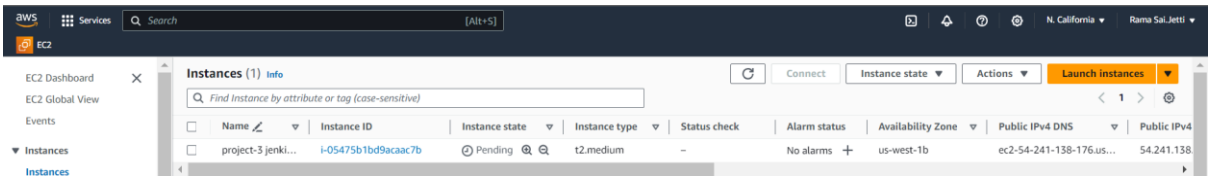
This project all about the **terraform integrate with the Jenkins**.

Prerequisites:

- lunch instance.
- This instance must have the installed Jenkins, terraform.



Script to install the Jenkins & terraform:

Then connect the instance. Then run this script using ./Jenkins.sh

vi Jenkins.sh

```
apt update -y
apt install default-jdk -y
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install Jenkins -y
```

#Terraform installtion

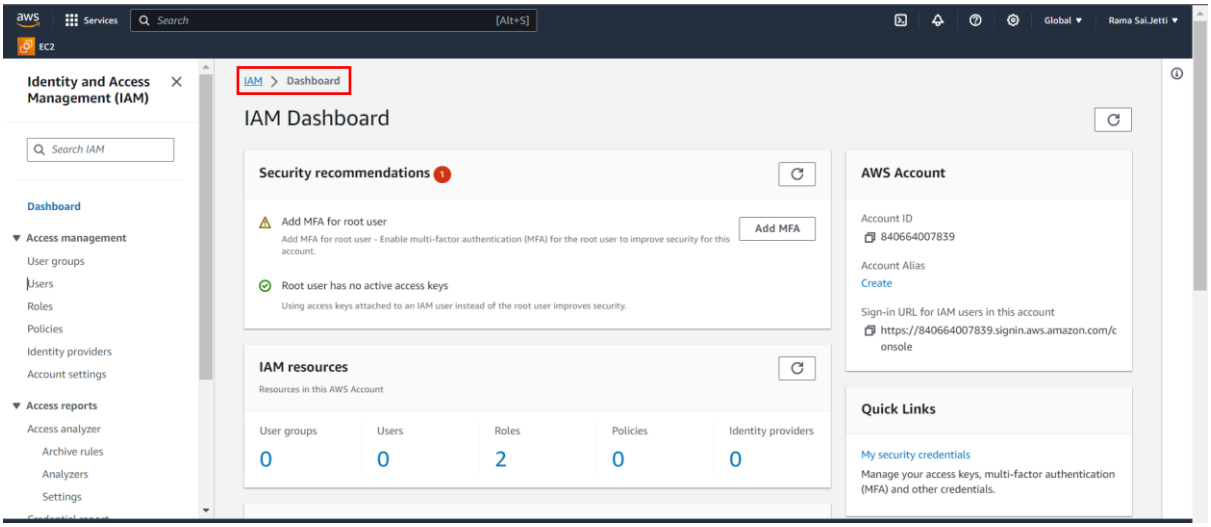
```
wget -O- https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg
```

echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d/hashicorp.list

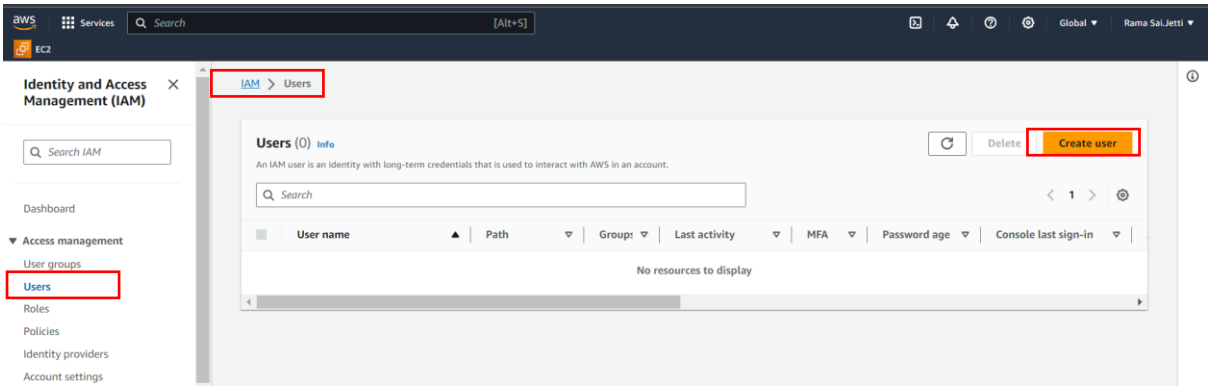
sudo apt update && sudo apt install terraform

