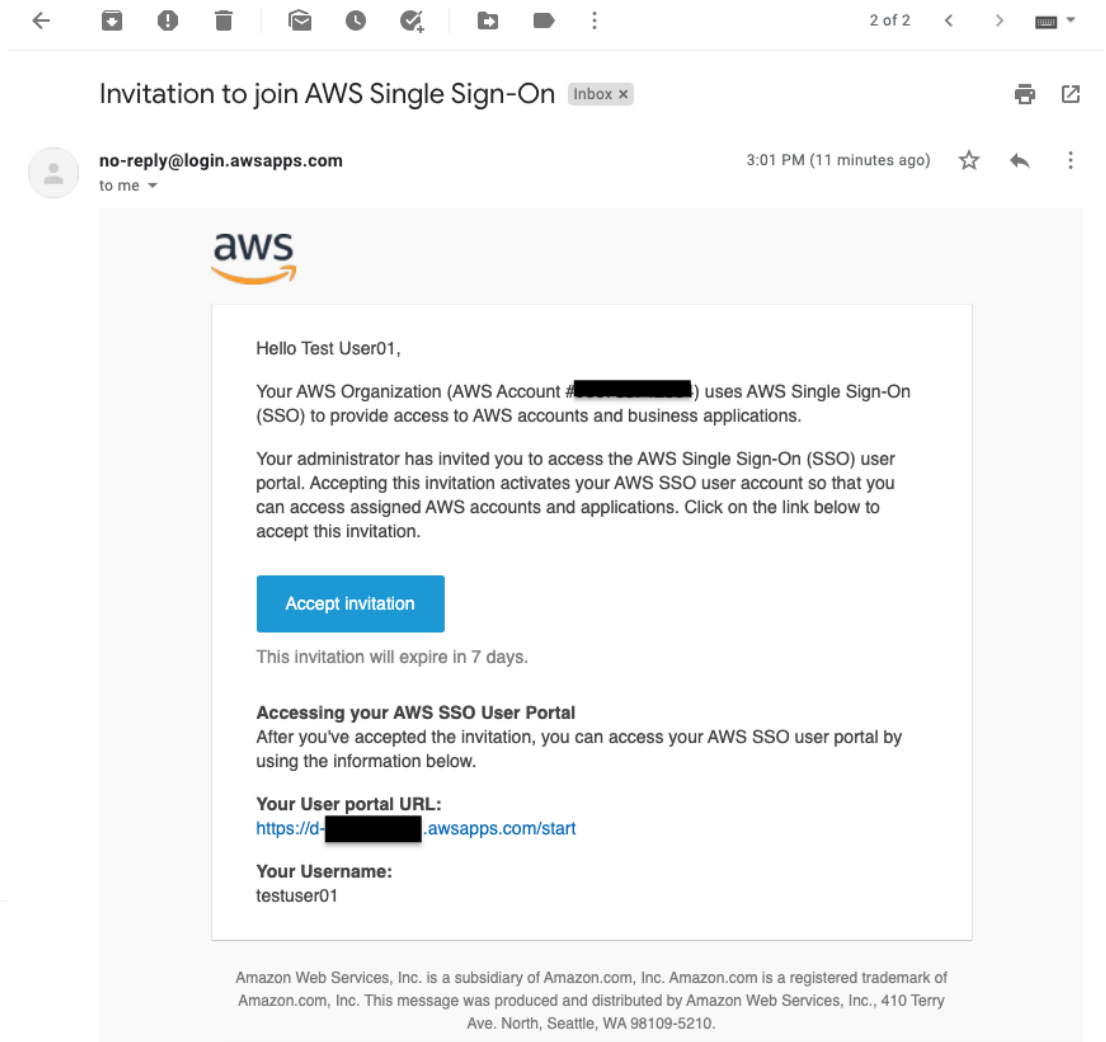


Figure 15 - Example invitation email to join AWS Single Sign-On

Choose **Accept invitation** to activate the AWS SSO user portal account. As part of the activation process, you are prompted to set a new password and then log in to the User Portal. At this point, the AWS SSO User Port does not display any accounts or services. In the next section, you grant AWS SSO access to your AWS account.

## Grant access to AWS account

Using AWS SSO, you can grant access to the users to access and manage all AWS accounts under the same AWS Organization. In this guide, you only use one active account. You will configure your user groups to manage this current account, specifically only to access AWS Systems Manager Session Manager.

First, create your **Permission sets**.

1. In the left navigation pane of the [AWS SSO console](#), choose **AWS accounts**.
2. Choose the Permission sets tab. Then, choose Create permission set.

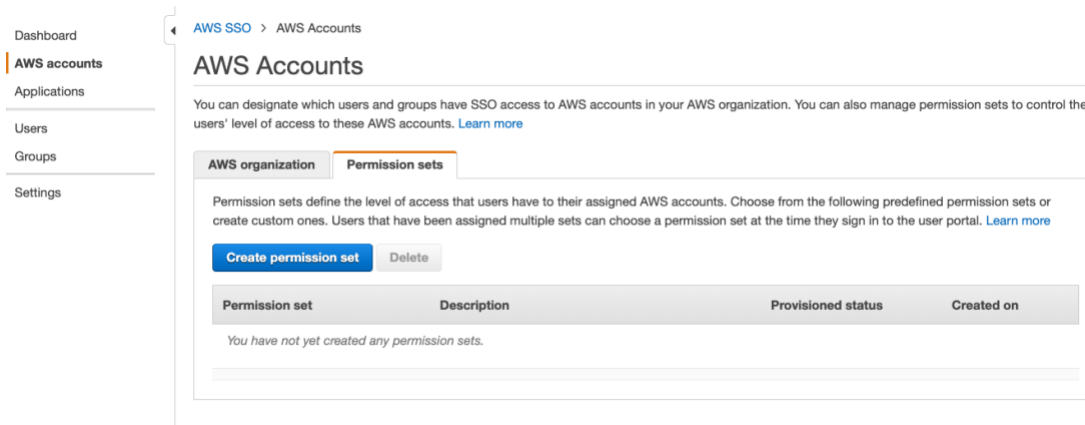


Figure 16 - Permission sets settings in the AWS Accounts menu

3. In the **Create new permission set type** page, select **Create a custom permission set** and choose **Next: Detail**.

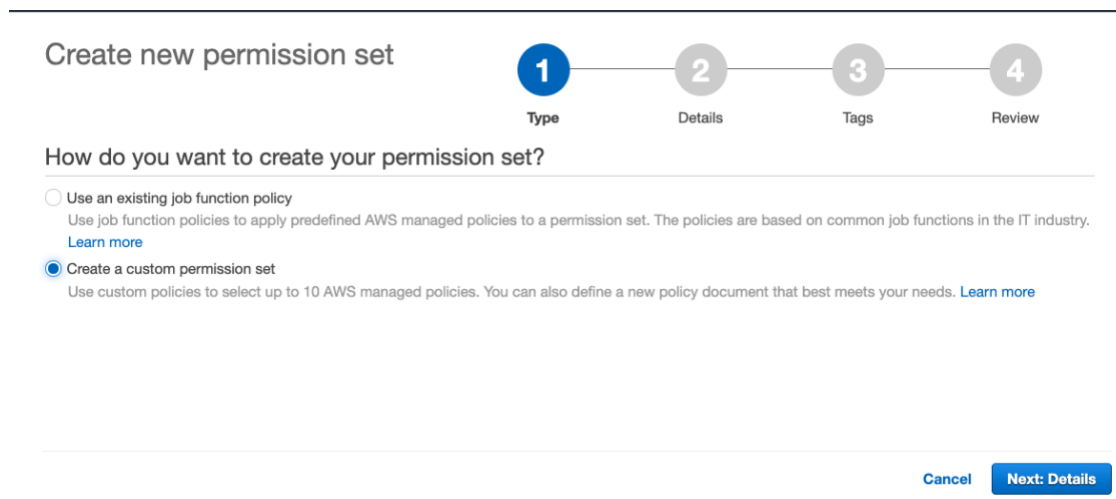
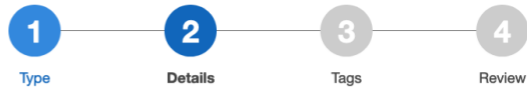


Figure 17 - First step of Permission set creation wizard with custom permission set being selected

4. On the Create new permission set Details page, do the following:
  - a. For **Name**, type *ProductionPermissionSet*
  - b. For **Description**, type *Permission set to access Session Manager limited to all instances with project tag = Production.*

- c. Keep the default options for **Session duration** and **Relay state**.
- d. Select the option for **Create a custom permissions policy**.

## Create new permission set



## Create a custom permission set

## Name

ProductionPermissionSet

Your users will see this name when they access this AWS account from the user portal. You cannot change this later.

## Description

Permission set to access Session Manager limited to all instances with project tag = Production

## Session duration

The length of time a user can be logged on before the console logs them out of their session. [Learn more](#)

1 hour

## Relay state

The value used in the federation process for redirecting users within the account. [Learn more](#)

## What policies do you want to include in your permission set?

Permission sets can contain links to AWS managed policies and custom policies. When your users sign in using this permission set, they are granted all permissions included in this set.

- ☐ Attach AWS managed policies
- ☒ Create a custom permissions policy

## Create a custom permissions policy

Paste a policy document that specifies custom permissions. This is useful for granting access to specific resources, a specific set of actions, or permissions that cannot be expressed by any combination of AWS managed policies. You can use the IAM policy simulator to test the effects of this policy before applying your changes. [Learn more](#)

```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "VisualEditor0",
6       "Effect": "Allow",
7       "Action": [
8         "ssm:DescribeSessions",
9         "ssm:DescribeInstanceProperties",
10        "ec2:DescribeInstances",
11        "ssm:GetConnectionStatus"
12      ],
13      "Resource": "*"
14    },
15    {
16      "Effect": "Allow",
  
```

Cancel

Previous

Next: Tags

Figure 18 - Second step of Permission set creation wizard to define name, description, session duration, and custom permission policy

- e. In the **Create a custom permission policy** editor, copy and paste the following code:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "ssm:DescribeSessions",
        "ssm:DescribeInstanceProperties",
        "ec2:DescribeInstances",
        "ssm:GetConnectionStatus"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:StartSession"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:instance/*"
      ],
      "Condition": {
        "StringLike": {
          "ssm:resourceTag/project": [
            "Production"
          ]
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:TerminateSession"
      ],
      "Resource": [
        "arn:aws:ssm:*:*:session/${aws:username}-*"
      ]
    }
  ]
}
```

5. Choose **Next: Tags**.
6. On the **Tags** page, create the following tag:
  - a. For Key, type *Name*.
  - b. For Value, type *ProductionAccess*.
7. Choose **Next: Review**

Create new permission set

1 Type 2 Details 3 Tags 4 Review

Add tags (optional)

Tags are key-value pairs that you can associate with your permission set. You can use tags to organize, track, or control access. Tag keys and values are case sensitive. [Learn more](#)

Key	Value (optional)	Remove
Name	ProductionAccess	x

Add new key Add new value

You can add 49 more tags.

Cancel Previous Next: Review

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates

Figure 19 - Third step of Permission set creation wizard to define Name tag

8. On the Review page, verify your selections and choose **Create**.

Create new permission set

1 Type 2 Details 3 Tags 4 Review

Review

Review your choices. After you create this permission set, you can view and edit the associated policies as needed.

Permission set details

Name	ProductionPermissionSet
Description	Permission set to access Session Manager limited to all instances with project tag = Production
Session duration	1 hour
Relay state	Not provided

Permissions policy

```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "VisualEditor0",
6       "Effect": "Allow",
7       "Action": [
8         "ssm:DescribeSessions",
9         "ssm:DescribeInstanceProperties",
10        "ec2:DescribeInstances",
11        "ssm:GetConnectionStatus"
12      ],
13       "Resource": "*"
14     }
15   ]
16 }

```

Cancel Previous Create

Figure 20 - Final step of Permission set creation wizard for review purpose

9. Repeat Step 1 through Step 8 to add a second permission set with the following attributes:

- **Name:** *DevelopmentPermissionSet*
- **Description:** *Permission set to access Session Manager limited to all instances with project tag = Development*
- **Custom Permission Policy:**

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "ssm:DescribeSessions",
        "ssm:DescribeInstanceProperties",
        "ec2:DescribeInstances",
        "ssm:GetConnectionStatus"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:StartSession"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:instance/*"
      ],
      "Condition": {
        "StringLike": {
          "ssm:resourceTag/project": [
            "Development"
          ]
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:TerminateSession"
      ],
      "Resource": [
        "arn:aws:ssm:*:*:session/${aws:username}-*"
      ]
    }
  ]
}
```

- Tags: Key=*Name* and Value=*DevelopmentAccess*.

On the **Permissions sets** tab of your AWS Accounts page, you should see two permissions sets: **ProductionPermissionSet** and **DevelopmentPermissionsSet**.

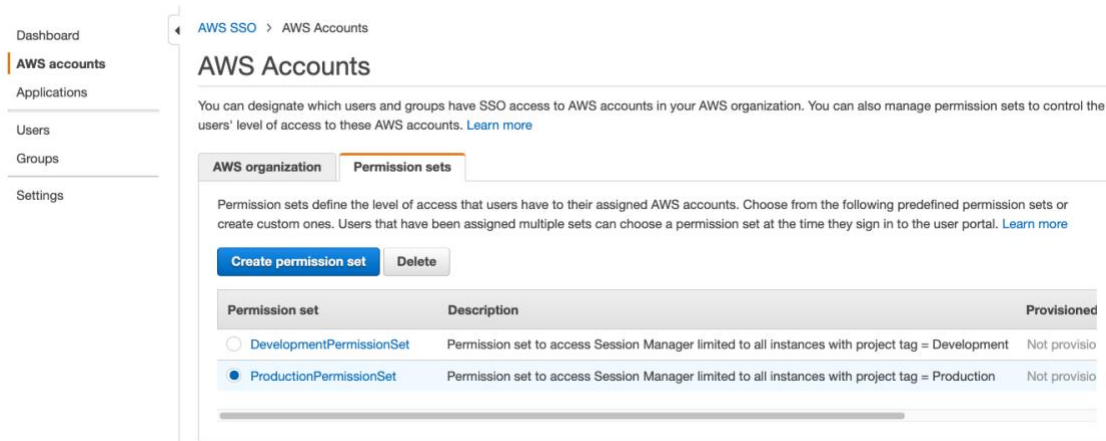


Figure 21 - AWS Accounts menu with 2 newly created Permission sets

**Note:** The two permission sets you just created only allow users to access EC2 instances. If you plan to provide remote access to on-premises servers, see the *(Optional) Configuring Systems Manager Session Manager to manage on-premises servers* section for steps.

After creating two **Permission sets**, now complete the following steps to grant your AWS Account access to your user group with the appropriate **permission sets**.

