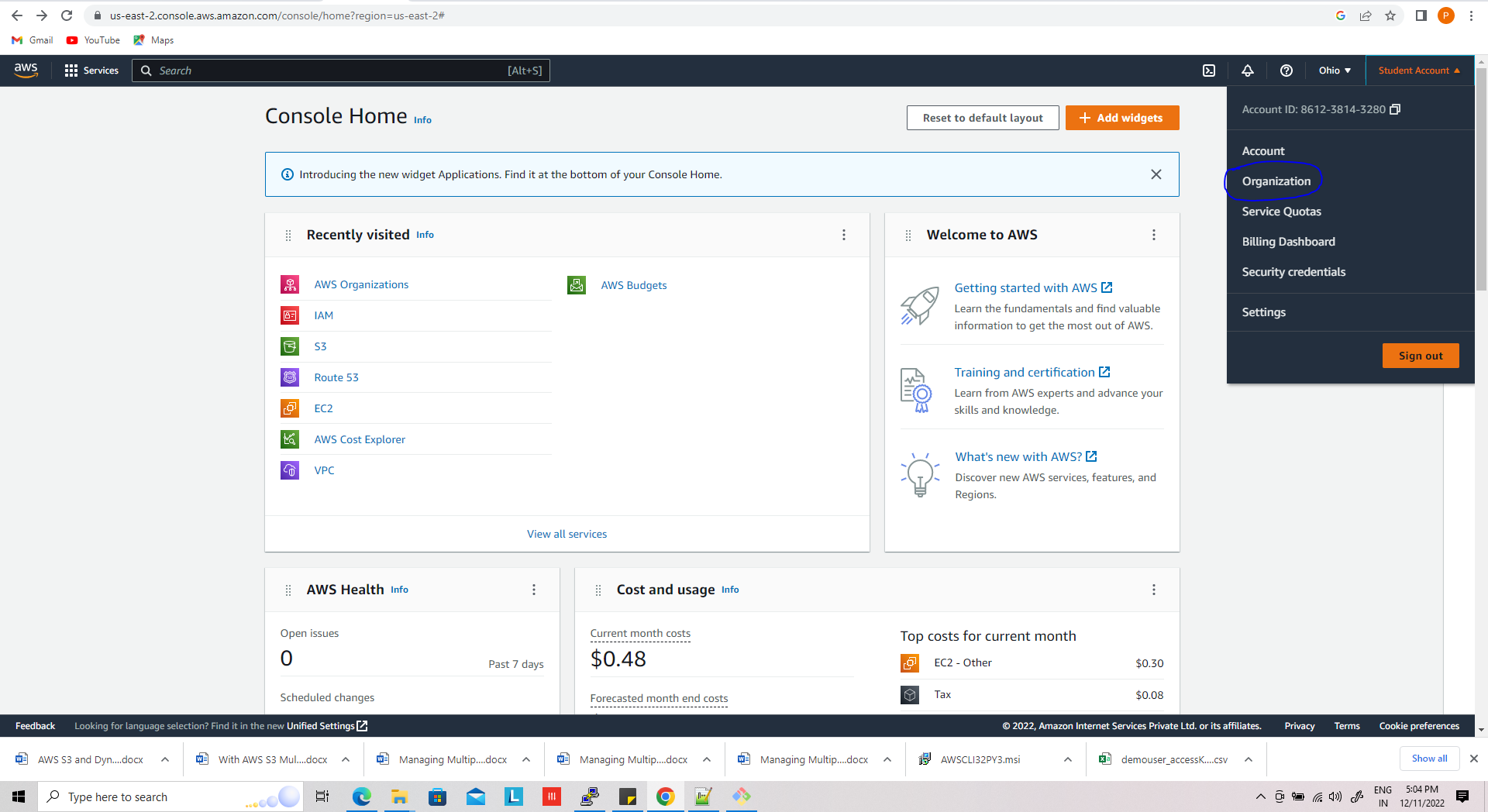
Create an AWS Org. for managing multiple accounts , grouping , budgeting and centralized way to handle all the things

**AWS Organizations solves this problem by allowing you to centrally manage and control 100s of AWS accounts efficiently from a single interface.**

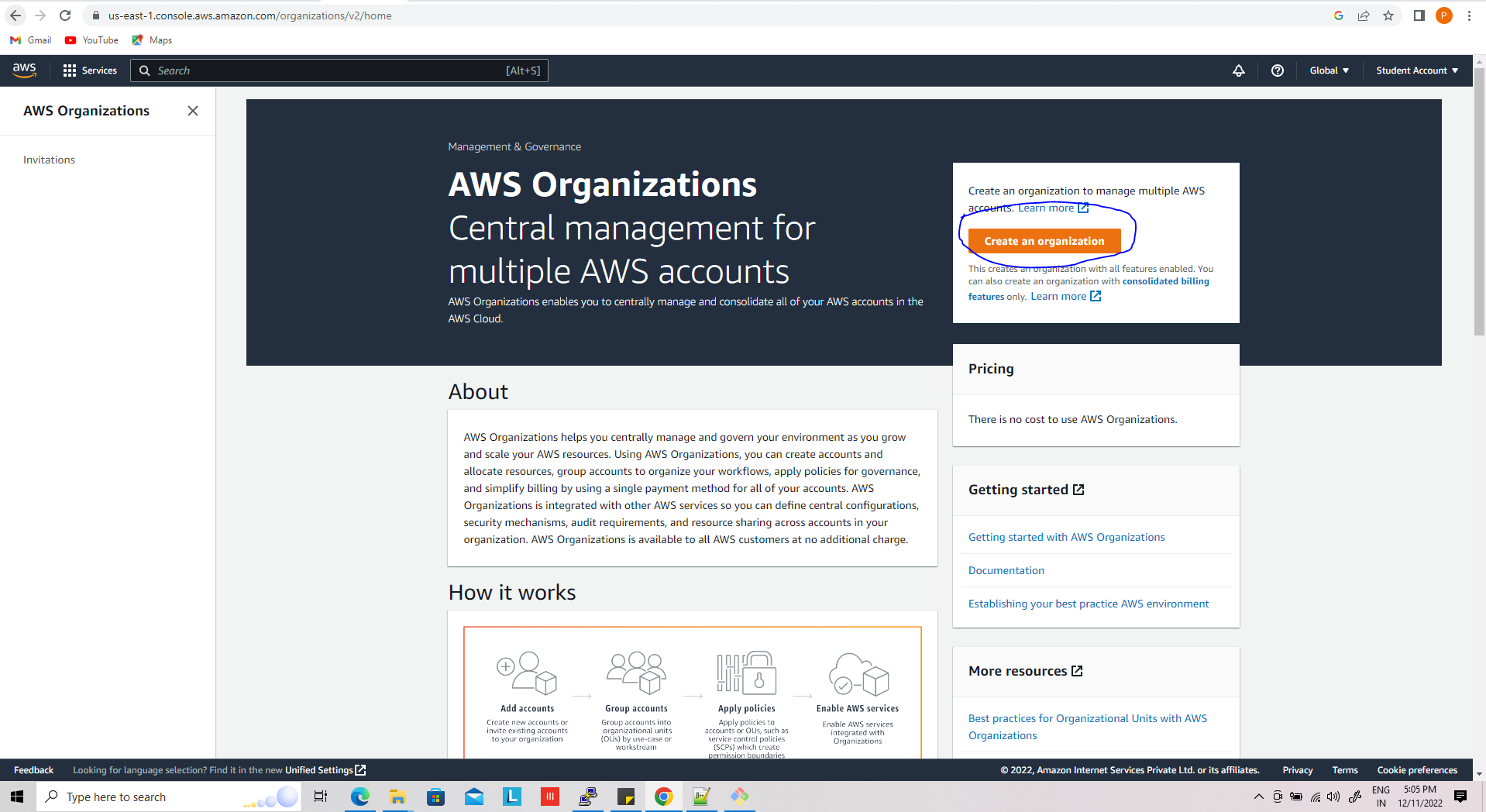
**AWS Organizations is an account management service that allows you to centrally manage multiple AWS accounts. It enables you to better meet budgetary, security, and compliance needs as an administrator of an organization.**

