AWS SSO setup

Contents

[Scope 3](#_Toc114660230)

[Problem Statement 3](#_Toc114660231)

[Introduction 3](#_Toc114660232)

[1. What benefits does AWS IAM Identity Center provide? 3](#_Toc114660233)

[2. What are the key features of AWS IAM Identity Center? 4](#_Toc114660234)

[Steps to Setup AWS SSO 5](#_Toc114660235)

[1 – Create AWS Account 5](#_Toc114660236)

[2‏‏‎‏‏‎ - Select Region 5](#_Toc114660237)

[3 – Search for AWS SSO service 6](#_Toc114660238)

[4 –Enable AWS SSO 7](#_Toc114660239)

[5 – Create AWS Organization 7](#_Toc114660240)

[6 – Open AWS organizations 8](#_Toc114660241)

[7 – Add an AWS account 9](#_Toc114660242)

[8 – Complete Configuration 10](#_Toc114660243)

[9 – Create permission set 11](#_Toc114660244)

[10 – Create Permission setes 12](#_Toc114660245)

[11 – Add user 13](#_Toc114660246)

[12 – Verify users 14](#_Toc114660247)

[13 – Testing 17](#_Toc114660248)

[14 - AWS console access through AWS SSO 17](#_Toc114660249)

Document Name: AWS SSO setup

Owner: Bhagyashree Patle

# Scope

AWS SSO is where you create, or connect, your workforce identities in AWS once and manage access centrally across AWS organization. You can choose to manage access just to your AWS accounts or cloud applications. You can create user identities directly in AWS SSO, or you can bring them from your Microsoft Active Directory or a standards-based identity provider, such as Okta Universal Directory or Azure AD. With AWS SSO, you get a unified administration experience to define, customize, and assign fine-grained access.

# Problem Statement

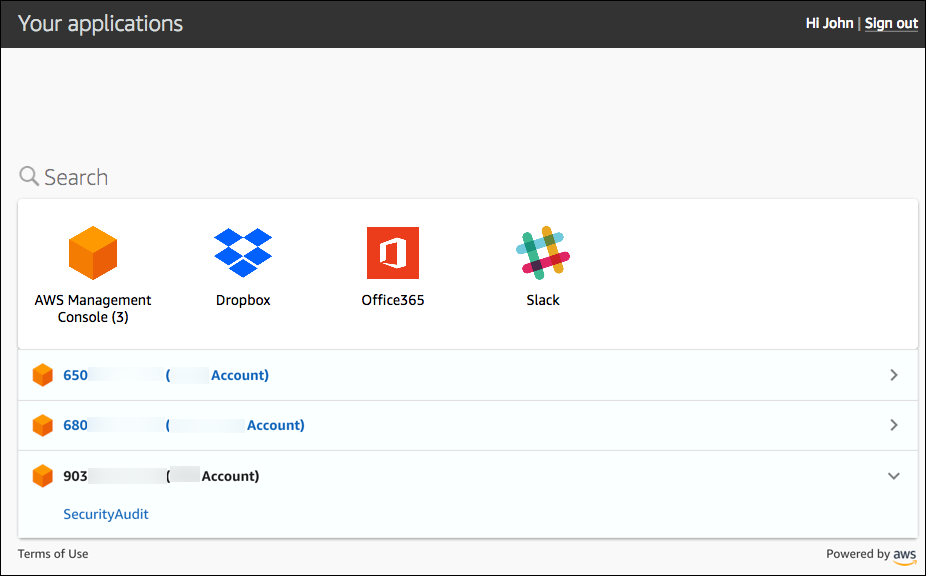
Configure a service where a user authenticates and can access multiple applications without having to log in to those other applications. So that users don’t have to log in to all of the applications they use every day.

# Introduction

AWS SSO is a cloud-based SSO service to easily manage access to all of the accounts configured in your AWS Organization. Using AWS SSO eliminates the need for operating your own SSO infrastructure. It easily manages your AWS Organizations account access, can integrate with many third-party applications, and it provides a Security Assertion Markup Language (SAML) configuration tool to extend AWS SSO to any SAML enabled application.

## What benefits does AWS IAM Identity Center provide?

Through AWS IAM Identity Center you can connect to your Microsoft Active Directory as a result of that users can sign in to the user portal with their user names and passwords to access the AWS accounts and applications to which you have granted them access. The following screenshot shows an example of the AWS IAM Identity Center user portal.



## What are the key features of AWS IAM Identity Center?

**AWS IAM Identity Center user portal:** In the user portal, users can easily find and access all applications and AWS accounts to which you have granted them access.

**Integration with AWS Organizations:**AWS IAM Identity Center is integrated with Organizations to enable you to manage access to all AWS accounts in your organization.

**Integration with on-premises Active Directory:**AWS IAM Identity Center integrates with on-premises Active Directory by using AWS Directory Service.

**Centralized permissions management:**With AWS IAM Identity Center, you can centrally manage the permissions granted to users when they access AWS accounts via the AWS Management Console.

Application configuration wizard: You can configure IAM Identity Center access to any SAML-enabled business application by using the AWS IAM Identity Center application configuration wizard.

Built-in IAM Identity Center integrations: AWS IAM Identity Center provides built-in IAM Identity Center integrations and step-by-step configuration instructions for many commonly used business applications such as Office 365, Salesforce, and Box.

Highly available multi-tenant IAM Identity Center infrastructure: AWS IAM Identity Center is built on a highly available, AWS managed IAM Identity Center infrastructure.

