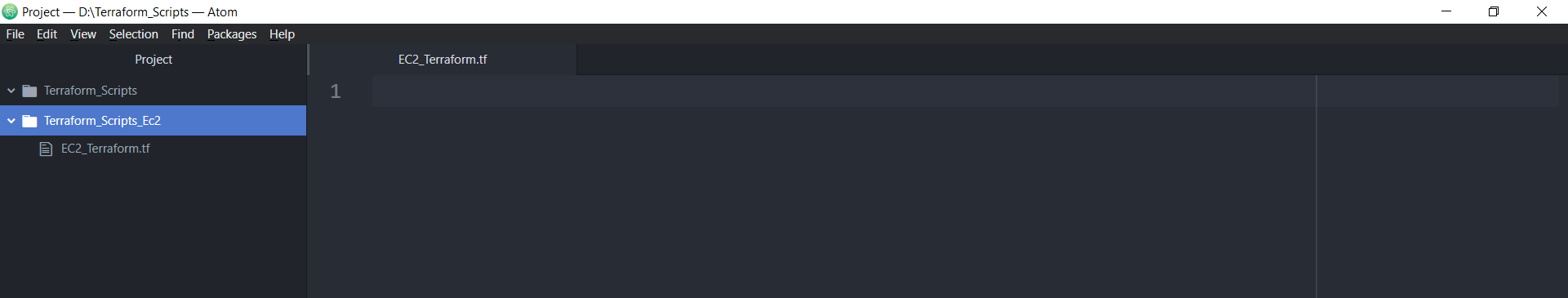
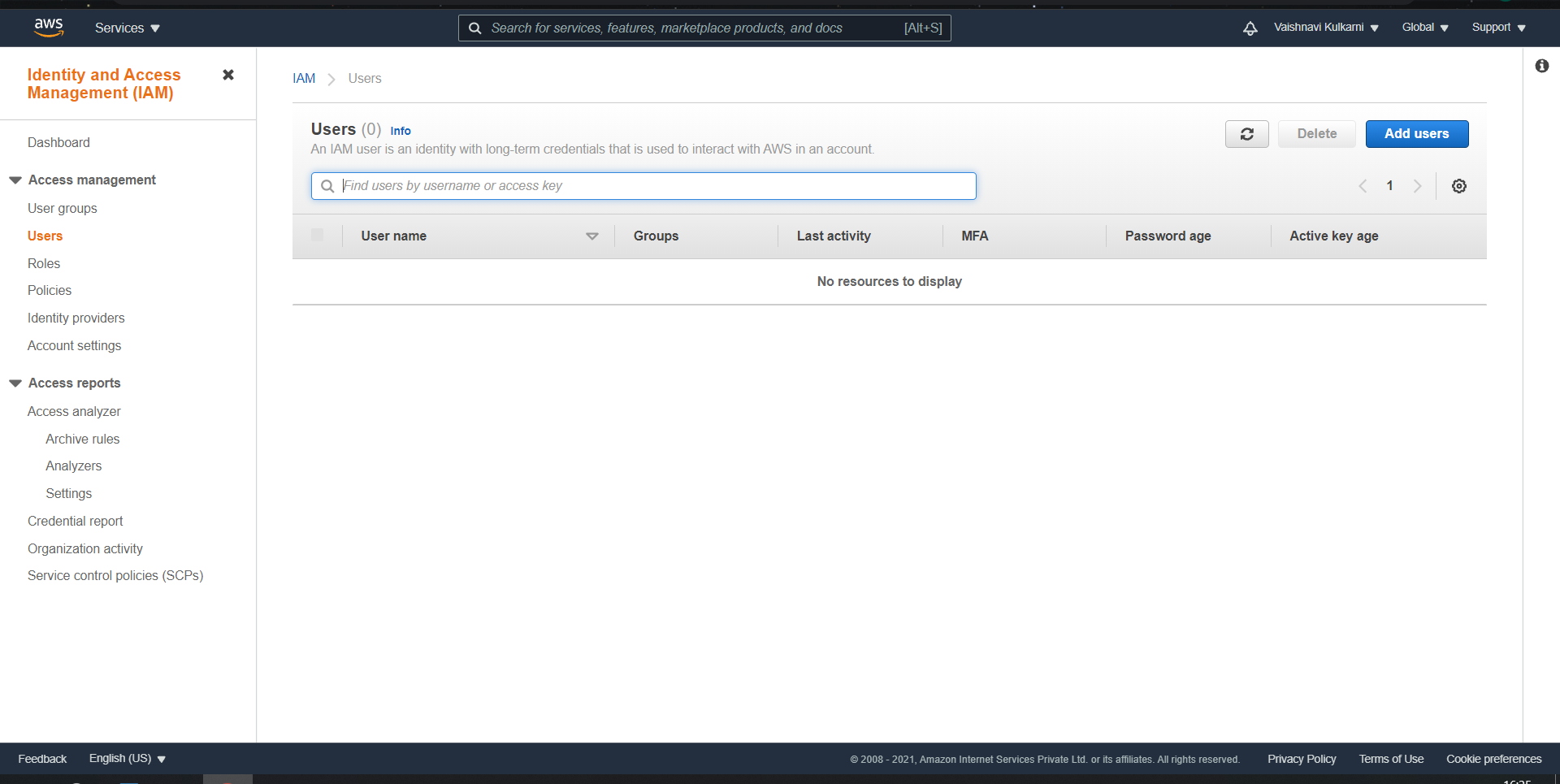
**Aim:** To create EC2 instance using Terraform.

**Steps With Screenshots:**

* Create a new folder**(“Terraform\_Scripts\_Ec2”)** in **“D”** drive and open that folder in Atom IDE also create a new .tf file in it **(“EC2\_Terraform.tf”)**

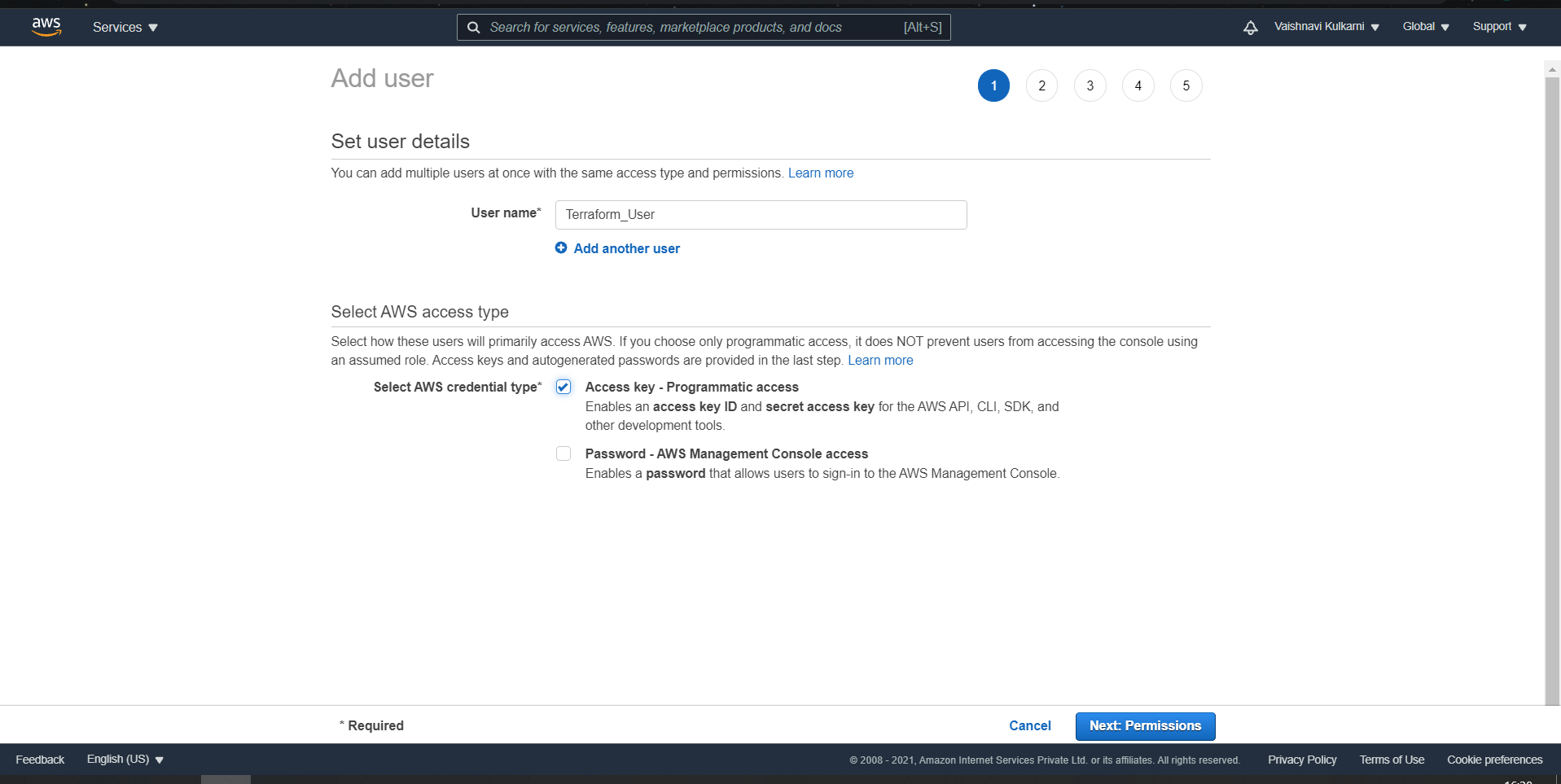


* Open **IAM Service** from AWS Console and select **USERS** option and then click on **“Add users”**