**Some of the features and benefits of using AWS Organizations include:**

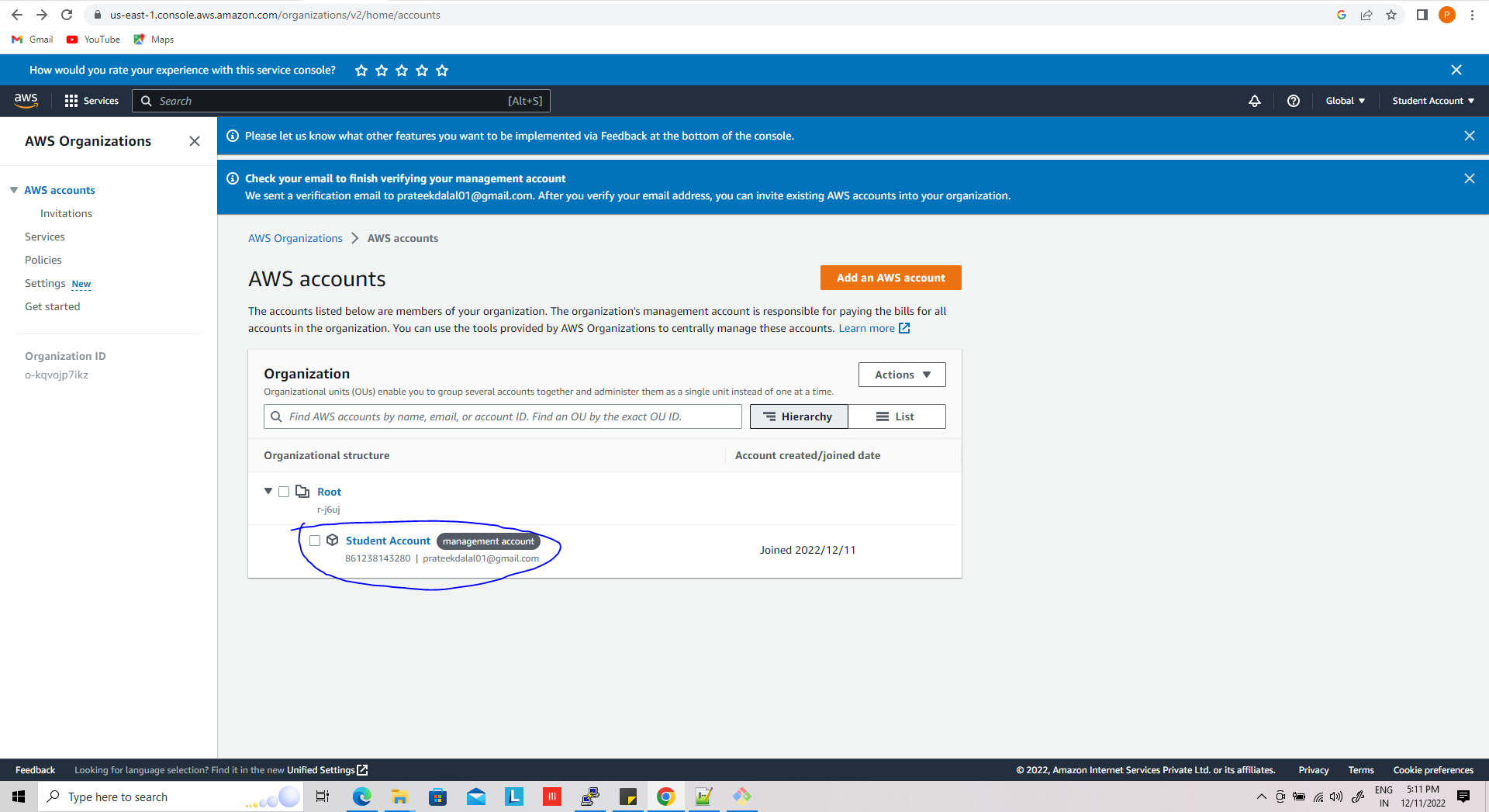
* **Centralized management of all of your AWS accounts**
* **Consolidated billing for all member accounts within the organization**
* **Hierarchical grouping of your accounts using Organization Units (OUs) to meet your budgetary, security, or compliance needs**
* **Policies to centralize control over the AWS services and API actions that each account can access**
* **Policies to standardize tags across the resources in your organization’s accounts**
* **Policies to control how AWS artificial intelligence (AI) and machine learning services can collect and store data.**
* **Policies that configure automatic backups for the resources in your organization’s accounts**
* **Integration and support for AWS Identity and Access Management (IAM)**
* **Running tasks and AWS services on all accounts that are members of the organization**
* **Its API plays a key role when you’re building a** [**landing zone**](https://towardsthecloud.com/landing-zone) **or a cloud center of excellence as part of your cloud adoption journey.**



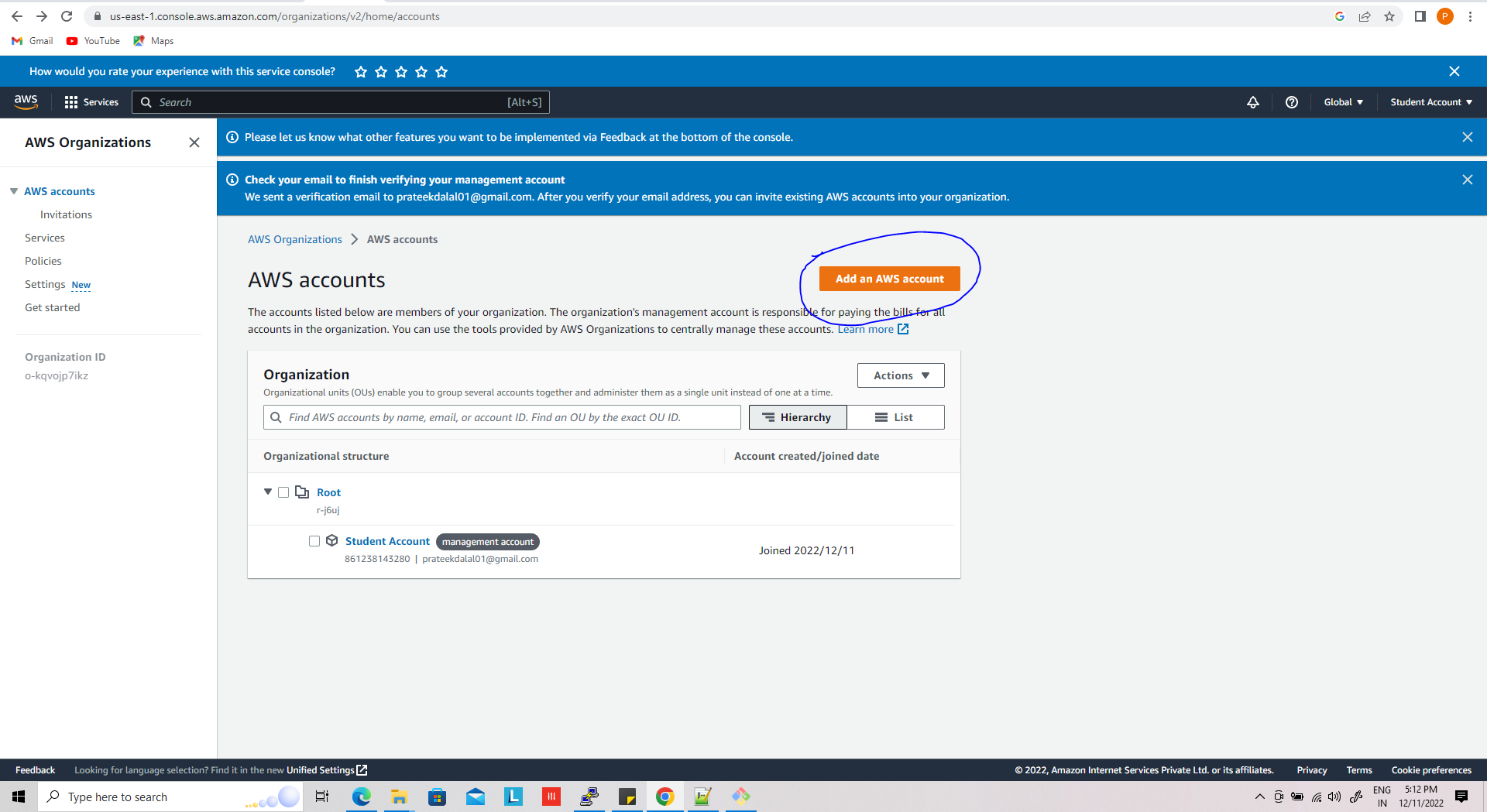
Click on drop down and choose organization