# Steps to Setup AWS SSO

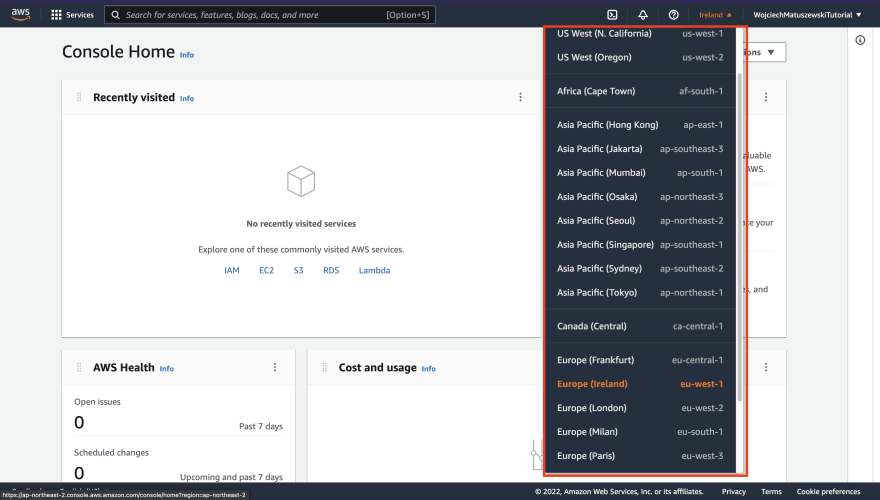
## 1 – Create AWS Account

If you do not have an AWS account already, create one. There are many great resources on how to do just that, for example, [this guide by A Cloud Guru](https://acloudguru.com/videos/acg-fundamentals/how-to-create-an-aws-account?utm_campaign=11244863417&utm_source=google&utm_medium=cpc&utm_content=469352928666&utm_term=_&adgroupid=115625160932).

* I would **strongly recommend this AWS account** be **used only for AWS SSO / AWS Organizations set up**. Consider **not** running any deployments on this account.

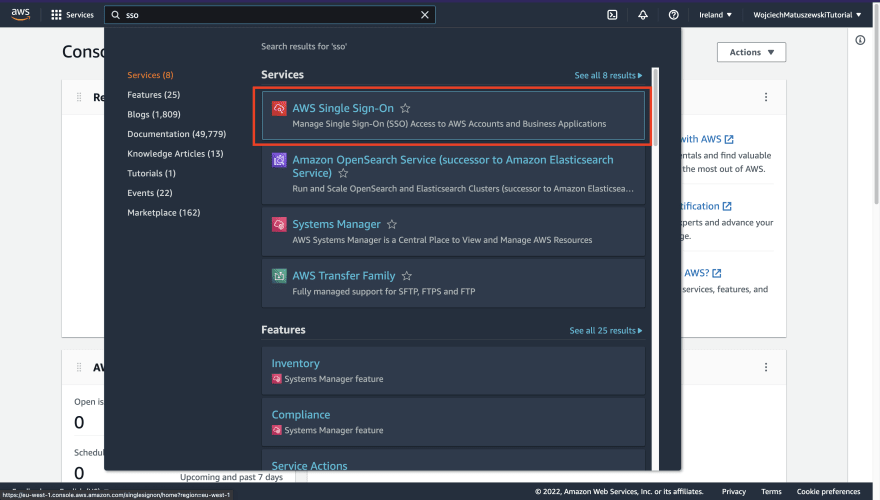
## ****2****‏‏‎‏‏‎ - Select Region

Pick the region you wish your AWS SSO configuration to be deployed into. To my best knowledge, the [AWS SSO is a region-bound service](https://docs.aws.amazon.com/singlesignon/latest/userguide/regions.html#region-data). **For demonstration purposes** I will be using the eu-west-1 region throughout this blog post.

[](https://res.cloudinary.com/practicaldev/image/fetch/s--sH-i5S89--/c_limit,f_auto,fl_progressive,q_auto,w_880/https:/dev-to-uploads.s3.amazonaws.com/uploads/articles/pwoctne9f5jh7y9o5jzh.png)

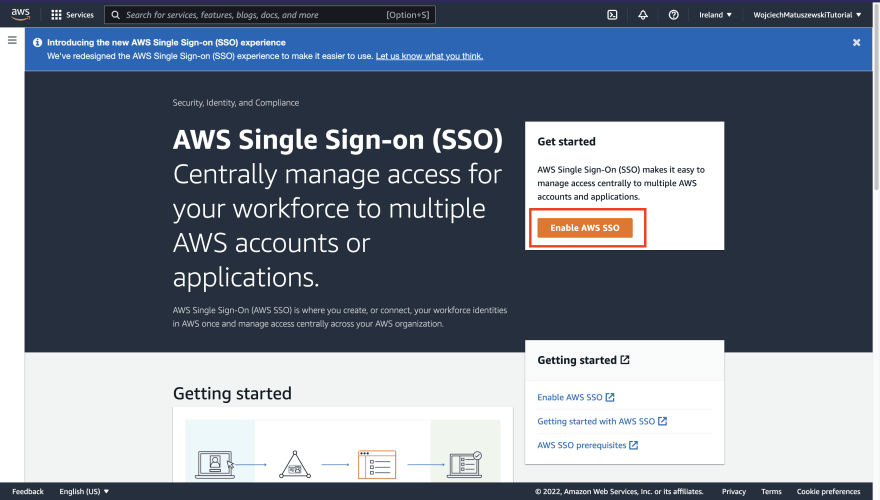
## ****3**** – Search for AWS SSO service

In the search bar, look for AWS SSO service. The sso search phrase should do. Click on the service tile to go to its dashboard.