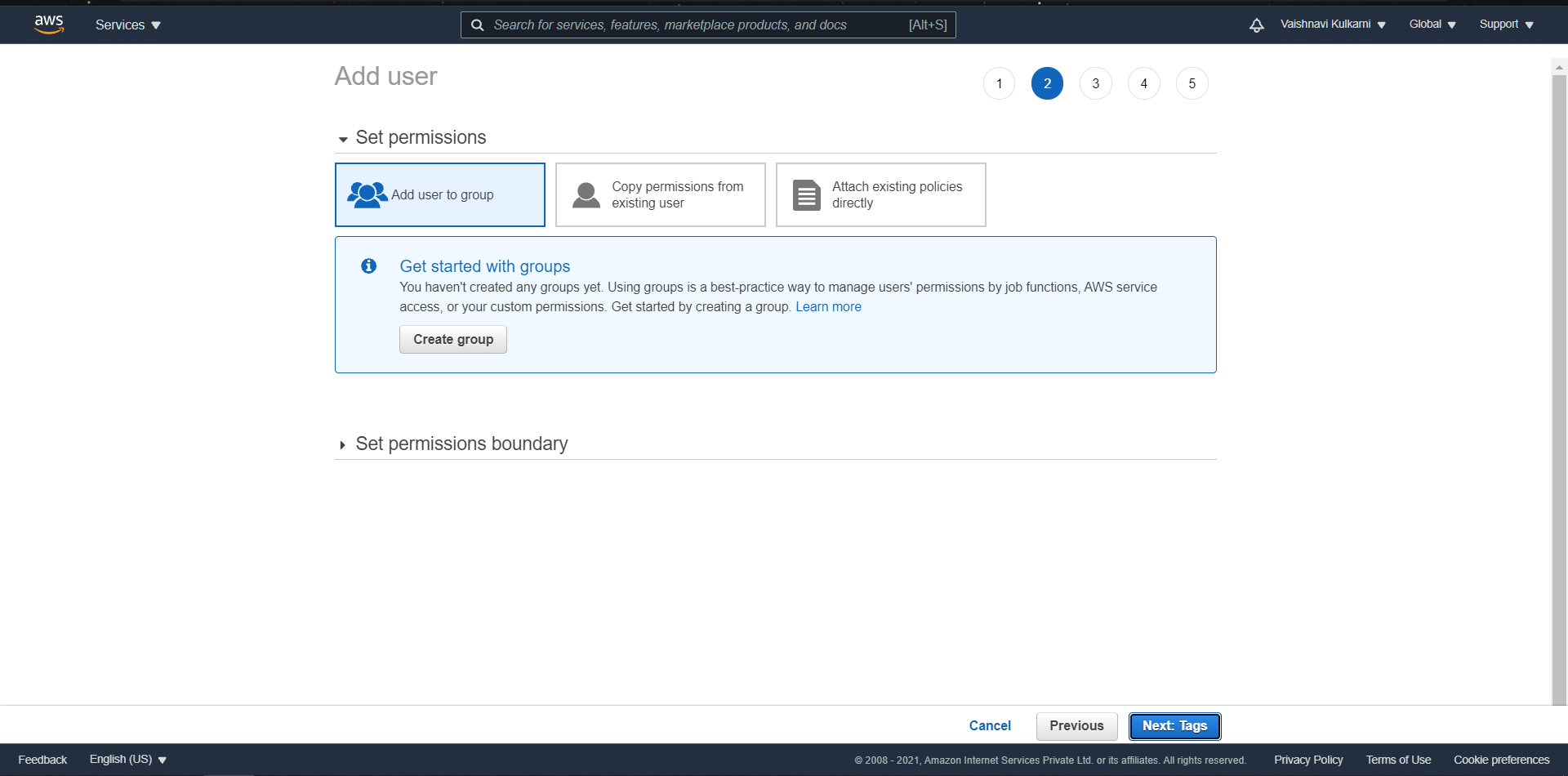


* Give any name to the user and in access type select the option

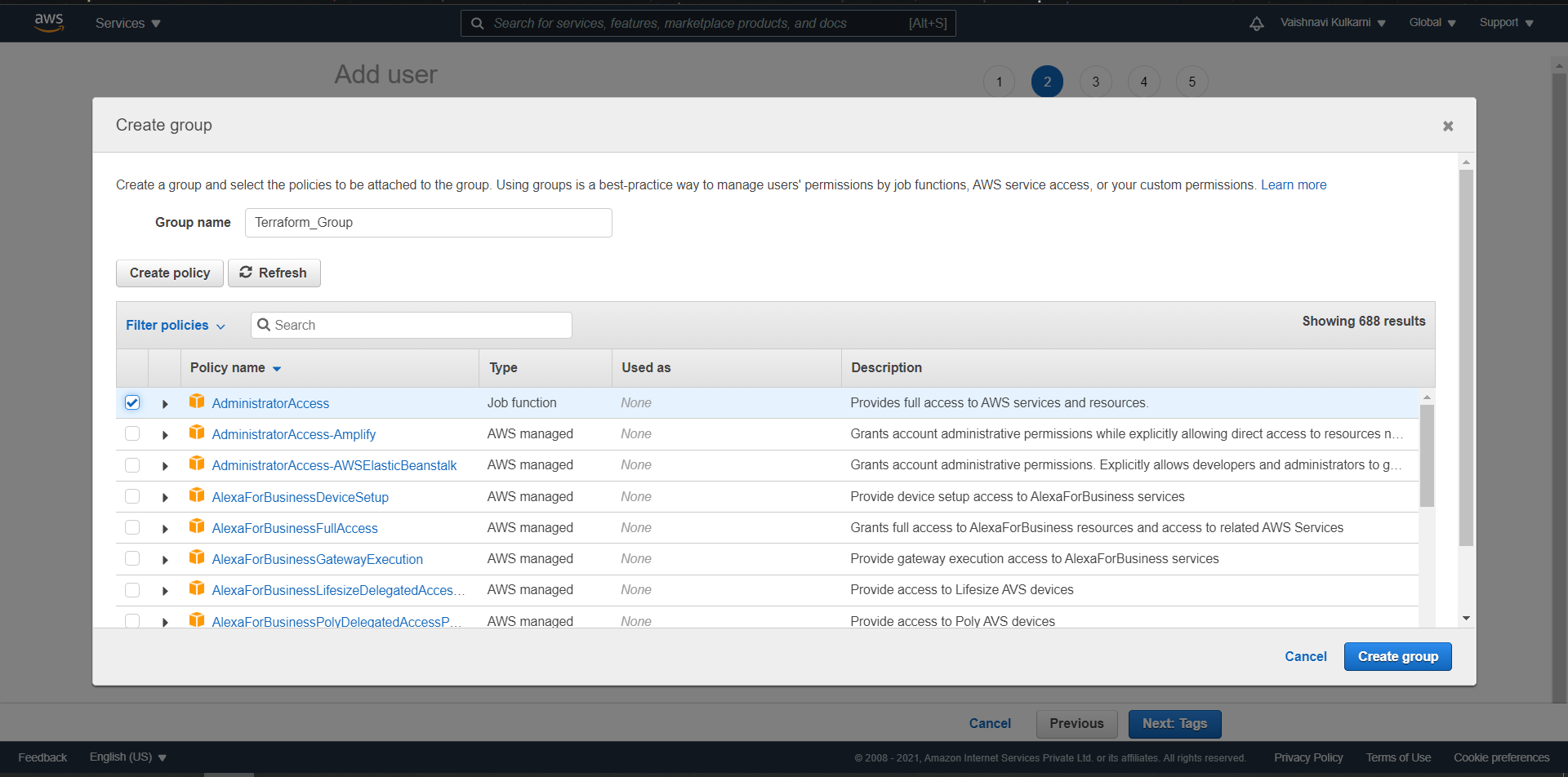
**“Access key - Programmatic access”** then click on **“Next: permissions”**



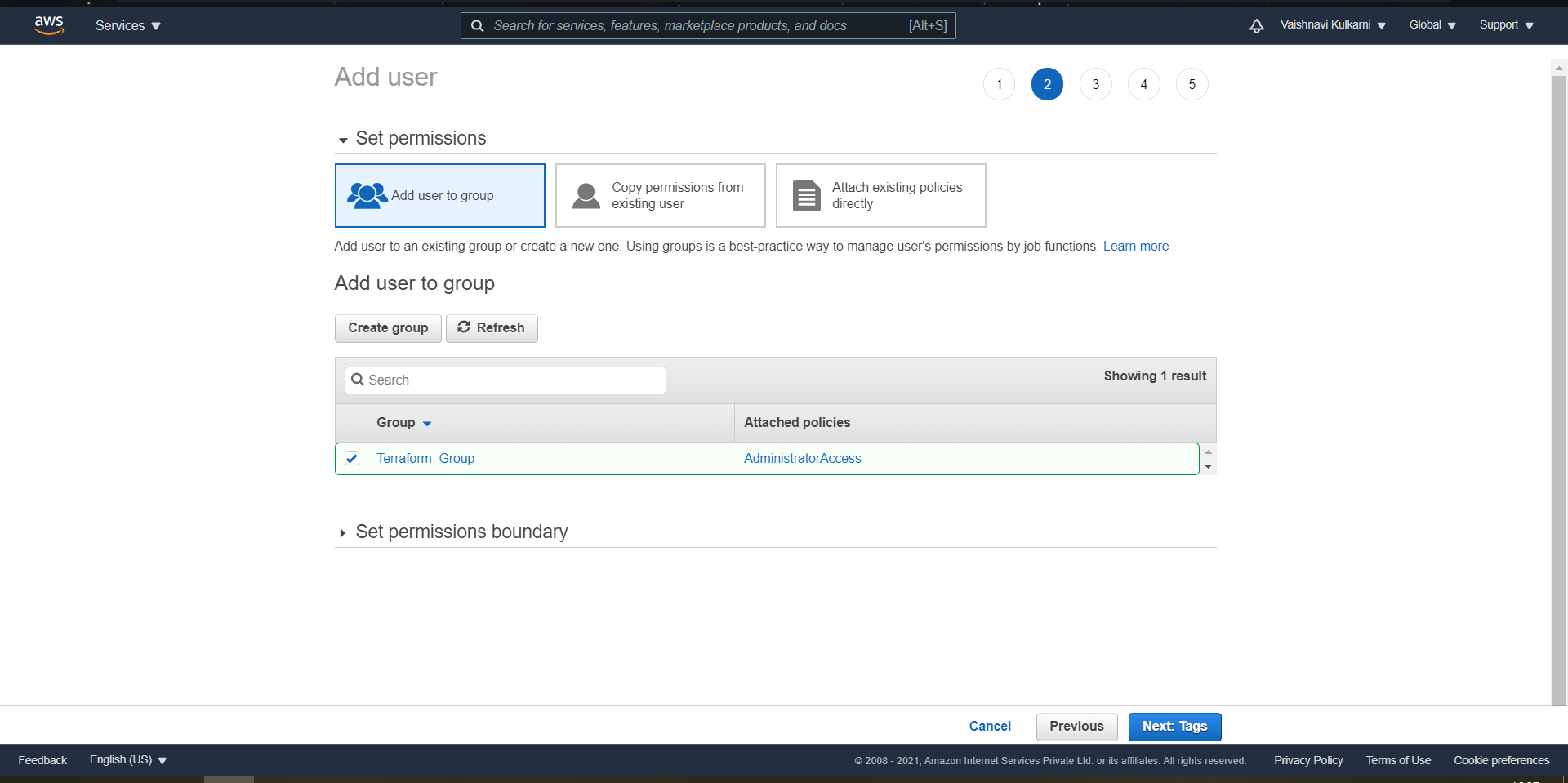
* Now we need to add specific **Groups** to the user for that click on **“create group”**