Once org has been created you see,



Now add new account just clicking on add aws account



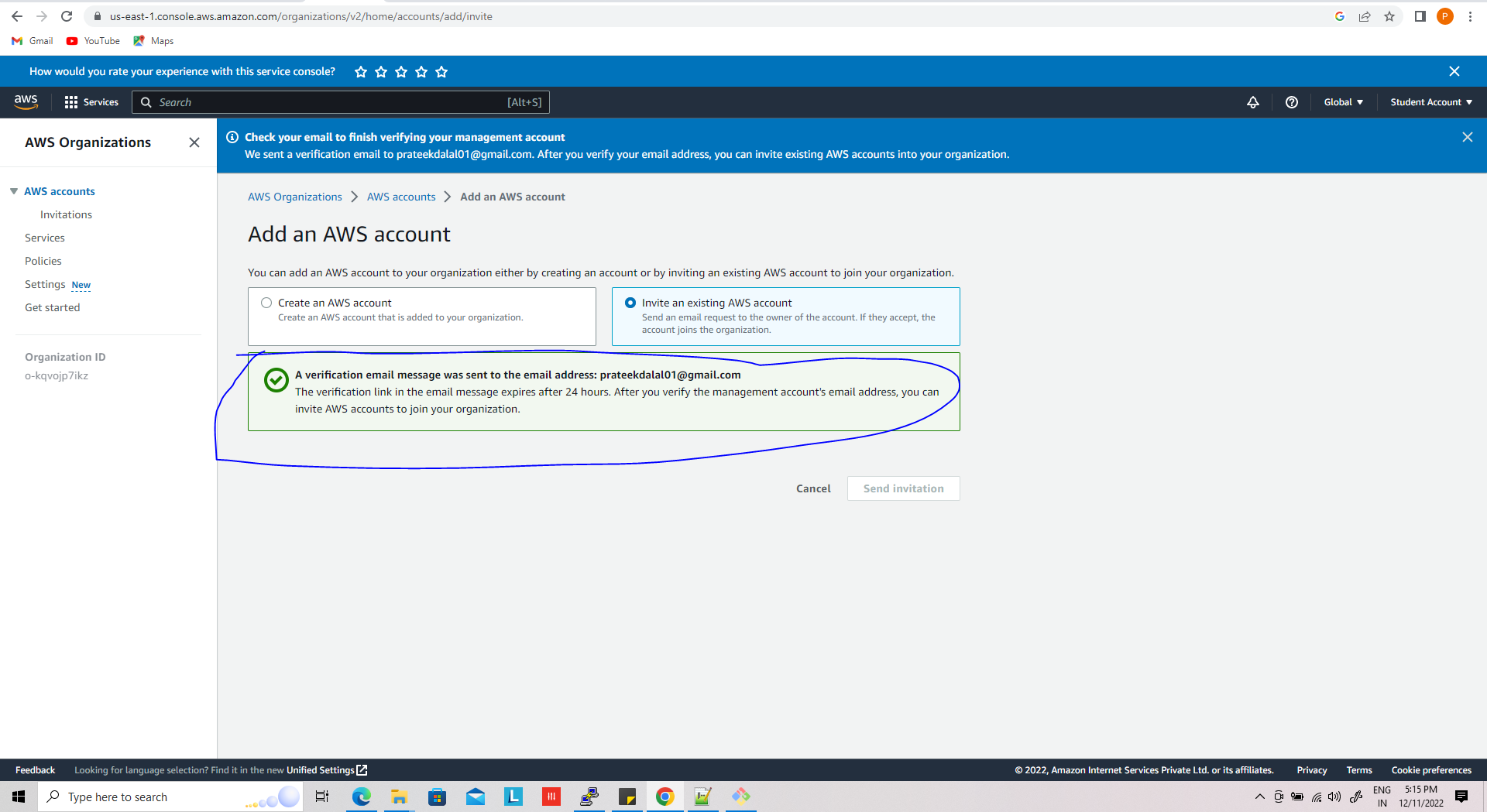
You can see two option , create new and add existing aws account,



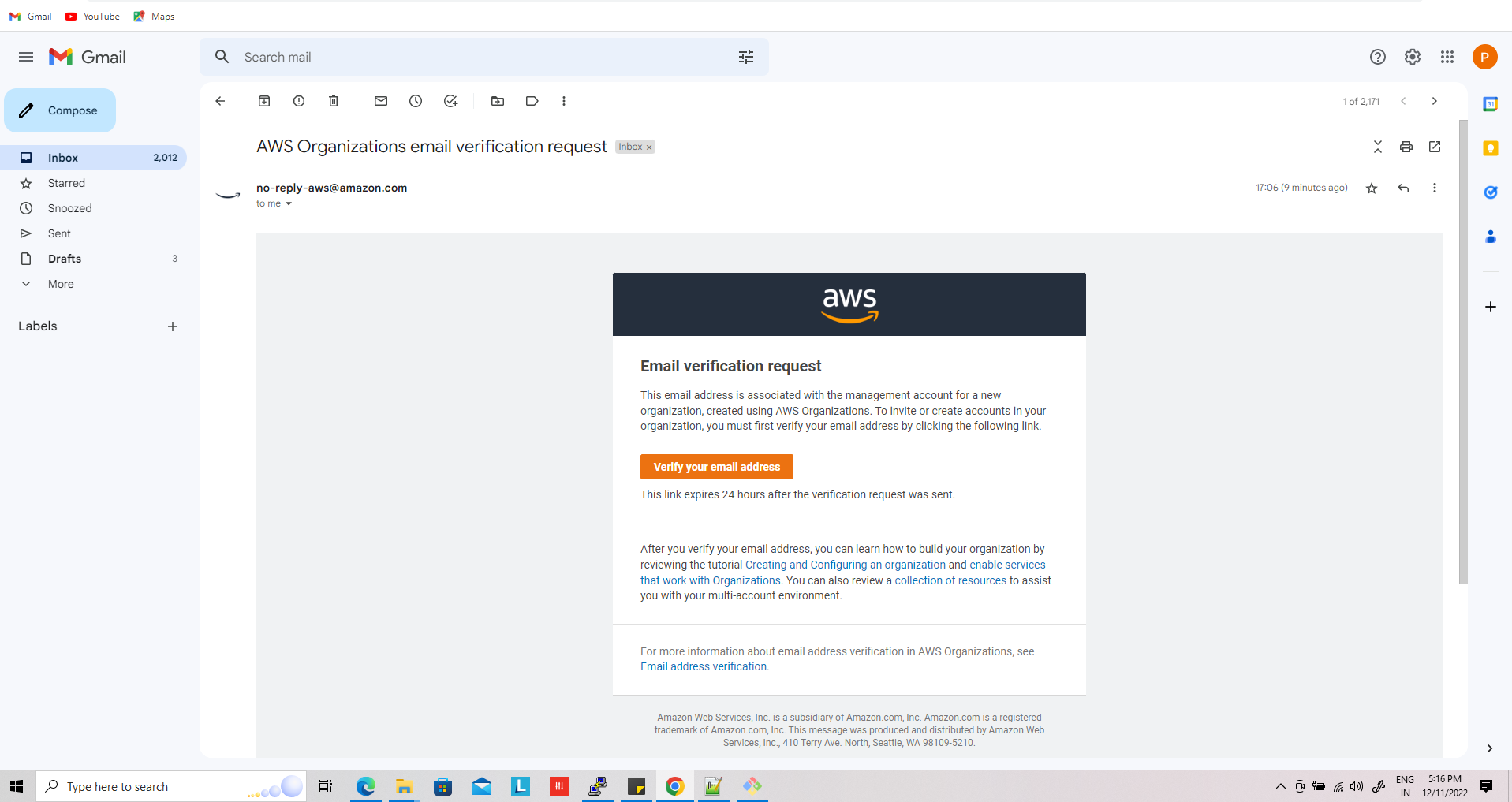
When you choose adding existing account you get following option,