1. In the left navigation pane of the [AWS SSO console](#), choose **AWS accounts** and choose your account.

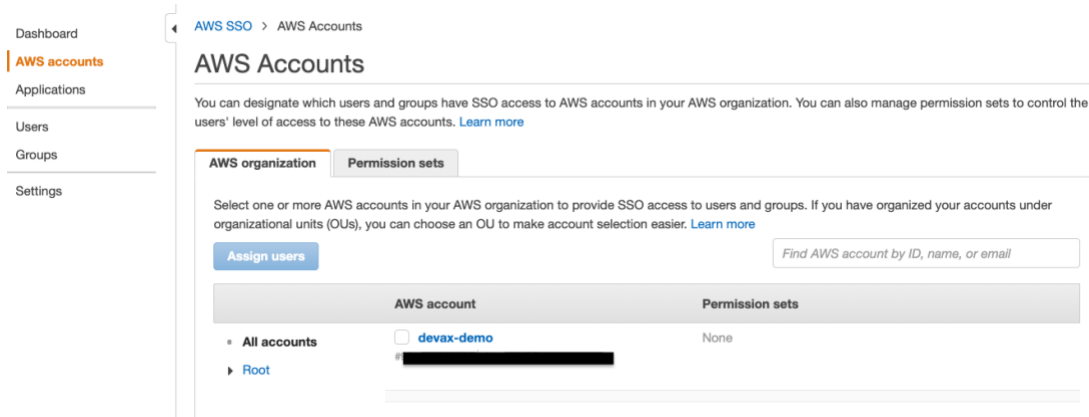


Figure 22 - AWS accounts menu on AWS SSO console



2. On the account detail page, choose **Assign users**.

Dashboard

**AWS accounts**

Applications

Users

Groups

Settings

◀ AWS SSO > AWS Accounts > devax-demo

## devax-demo

### Details

Account name	devax-demo
Account ID	[REDACTED]
Email	[REDACTED]

### Assigned users and groups

The following users or groups can access this AWS account from their user portal. [Learn more](#)

**Assign users**

User/group	Permission sets
You have not yet assigned any users or groups to this account.	

▶ Permission sets

▶ IAM identity provider

Figure 23 - Menu to assign users inside AWS account detail

3. On the **Assign Users** page, choose the **Groups** tab, and then select the **ProductionAccess** group. Choose **Next: Permission sets**.

## Assign Users

1 **Users and groups**
2 **Permission sets**

### Select users or groups

You can search for the users and groups that you want to give SSO access to.

**Users** **Groups**
1 group selected | [Deselect](#)

Select one or more groups to assign to this AWS account. Find groups by name

Group name ▲	Description
<input type="checkbox"/> DevelopmentAccess	This is user group that can manage all development servers
<input checked="" type="checkbox"/> ProductionAccess	This is user group that can manage all production servers

Cancel
Next: Permission sets

e?region=ap-southeast-1#
© 2008 - 2020, Amazon Web Services, Inc. or its affili

Figure 24 - Groups selection that you want to give SSO access to

4. On the **Permissions sets** page, select the **ProductionPermissionSet** and choose **Finish**.

## Assign Users

1 **Users and groups**
2 **Permission sets**

### Select permission sets

Permission sets define the level of access that users and groups have to an AWS account. Permission sets are stored in AWS SSO and appear in the AWS account as IAM roles. You can assign more than one permission set to a user. To ensure least privilege access to AWS accounts, users with multiple permission sets on an AWS account must pick a specific permission set when accessing the account and then return to the user portal to pick a different set when necessary. [Learn more](#)

Create new permission set

<input type="checkbox"/> Permission set	Description	Provisioned status
<input type="checkbox"/> DevelopmentPermissionSet <a href="#">↗</a>	Permission set to access Session Manager limited to all instances with project tag = Development	Not provisioned
<input checked="" type="checkbox"/> ProductionPermissionSet <a href="#">↗</a>	Permission set to access Session Manager limited to all instances with project tag = Production	Not provisioned

Cancel
Previous
Finish

© 2008 - 2020, Amazon Web Services, Inc. or its affili

Figure 25 - Permission sets selection that you want to give to user/group

Wait while your AWS account is configured with the permission set.

### ⚙️ Configuring your AWS account...

Do not leave this page while we are configuring your AWS accounts. This process may take a few minutes based on the accounts and permission sets being configured. If you close this window before the process is complete, you may need to start it again.

Account	Status	
devax-demo ██	Processing	<a href="#">Show details</a>

Figure 26 - Notification when AWS SSO in the progress configuring the account

Optionally, choose **Show details** to see all the processes that happen in the background.

### Complete

We have successfully configured your AWS account. Your users can access this AWS account with the permissions you assigned.

[Proceed to AWS accounts](#)

Account	Status	
devax-demo #956 ████████   tirta ████████	Complete	<a href="#">Hide details</a>
<ul style="list-style-type: none"><li>✓ Provisioning account</li><li>✓ Setting up SAML federation into this account</li><li>✓ Create role "ProductionPermissionSet" for permission set <a href="#">ProductionPermissionSet</a></li><li>✓ Assign group "ProductionAccess" access to <a href="#">ProductionPermissionSet</a></li></ul>		

© 2008 - 2020, Amazon Web Services, Inc. or its affil

Figure 27 - Notification when AWS account has been configured successfully for the user/group to access it

- Repeat Step 1 through Step 4 to assign the **DevelopmentPermissionSet** to the **DevelopmentAccess** group.

Once completed, your AWS account shows two user groups on the account detail page: **ProductionAccess** and **DevelopmentAccess**:

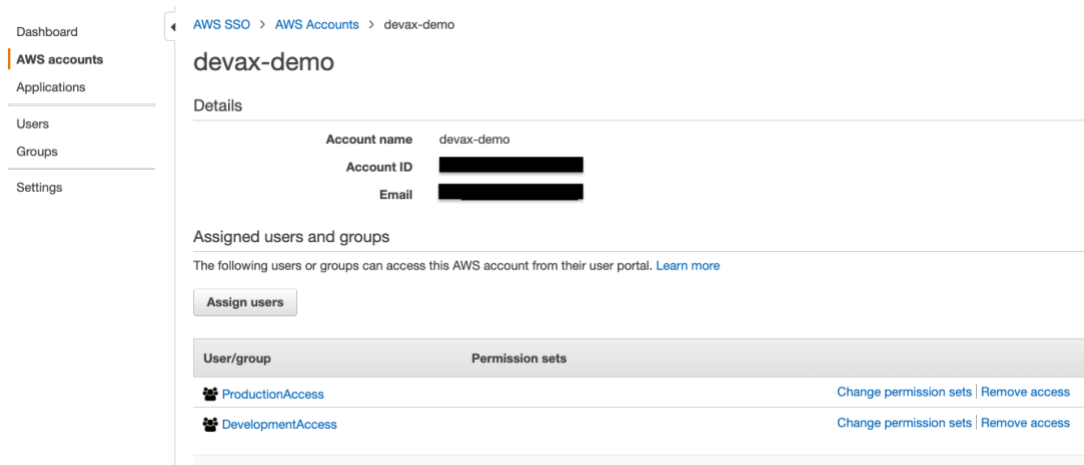


Figure 38 - Account detail page with assigned user groups

At this point, if you refresh the AWS SSO User Portal, you can see the AWS account you just configured with permissions.

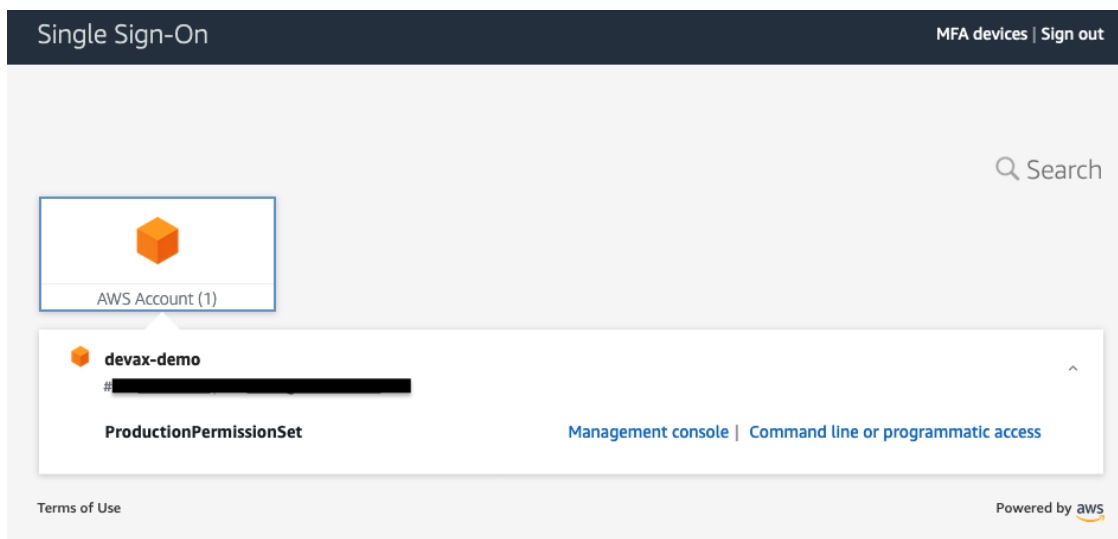


Figure 39 - User portal with AWS Accounts that has been granted to the active user

## Enable Multi-Factor Authentication

As the account administrator, you can configure AWS SSO to enforce multi-factor authentication (MFA) usage for all users. To enable MFA for all users, follow these steps:

1. In the left navigation pane of the [AWS SSO console](#), choose **Settings**. In the **Multifactor authentication** section, choose **Configure**.

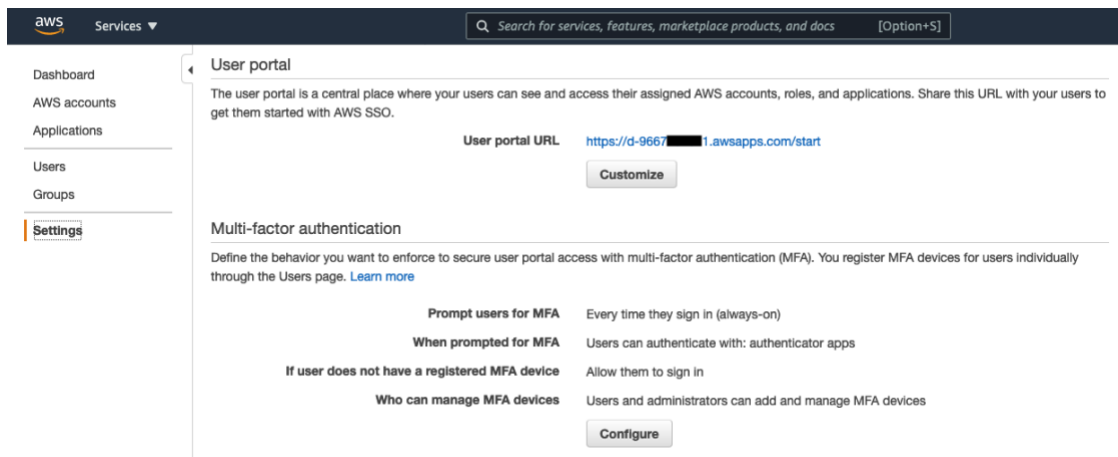


Figure 28 - AWS SSO Settings menu has Multi-factor authentication section

2. On the **Configure multi-factor authentication** page, make the following selections:
  - a. For **Users should be prompted for MFA**, choose **Every time they sign in (always-on)**.
  - b. For **Users can authenticate with these MFA types**, choose **Authenticator apps**.
  - c. For **If a user does not yet have a registered MFA device**, choose **Require them to provide a one-time password sent by email to sign in**.  
With this option, the user must enter the one-time password after signing in.

**Note:** The option **Block their sign-in** is only used if the administrator manages the MFA device activation.

3. Choose **Save changes**.

AWS SSO > Settings > Configure multi-factor authentication

## Configure multi-factor authentication

Choose how often users should be prompted for multi-factor authentication (MFA) and which types of devices can be used for signing in to the user portal.  
[Learn more](#)

### Users should be prompted for MFA

☐ Only when their sign-in context changes (context-aware)  
Users with a registered MFA device will only be prompted when their sign-in context changes (e.g. new device, location, anomalous behavior). Users can remember devices when this mode is selected.

☒ Every time they sign in (always-on)  
Users with a registered MFA device will be prompted every time they sign in.

☐ Never (disabled)  
All users sign in with their standard user name and password only. Choosing this option disables MFA.

### Users can authenticate with these MFA types

☐ Security keys and built-in authenticators  
Users can verify their identity using any FIDO2 or U2F capable device such as a physical security keys (e.g. YubiKey, Feitian, etc) or built-in authenticators (e.g. Apple TouchID, Windows Hello).

☒ Authenticator apps  
Users can verify their identity by entering a code generated from a time-based one-time password authenticator app (e.g. Authy, Google Authenticator, Microsoft Authenticator).

### If a user does not yet have a registered MFA device

☐ Require them to register an MFA device at sign in

☒ Require them to provide a one-time password sent by email to sign in

☐ Block their sign-in

☐ Allow them to sign in

### Who can manage MFA devices

☒ Users can add and manage their own MFA devices

[Cancel](#) [Save changes](#)

Figure 29 - Available settings for multi-factor authentication

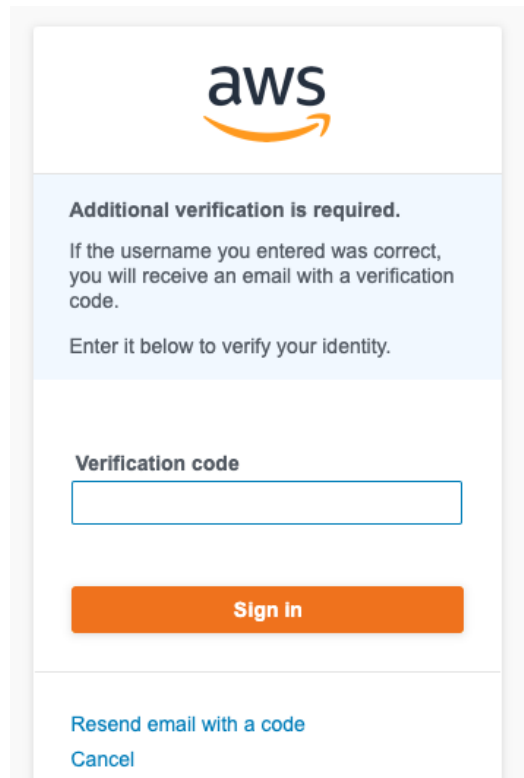


Figure 30 - Example display when we enable one-time password via email for user without MFA device

## Register the MFA device

As the user, follow these steps to register the MFA device:

1. On the top navigation bar of your **AWS SSO user portal**, choose **MFA devices**, and then choose **Register device**.