* Give a name to the **group(“Terraform\_Group”)** and add **“Administrator Access”** Policy which provides full access to aws services. Then click on **“Create group”**

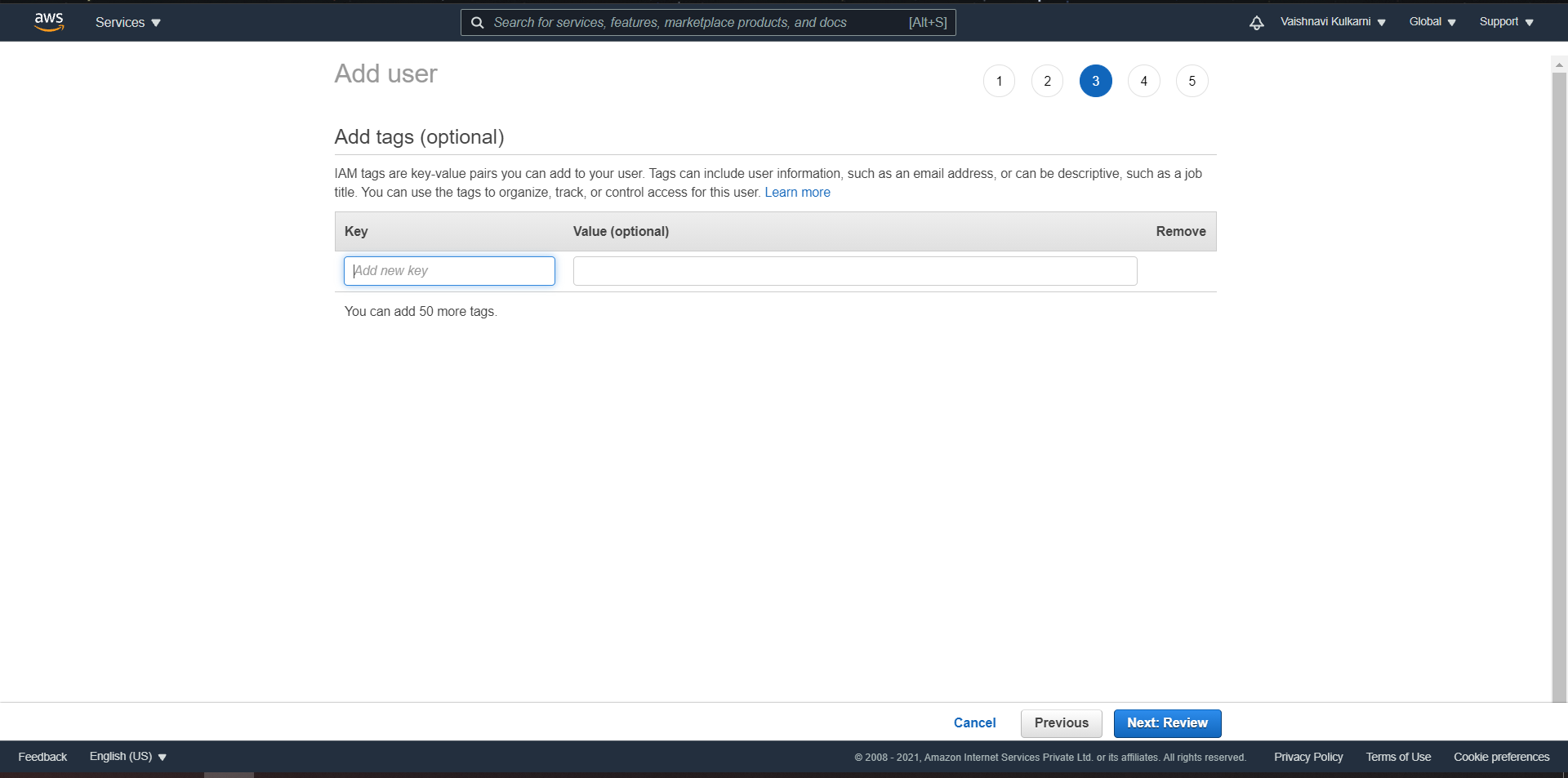


* You can see the policy is successfully added to your group.Click on **“Next: Tags”**