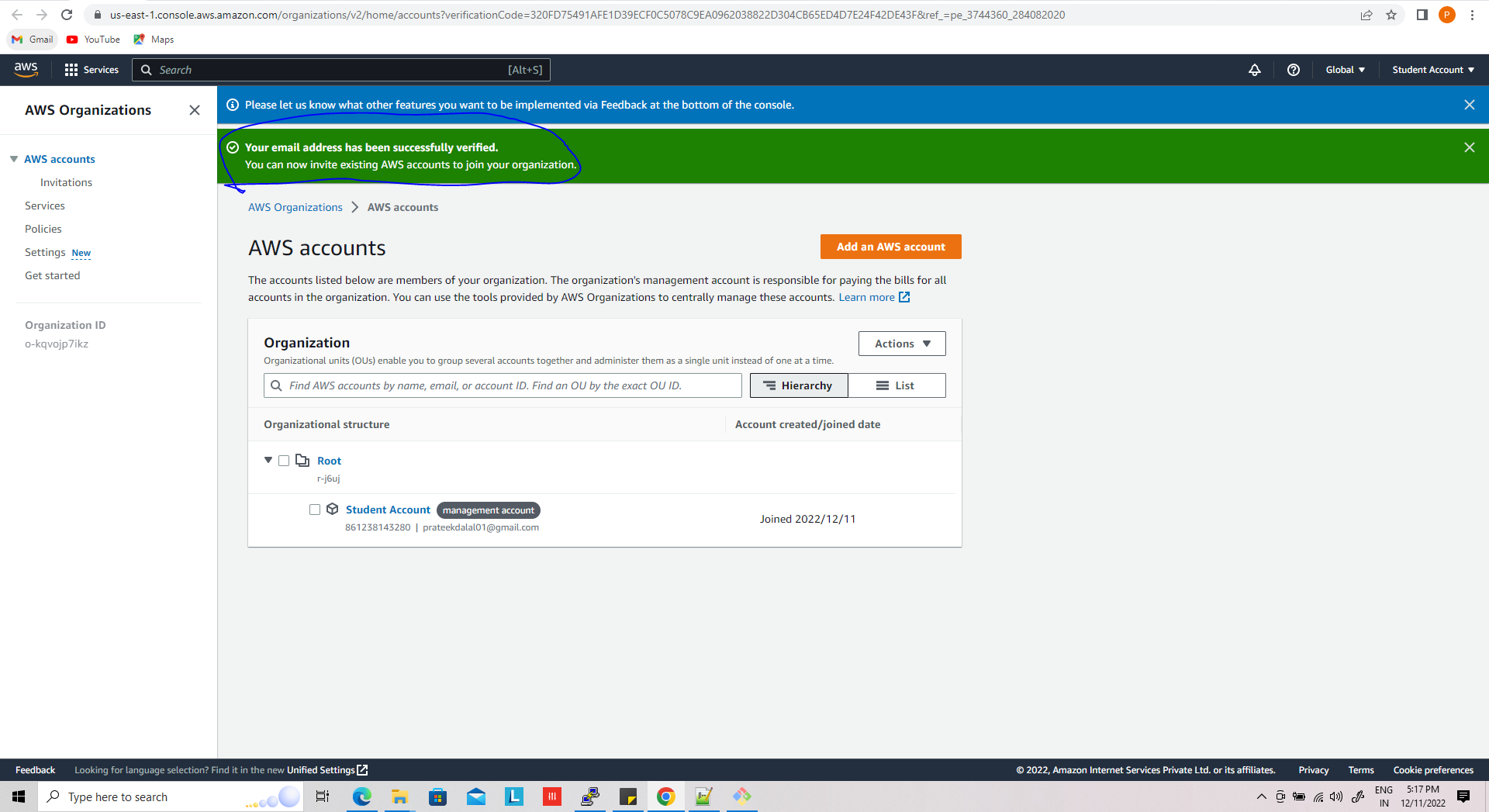
This will send invitation to your email id just accept and be a part of this aws org.