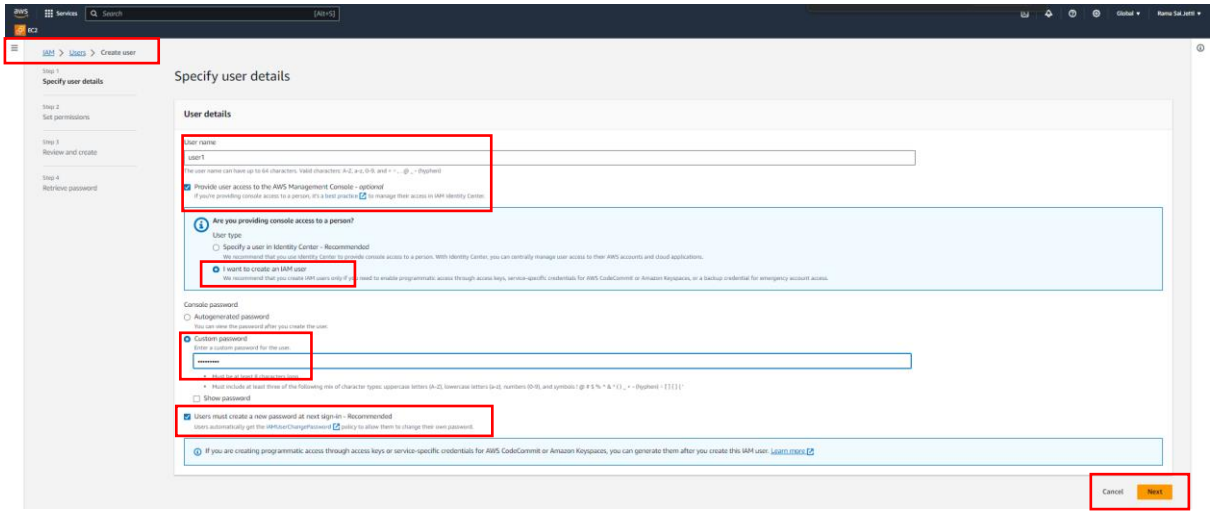
Note: to check the Jenkins& terraform on instance check **versions** or type **whereis Jenkins or terraform.**

After that create a IAM user, click on user follow below steps:

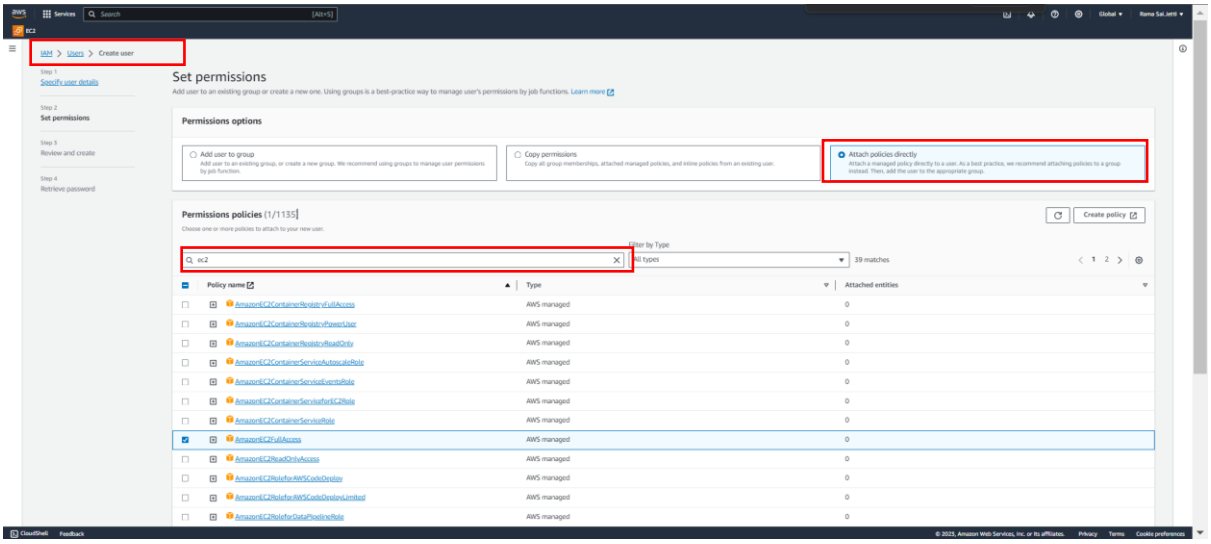


Create user here

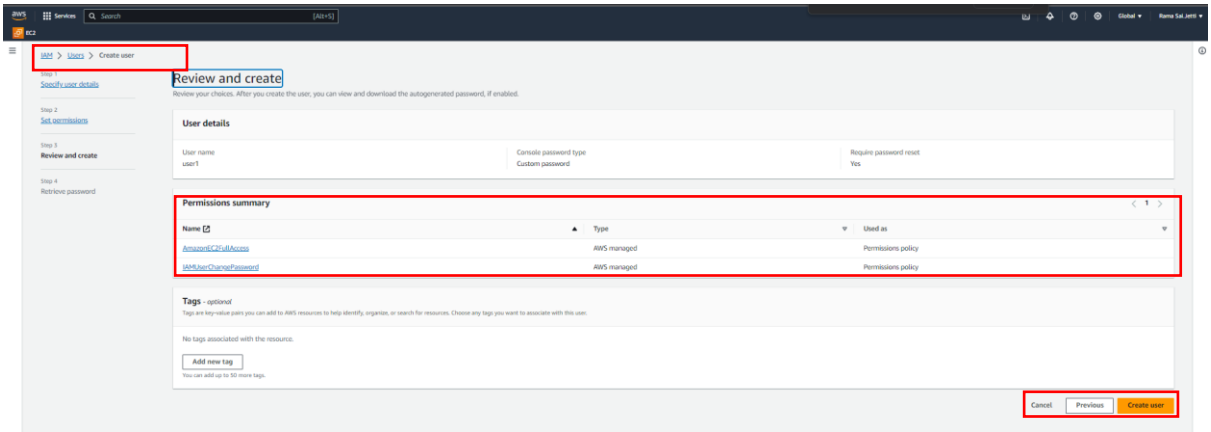


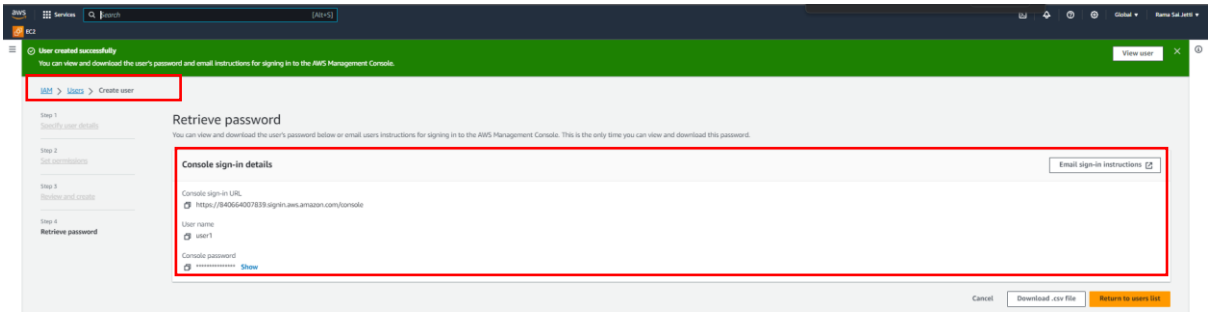