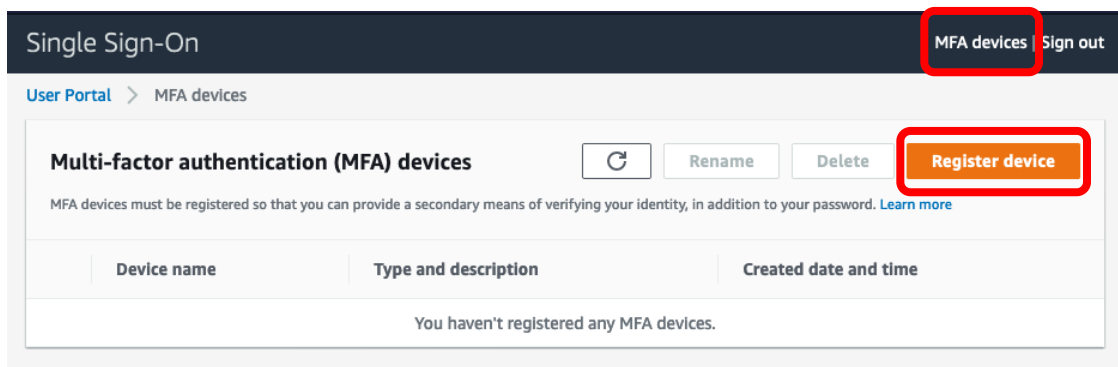



Figure 31 - MFA devices menu in the user portal

2. On the Multi-factor authentication (MFA) page, choose Show QR code.

Single Sign-On


User Portal > Multi-factor authentication (MFA)



### Set up the authenticator app

Username: testuser01

1



Install either the Google Authenticator, Duo Mobile, or Authy app on your mobile device or computer. [See a list of compatible apps](#)

2

[Show QR code](#)

Use your virtual MFA app or your device's camera to scan the QR code ([show secret key](#))

3

Please enter the six digit code from your authenticator app

Authenticator code

Cancel

Assign MFA

Figure 32 - MFA device setup process with link to display QR Code


3. Scan the QR code using your phone authentication application to generate the six-digit authenticator code. Type this code in the **Authenticator code** field.

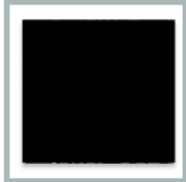


**aws**

### Set up the authenticator app

Username: testuser01

- 

1. Install either the Google Authenticator, Duo Mobile, or Authy app on your mobile device or computer. [See a list of compatible apps](#)
- 

2. Use your virtual MFA app or your device's camera to scan the QR code ([show secret key](#))
3. Please enter the six digit code from your authenticator app

Authenticator code

Figure 33 - After scanning QR code, input authenticator code

4. Choose **Assign MFA**.
5. On the **Authenticator app registered** confirmation dialog box, choose **Done**.

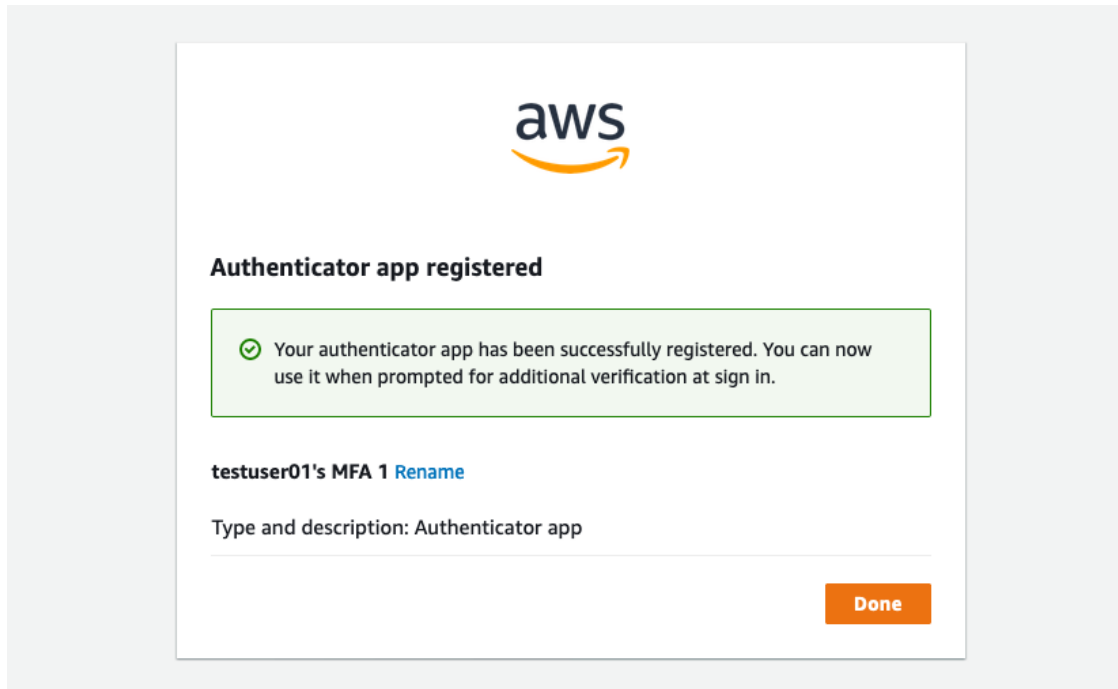


Figure 34 - Notification when MFA device registration completed successfully

6. On the Multi-factor authentication (MFA) devices page, test your MFA. Choose Sign out.

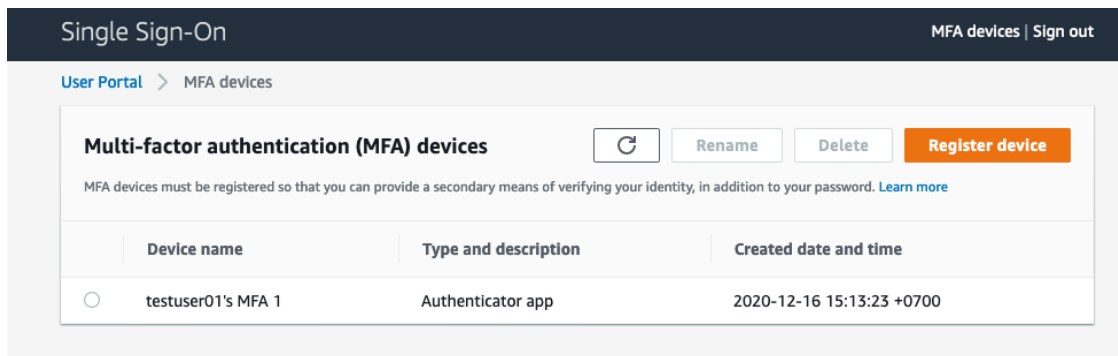
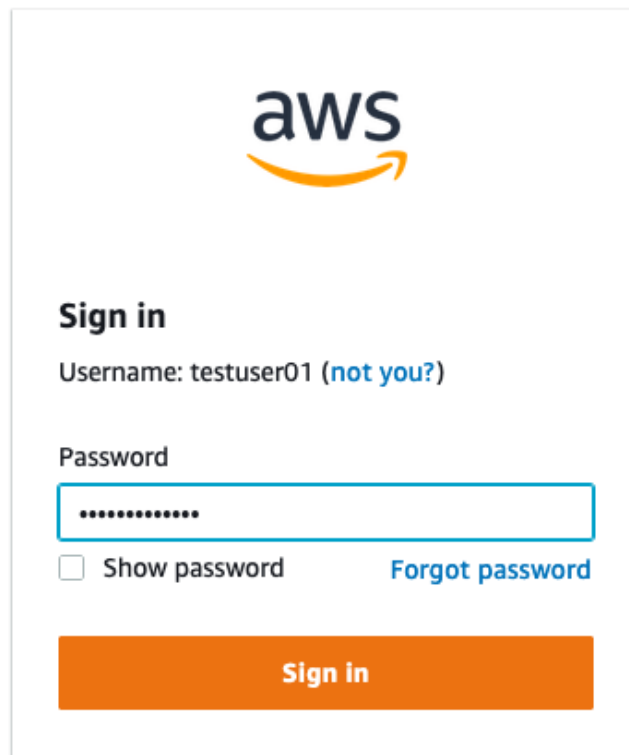


Figure 35 - New device has been listed in the MFA device menu

7. On the **Sign in** page, type your password and choose **Sign in**.

The image shows the AWS Sign in page. At the top is the AWS logo. Below it is the heading "Sign in". Under the heading, the text "Username: testuser01 (not you?)" is displayed. Below that is the label "Password" followed by a password input field containing eight dots. To the left of the input field is a checkbox labeled "Show password". To the right of the input field is a link labeled "Forgot password". At the bottom of the form is a large orange button labeled "Sign in".

aws

**Sign in**

Username: testuser01 (not you?)

Password

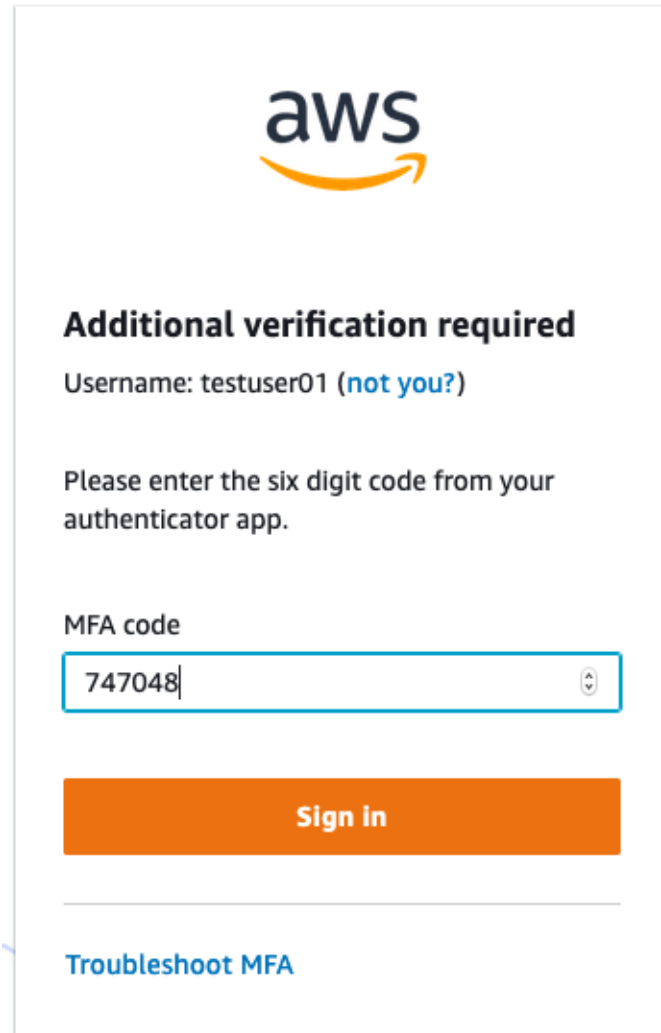
.....

☐ Show password [Forgot password](#)

**Sign in**

*Figure 36 - User password prompt during SSO login*

After you choose **Sign in**, you should be prompted to enter your six-digit **MFA code**.



**aws**

**Additional verification required**

Username: testuser01 (not you?)

Please enter the six digit code from your authenticator app.

MFA code

747048

**Sign in**

[Troubleshoot MFA](#)

Figure 37 - MFA page prompted after user login

## Set up AWS Systems Manager Session Manager

In this section, you set up AWS Systems Manager to enable Session Manager capability. Complete the following steps to enable Systems Manager:

### Create AWS Systems Manager Host Management configuration

1. Open the [Systems Manager console](#) and choose **Get Started with Systems Manager**.

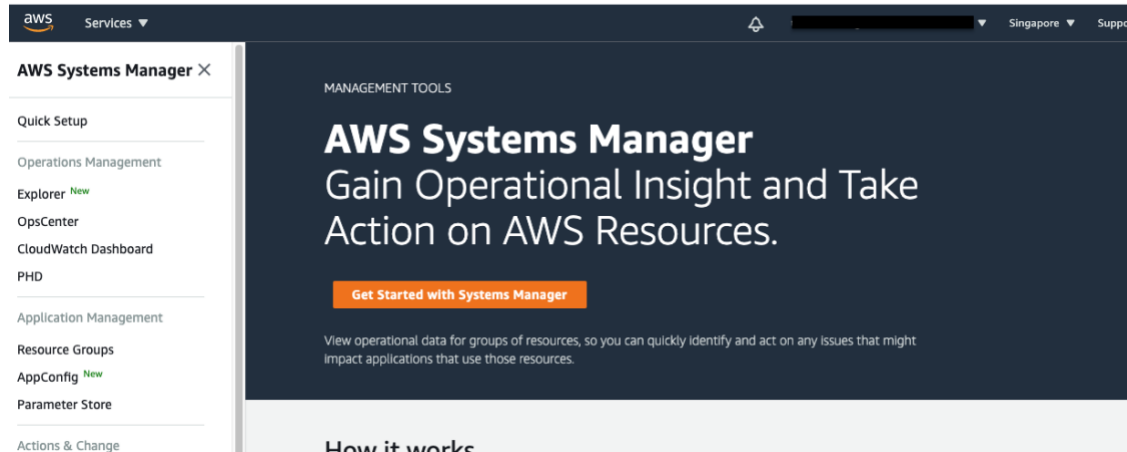


Figure 38 - Default Systems Manager page

2. On the **AWS Quick Setup** page, choose **Get started**.

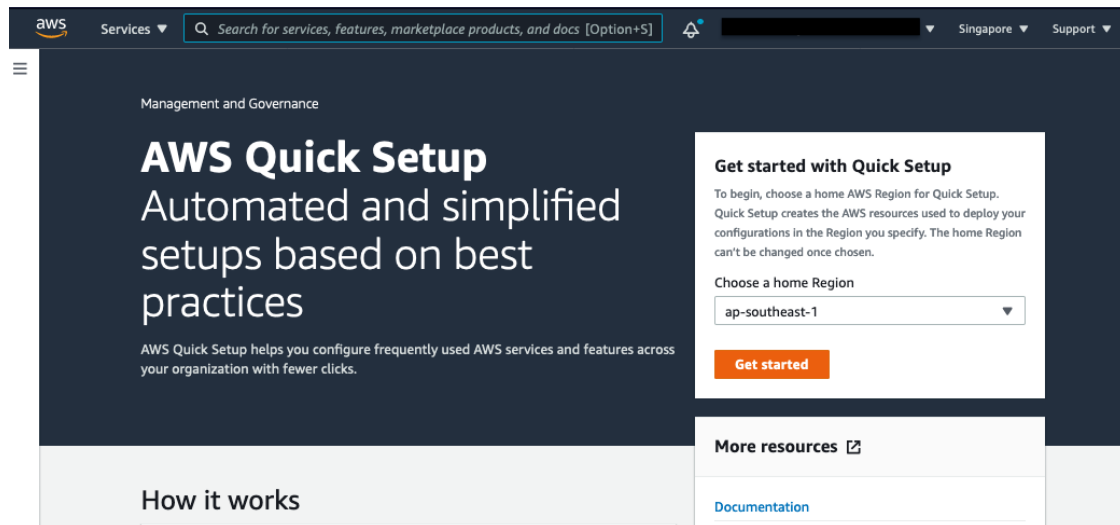


Figure 39 - AWS Quick Setup default page

3. On the **Quick Setup** page, choose **Create**.

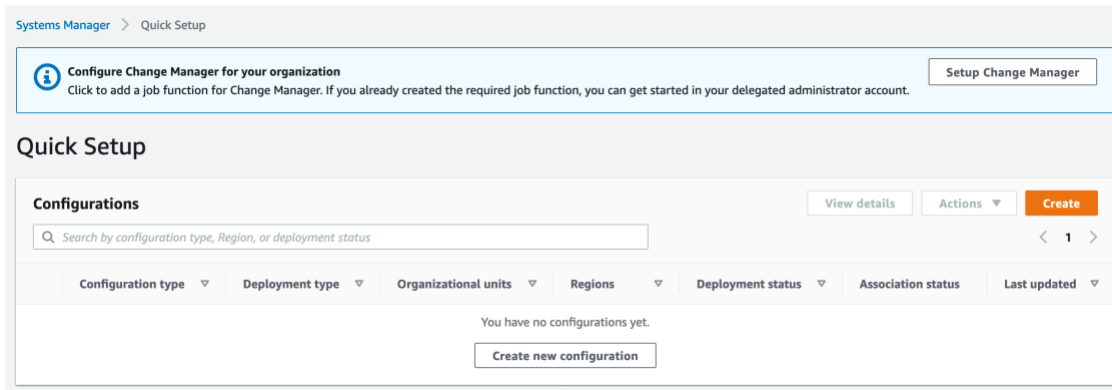


Figure 40 - No configuration created by default in the Quick Setup page

4. On the **Choose a configuration type** page, choose **Host Management** and then choose **Next**.

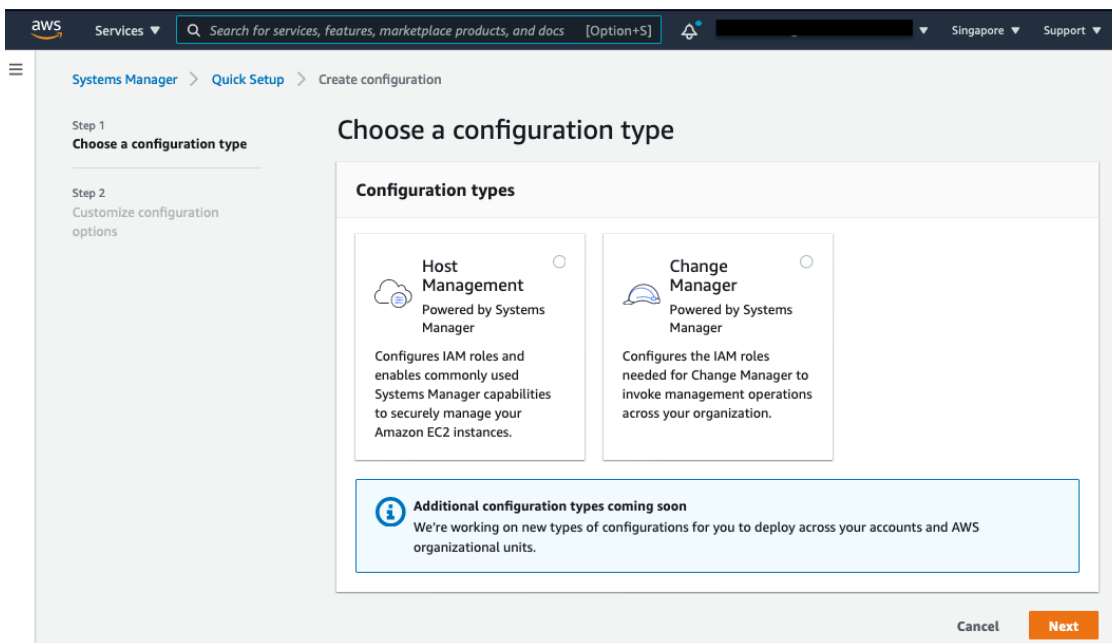


Figure 41 - Configuration type options for Systems Manager Quick Setup

5. On the Customize Host Management configuration options page, make the following selections:
  - d. For **Configuration options**, keep the default settings.
  - e. In the **Targets** section, select **Current account** and **Current Region**. Keep the default target instance setting of **All instances**.

Systems Manager > Quick Setup > Create configuration

Step 1  
Choose a configuration type

Step 2  
Customize Host Management configuration options

### Customize Host Management configuration options

**Configuration options**

Quick Setup configures the following Systems Manager components based on best practices. Select the check boxes for actions you want to schedule. [Learn more](#)

**Systems Manager**

- ☒ Update Systems Manager (SSM) Agent every two weeks.
- ☒ Collect inventory from your instances every 30 minutes.
- ☒ Scan instances for missing patches daily.

**Amazon CloudWatch**

- ☐ Install and configure the CloudWatch agent.
- ☐ Update the CloudWatch agent once every 30 days.

If you run this configuration, [Systems Manager Explorer](#) is enabled.

Learn more about the metrics included in [the CloudWatch agent's basic configuration](#) and [Amazon CloudWatch pricing](#).

**Targets**

Targets determine where this configuration will be deployed.

Choose the accounts and Regions where this configuration will be deployed

- ☐ Entire organization  
Deploys your configuration to all OUs and Regions in your organization.
- ☐ Custom  
Choose the OUs and Regions where this configuration will be deployed.
- ☒ Current account  
Choose the Regions to deploy this configuration to within the currently signed in account.

Choose between deploying to the current region or a custom set of regions

- ☒ Current Region  
Deploy configuration to the current Region.
- ☐ Choose Regions  
Choose the Regions where this configuration will be deployed.

Choose how you want to target instances

- ☒ All instances  
Deploy your configuration to all instances in the target account and Regions.
- ☐ Specify instance tag  
Specify a tag key-value pair to select instances that share that tag.
- ☐ By Resource Group  
Specify a resource group. Only instances in that group will be configured.
- ☐ Manual  
Manually specify the instances you want to configure.

Figure 42 - Customization available for Host Management configuration options

## 6. Choose **Create**.

**Summary**

Choose "Create" to perform the following actions:

- Enable Systems Manager Explorer in all targeted accounts and regions.
- Deploy IAM roles which enabling State Manager to invoke Automation documents that apply selected configuration options
- Create a State Manager association for each configuration option you have selected.
- Attach instance profiles or IAM roles with required Systems Manager permissions to targeted instances

Cancel **Create**

Figure 43 - Summary of Quick Setup actions with Create button

A confirmation message appears once the setup is updated.

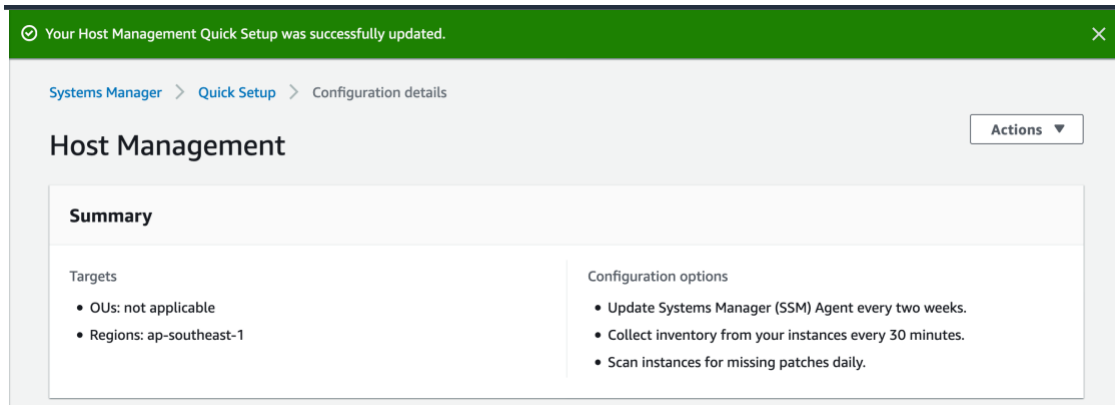


Figure 59 - Notification when Quick Setup has been updated successfully

Behind the scenes, Systems Manager runs several activities during the quick setup process:

- Creates four AWS Identity and Access Management (IAM) roles:
  - AmazonSSMRoleForInstancesQuickSetup
  - AWS-QuickSetup-SSM-RoleForEnabling Explorer
  - AWSServiceRoleForAmazonSSM
  - AWS-QuickSetup-HostMgmtRole-<Region>-<GUID> (Region and GUID are specific to your account)

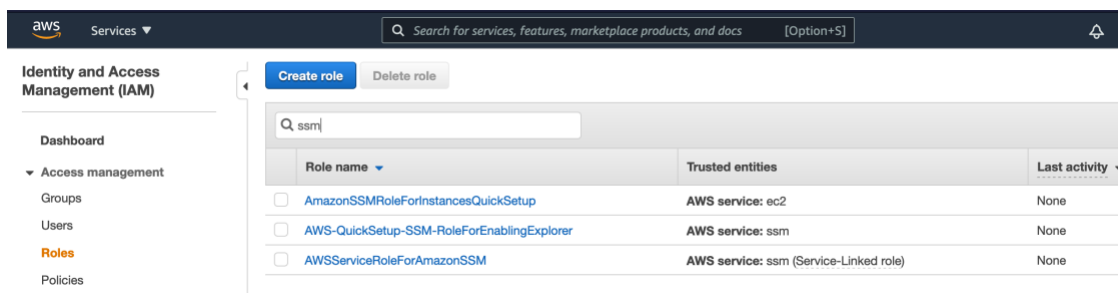


Figure 44 - IAM Roles related for SSM Quick Setup

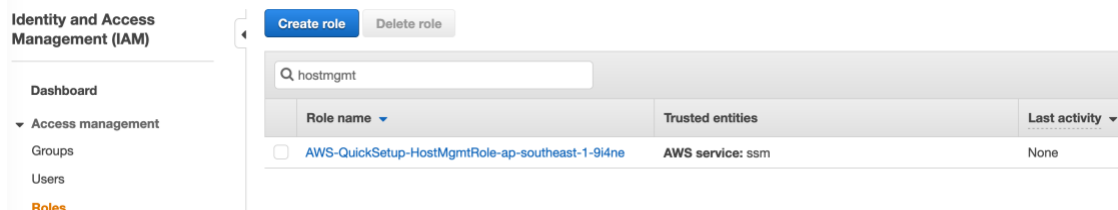


Figure 45 - IAM Role for Host Management has dynamic values (Region and GUID) in the name



- Creates six AWS Systems Manager State Manager associations with the naming format *AWS-QuickSetup-SSMHostMgmt-<TaskName>-<GUID>*. See the following table for descriptions of each of these associations.

AWS Systems Manager > State Manager

Associations View details Apply association now Edit Delete Create association

	Association id	Association name	Document name	Last execution date	Status	Association version	Resource status count
<input type="radio"/>	30466a49-f558-4396-85c4-dcc3664ec250	AWS-QuickSetup-SSMHostMgmt-UpdateSSMAgent-9i4ne	AWS-UpdateSSMAgent	Tue, 05 Jan 2021 15:46:52 GMT	Success	1	Success:1
<input type="radio"/>	7b52ce83-66a0-41a7-a9cd-535ce8bf60a0	AWS-QuickSetup-SSMHostMgmt-EnableExplorer-9i4ne	AWS-EnableExplorer	Tue, 05 Jan 2021 15:43:04 GMT	Success	1	
<input type="radio"/>	801386ea-8114-4e61-9ec3-e749b5b7867c	AWS-QuickSetup-SSM-EnableExplorer	AWS-EnableExplorer	Tue, 05 Jan 2021 15:42:19 GMT	Success	1	
<input type="radio"/>	aed1f6de-7697-4637-aaf0-cf8f733c2f5f	AWS-QuickSetup-SSMHostMgmt-ScanForPatches-9i4ne	AWS-RunPatchBaselineAssociation	Tue, 05 Jan 2021 15:46:50 GMT	Success	1	Success:1
<input type="radio"/>	ead375e2-122e-45f4-ab3b-37e28eb0d0bc	AWS-QuickSetup-SSMHostMgmt-AttachIAMToInstance-9i4ne	AWSQuickSetup-CreateAndAttachIAMToInstance-9i4ne	Tue, 05 Jan 2021 15:43:16 GMT	Success	1	Success:1
<input type="radio"/>	ff483cf5-c293-4d67-94a7-0d5266fdb142	AWS-QuickSetup-SSMHostMgmt-CollectInventory-9i4ne	AWS-GatherSoftwareInventory	Tue, 05 Jan 2021 15:47:06 GMT	Success	1	Success:1

Figure 46 - AWS SSM State Manager with 6 newly created associations

Association	Description
<b>AWS-QuickSetup-SSMHostMgmt-AttachIAMToInstance-&lt;GUID&gt;</b>	Checks that all EC2 instances have the correct IAM role attached. For instances that do not have an IAM profile attached by default, the AmazonSSMRoleForInstancesQuickSetup is attached.
<b>AWS-QuickSetup-SSMHostMgmt-UpdateSSMAgent-&lt;GUID&gt;</b>	Checks that all the Systems Manager agents already use the latest update.
<b>AWS-QuickSetup-SSMHostMgmt-CollectInventory-&lt;GUID&gt;</b>	Gathers inventory data.
<b>AWS-QuickSetup-SSMHostMgmt-EnableExplorer-&lt;GUID&gt;</b>	Both associations function together as unified operations dashboard to enable Explorer.
<b>AWS-QuickSetup-SSM-EnableExplorer</b>	
<b>AWS-QuickSetup-SSMHostMgmt-ScanForPatches-&lt;GUID&gt;</b>	Performs a patch scan.

You can check your existing EC2 instance and verify that the newly attached IAM role appears. On the **Instances** dashboard, choose the **Security** tab to view the attached role.