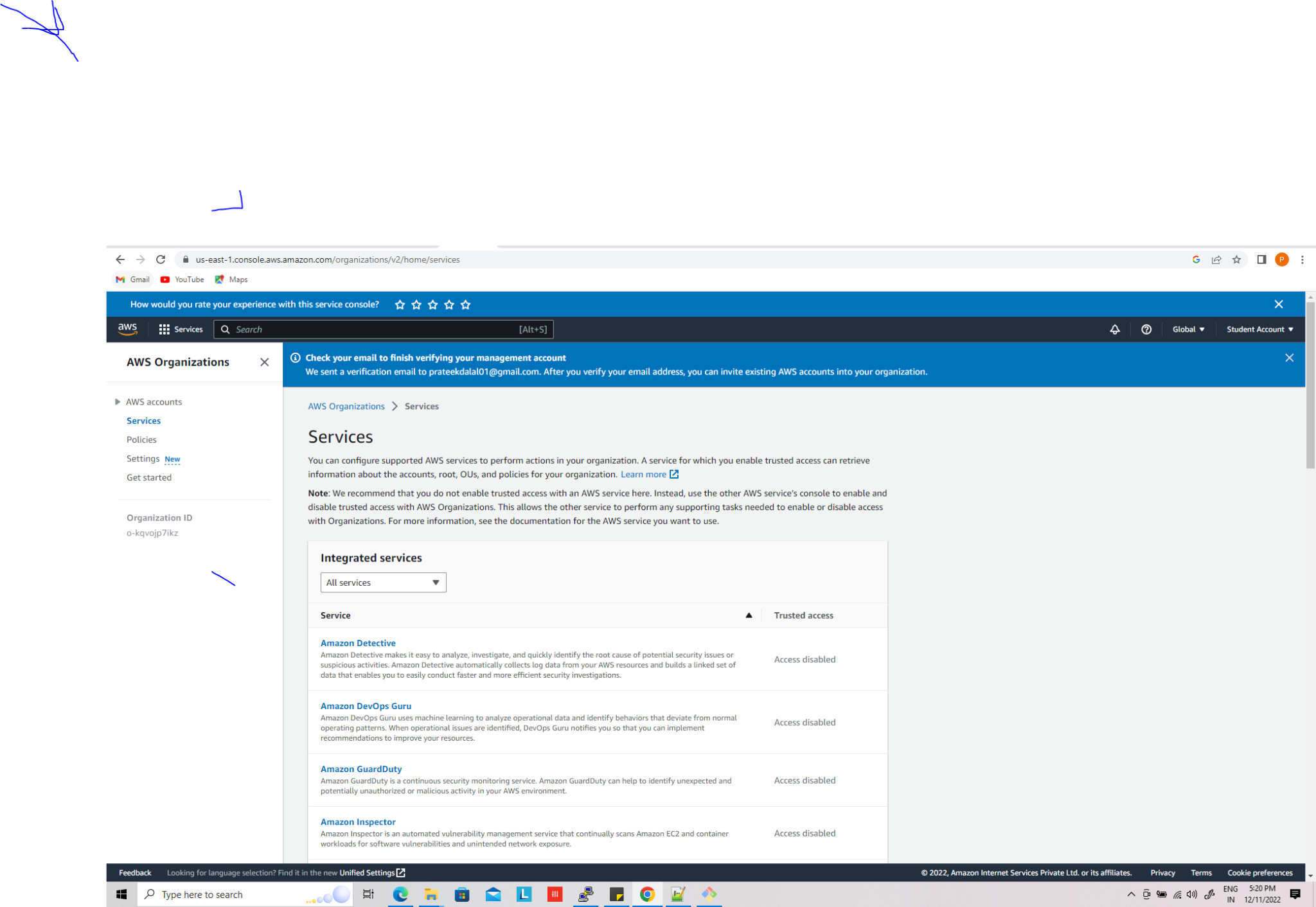
You can see in the gmail,



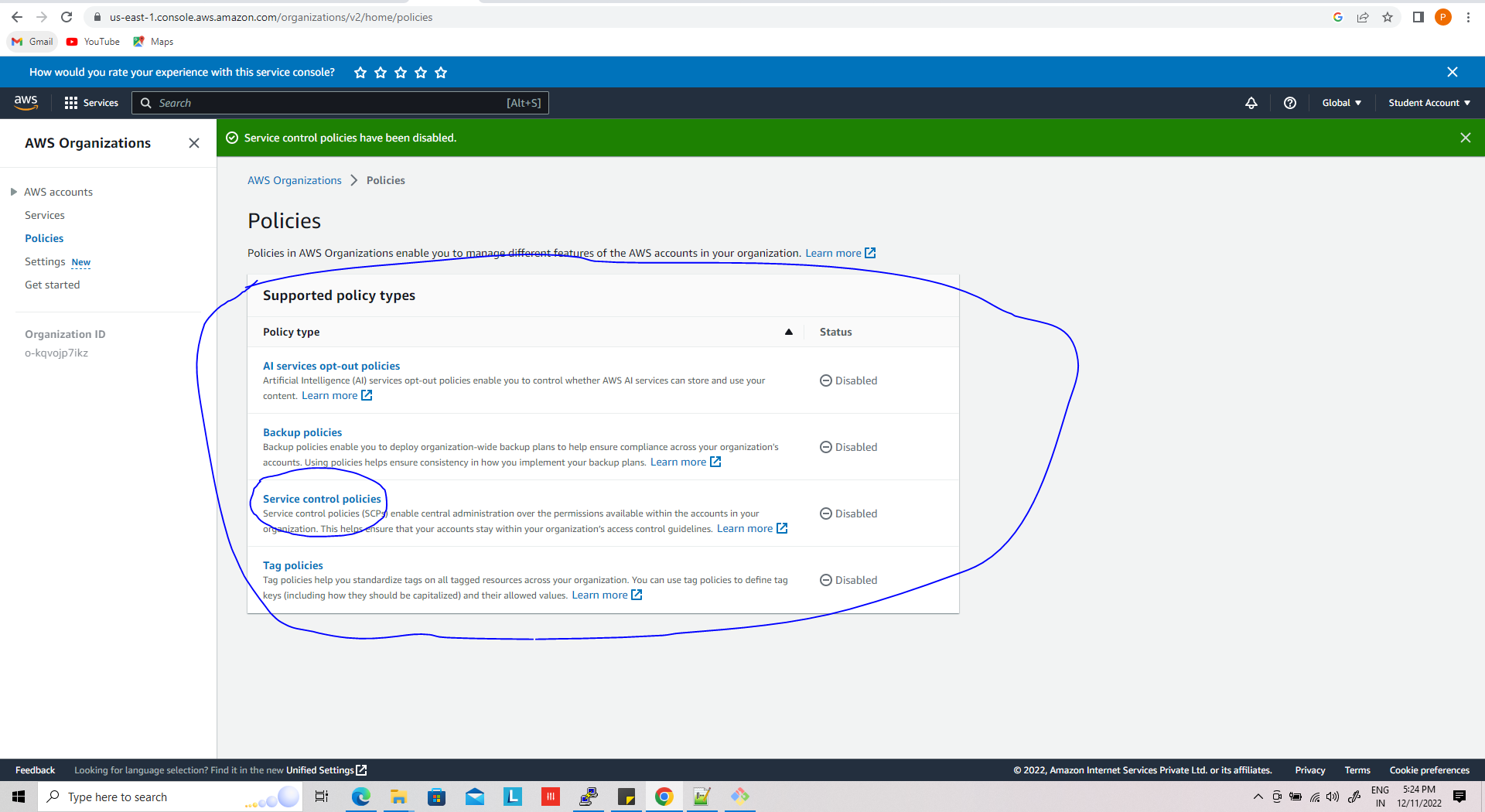
Once verification has been done you see like this



After verification , you can add as many as services to the account,



Now you can choose policies , and enable as per your need



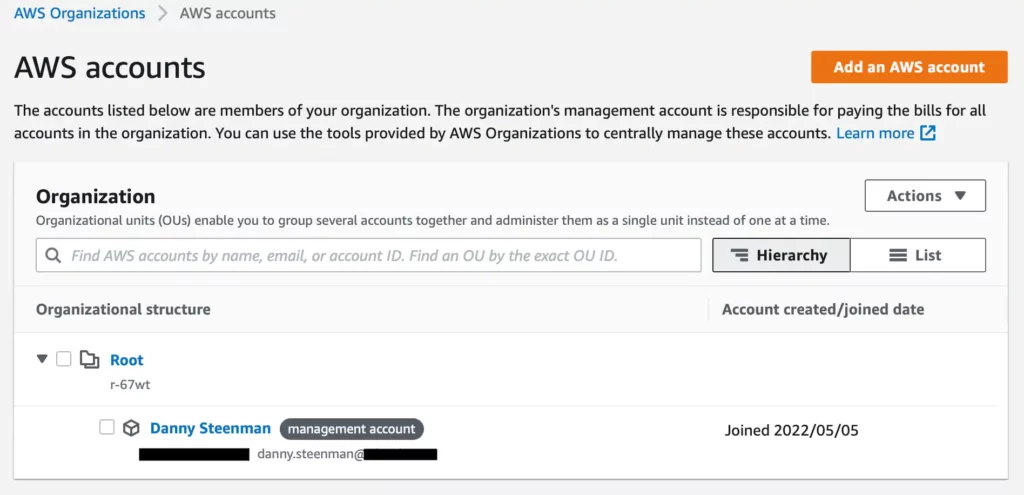
**Advance concept in AWS ORG**

## **Create an AWS Organization on the management account**

**To start an AWS Organization, simply head over to** [**AWS Organizations in the AWS console**](https://us-east-1.console.aws.amazon.com/organizations/v2/home/) **and press Create an organization**

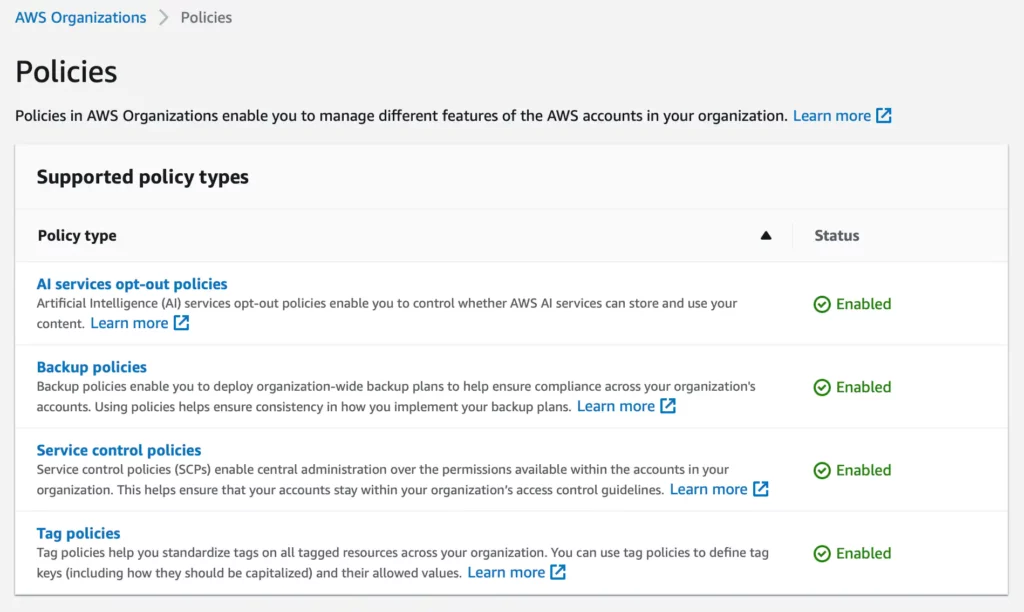
****

**Once you’ve created an AWS Organization you’ll see the organization structure in the** [**AWS Organizations > AWS accounts page**](https://us-east-1.console.aws.amazon.com/organizations/v2/home/accounts)