[](https://res.cloudinary.com/practicaldev/image/fetch/s--VK_aWobz--/c_limit,f_auto,fl_progressive,q_auto,w_880/https:/dev-to-uploads.s3.amazonaws.com/uploads/articles/mjwcq49cy7v7mzyp4asu.png)

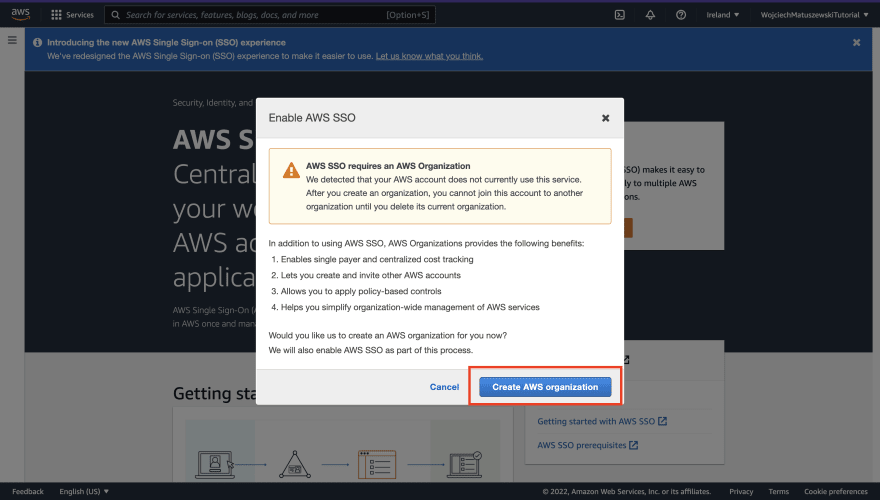
## ****4**** –Enable AWS SSO

After landing on the AWS SSO dashboard, assuming you do not have it enabled in any other AWS regions, you should be presented with a landing page that contains a big "Enable AWS SSO" button – click on it.

[](https://res.cloudinary.com/practicaldev/image/fetch/s--f-l__nPG--/c_limit,f_auto,fl_progressive,q_auto,w_880/https:/dev-to-uploads.s3.amazonaws.com/uploads/articles/rd7fdqgvciuao3eqr7ko.png)

## ****5**** – Create AWS Organization

Assuming you do not have any [AWS Organizations](https://aws.amazon.com/organizations/) created on this account, the AWS console will greet you with the popup to create one. Click on the "Create AWS Organization" button.

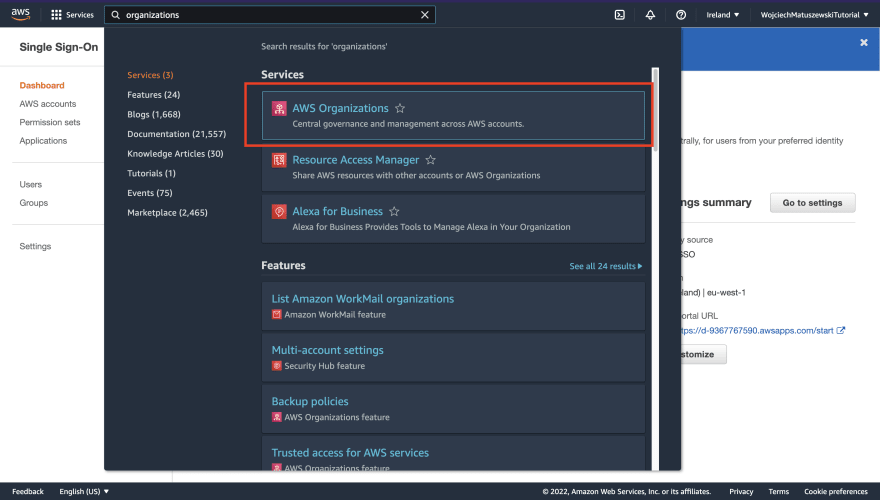
[](https://res.cloudinary.com/practicaldev/image/fetch/s--hGd4V9uF--/c_limit,f_auto,fl_progressive,q_auto,w_880/https:/dev-to-uploads.s3.amazonaws.com/uploads/articles/zlf3vk1l2hguzwjgiv8o.png)

* **If you are new to AWS and have no idea what AWS Organizations are**, consider them a way of creating multiple AWS accounts with ease. **You do NOT have to know a lot about them to configure AWS SSO**. Consult the [AWS Organizations documentation](https://docs.aws.amazon.com/organizations/latest/userguide/orgs_introduction.html) for more information.

After waiting a bit for the AWS console to create the AWS Organization, you will be redirected to the AWS SSO service dashboard.

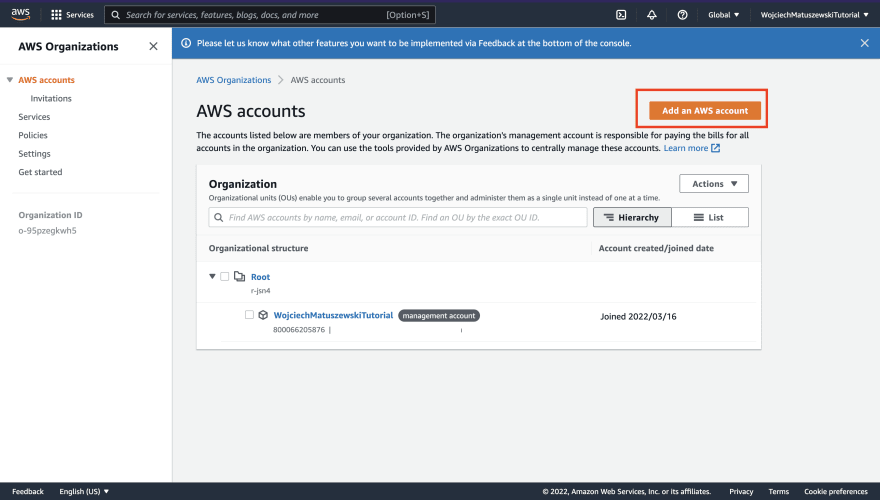
## ****6**** – Open AWS organizations

Next up, we need to create an AWS account we will be SSOing into (and assuming given policy in that account, more on that later). As I eluded earlier, we will be using AWS Organizations to manage AWS accounts. AWS SSO service created an AWS Organization for us in the previous step. Go to the created AWS Organization by searching for the service name.