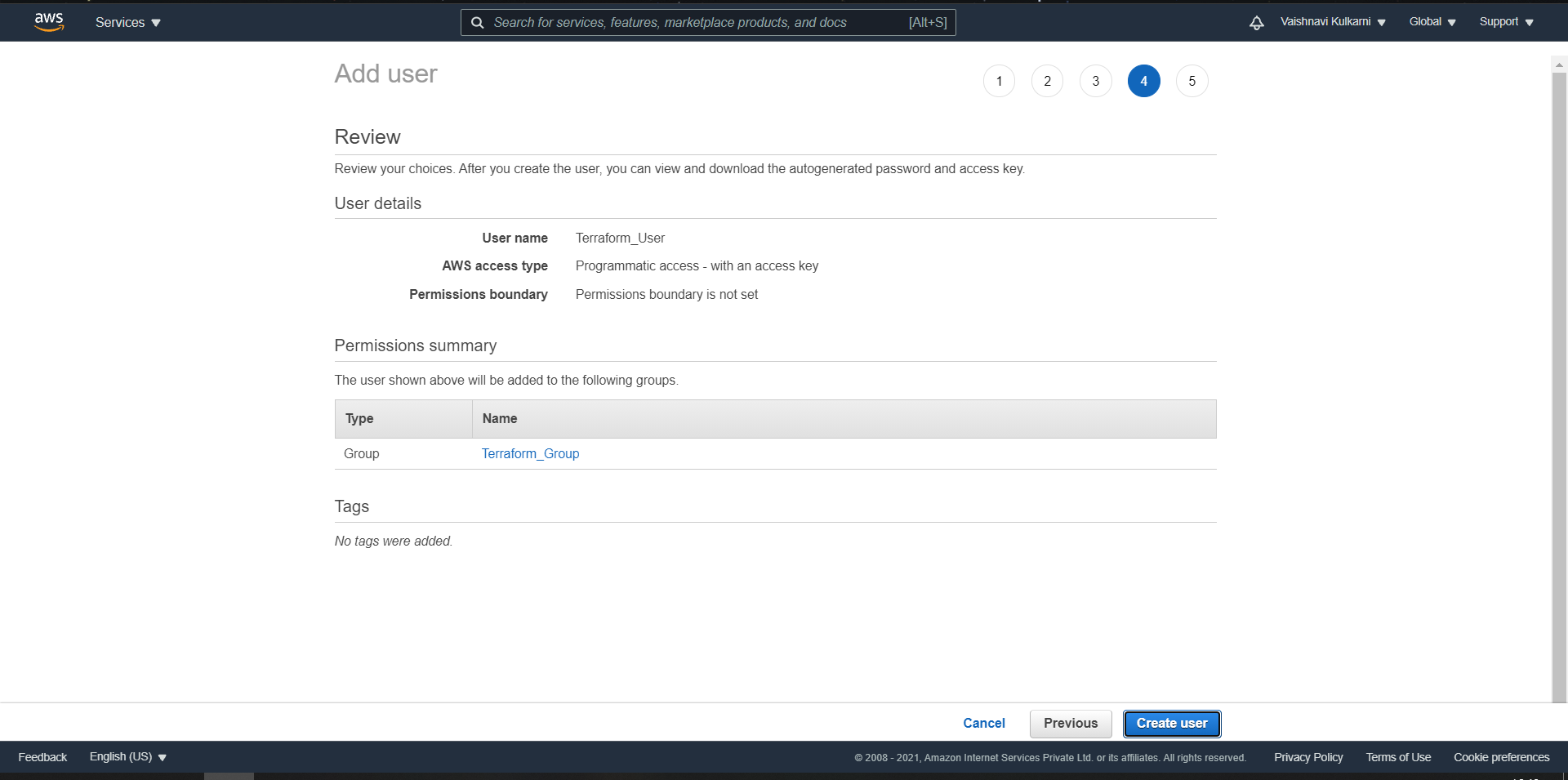


* If you want you can add tag name in this section but it is optional. Click on

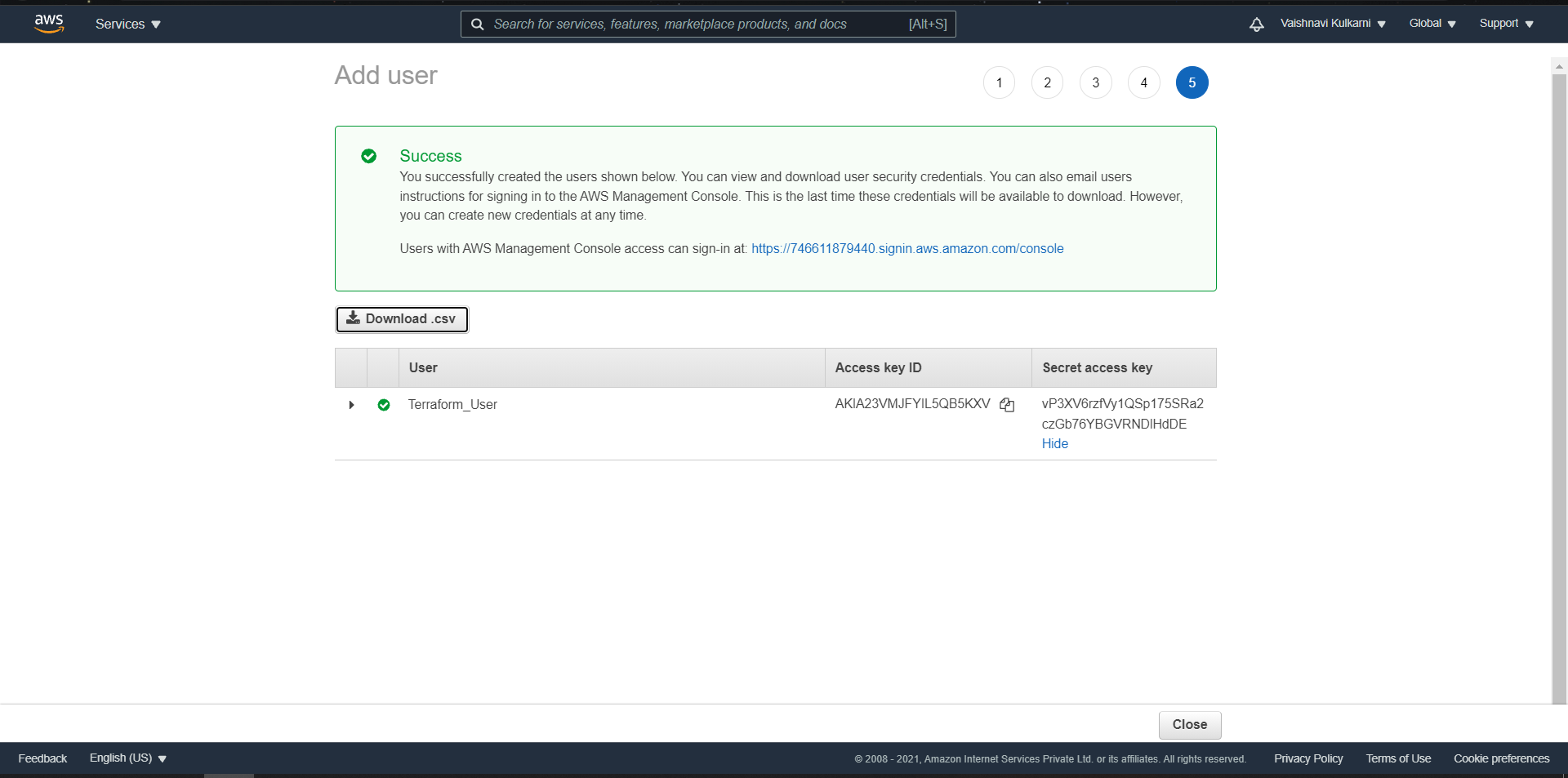
**“Next: Review”**



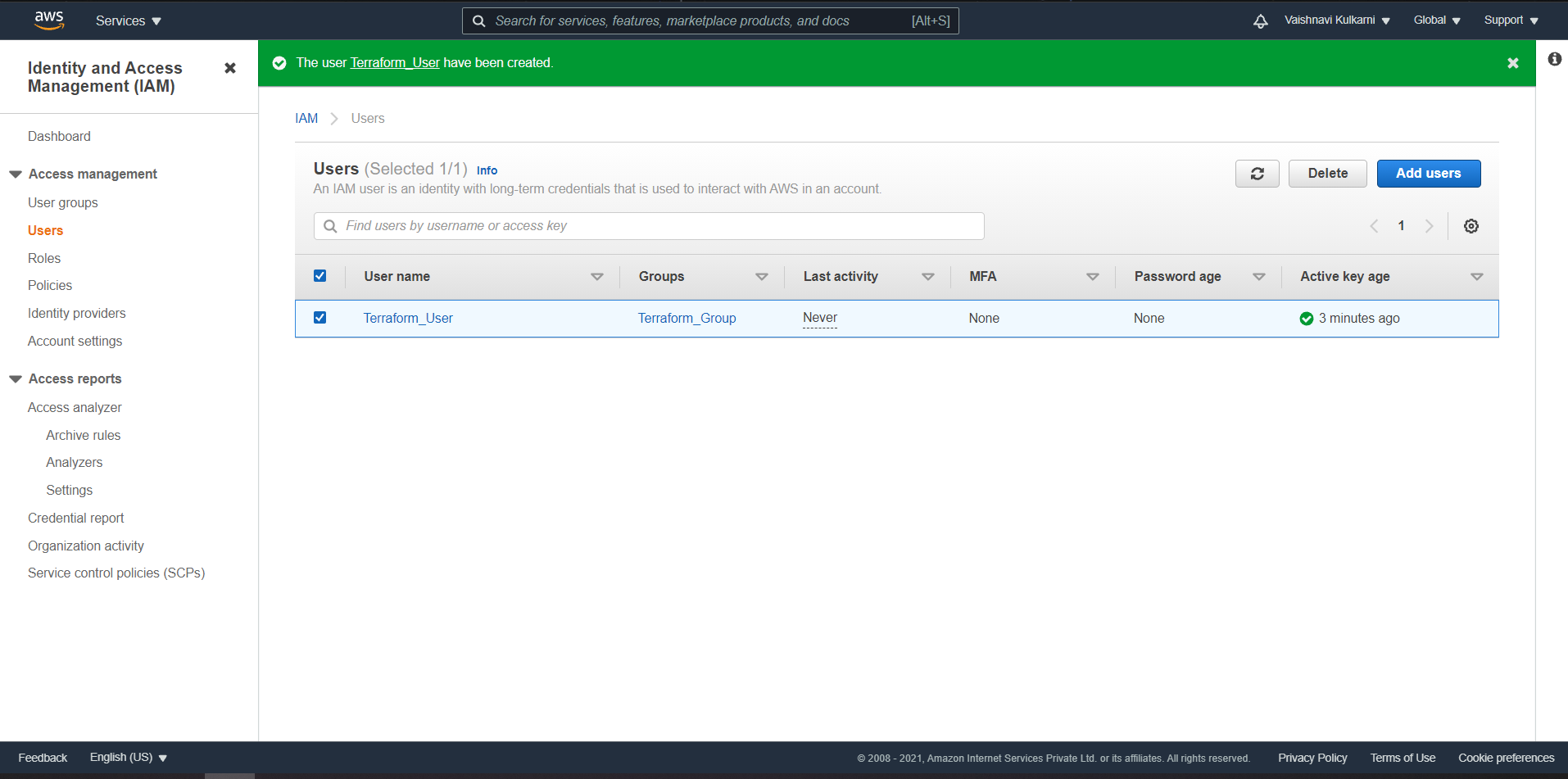
* Next section will show the review of the user and the policy added to it. Also, groups added will be displayed now click “Create user” to get access key and secret key.



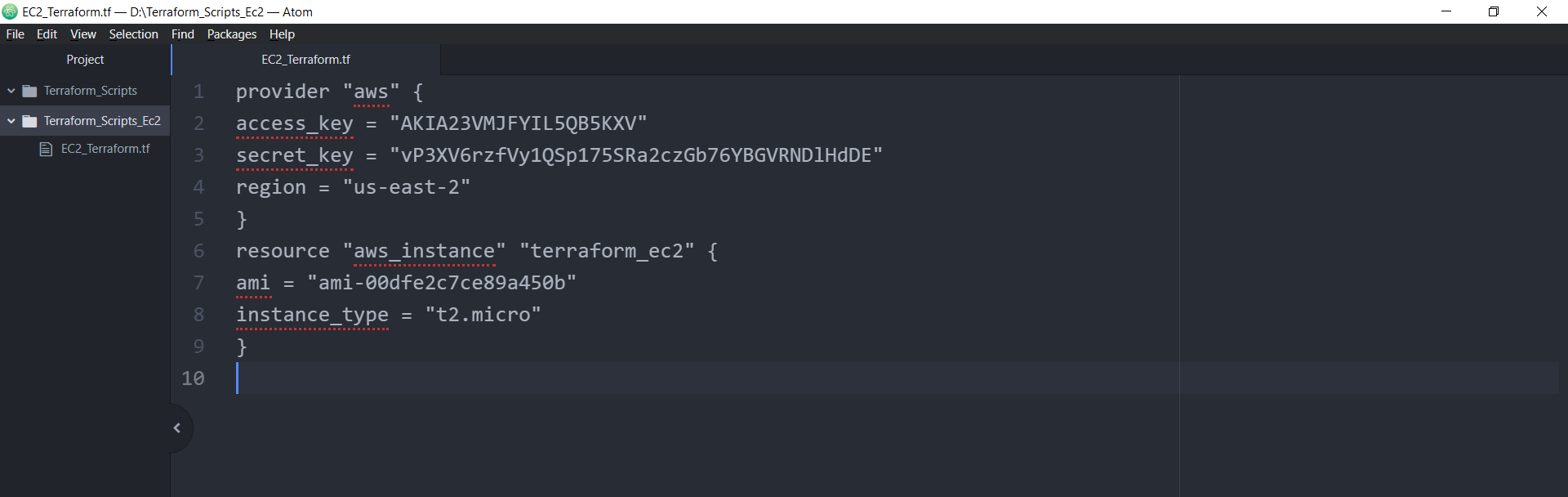
* User is successfully added, access key and secret key of that user will be generated copy paste that access key and secret key in a word file because they will be get added in terraform script. Then click **“Close”.**



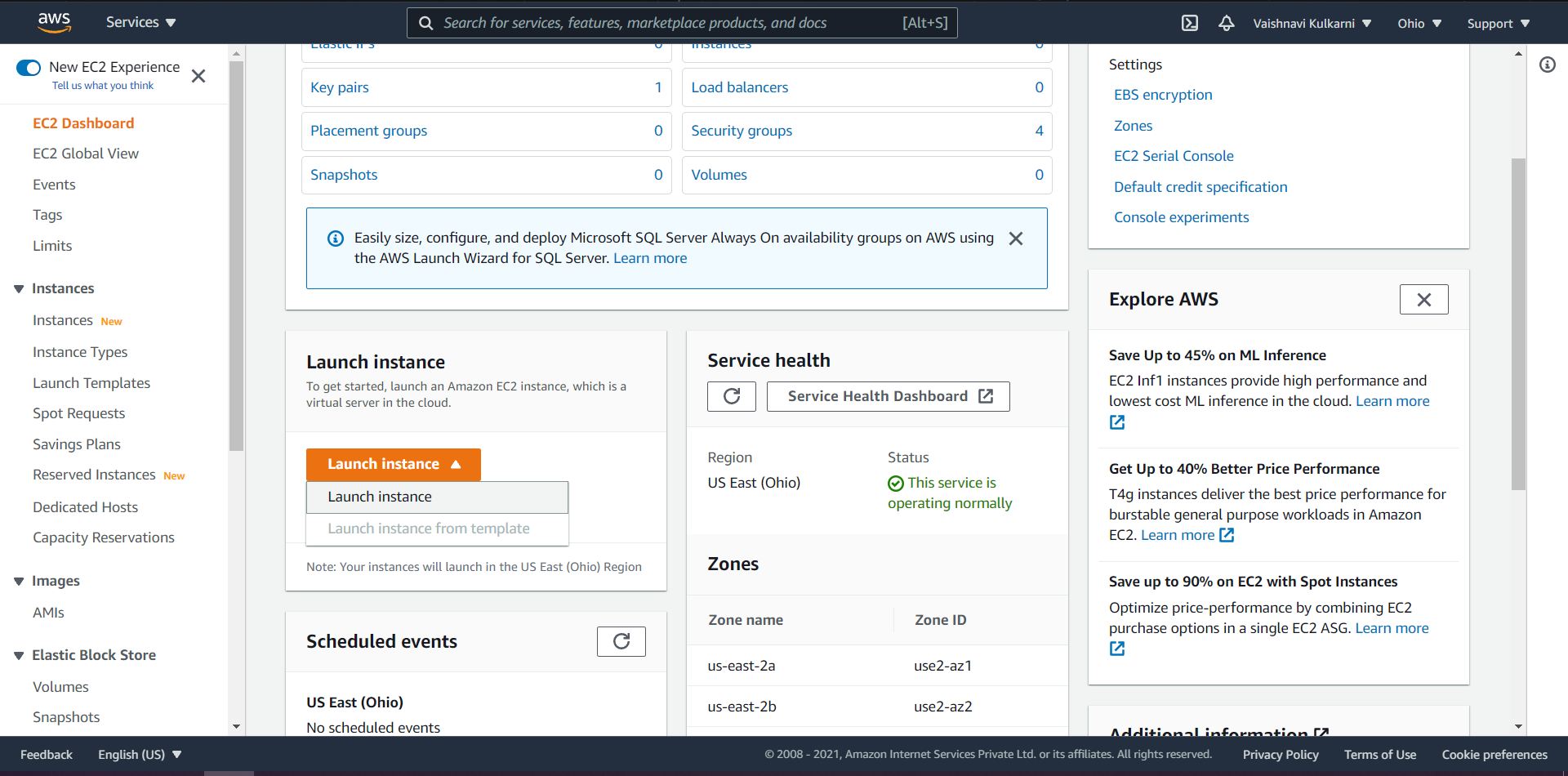
* You can see that the user is added successfully in IAM Console user’s list, also groups which are included in that user.