****

### **Enable all AWS Organization policies**

**After creating the AWS Organization you need to enable all the policies in the** [**AWS Organizations > Policies page**](https://us-east-1.console.aws.amazon.com/organizations/v2/home/policies)**, so you can make effective use of all its features.**

****

**The following policies are available:**

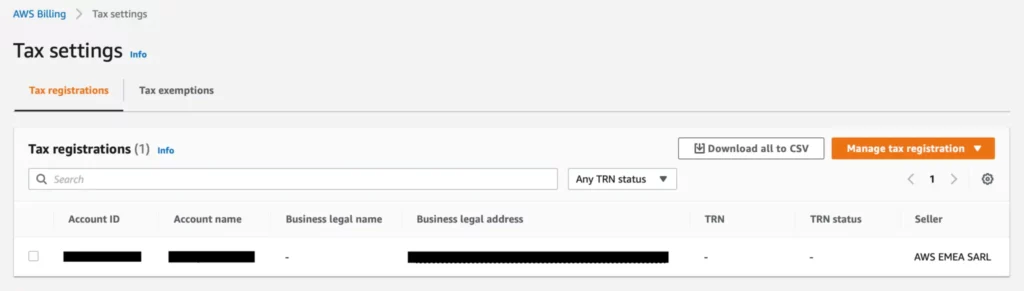
* **AI services opt-out policies – Artificial Intelligence (AI) services opt-out policies enable you to control whether AWS AI services can store and use your content.**
* **Backup policies – Backup policies enable you to deploy organization-wide backup plans to help ensure compliance across your organization’s accounts. Using policies helps ensure consistency in how you implement your backup plans.**
* **Service Control Policies – Service control policies (SCPs) enable central administration over the permissions available within the accounts in your organization. This helps ensure that your accounts stay within your organization’s access control guidelines.**
* **Tag policies – Tag policies help you standardize tags on all tagged resources across your organization. You can use tag policies to define tag keys (including how they should be capitalized) and their allowed values.**

### **Turn on tax inheritance for AWS billing in your AWS Organization**

**Tax inheritance allows you to configure tax settings once for the whole organization and consolidate it for all existing and new AWS accounts that join the organization. This effectively saves you time that you don’t have to spend doing administrative tax tasks when you create a new AWS account.**

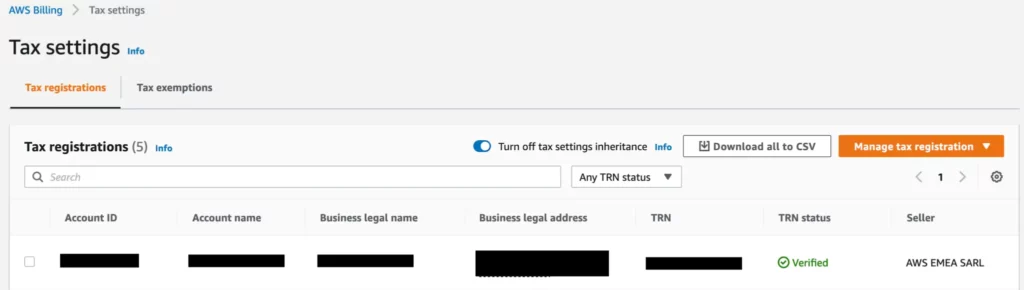
**To enable it, go to the** [**billing dashboard**](https://us-east-1.console.aws.amazon.com/billing/home#/tax-settings) **and configure the tax settings first.**

**As you can see at first, the tax settings haven’t been verified yet.**

****

**Click on Manage tax registration in the top right corner to configure the customer’s tax settings. Follow the wizard and save the settings.**

**It takes a couple of minutes for AWS to process and verify your tax registration. Once that succeeded you can proceed and enable the toggle Turn on tax settings inheritance**

****

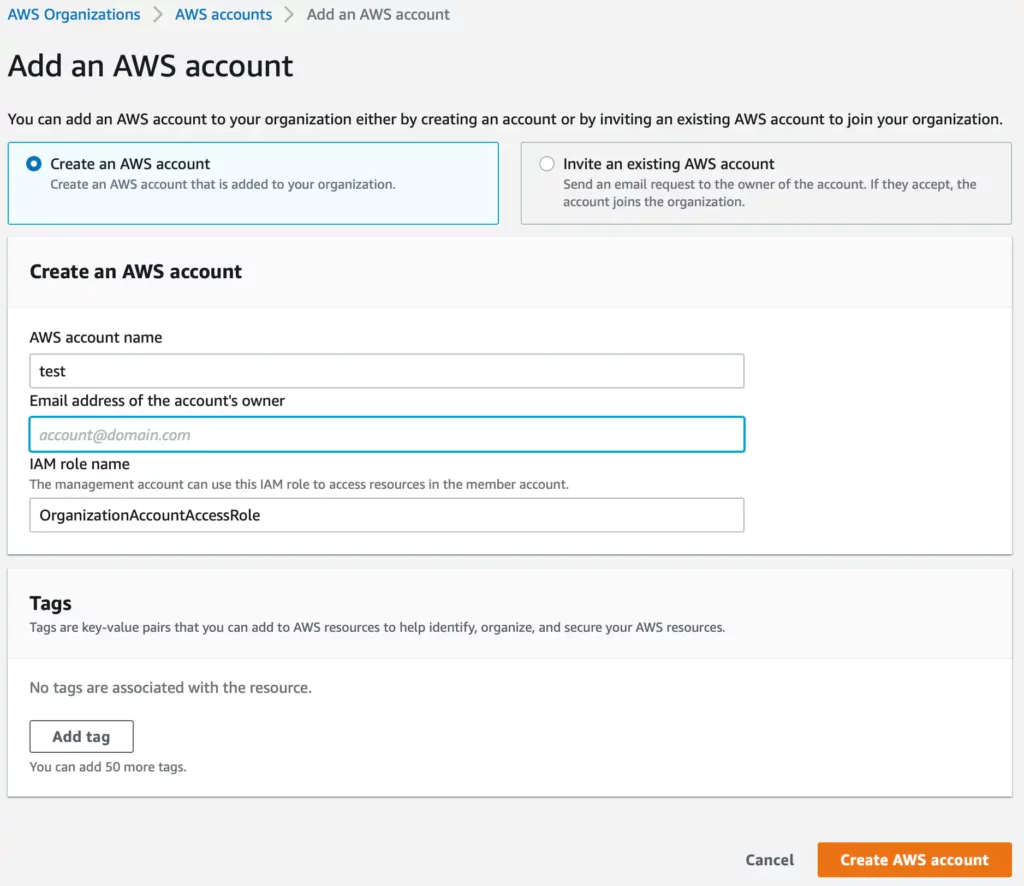
**By turning on tax inheritance, it will automatically fill in the tax settings for any new account that you create within an AWS Organization.**