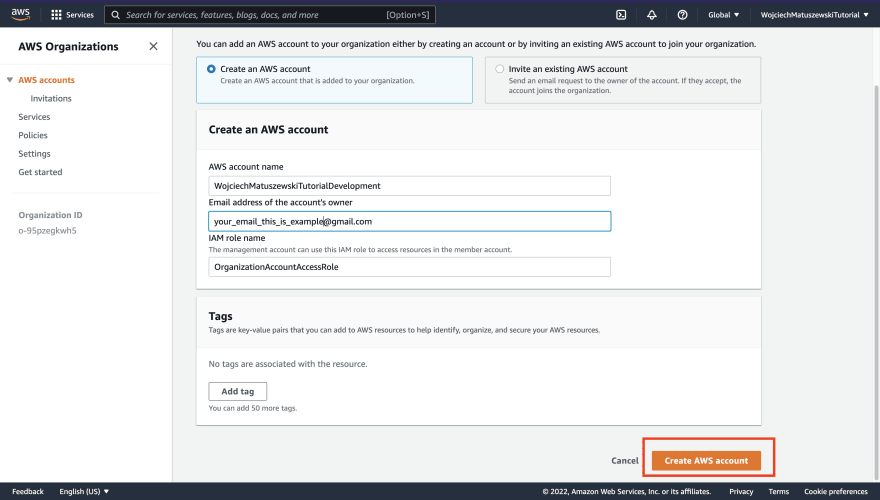
[](https://res.cloudinary.com/practicaldev/image/fetch/s--t7yN3xje--/c_limit,f_auto,fl_progressive,q_auto,w_880/https:/dev-to-uploads.s3.amazonaws.com/uploads/articles/atlo5dc1yneyf7apnyd5.png)

## ****7**** – Add an AWS account

Next up, click on the "Add an AWS account" button.

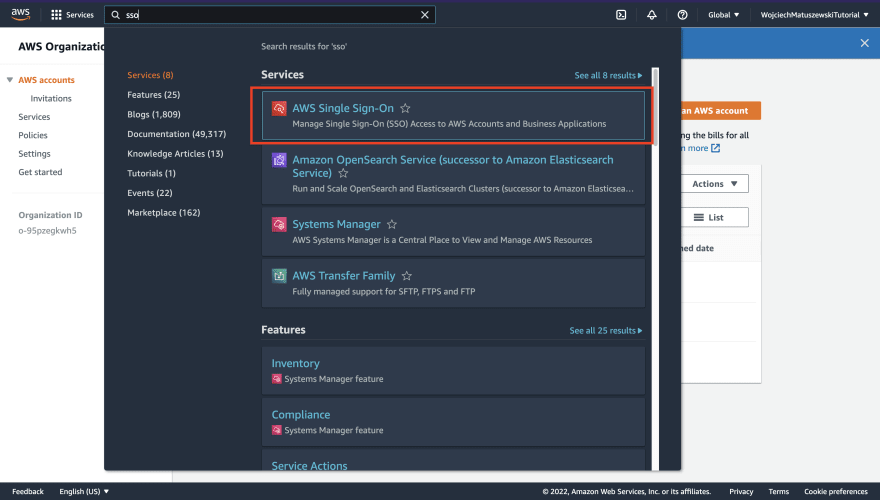
[](https://res.cloudinary.com/practicaldev/image/fetch/s--ku3o24SI--/c_limit,f_auto,fl_progressive,q_auto,w_880/https:/dev-to-uploads.s3.amazonaws.com/uploads/articles/om9qhk3uhtf2qo7ksjdi.png)

* We want to create a **new AWS account**.
* Pick whatever name you want. My recommendation would be to add the "Development" suffix to the name to signal that this is the account I'm using for playing around – i.e., it contains deployed AWS resources.
* I would leave the "IAM role name" default value. The permission model of AWS Organizations goes beyond the scope of this article.
* I would not bother with any Tags for personal-only use.

[](https://res.cloudinary.com/practicaldev/image/fetch/s--p09fBxFB--/c_limit,f_auto,fl_progressive,q_auto,w_880/https:/dev-to-uploads.s3.amazonaws.com/uploads/articles/0zheae95pb8izlk1zhlk.png)