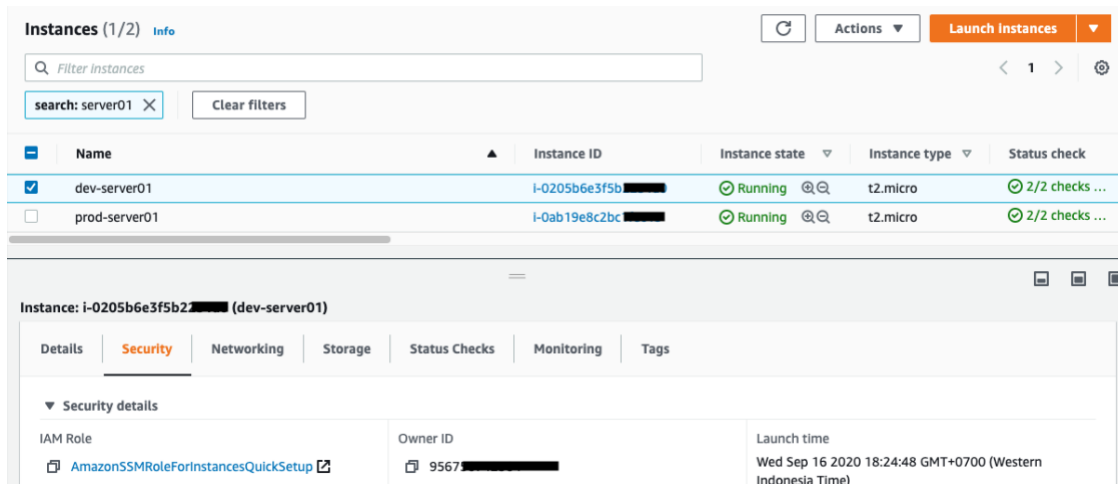


Figure 47 - An EC2 with AmazonSSMRoleForInstancesQuickSetup role attached

The **IAM Role** lists the **AmazonSSMRoleForInstanceQuickSetup** that the Quick Setup process created. If the **AmazonSSMRoleForInstanceQuickSetup** role does not appear on the Security tab, you need to attach the role manually. (For steps, see [Attach an IAM role to an instance](#).) Without this role, you cannot access your EC2 instance from Session Manager.

On the **Managed instances** page of the AWS Systems Manager console, you can see all of your EC2 instances.

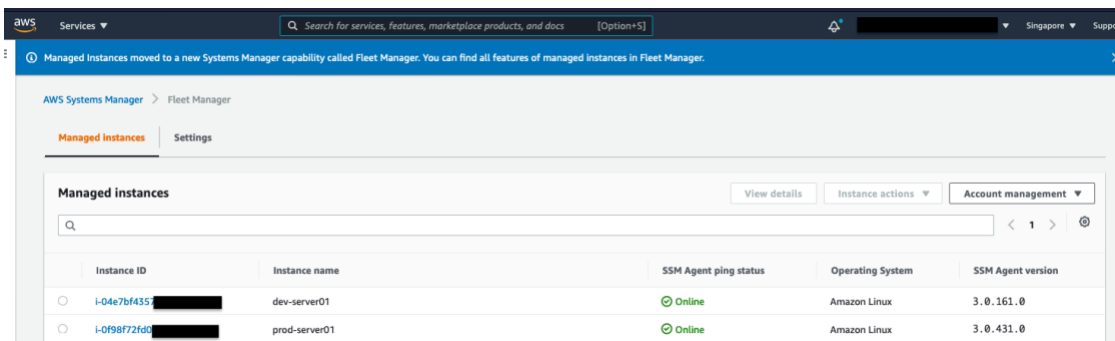


Figure 48 - Managed instances menu contains all registered EC2 instances

If you don't see your EC2 instances on this page, check the following items:

- Make sure that the Systems Manager Agent is installed and running. For details, see [Checking SSM Agent status and starting the agent](#).

- Ensure that the EC2 instance can access the Systems Manager endpoints via the public internet or VPC Endpoints. VPC Endpoints can be configured in an account that does not need internet access. For steps, see [How do I create VPC endpoints so that I can use Systems Manager to manage private EC2 instances without internet access?](#)
- Restart the instance. For steps, see [Reboot your instance](#).

## Enable Session Manager logging

Session Manager provides logging functionality that you can use to record and store all user session activity, including typed commands and outputs. You can also generate notifications of session activity in your AWS account with [Amazon Simple Notification Service \(Amazon SNS\)](#). These capabilities can be useful for audit purposes.

Session Manager logs can be stored in an Amazon S3 bucket or streamed into Amazon CloudWatch Logs. To enable logging, you need to specify the Amazon S3 bucket and Amazon CloudWatch Log Group.

1. In the left navigation pane of the [AWS Systems Manager console](#), choose **Session Manager**. Then, choose the **Preferences** tab.

Notice that, by default, Session Manager logging is disabled. When you enable logging, every user remote session (i.e., user typed commands and outputs) is captured and store in the designation location.

2. Choose **Edit**.

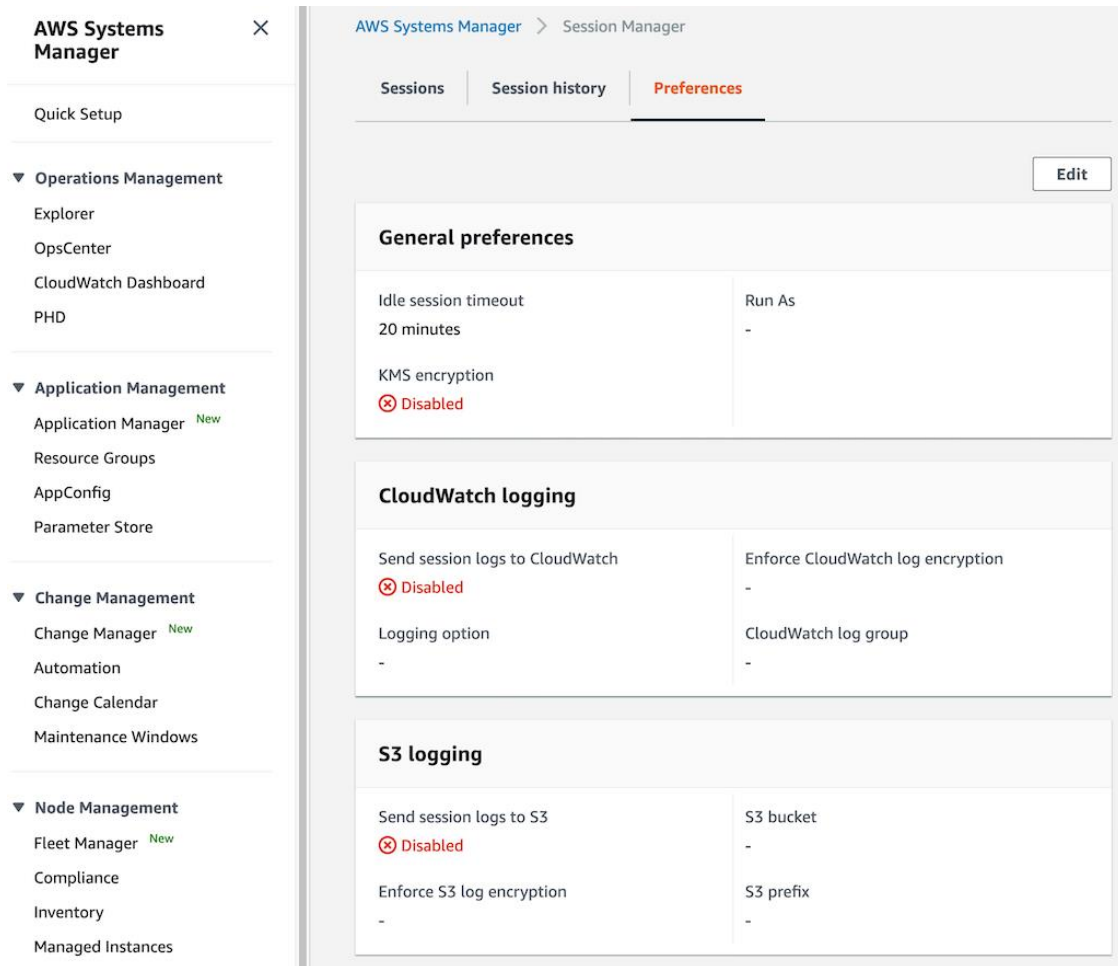
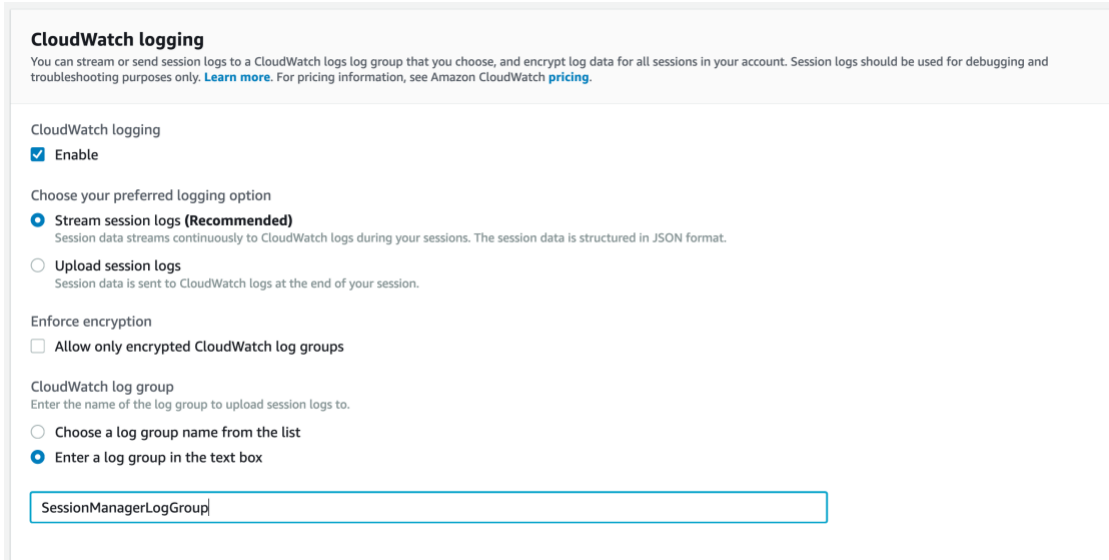


Figure 49 - Default Session Manager logging configuration is disabled

3. In the **CloudWatch logging** section, for **CloudWatch logging**, choose **Enable**. For **CloudWatch log group**, choose **Enter a log group in the text box** and type `SessionManagerLogGroup`.



**CloudWatch logging**

You can stream or send session logs to a CloudWatch logs log group that you choose, and encrypt log data for all sessions in your account. Session logs should be used for debugging and troubleshooting purposes only. [Learn more](#). For pricing information, see Amazon CloudWatch [pricing](#).

CloudWatch logging

☒ Enable

Choose your preferred logging option

☒ **Stream session logs (Recommended)**  
Session data streams continuously to CloudWatch logs during your sessions. The session data is structured in JSON format.

☐ Upload session logs  
Session data is sent to CloudWatch logs at the end of your session.

Enforce encryption

☐ Allow only encrypted CloudWatch log groups

CloudWatch log group

Enter the name of the log group to upload session logs to.

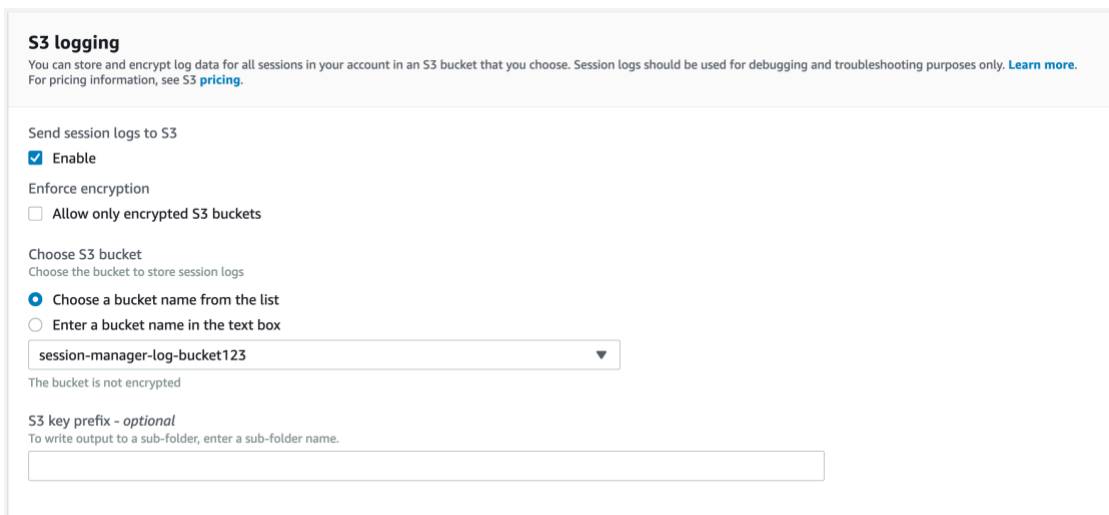
☐ Choose a log group name from the list

☒ Enter a log group in the text box

SessionManagerLogGroup

Figure 50 - Menu to configure CloudWatch logging

4. In the **S3 logging** section, for **Send session logs to S3**, choose **Enable**. For **Choose S3 bucket**, select the S3 bucket name from the list or manually specify the bucket name. Optionally, for **S3 key prefix**, specify a prefix to add to the logs.



**S3 logging**

You can store and encrypt log data for all sessions in your account in an S3 bucket that you choose. Session logs should be used for debugging and troubleshooting purposes only. [Learn more](#). For pricing information, see S3 [pricing](#).

Send session logs to S3

☒ Enable

Enforce encryption

☐ Allow only encrypted S3 buckets

Choose S3 bucket

Choose the bucket to store session logs

☒ Choose a bucket name from the list

☐ Enter a bucket name in the text box

session-manager-log-bucket123

The bucket is not encrypted

S3 key prefix - optional

To write output to a sub-folder, enter a sub-folder name.

Figure 51 - Menu to configure S3 logging

5. Choose **Save**.

## Add permissions to IAM role

To enable log delivery to both the CloudWatch Logs and the S3 bucket, you must add permissions to the **AmazonSSMRoleForInstancesQuickSetup** instance role you created in [Create AWS Systems Manager Host Management configuration](#).

Create a new IAM policy named *SessionManagerLogPolicy* and attach it to **AmazonSSMRoleForInstancesQuickSetup** role. The required permission for the *SessionManagerLogPolicy* is as follows:

**Note:** Make sure to replace the `YOUR_REGION`, `YOUR_ACCOUNT_ID`, `YOUR_BUCKET_NAME`, and `YOUR_LOG_GROUP` with your information.

```
{
  "Sid": "describeLogGroups",
  "Effect": "Allow",
  "Action": [
    "logs:DescribeLogGroups"
  ],
  "Resource": [
    "arn:aws:logs:<YOUR_REGION>:<YOUR_ACCOUNT_ID>:log-group:*"
  ]
},
{
  "Sid": "describeLogStreams",
  "Effect": "Allow",
  "Action": [
    "logs:DescribeLogStreams"
  ],
  "Resource": [
    "arn:aws:logs:<YOUR_REGION>:<YOUR_ACCOUNT_ID>:log-group:*:log-stream:*"
  ]
},
{
  "Sid": "createLogStream",
  "Effect": "Allow",
  "Action": [
    "logs:CreateLogStream",
    "logs:PutRetentionPolicy"
  ],
  "Resource": [
    "arn:aws:logs:<YOUR_REGION>:<YOUR_ACCOUNT_ID>:log-group:<YOUR_LOG_GROUP>:*"
  ]
},
{
  "Sid": "putEvents",
  "Effect": "Allow",
  "Action": [
    "logs:PutLogEvents",
    "logs:GetLogEvents"
  ],
  "Resource": [
```

```

    "arn:aws:logs:<YOUR_REGION>:<YOUR_ACCOUNT_ID>:log-
group:<YOUR_LOG_GROUP>:log-stream:*"
  ],
  {
    "Sid": "listBucket",
    "Effect": "Allow",
    "Action": "s3:ListBucket",
    "Resource": "arn:aws:s3:::<YOUR_BUCKET_NAME>"
  },
  {
    "Sid": "putObject",
    "Effect": "Allow",
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::<YOUR_BUCKET_NAME>/*"
  }
}

```

Your **AmazonSSMRoleForInstancesQuickSetup** role should now include two permissions policies, including the newly added **SessionManagerLogPolicy**.