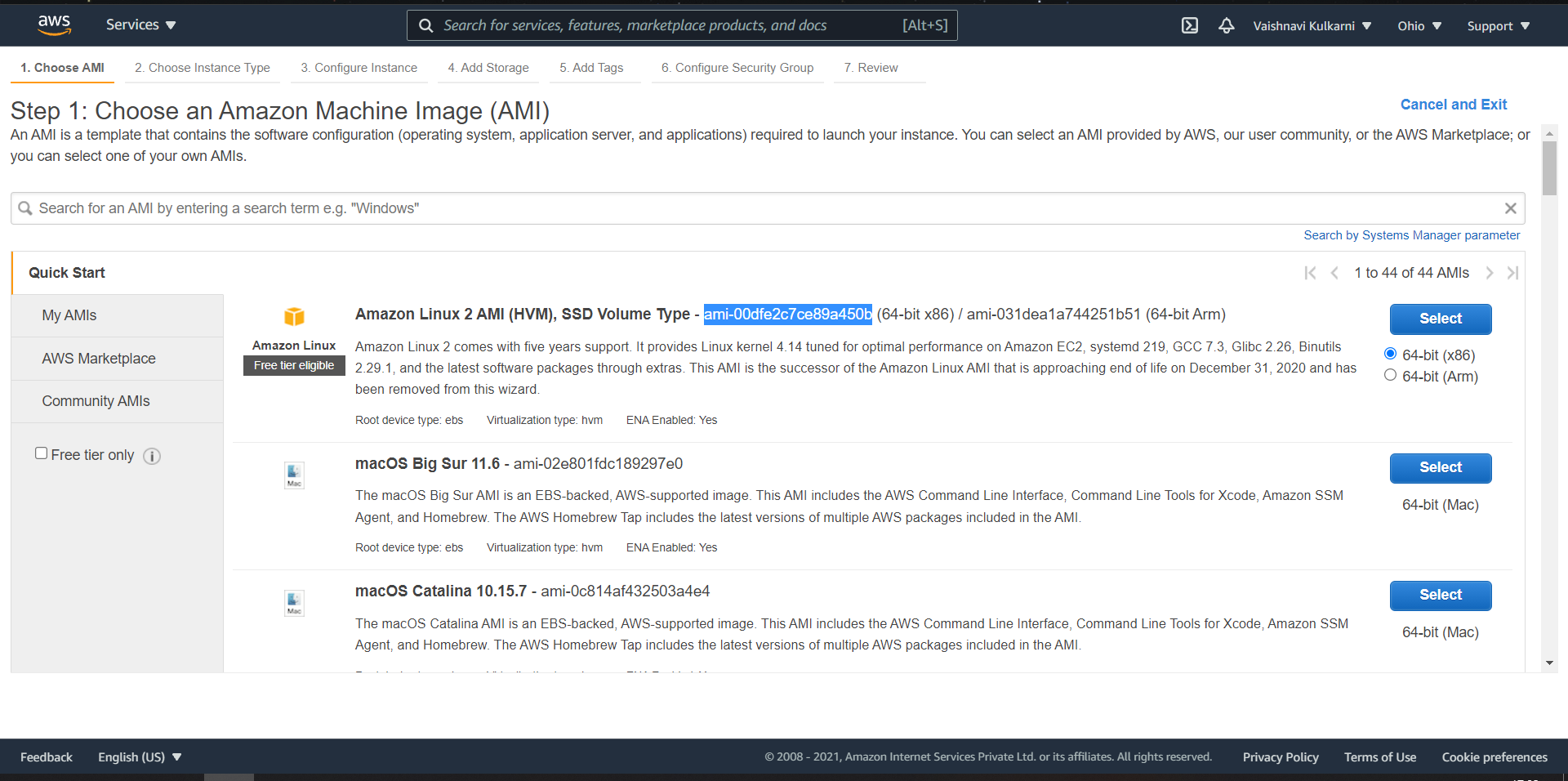
* Now open atom IDE and start typing terraform script in the file which we created in previous steps. While typing the terraform code make sure you copy paste access\_key and secret\_key of the user which was generated in previous steps.



* To get **“ami key”** select EC2 Service from AWS Console 🡪 Select **EC2 Dashboard** Option **🡪** Then click on **“Launch instance”**

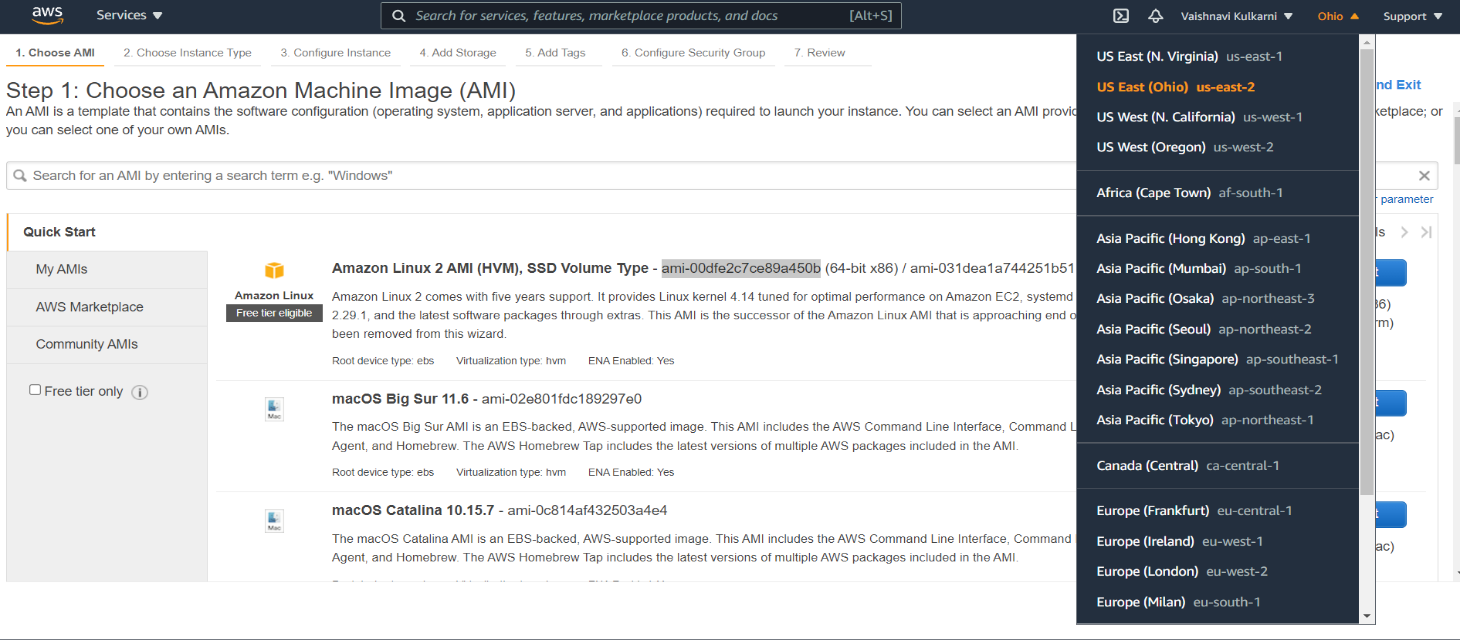


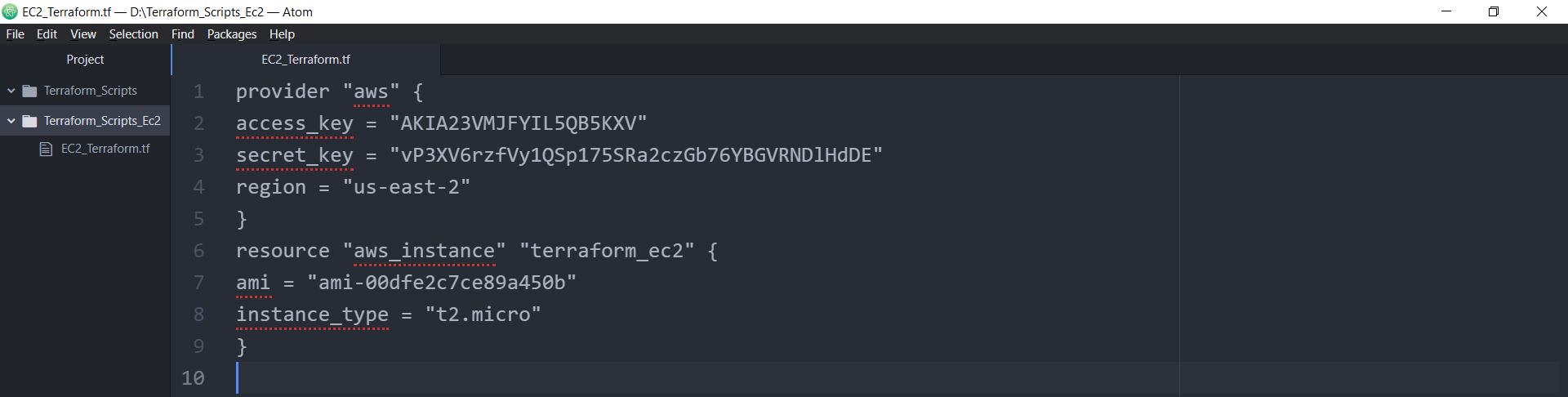
* You can select any **“free tier eligible” AMI and copy the ami key provided after the name of AMI as shown below. Here I have selected “Amazon Linux” AMI and copied the ami key.**