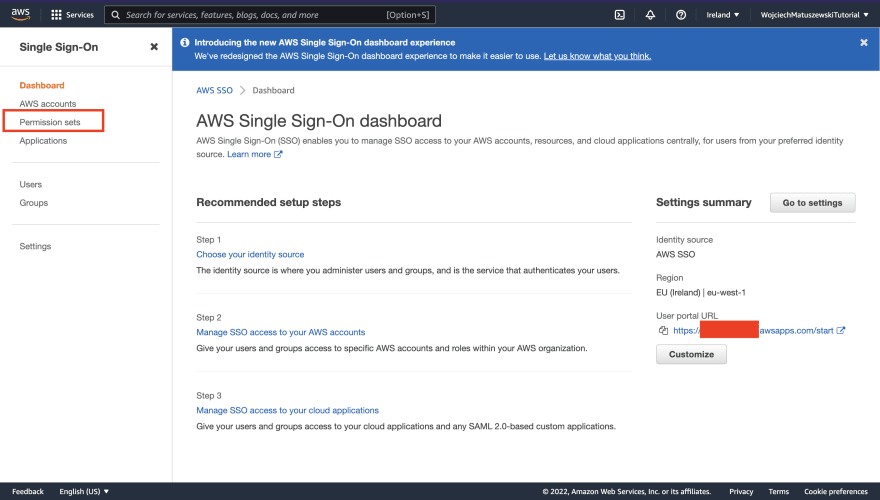
## ****8**** – Complete Configuration

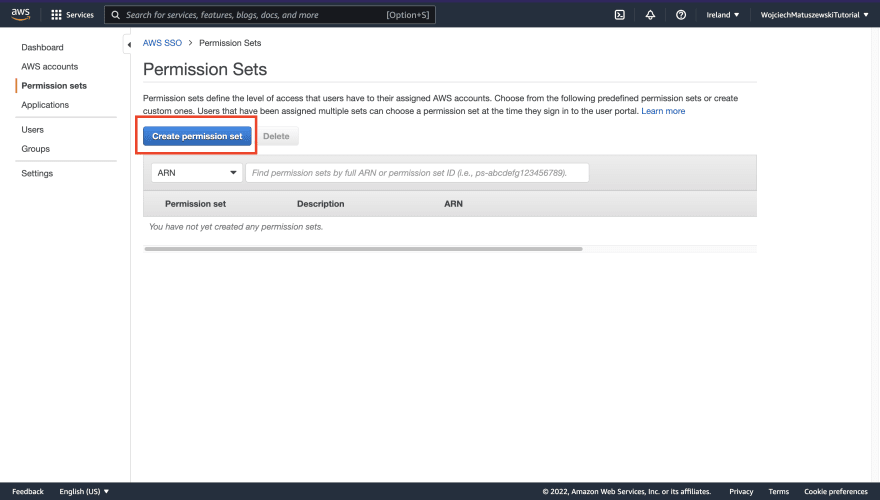
After you have successfully created the AWS account and set the password (the email you have received from AWS), it is time to go back to the AWS SSO dashboard to finish the configuration.

[](https://res.cloudinary.com/practicaldev/image/fetch/s--6jkqUDjp--/c_limit,f_auto,fl_progressive,q_auto,w_880/https:/dev-to-uploads.s3.amazonaws.com/uploads/articles/gnt5r6kqdm1bqj2jsnin.png)

## ****9**** – Create permission set

Next, we have to create a [permission set](https://docs.aws.amazon.com/singlesignon/latest/userguide/permissionsetsconcept.html) (a scary name for a collection of IAM Policies) you will be acquiring when SSOing into a given AWS account. **This part is what allows us to ditch the long-lived credentials**. Switch to the "Permission sets" tab and click on "Create permission set".

[](https://res.cloudinary.com/practicaldev/image/fetch/s--KqJRAXGM--/c_limit,f_auto,fl_progressive,q_auto,w_880/https:/dev-to-uploads.s3.amazonaws.com/uploads/articles/06u7wwpnxogsaqvpdzvi.png)

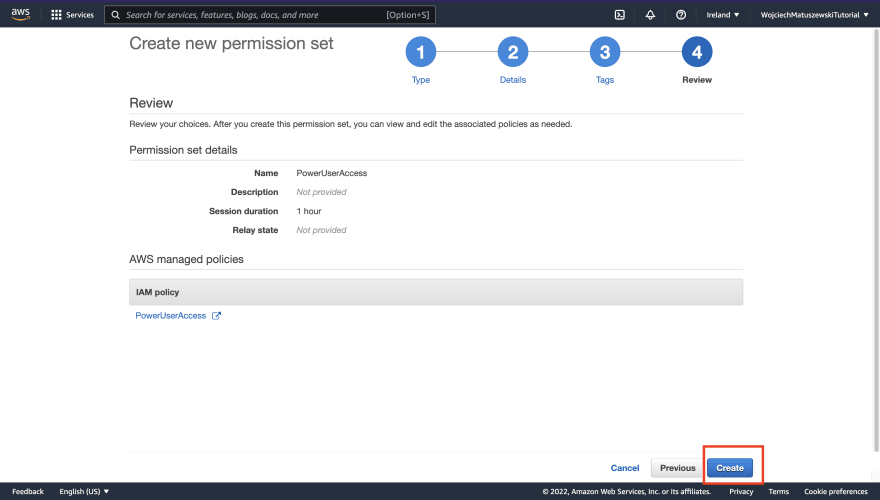
[](https://res.cloudinary.com/practicaldev/image/fetch/s--xw1Yhpjr--/c_limit,f_auto,fl_progressive,q_auto,w_880/https:/dev-to-uploads.s3.amazonaws.com/uploads/articles/iy8et8k4msvn33mch6oh.png)

## ****10**** – Create Permission setes

Now comes the permission set creation wizard. The following are my recommendations for each step.