Set the permissions to the user

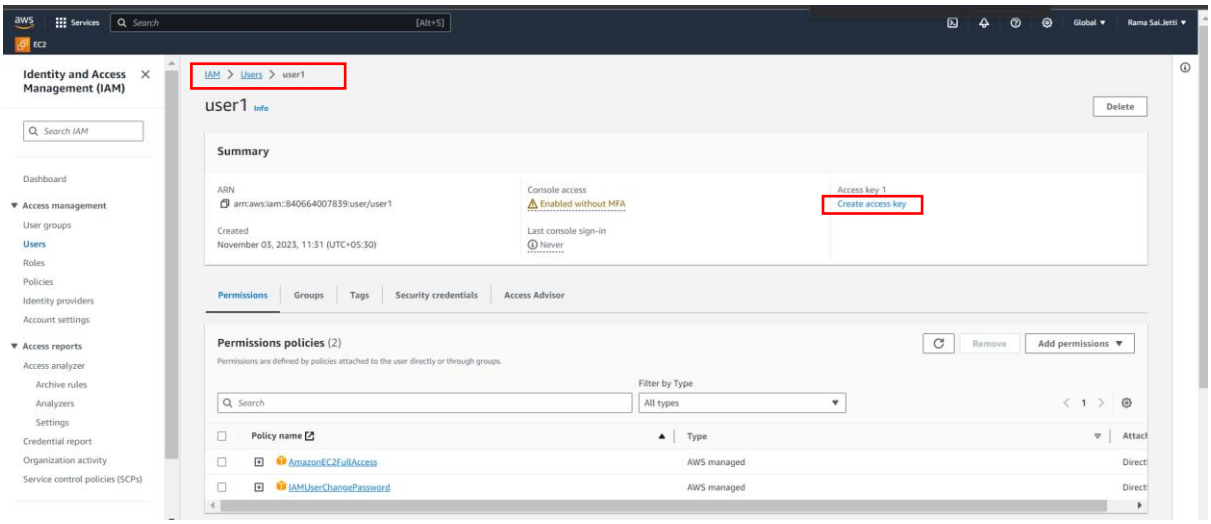


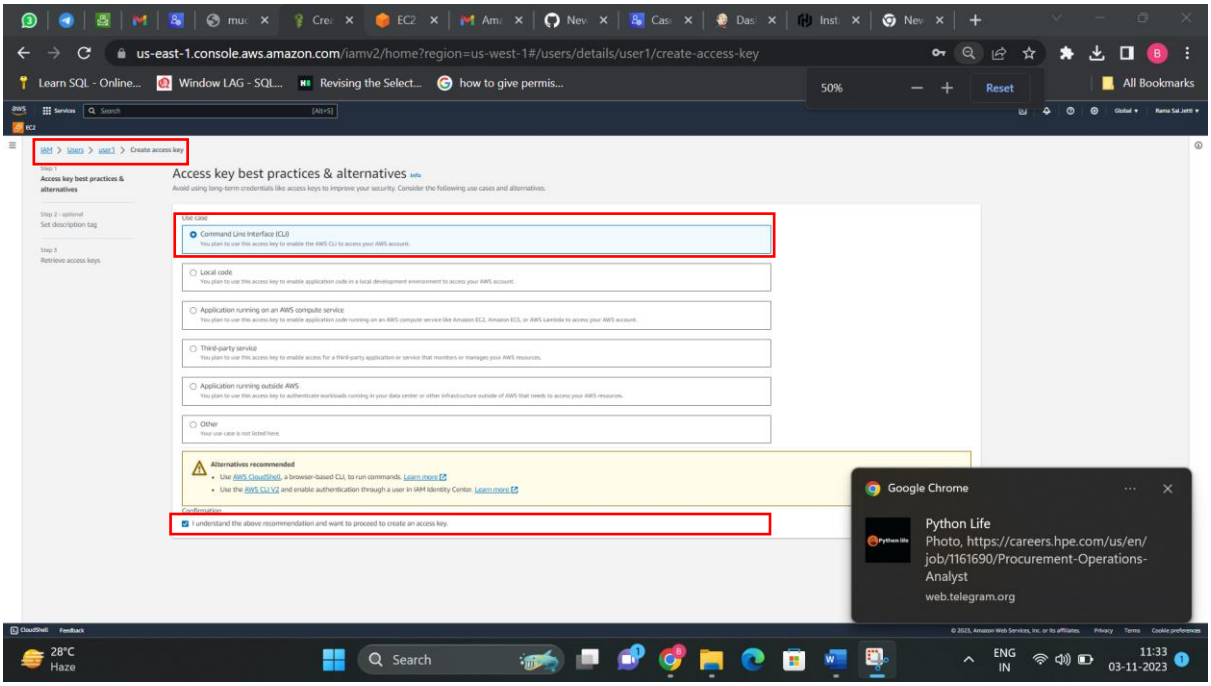
Here able to see the permissions summary