**Paste that AMI Key in the terraform code**

* After pasting the AMI Key give instance\_type and **make sure that you give the same region which is in your aws console as shown**

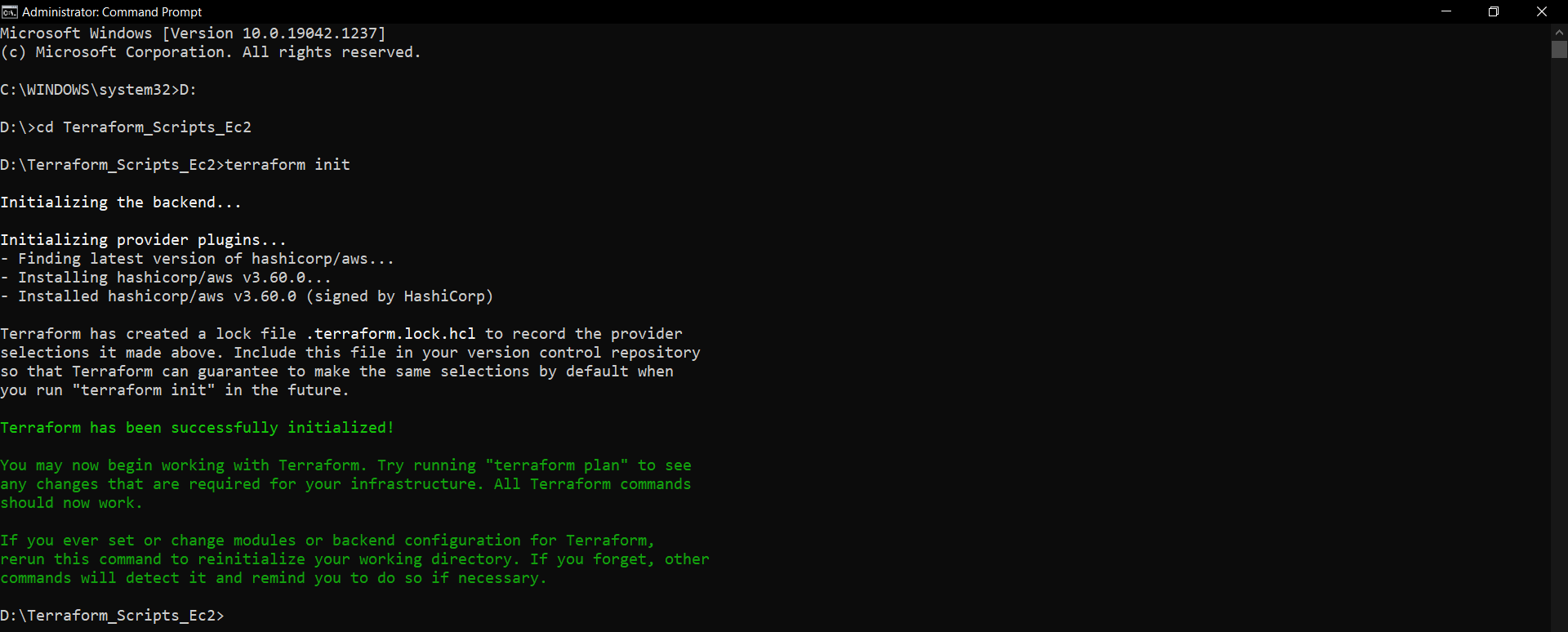
****



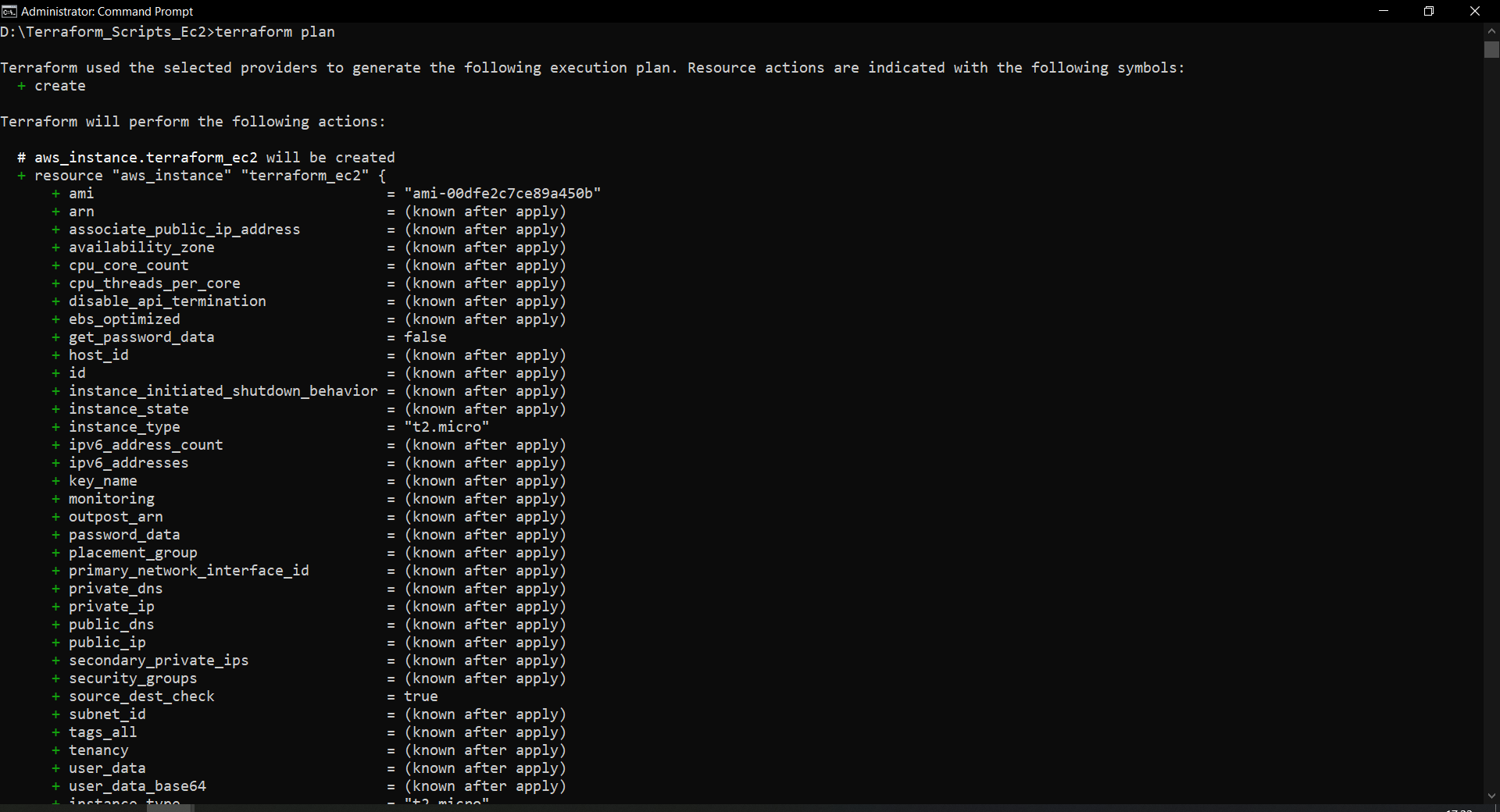
* **After completing the code open command prompt and go to the folder where EC2\_Terraform.tf file is located**