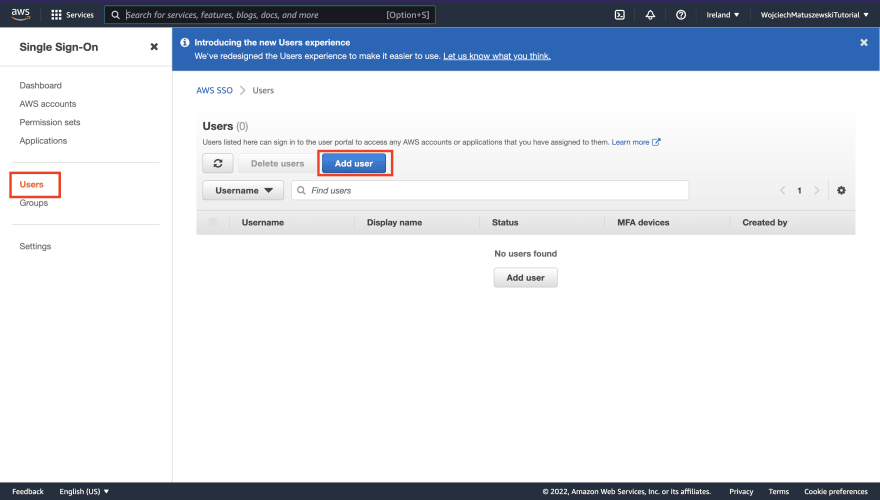
* For the "Type" step, I recommend sticking with "Use an existing job function policy" unless you are adept at writing AWS IAM Policies.
* For the "Detail" step, I recommend picking the "PowerUserAccess" policy. **Make sure you understand what a given policy grants**.
* When it comes to the "Tags" step, I would not bother setting them for personal use, but I leave that decision up to you.

Tags are essential in a work environment where multiple teams operate on different resources.

[](https://res.cloudinary.com/practicaldev/image/fetch/s--dzUZ92Y7--/c_limit,f_auto,fl_progressive,q_auto,w_880/https:/dev-to-uploads.s3.amazonaws.com/uploads/articles/lm9edogddajjuupkoida.png)

## ****11**** – Add user

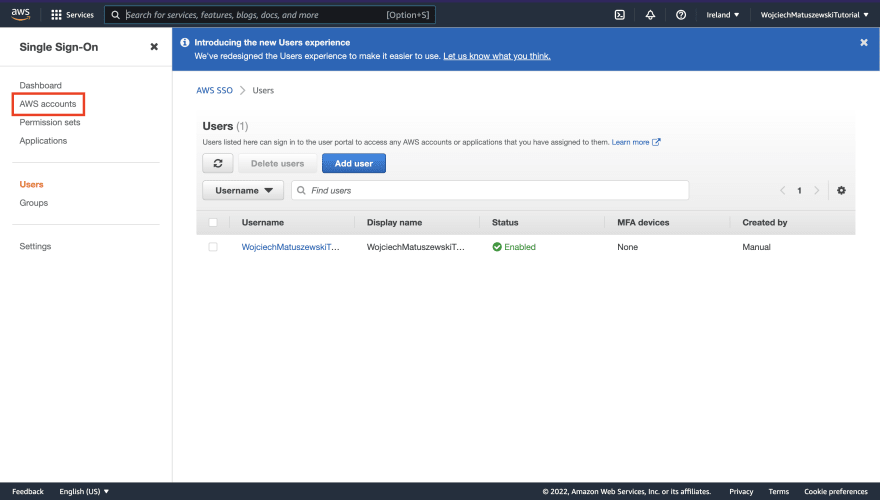
Next, we have to create a user that can sign in to an AWS SSO user portal. We will be associating the user with a permission set and AWS account later. To create an AWS SSO user, navigate to the "Users" tab and click the "Add user" button.

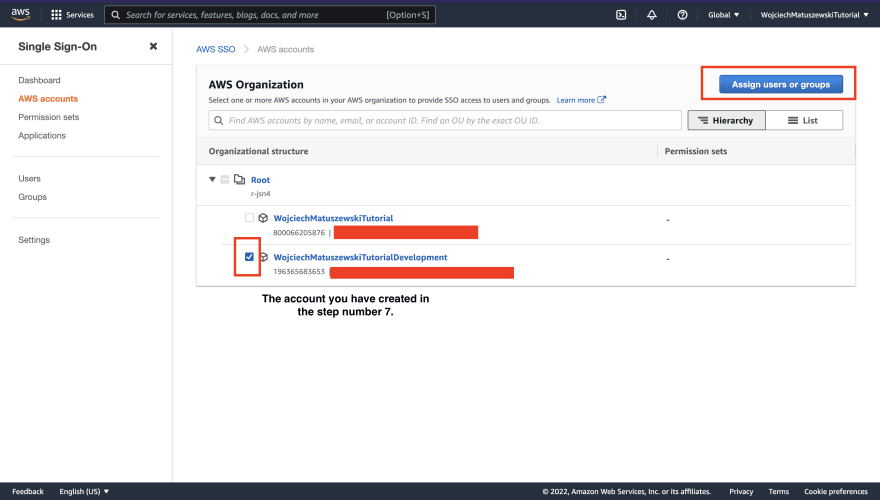
[](https://res.cloudinary.com/practicaldev/image/fetch/s--fth-Nd1t--/c_limit,f_auto,fl_progressive,q_auto,w_880/https:/dev-to-uploads.s3.amazonaws.com/uploads/articles/r97sl1scuofoi5v1s9f6.png)

* Make sure to save the username you specified in the "Specify user details" step – you will need it later on.
* I would suggest omitting creating groups for simplicity's sake, but that is up to you.
* After you have created the user, **I would strongly consider adding an MFA for that user**.