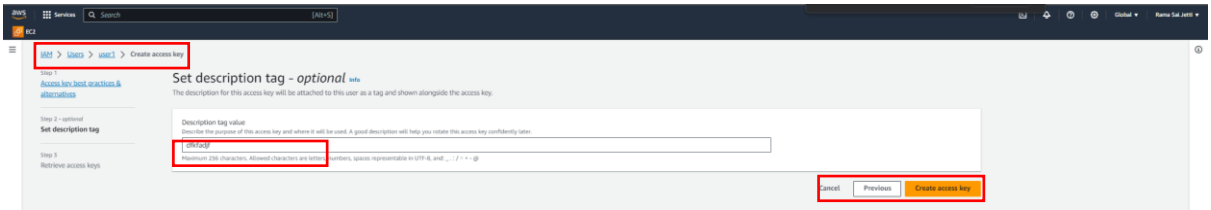


Create access key

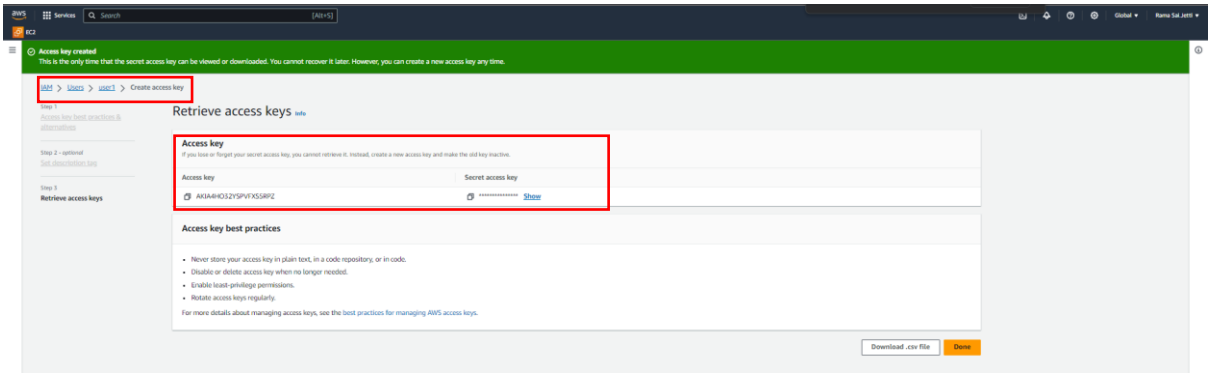




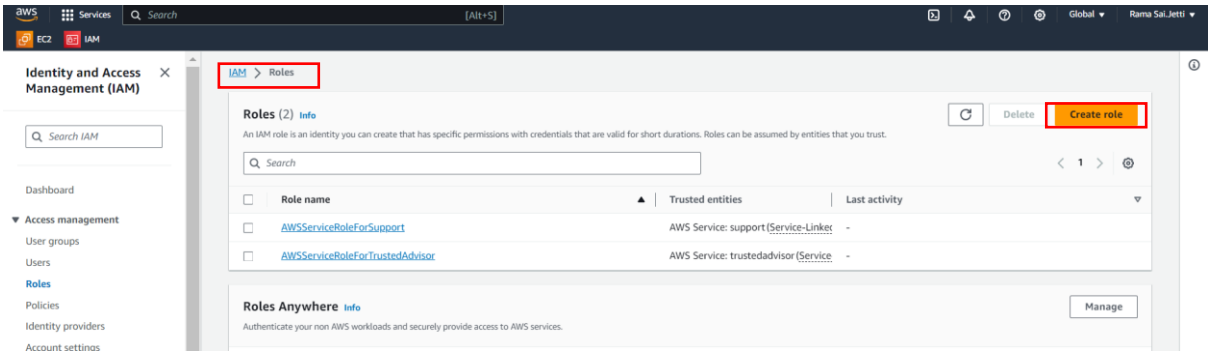
Click on the create access key



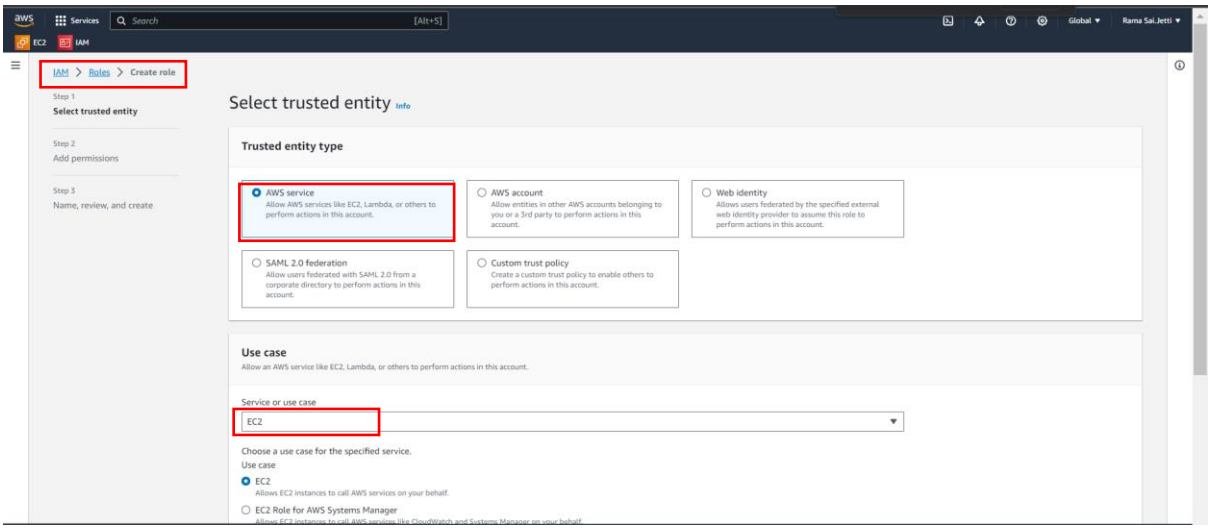
Here can get the access key & secret access key