* **Initialize the terraform with “terraform init” command.**

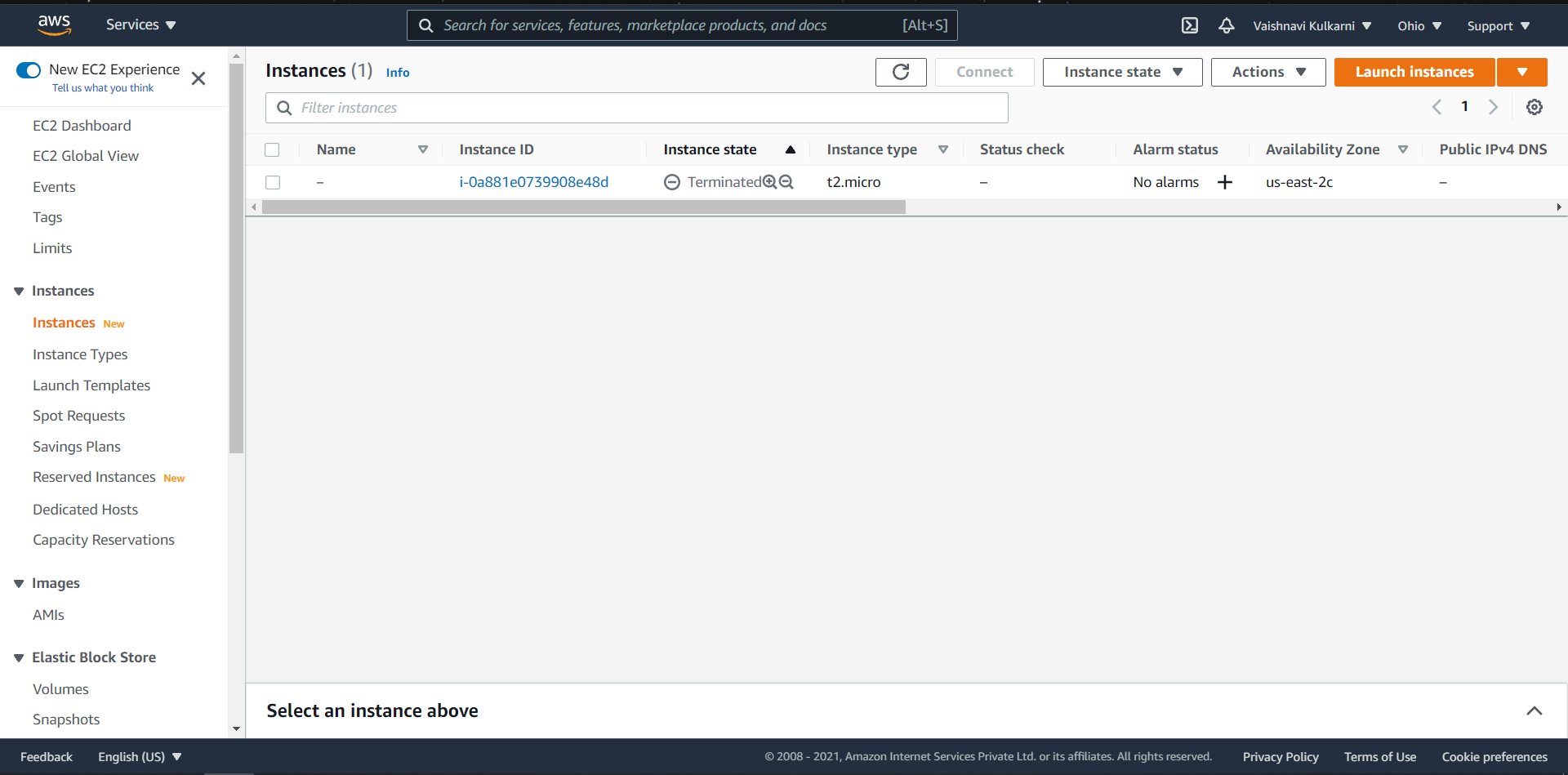


* **To execute the plan run “terraform plan” command.**

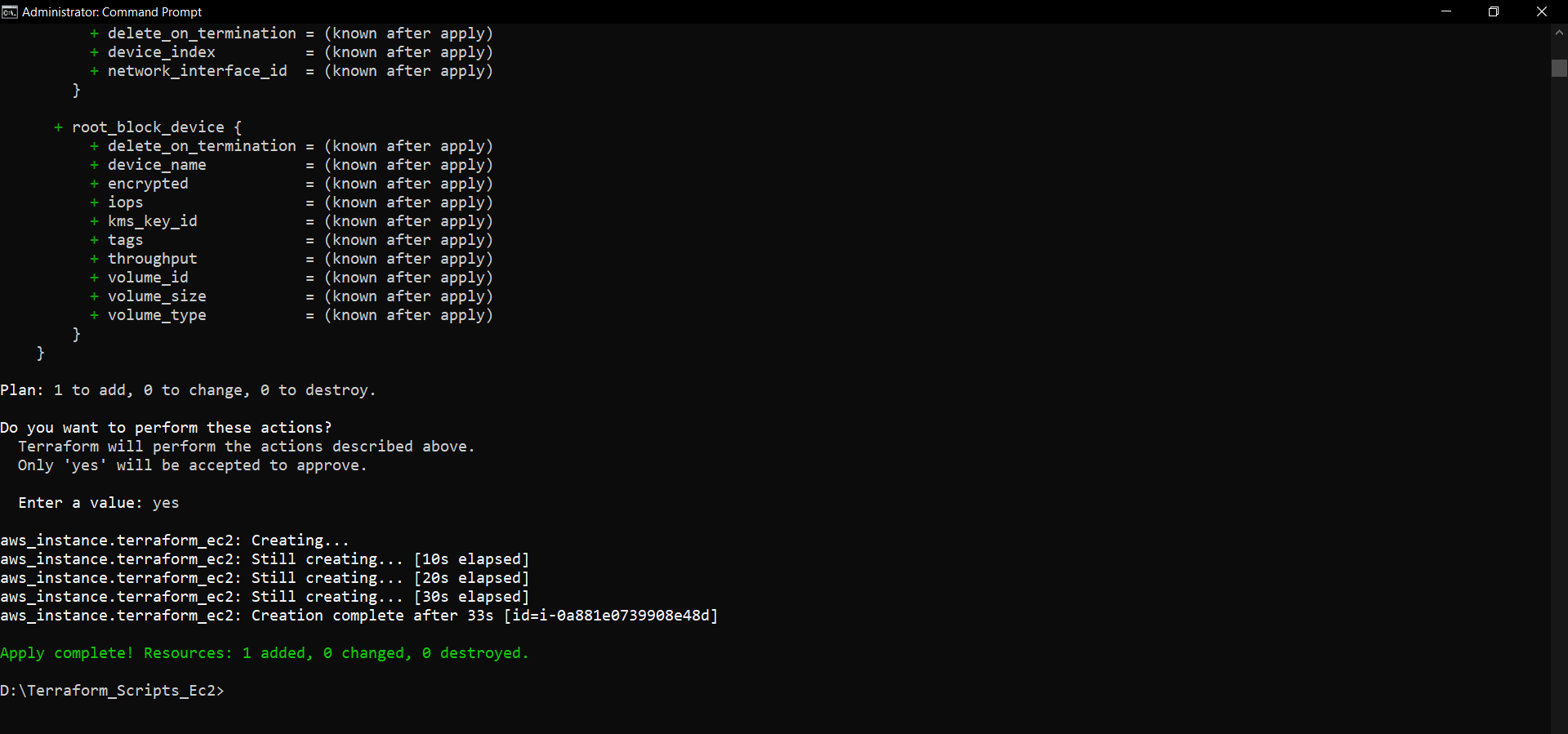
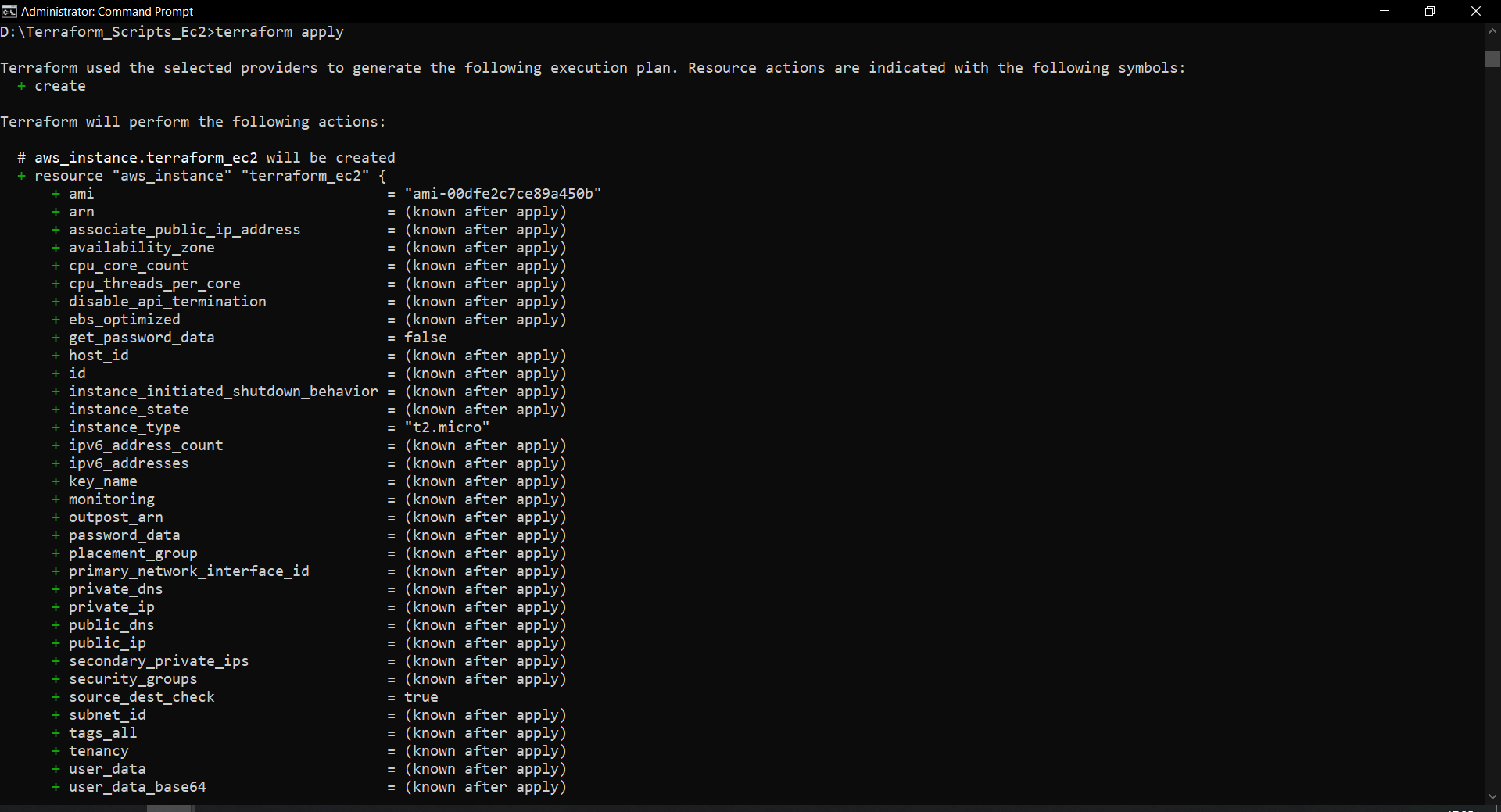




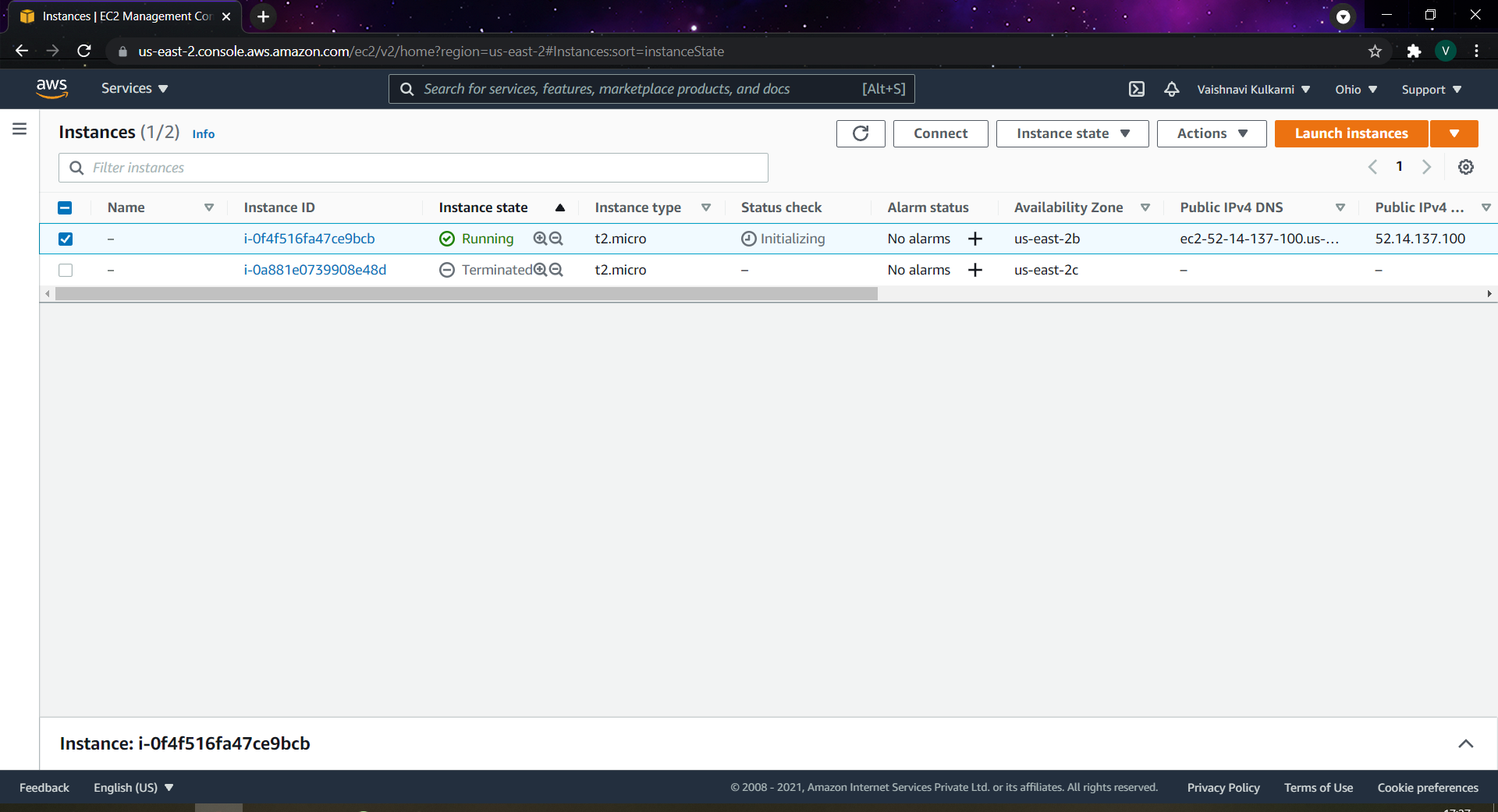
* **Before applying next command, you can see that there is no other instance in running state in my AWS EC2 Service console**