## **Create an AWS account in AWS Organizations**

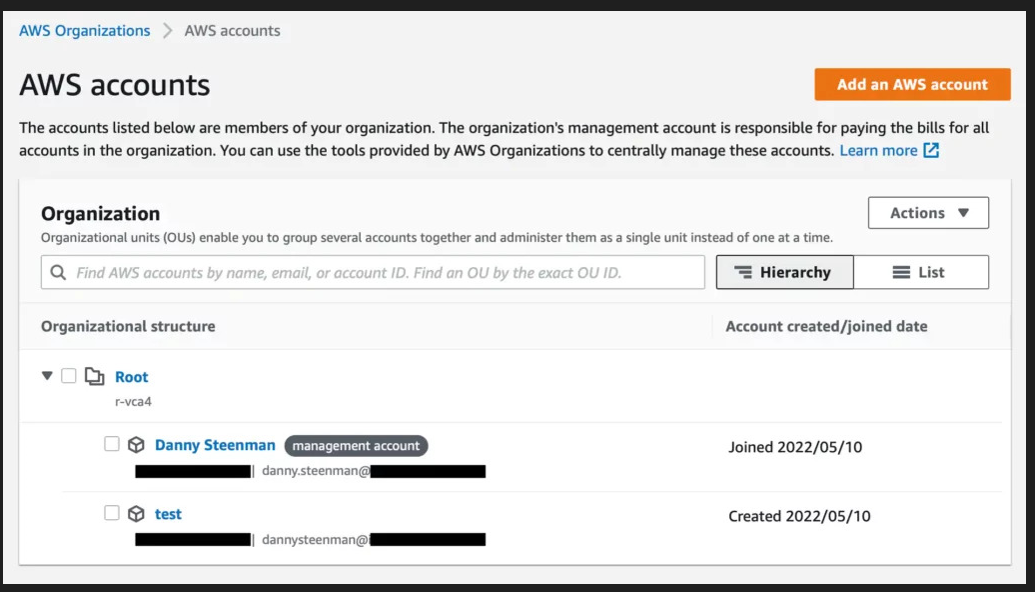
**Now that we’ve enabled AWS Organizations and set up the required tax settings we can easily create new AWS accounts in the** [**AWS Organizations dashboard**](https://us-east-1.console.aws.amazon.com/organizations/v2/home/accounts) **by pressing the Add an AWS account button.**

****

**You can proceed and fill in the account name and email address.**

****

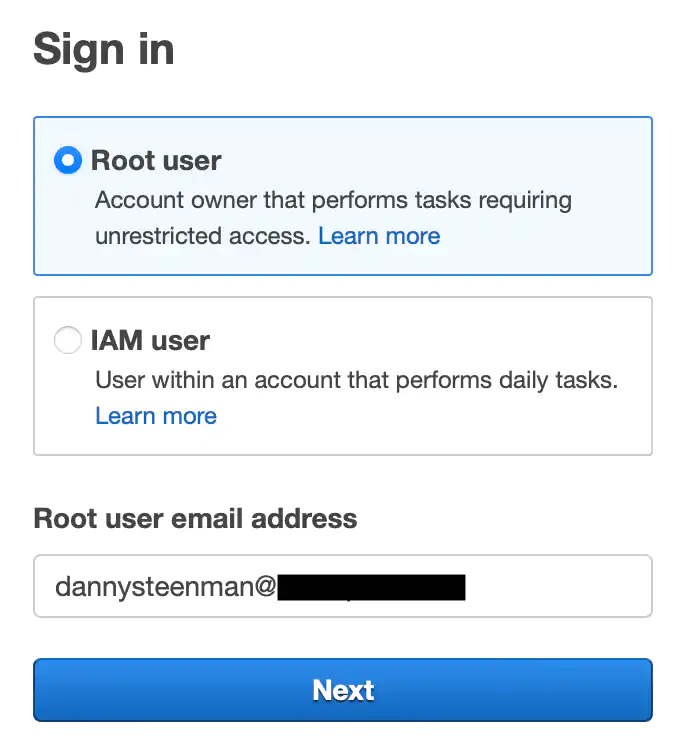
**Once you click the Create AWS account button it will proceed and create the account for you and add it to the root of your AWS Organization.**

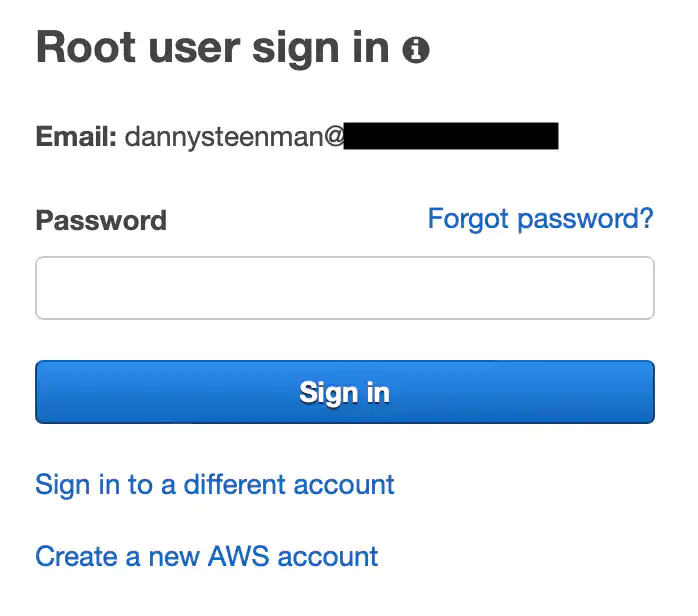


As you can see once, the AWS account is created it will be visible on the **AWS Organizations > AWS Accounts** page in the AWS Console.

### **Sign in to the newly created AWS account**

In the meantime, you’ll receive an automated email from Amazon stating that the new AWS account is ready. You can go ahead and activate it by going to the [AWS Console sign-in page](https://console.aws.amazon.com/console/home?nc2=h_ct&src=header-signin). Select **sign-in as root** using the email of the new AWS account you created and then follow up with the **forgot password** step.