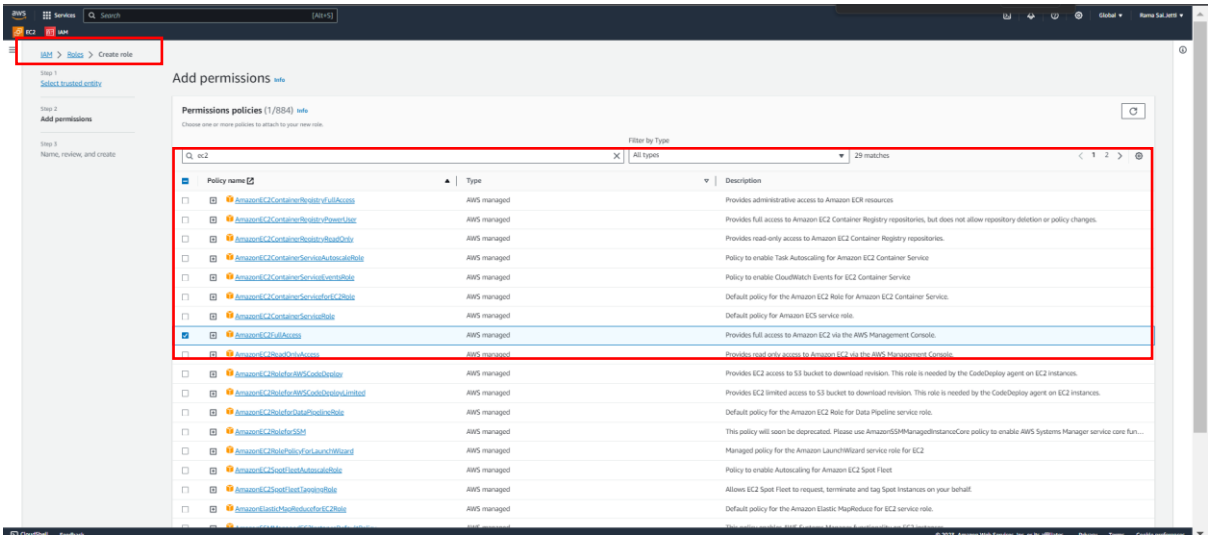


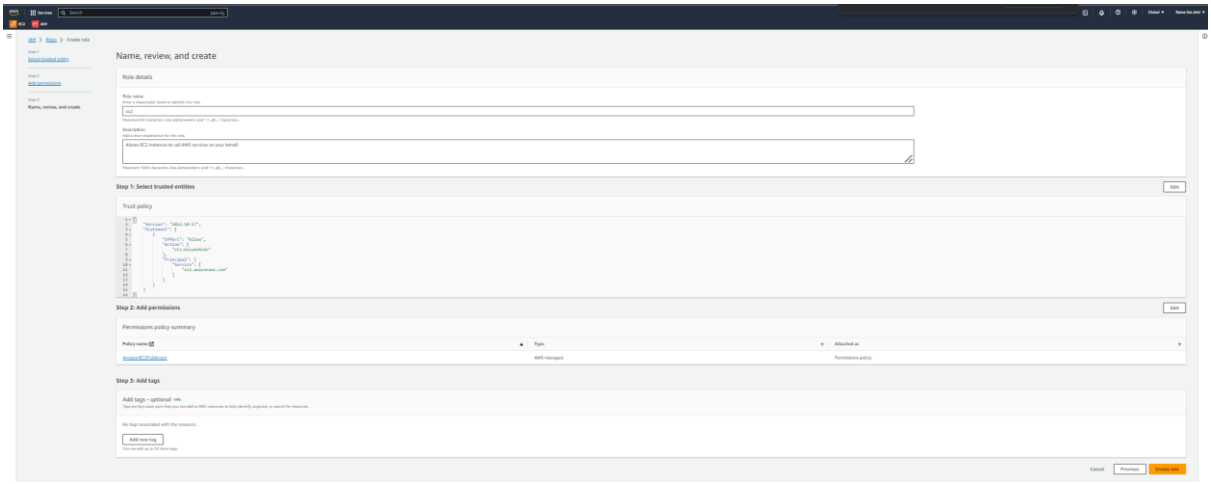
Then create role

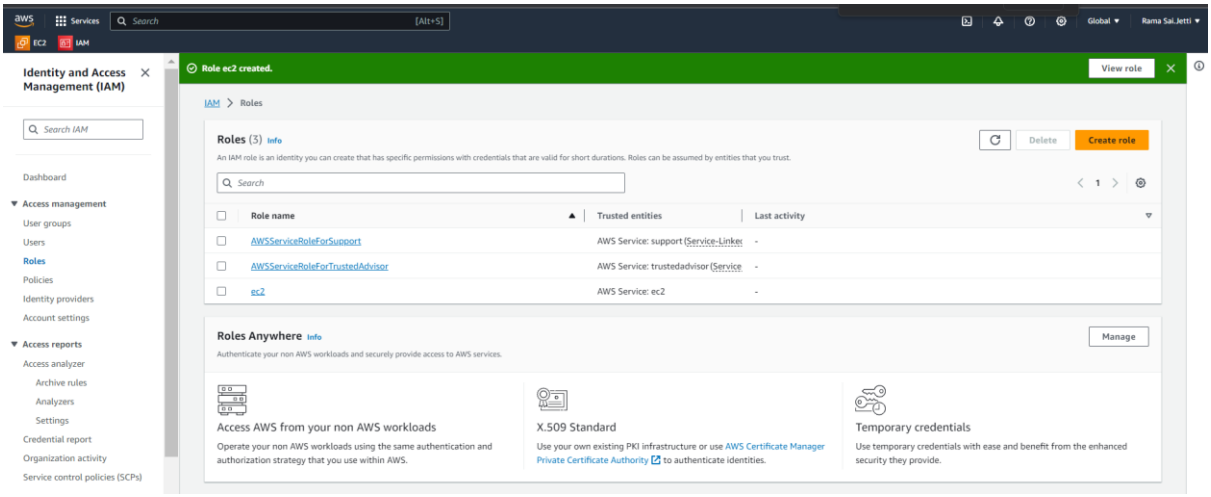


Follow the bellow steps

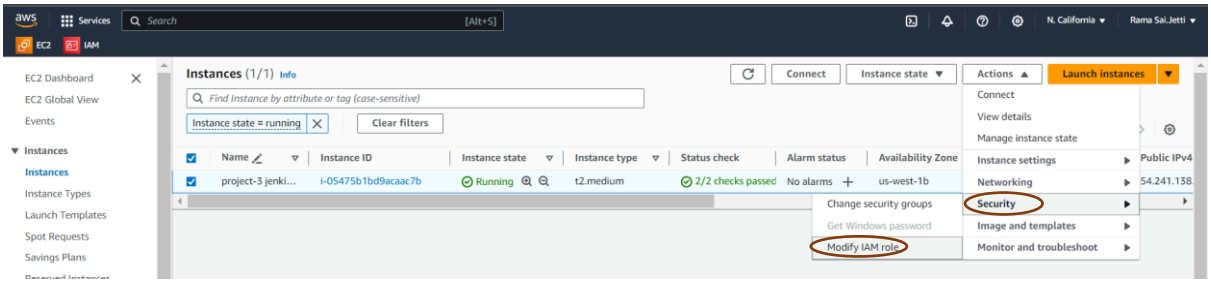