The screenshot shows the AWS IAM console for the role **AmazonSSMRoleForInstancesQuickSetup**. The **Summary** tab is active, displaying the role's ARN, description, instance profile ARNs, path, creation time, last activity, and maximum session duration. Below the summary, the **Permissions** tab is selected, showing a list of permissions policies applied to the role. Two policies are listed: **SessionManagerLogPolicy** (Managed policy) and **AmazonSSMManagedInstanceCore** (AWS managed policy). Both policies have a delete icon (X) in the right column.

Policy name	Policy type	
SessionManagerLogPolicy	Managed policy	X
AmazonSSMManagedInstanceCore	AWS managed policy	X

Figure 52 - Example of AmazonSSMRoleForInstancesQuickSetup role with new policy

## Test Session Manager capability

In this section, you test Session Manager capability as the account administrator. Using Session Manager, you can access the instance's console without opening the management port to the internet.

1. In the left navigation pane of the [Systems Manager console](#), choose **Session Manager**. Then, choose **Start Session**.



Figure 53 - Default Session Manager page

2. On the **Start a session** page, select the target instance and choose **Start session**.

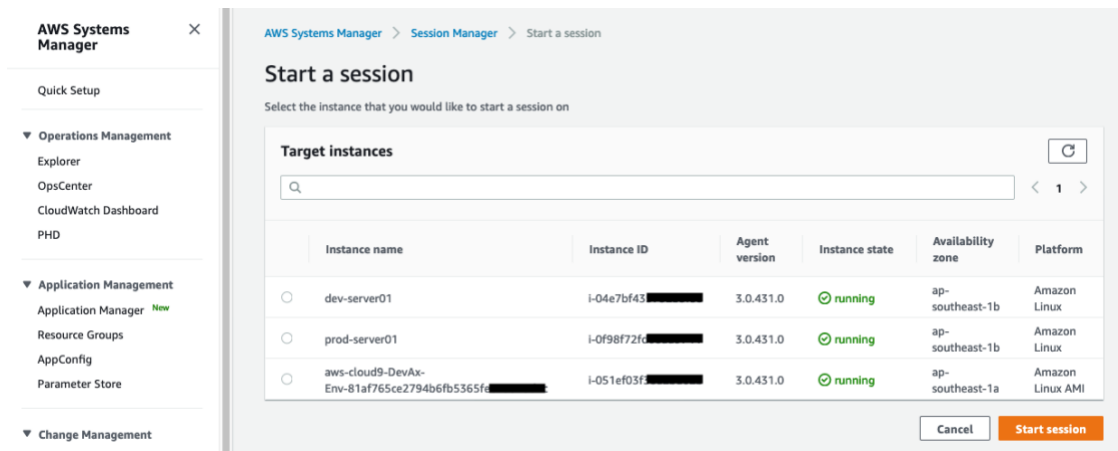
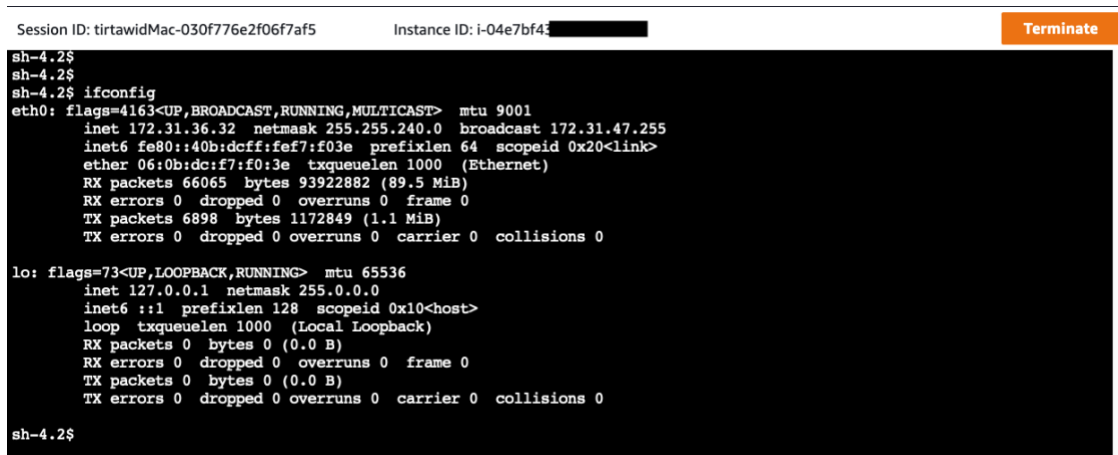


Figure 54 - List of servers in Start a session menu

Session Manager will open new browser tab and connect to the instance.



The screenshot shows a terminal window titled "Session ID: tirtawidMac-030f776e2f06f7af5" and "Instance ID: i-04e7bf4...". A "Terminate" button is in the top right. The terminal output shows the user running 'ifconfig' on an 'eth0' interface, displaying details like IP (172.31.36.32), netmask, broadcast, and statistics. Then, the user runs 'lo:' to show the 'lo' (loopback) interface details, including IP (127.0.0.1) and statistics. The prompt 'sh-4.2\$' is visible at the end.

```
Session ID: tirtawidMac-030f776e2f06f7af5 Instance ID: i-04e7bf4... Terminate
sh-4.2$
sh-4.2$
sh-4.2$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
    inet 172.31.36.32 netmask 255.255.240.0 broadcast 172.31.47.255
    inet6 fe80::40b:dcff:fef7:f03e prefixlen 64 scopeid 0x20<link>
    ether 06:0b:dc:f7:f0:3e txqueuelen 1000 (Ethernet)
    RX packets 66065 bytes 93922882 (89.5 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 6898 bytes 1172849 (1.1 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

sh-4.2$
```

Figure 55 - Example of connected remote session to an EC2

**Note:** Although the session experience is similar to SSH, the session is not using SSH Daemon (sshd).

With this test, you verified that Session Manager works as expected. This is apparent when you see the operating system prompt and you can run commands. In the following [Test your configuration](#) section, you verify Session Manager functionality again but in a scenario where a user accesses it through AWS Single Sign-On.

## Test your configuration

At this stage, you have completed these steps:

1. Created separate users and groups with specific permission set in AWS SSO.
2. Configured AWS Systems Manager and validated Session Manager functionality.

In this section, you test Session Manager functionality from within the AWS SSO user portal page.

### Test 1: ProductionAccess group

For this test, you use the **testuser01** login to access the Production EC2 instance remotely.

1. Sign in to your Single-Sign-On user portal with your **testuser01** login.
2. Choose AWS Account, and then choose Management console.

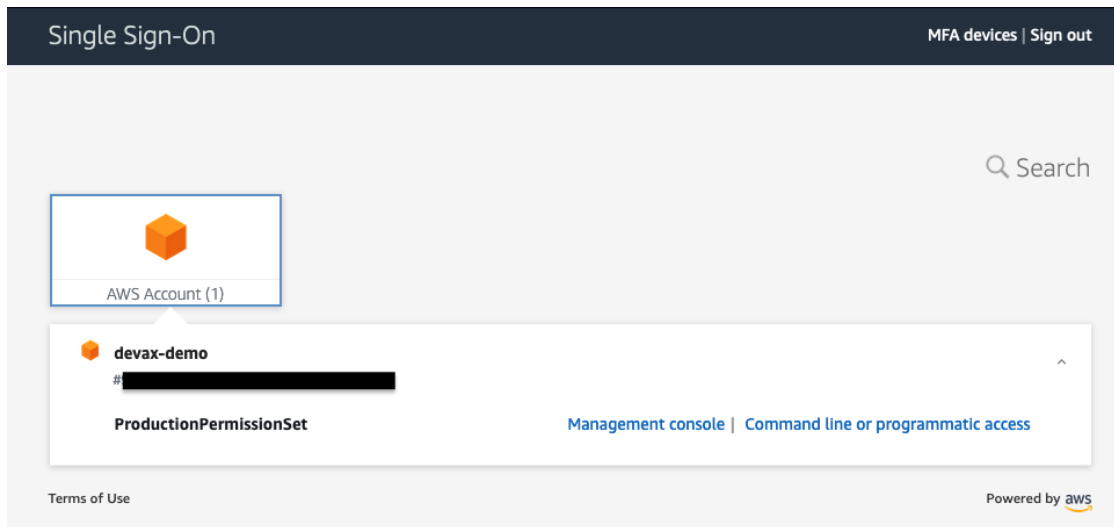


Figure 56 - AWS SSO display Management console menu for assigned account

**Single Sign On** configures the session and opens the **AWS Management Console**.

In the top right navigation bar, notice that **testuser01** signed in to the console as federated user with the name

**AWSReservedSSO\_ProductionPermissionSet\_**. This shows that AWS SSO takes care of the user federation with the corresponding Permission Set.

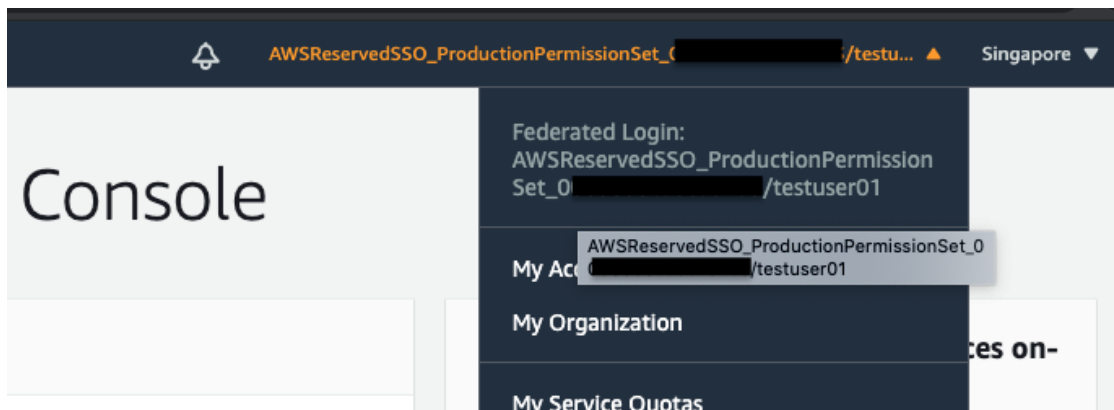


Figure 57 - Federated login information when user access the AWS Console with Production Permission Set

3. In the search bar, type **Systems Manager** to open the **Systems Manager** console.
4. In the left navigation pane of the Systems Manager console, choose **Session Manager**, then choose **Start Session**.



Figure 58 - Location of Session Manager menu in the Systems Manager console

- On the Start a session page, in the list of Target instances, select **prod-server01** and then choose Start session.

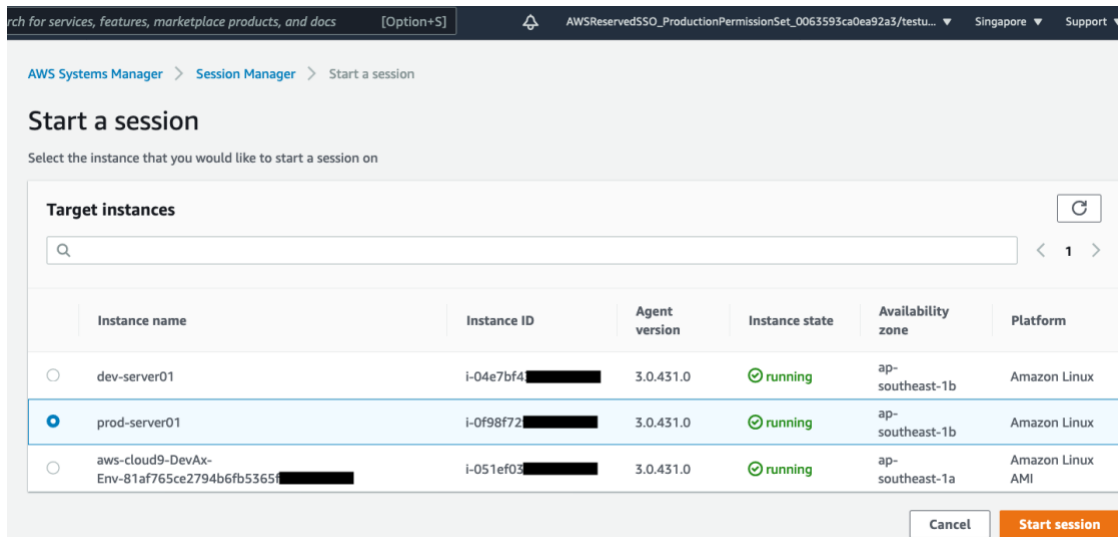
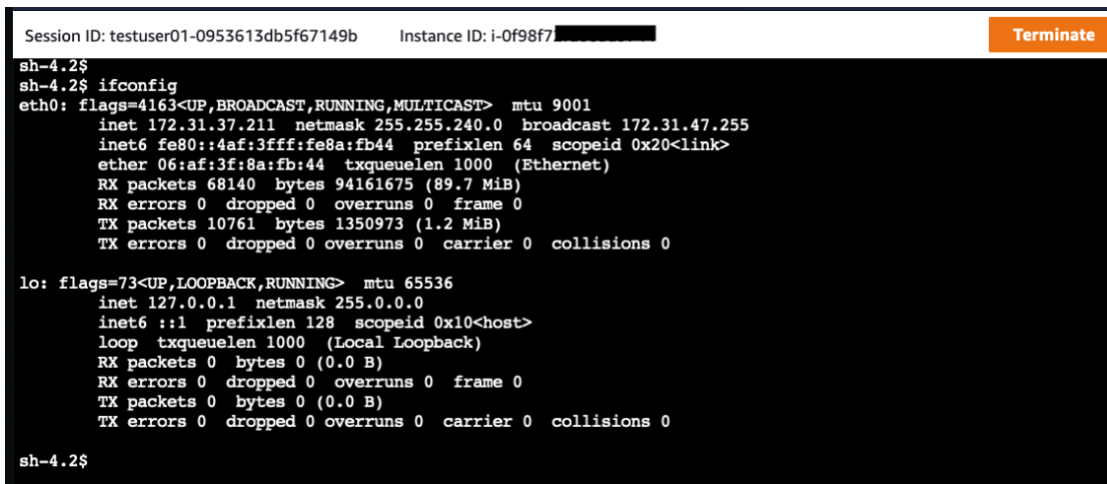


Figure 59 - List of servers in Start a session menu with Production server selected

The session opens in a new window.



```

Session ID: testuser01-0953613db5f67149b Instance ID: i-Of98f7
sh-4.2$
sh-4.2$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
    inet 172.31.37.211 netmask 255.255.240.0 broadcast 172.31.47.255
    inet6 fe80::4af:3fff:fe8a:fb44 prefixlen 64 scopeid 0x20<link>
    ether 06:af:3f:8a:fb:44 txqueuelen 1000 (Ethernet)
    RX packets 68140 bytes 94161675 (89.7 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 10761 bytes 1350973 (1.2 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

sh-4.2$

```

Figure 60 - Example of connected remote session to a production EC2 instance

If you repeat Step 5 and select **dev-server01** for the target instance, you will encounter the following error.