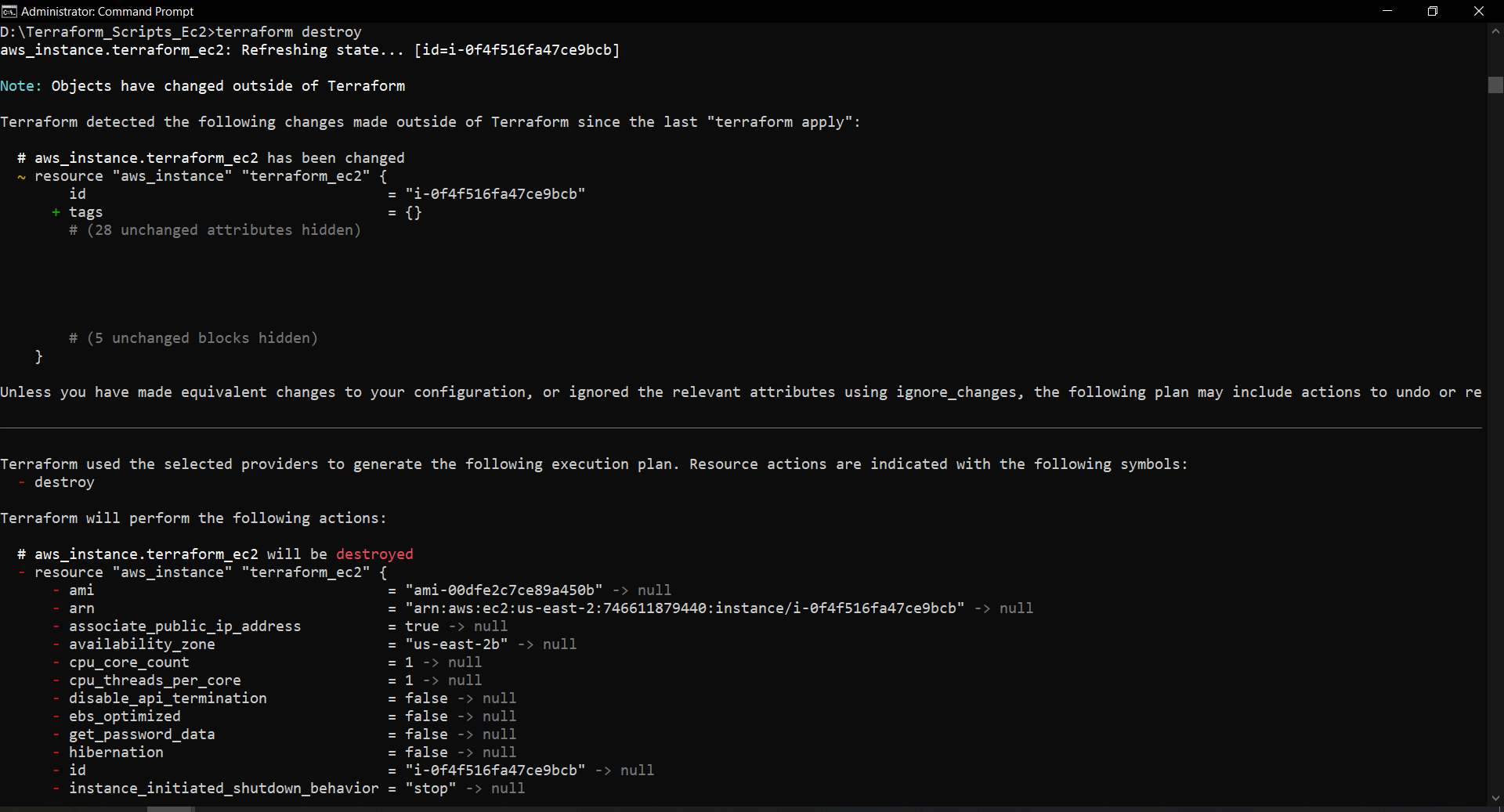
* **To apply the changes of execution run “terraform apply” command which will create EC2 Instance in aws console**

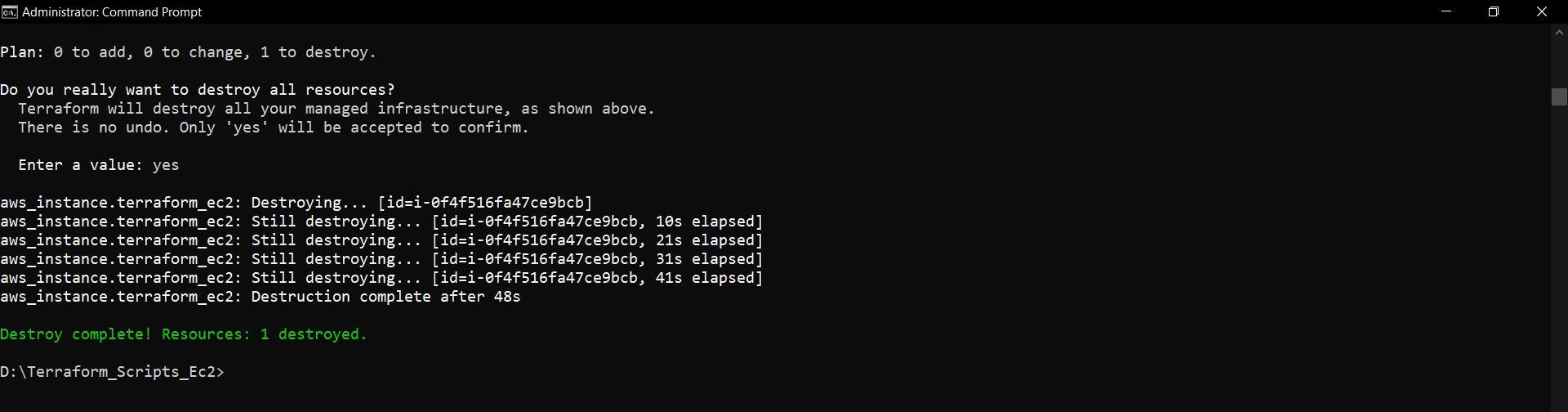


* **After completing apply command you can see that EC2 instance is running in my AWS EC2 Console.**

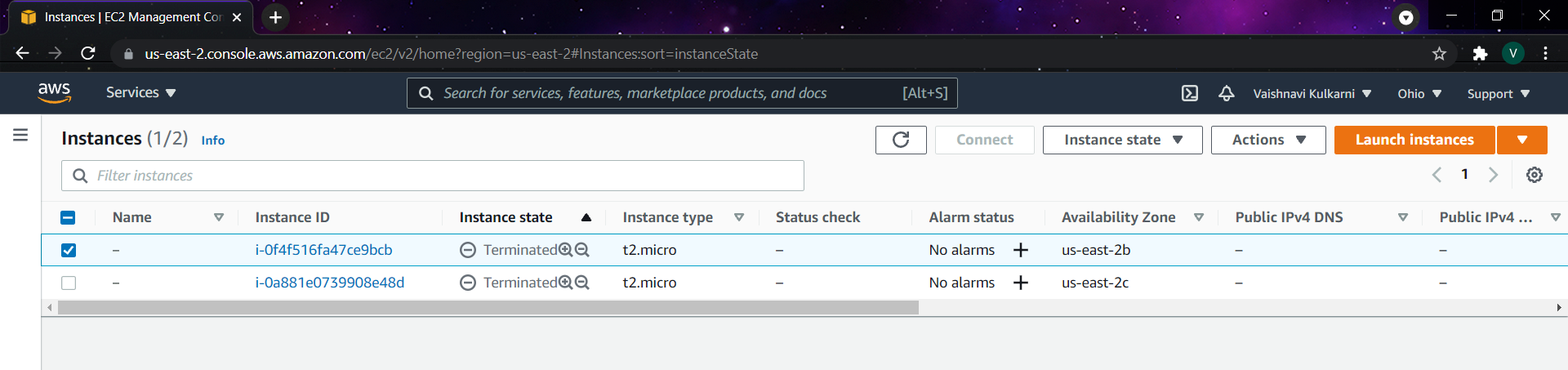


* You also terminate the instance with the help of terraform using **“terraform destroy”** command.
* Before applying terraform destroy command you can see in the above image selected instance in running





* After applying **“terraform apply”** command you can now see the EC2 instance got terminated.



**Conclusion:**

In this experiment we created a IAM User added groups to it and gave important policies and we created EC2 instance with the help of terraform script by using access key and secret key on IAM User and AMI KEY of a AMI image from EC2 Instance also we terminated the instance with the help of terraform command.

**Lab Outcome: ITL504.3**

To apply best practices for managing infrastructure as code environments and use terraform to define and deploy cloud infrastructure.