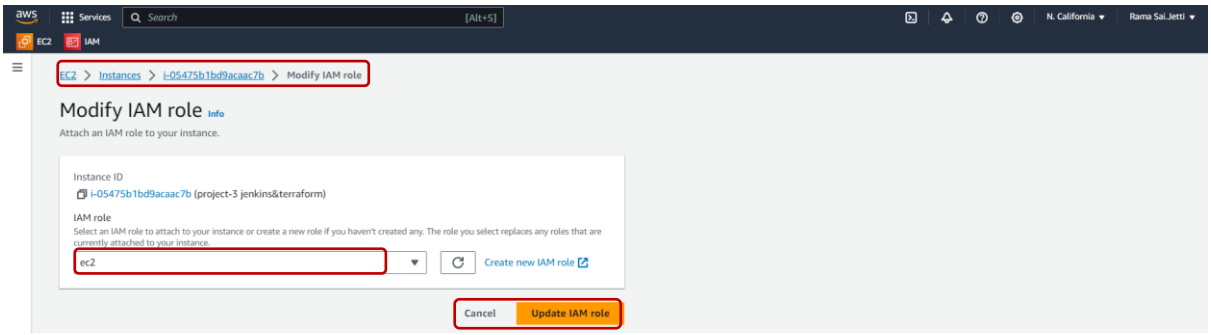

IAM > Roles > Create role
Add permissions
Permissions policies (1/884)
ec2
AmazonEC2FullAccess
Provides full access to Amazon EC2 via the AWS Management Console.