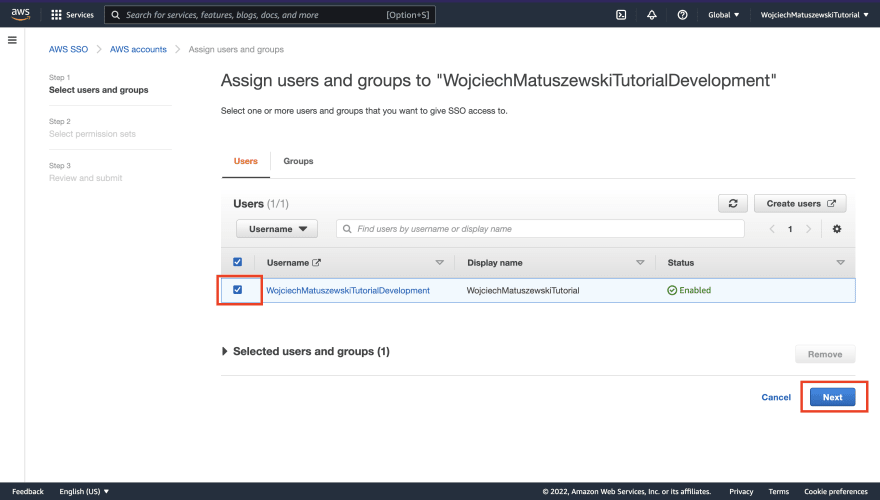
## ****12**** – Verify users

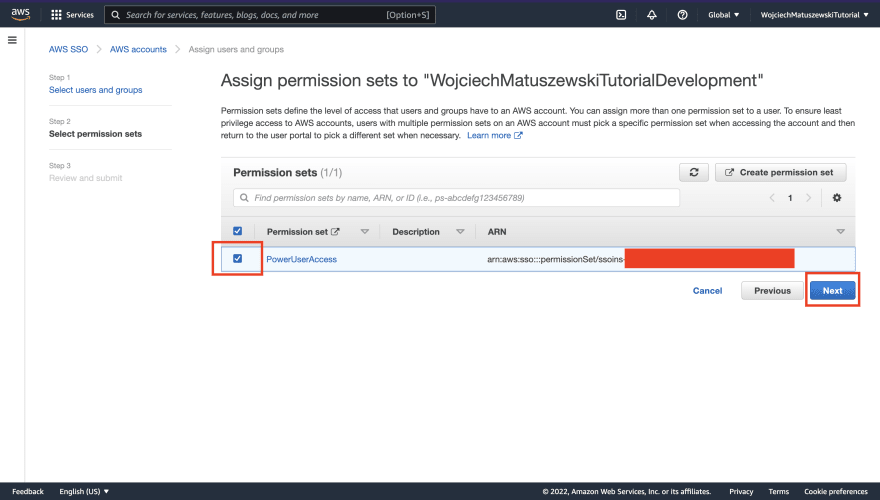
With the AWS SSO user created, it is time to tie everything we have been configuring together. Navigate to the "AWS accounts" tab. We will be associating the AWS SSO user with a given permission set and an AWS account.

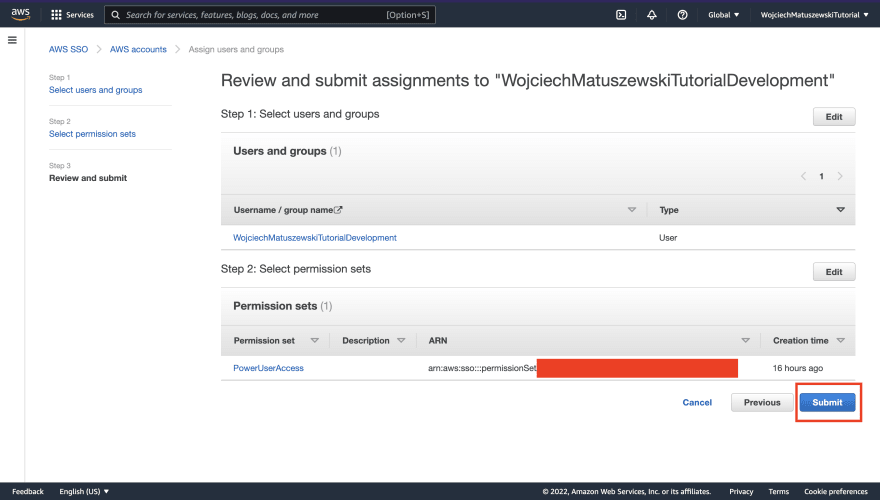
[](https://res.cloudinary.com/practicaldev/image/fetch/s--wSEIBQCz--/c_limit,f_auto,fl_progressive,q_auto,w_880/https:/dev-to-uploads.s3.amazonaws.com/uploads/articles/rrm7arirt8sse375f31r.png)

[](https://res.cloudinary.com/practicaldev/image/fetch/s--S4QZF5Zo--/c_limit,f_auto,fl_progressive,q_auto,w_880/https:/dev-to-uploads.s3.amazonaws.com/uploads/articles/6fexjk8o3pyu6ckmeyls.png)

As with AWS SSO users, the AWS console uses a wizard-style form for this configuration. The following are screenshots that should help you navigate the steps.

[](https://res.cloudinary.com/practicaldev/image/fetch/s--LG7EKpfc--/c_limit,f_auto,fl_progressive,q_auto,w_880/https:/dev-to-uploads.s3.amazonaws.com/uploads/articles/q6zp8z6hlmodgwts0tb5.png)

[](https://res.cloudinary.com/practicaldev/image/fetch/s--YRMfVh3H--/c_limit,f_auto,fl_progressive,q_auto,w_880/https:/dev-to-uploads.s3.amazonaws.com/uploads/articles/x7pvtl0vembpbaptckpk.png)

[](https://res.cloudinary.com/practicaldev/image/fetch/s--KN7GPks4--/c_limit,f_auto,fl_progressive,q_auto,w_880/https:/dev-to-uploads.s3.amazonaws.com/uploads/articles/krzhufd5c2hozf7zjca7.png)