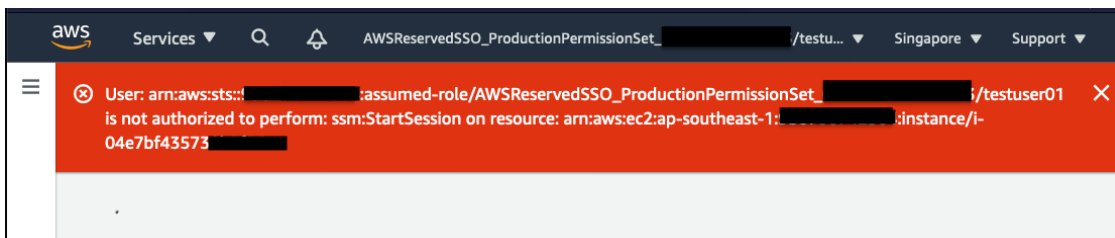


Figure 75 - Error message when user from ProductionAccess group try to access Development server

This error occurs because **testuser01** belongs to the **ProductionAccess** group with **ProductionPermissionSet** permission set. This user doesn't have access to **DeveloperPermissionSet**.

## Test 2: DevelopmentAccess group

1. Sign in to your Single-Sign-On user portal with your **testuser02** login.
2. Choose AWS Account, and then choose Management console.

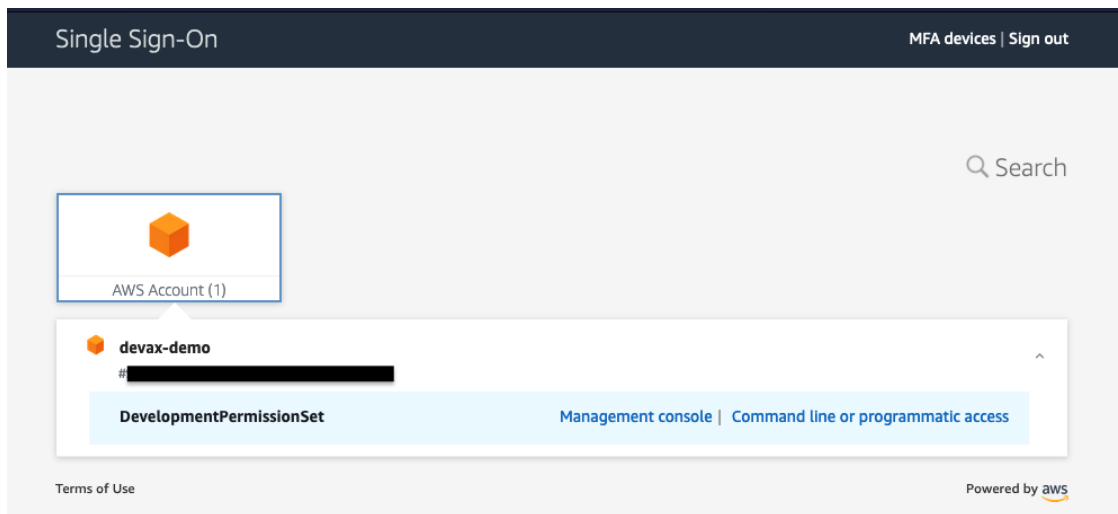


Figure 76 - AWS SSO display Management console menu for assigned account

In the top right navigation bar, notice that **testuser02** signed in to the console as federated user with the name **AWSReservedSSO\_DevelopmentPermissionSet\_**. This shows that AWS SSO takes care of the user federation with the corresponding Permission Set.

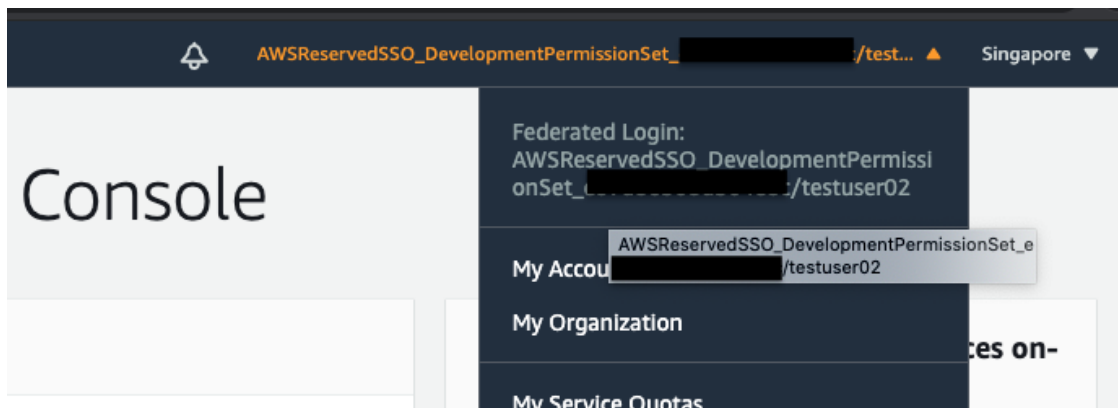


Figure 61 - Federated login information when user access the AWS Console with Development Permission Set

3. In the search bar, type **Systems Manager** to open the **Systems Manager** console.
4. In the left navigation pane of the Systems Manager console, choose **Session Manager**, then choose **Start Session**.
5. On the **Start a session** page, in the list of **Target instances**, select **dev-server01** and then choose **Start session**.

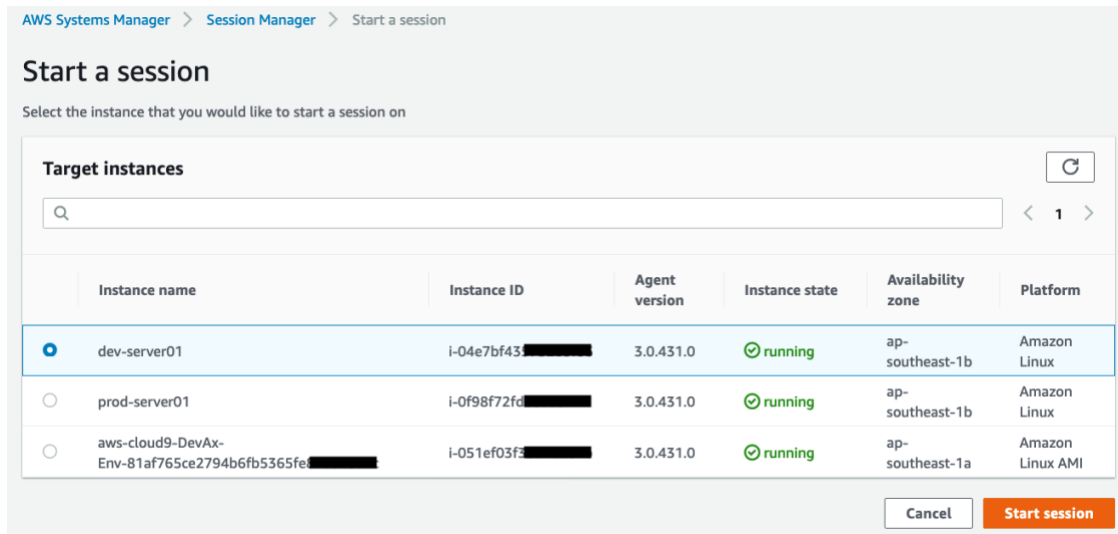


Figure 62 - List of servers in Start a session menu with Development server selected

The session opens in a new window.

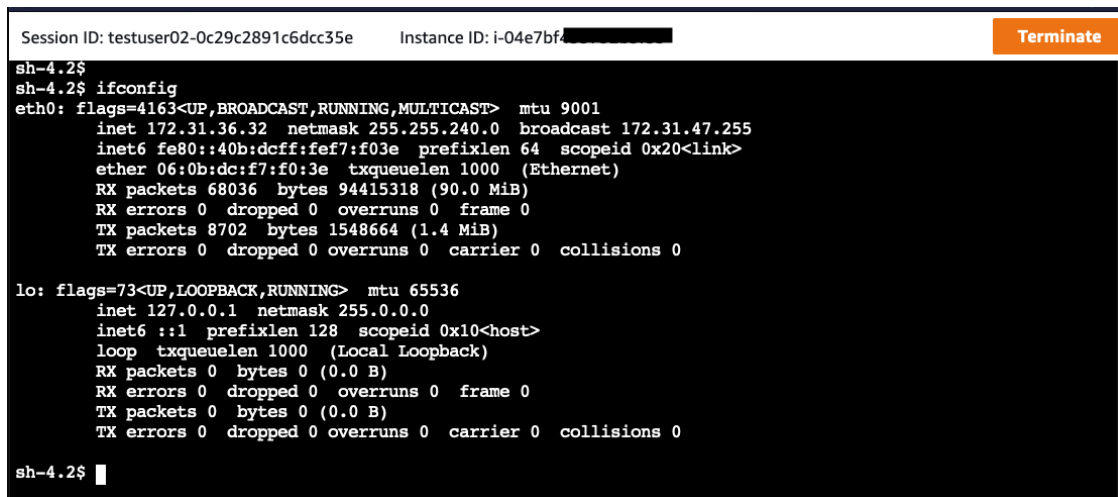


Figure 63 - Example of connected remote session to a Development EC2

If you repeat Step 5 and select **prod-server01** for the target instance, you will encounter the following error.

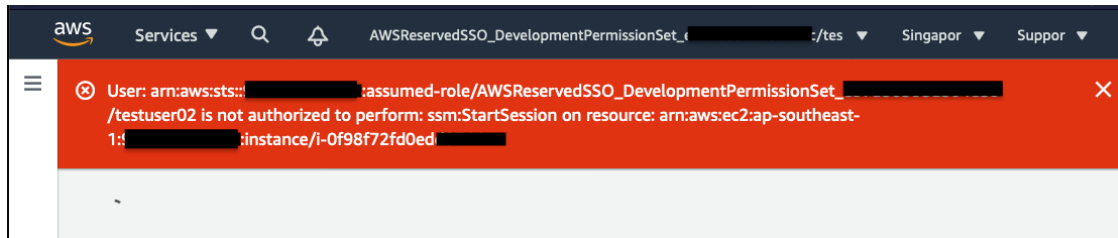


Figure 64 - Error message when user from *DevelopmentAccess* group try to access *Production* server

This error occurs because **testuser02** belongs to the **DevelopmentAccess** group with **DevelopmentPermissionSet** permission set. This user doesn't have access to **DeveloperPermissionSet**.

### Test 3: Validate the logs

1. In the left navigation pane of the [CloudWatch console](#), choose **Logs**, then choose **Log groups**. Choose the **SessionManagerLogGroup** you created in the [Enable Session Manager logging](#) section.

The **Log streams** section lists two logs corresponding to the previous tests.

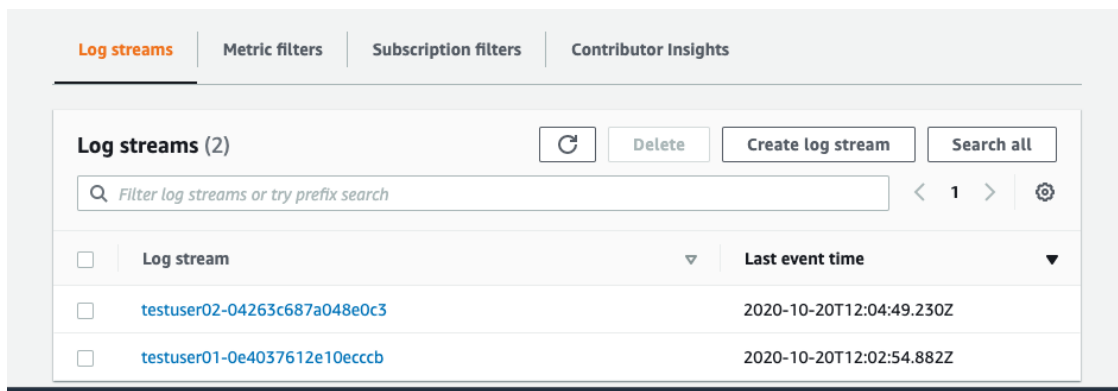


Figure 651 - Session Manager logs stored in CloudWatch Logs

Every keystroke and command output are recorded for every session. Choose a **Log stream** to see the log events:

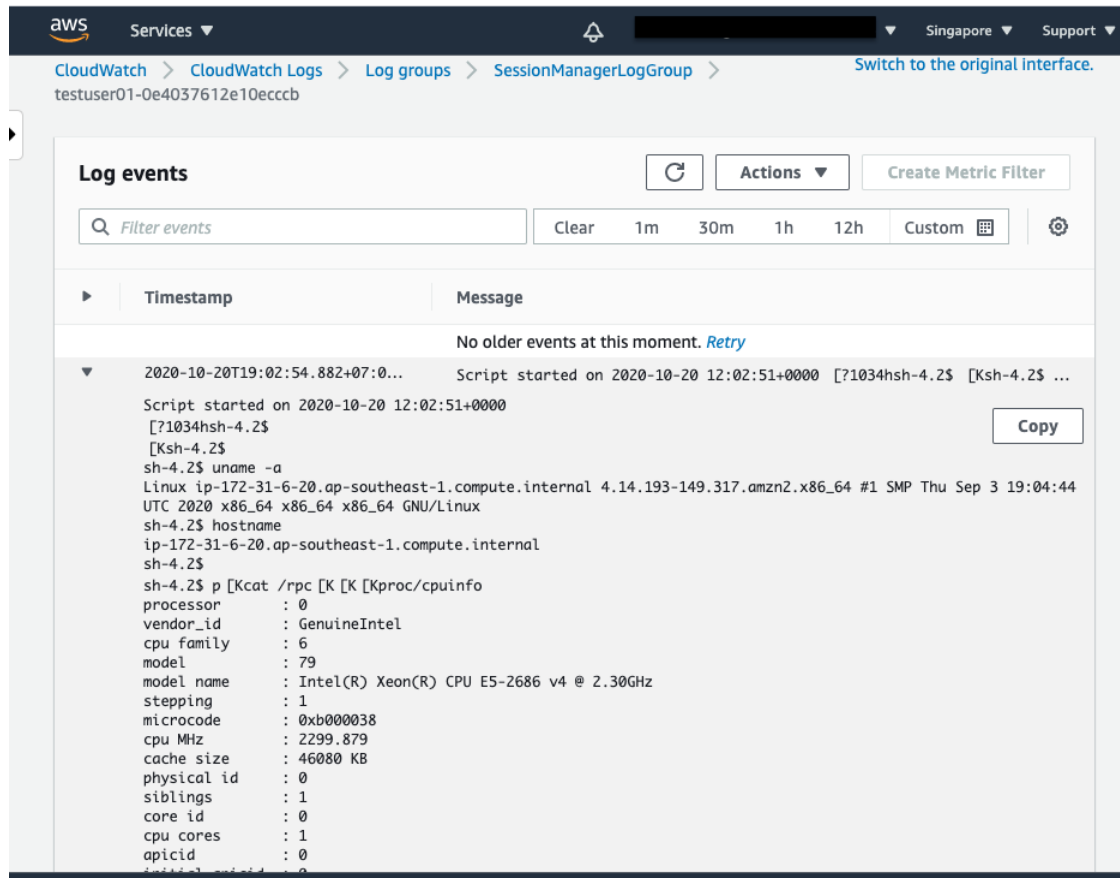


Figure 662 - Example of console output recorded in CloudWatch Logs

These same logs also stored in the Amazon S3 bucket you created in the [Enable Session Manager logging](#) section.