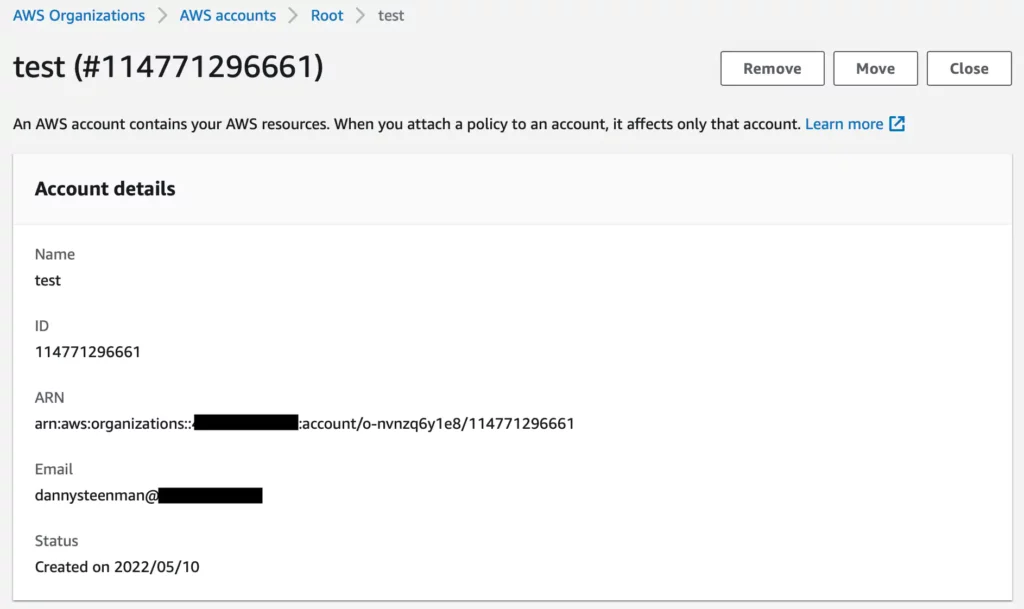




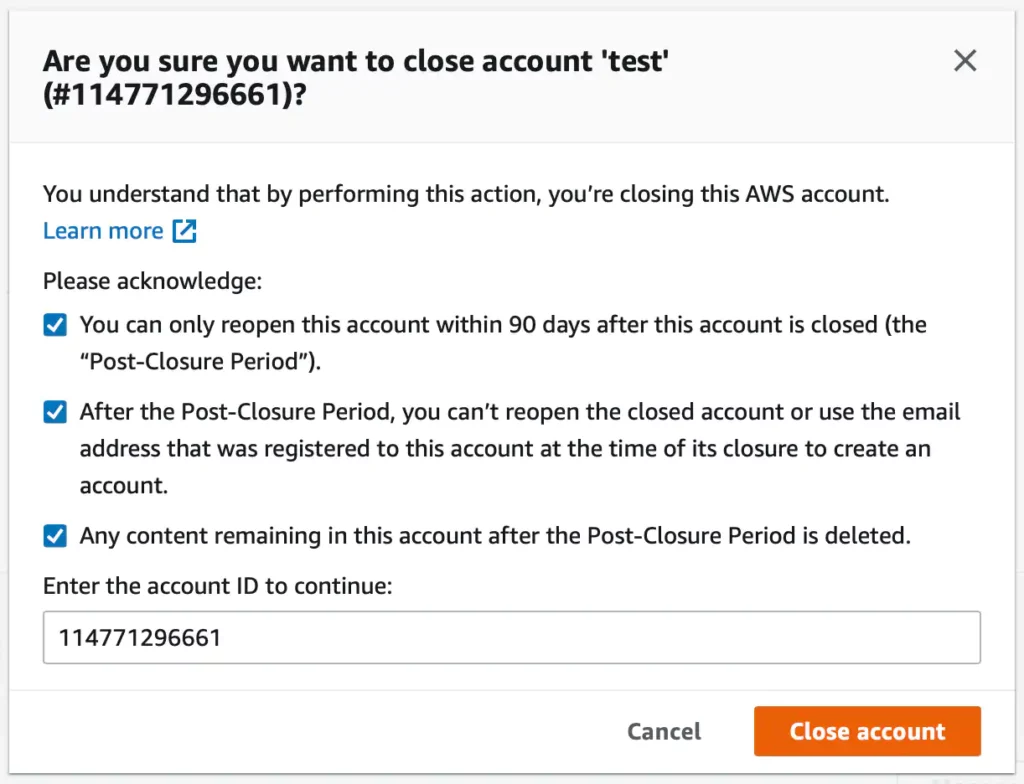
You’ll receive an email with a new password and then you can proceed to sign in. Once you’re signed in make sure to secure the new root account again by following the instructions as explained in the step [Secure root user](https://towardsthecloud.com/blog/aws-organizations-guide#secure-root-user)

## **Delete an AWS account in AWS Organizations**

Deleting an AWS account can be done from the [AWS Organizations > AWS accounts page](https://us-east-1.console.aws.amazon.com/organizations/v2/home/accounts). Click on the account you wish to delete and press close



You’ll get a confirmation prompt to validate if you want to close the AWS account permanently. Check the boxes and click **Close account**

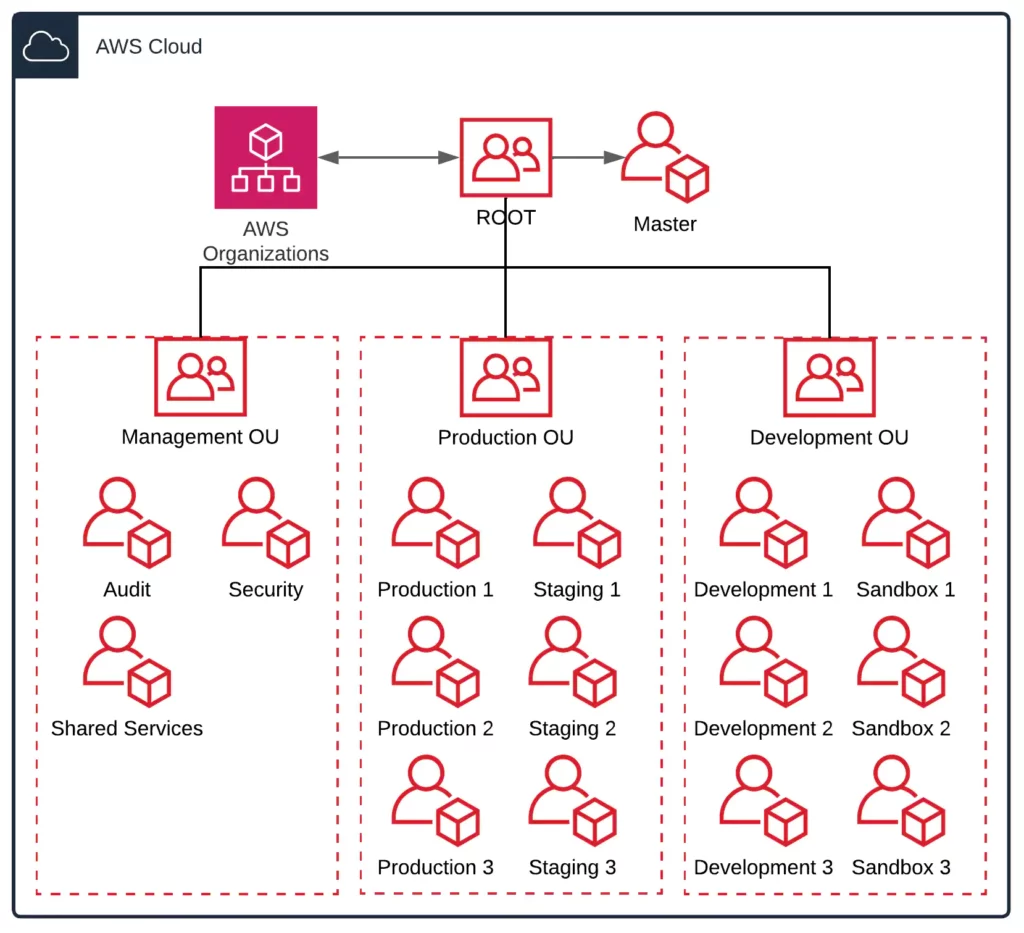
****

## **Best practices for organizing AWS accounts in AWS Organizations**

**Now that you know how you can set up AWS Organizations for your** [**own accounts**](https://towardsthecloud.com/how-to-find-aws-account-id)**, it’s time to dive deeper and focus on setting up a correct structure that allows you to control your accounts in a more efficient and effective way.**

**When you initially create an AWS Organization you start out with the root of that organization that holds all member AWS accounts. As a best practice, you’ll want to create a hierarchical structure and group similar accounts based on their function using Organizational Units (OUs).**

**A typical medium size business that manages 10-30 AWS accounts can use an AWS Organization structure like the below:**

****

**A few key takeaways:**

* **Avoid nested OUs when possible – You can nest up to 5 levels deep under the root within the organization. Setting up a nested structure can cause confusion on what restrictions apply to certain member accounts that are nested in multiple OUs. For example, if you apply a Service Control Policy on the parent OU, the nested OU automatically inherits all the restrictions that you apply to the parent OU. You can imagine that adding restrictions to multiple layers can add a lot of complexity.**
* **Keep It Simple, Stupid (KISS) – Use this method to create a single level of OUs like in the example diagram and expand when there is an actual demand for a new OU.**
* **Organize based on security and operational needs – You’ll need to find a balance between having the freedom to quickly develop and having a stable and secure environment. Therefore it’s a good practice to restrict Production and Staging environments in their own OU as opposed to Development and Sandbox accounts that rely on speed and trying out new services.**
* **Apply restrictions to OUs instead of accounts – It’s way easier to manage a group of accounts than multiple standalone accounts. So as explained in the previous point, try to organize accounts based on their security and operational needs.**

## **Conclusion**

**Simply put, AWS Organizations is a great way to consolidate multiple AWS accounts into a single organization that you can easily manage. It’s a bit complicated to get started at first, but if you plan ahead and followed this guide it can save you loads of time, energy, and money in the long run.**

**If you followed this guide and you’re looking to add security to your OUs within your AWS Organization.**