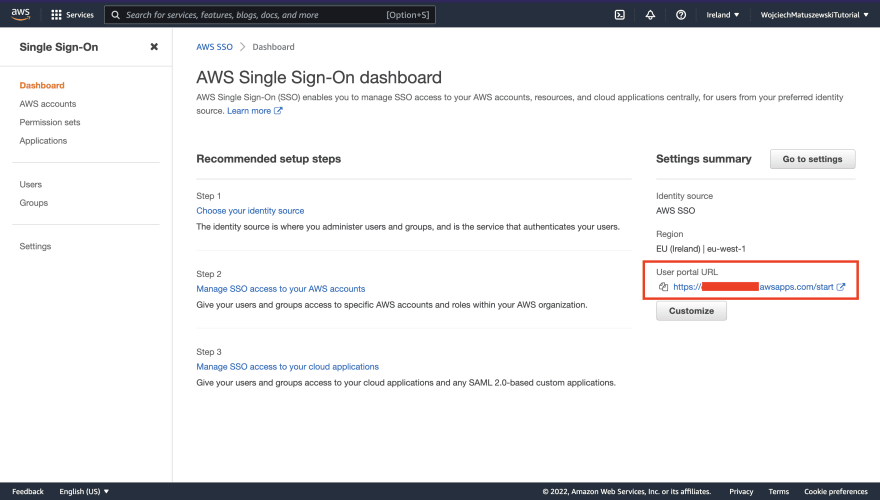
## 

## 13 – Testing

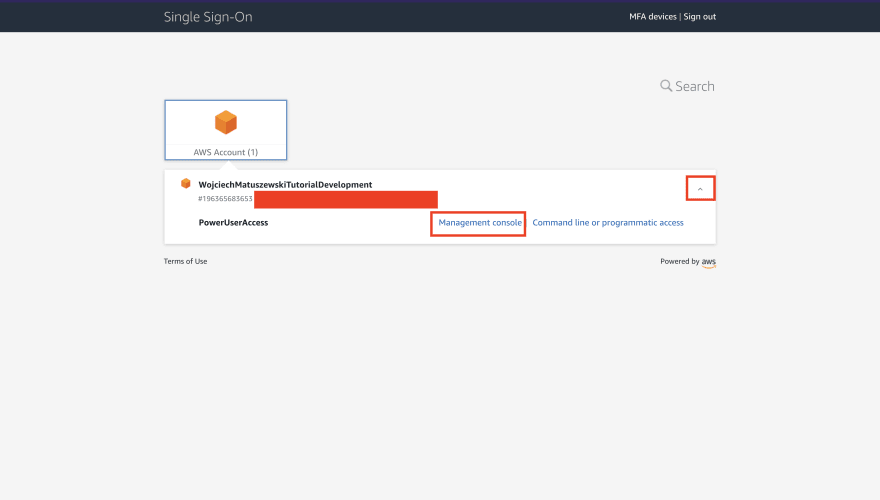
Okay, we have done everything we need to make the AWS SSO work in its simplest, from – having access to an AWS account without using long-lived credentials. Now it is time to test our setup to ensure everything is working correctly. Let us start with AWS console access using AWS SSO.

## 14 - AWS console access through AWS SSO

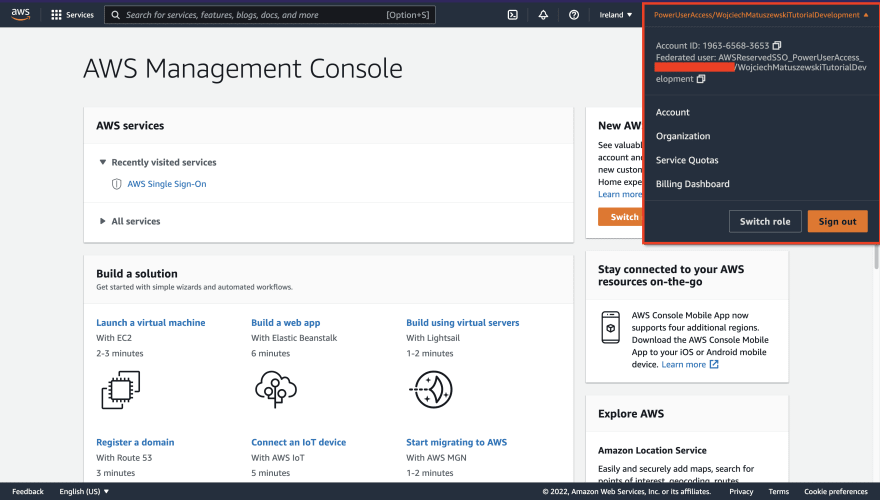
After finishing step 12 from the The steps section of this article, navigate to the AWS SSO "Dashboard" tab and copy the "User portal URL"

[](https://res.cloudinary.com/practicaldev/image/fetch/s--Ugjul8fc--/c_limit,f_auto,fl_progressive,q_auto,w_880/https:/dev-to-uploads.s3.amazonaws.com/uploads/articles/i3n4yha2z75hgudy593g.png)

After signing in successfully (**use credentials for AWS SSO user and NOT the AWS account**), you should be able to pick the AWS account and the permission set combo, we have configured in the previous section.

[](https://res.cloudinary.com/practicaldev/image/fetch/s--mT2w6cpn--/c_limit,f_auto,fl_progressive,q_auto,w_880/https:/dev-to-uploads.s3.amazonaws.com/uploads/articles/6lbapxp3x1o0fce0i6vl.png)

The "Management console" link should redirect you to a given AWS account as a federated user scoped to a given permission set.

[](https://res.cloudinary.com/practicaldev/image/fetch/s--E-kSWawM--/c_limit,f_auto,fl_progressive,q_auto,w_880/https:/dev-to-uploads.s3.amazonaws.com/uploads/articles/ra74zap8vfdk7s90uesf.png)