2. Open [Amazon S3 console](#) and choose the S3 bucket you specified for this walkthrough. The S3 bucket contents lists two logs corresponding to the testing activities you performed earlier.



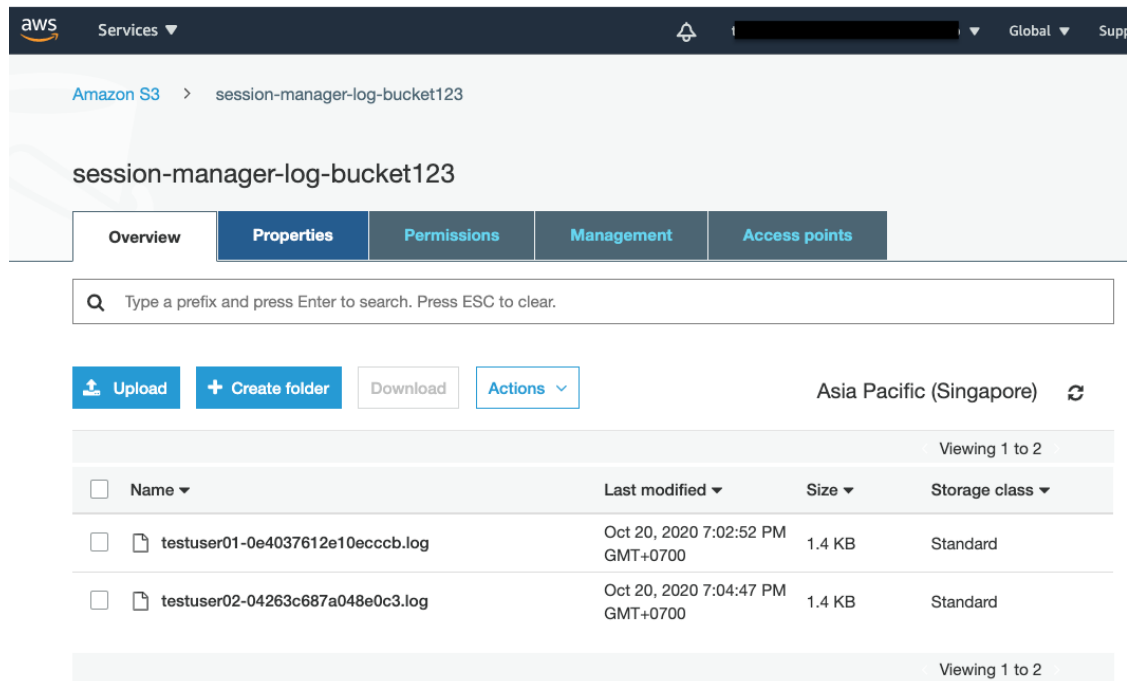


Figure 67 - Session Manager logs stored in S3 bucket

## (Optional) Configure Session Manager to manage on-premises servers

**Note:** If you don't have on-premises servers or don't have a requirement to access on-premises servers remotely via Session Manager, you can skip this section.

AWS Systems Manager can manage on-premises servers running SSM Agent.

You must complete additional steps to register an on-premises server to be managed by AWS Systems Manager. For instructions, see [Setting up AWS Systems Manager for hybrid environments](#).

However, to enable Session Manager functionality (to manage on-premises servers remotely), you must use the [advanced instances tier](#) during Hybrid Activation. The advanced instances tier enables user to register more than 1,000 on-premises servers or VMs in a single account and Region; as well as enable Session Manager for on-premises servers. When you create an activation, you must choose the **Change setting** option to use advanced instances in your account.


AWS Systems Manager > Activations > Create activation

## Create activation

**Activation setting**  
Create a new activation. After you complete the activation, you receive an activation code and ID. Use the code and ID to register SSM Agent on hybrid and on-premises servers or virtual machines. [Learn more](#)

Activation description- *Optional*  
  
Maximum 256 characters.

**Instance limit**  
Specify the total number of servers and VMs that you want to register with AWS. The maximum is 1000.  
  
Maximum number is 1000.

 To register more than 1,000 managed instances in the current AWS account and Region, change your account settings to use advanced instances. [Learn more](#)

[Change setting](#)

**IAM role**  
To enable communication between SSM Agent on your managed instances and AWS, specify an IAM role

- ☒ **Create a system default command execution role that has the required permissions**  
If you select this option, AWS creates a new role for you named AmazonEC2RunCommandRoleForManagedInstances. The role uses the existing public managed policy AmazonEC2RoleForSSM and grants AssumeRole permission to the SSM service.
- ☐ Select an existing custom IAM role that has the required permissions

Figure 68 - Hybrid activation menu

Note that enabling **advanced-instances tier** incurs charges. The charge is applied per the advanced on-premises instance per hour. You view pricing information for the **advanced-instances tier** on the [Systems Manager pricing page](#).

When you choose **Change setting** to allow advanced instance, you are prompted to acknowledge [advanced instance tier](#) enablement.

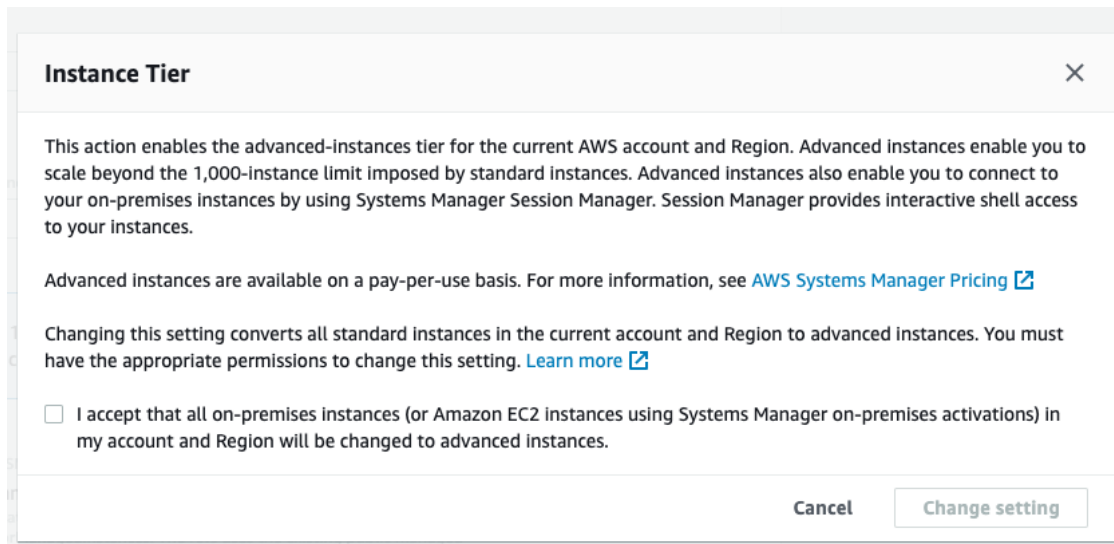


Figure 85 - Confirmation pop-up to enable Advance Instance Tier

After you have completed the steps to register an on-premises server, you can see the servers listed in the Systems Manager Managed Instances with the prefix **mi-**.

To allow SSO users access all on-premises servers, make the following changes:

1. Assign the same tags to the on-premises server as shown in [Table 1 - EC2 Required Tags](#). Use the **project** tag with possible values of **Development** and **Production**.

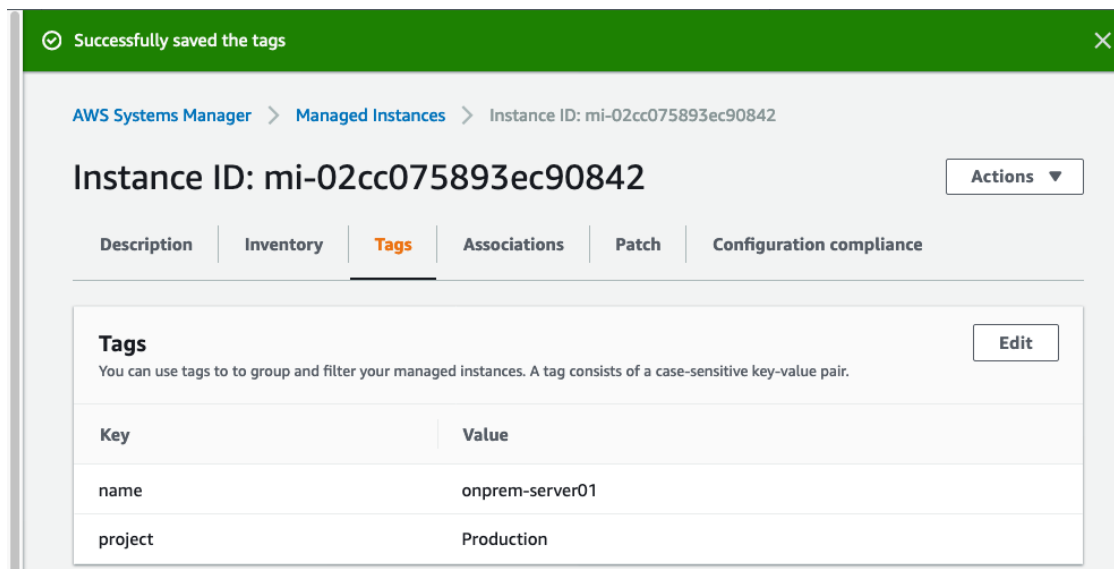


Figure 69 - On-premises server also has both project tags

2. Modify the AWS SSO Permission Set (for both **ProductionPermissionSet** and **DevelopmentPermissionSet**) to add extra permissions. These custom policies add a new resource ARN `arn:aws:ssm:*:*:managed-instance/*`. The on-premises server didn't use EC2 ARN format but uses Systems Manager ARN format.

See the following code snippets for the custom policies

#### Custom Permission Policy for **ProductionPermissionSet**:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:GetServiceSetting"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Deny",
      "Action": [
        "ssm:ResetServiceSetting",
        "ssm:UpdateServiceSetting"
      ],
      "Resource": "*"
    },
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "ssm:DescribeSessions",
        "ssm:DescribeInstanceProperties",
        "ec2:DescribeInstances",
        "ssm:GetConnectionStatus"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:StartSession"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:instance/*",
        "arn:aws:ssm:*:*:managed-instance/*"
      ],
      "Condition": {
        "StringLike": {
          "ssm:resourceTag/project": [
            "Production"
          ]
        }
      }
    }
  ]
}
```

```

    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "ssm:TerminateSession"
    ],
    "Resource": [
      "arn:aws:ssm:*:*:session/${aws:username}-*"
    ]
  }
]
}

```

### Custom Permission Policy for **DevelopmentPermissionSet**:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:GetServiceSetting"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Deny",
      "Action": [
        "ssm:ResetServiceSetting",
        "ssm:UpdateServiceSetting"
      ],
      "Resource": "*"
    },
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "ssm:DescribeSessions",
        "ssm:DescribeInstanceProperties",
        "ec2:DescribeInstances",
        "ssm:GetConnectionStatus"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:StartSession"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:instance/*",
        "arn:aws:ssm:*:*:managed-instance/*"
      ],
      "Condition": {
        "StringLike": {

```

```

        "ssm:resourceTag/project": [
            "Development"
        ]
    }
},
{
    "Effect": "Allow",
    "Action": [
        "ssm:TerminateSession"
    ],
    "Resource": [
        "arn:aws:ssm:*:*:session/${aws:username}-*"
    ]
}
]
}

```

3. Allow `ssm:GetServiceSetting` action so that user can check the managed instances status especially to let the user able to query Parameter Store `/ssm/managed-instance/activation-tier`. See the following code for the custom policy.

```

{
    "Effect": "Allow",
    "Action": [
        "ssm:GetServiceSetting"
    ],
    "Resource": "*"
},
{
    "Effect": "Deny",
    "Action": [
        "ssm:ResetServiceSetting",
        "ssm:UpdateServiceSetting"
    ],
    "Resource": "*"
},

```

Make sure you reapply the permission set to see the new policy applied. For steps, see [Grant access to AWS account](#).

## Clean up

It is a best practice to clean up your resources to avoid incurring further charges to your account. To avoid unnecessary charges, clean up the following resources:

- Terminate your test EC2 instances. For steps, see [Terminating instances](#).
- Remove the AWS account assignment in the AWS SSO console. See [Remove user access](#) section for detail steps.

- Remove both test users and the groups. To remove user, open AWS SSO console, choose **Users**, tick on the user and choose **Delete users**. To remove group, open AWS SSO console, choose **Groups**, tick on the group and choose **Delete groups**.

## Conclusion

The steps in this guide walked you through securely limiting remote access sessions using AWS Systems Manager and AWS Single Sign-On with multi-factor authentication (MFA).

As a follow-up, you can try to apply the same approach to serve your organization's needs. For example, set up additional permission sets for additional user groups or add more permissions to allow users to access other AWS services.

## Contributors

Contributors to this document include:

- Tedy Tirtawidjaja, Senior Solution Architect, Amazon Web Services
- Erik Weber, Senior Management Tools Specialist Solution Architect, Amazon Web Services
- Colin Igbokwe, Senior Security Solution Architect, Amazon Web Service

## Additional Resources

For additional information, see:

- [AWS Systems Manager User Guide](#)
- [AWS Systems Manager Session Manager](#)
- [AWS Single Sign-On User Guide](#)

## Document Revisions

Date	Description
March 1, 2021	First publication.