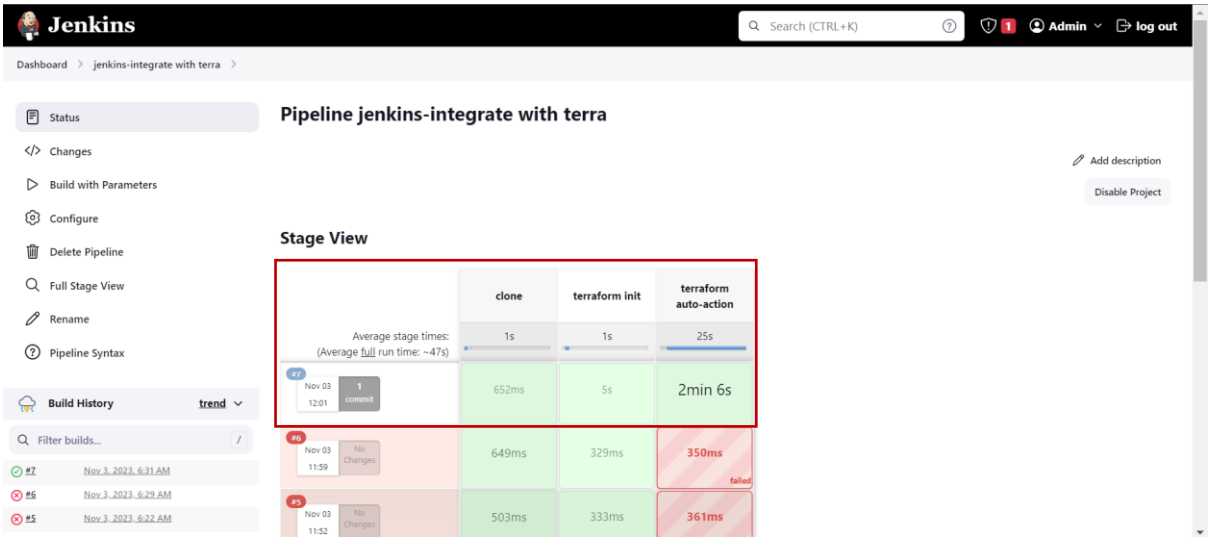

Name, review, and create


Role ec2 created.
View role
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access analyzer
Archive rules
Analyzers
Settings
Credential report
Organization activity
Service control policies (SCPs)
IAM > Roles
Roles (3)
An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.
Delete
Create role
Role name
Trusted entities
Last activity
AWSServiceRoleForSupport
AWS Service: support (Service-Linke
AWSServiceRoleForTrustedAdvisor
AWS Service: trustedadvisor (Service
ec2
AWS Service: ec2
Roles Anywhere
Authenticate your non AWS workloads and securely provide access to AWS services.
Manage
Access AWS from your non AWS workloads
Operate your non AWS workloads using the same authentication and authorization strategy that you use within AWS.
X.509 Standard
Use your own existing PKI infrastructure or use AWS Certificate Manager Private Certificate Authority to authenticate identities.
Temporary credentials
Use temporary credentials with ease and benefit from the enhanced security they provide.

After create the role, attach IAM role to the instance.







Here the Jenkins script for pipeline:

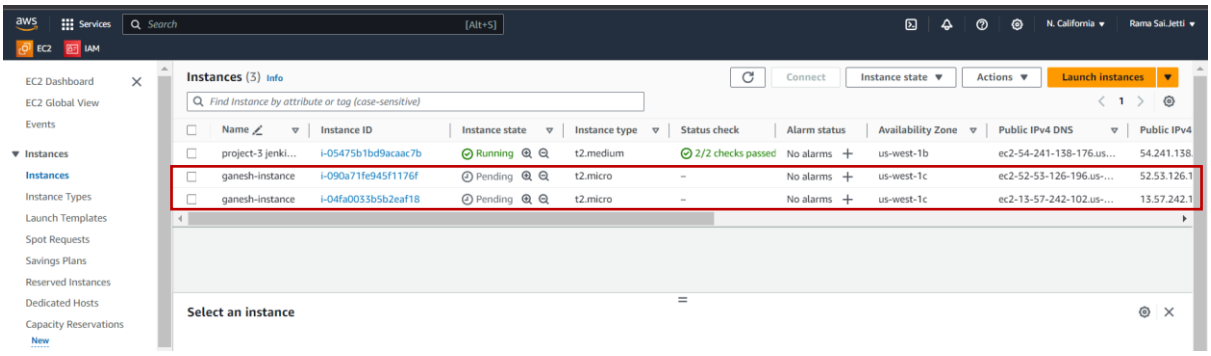
```
pipeline {
   agent any
   stages{
      stage("clone"){
         steps{
            git "https://github.com/Ganeshbayya/prtgit.git"
         }
      }

      stage('terraform init'){
         steps{
            sh 'terraform init'
```

```
        }
      }

      stage('terraform auto-action'){
        steps{
          echo 'terraform action is  --> ${action}'
          sh ('terraform ${action} --auto-approve')
        }
      }
   }
}
```

Here 2 instances created through terraform script, using Jenkins pipeline



Here is (IAC)

```
provider "aws" {
 region = "us-west-1"
 access_key = "AKIA4HO32YSPVFXxxxxx"
 secret_key = "ez4mNis+PVwoeYd+9G/a6vYun+v2iuatSJvxxxxx"
}
resource "aws_instance" "ganesh" {
 ami = "ami-0cbd40f694b804622"
 instance_type = "t2.micro"
 count = 2
tags = {
      Name = "ganesh-instance"
}
}
```

NOTE: Don't Expose Your **access_key** & **secret key**, it's leads to suspend aws account

----------EnD----------