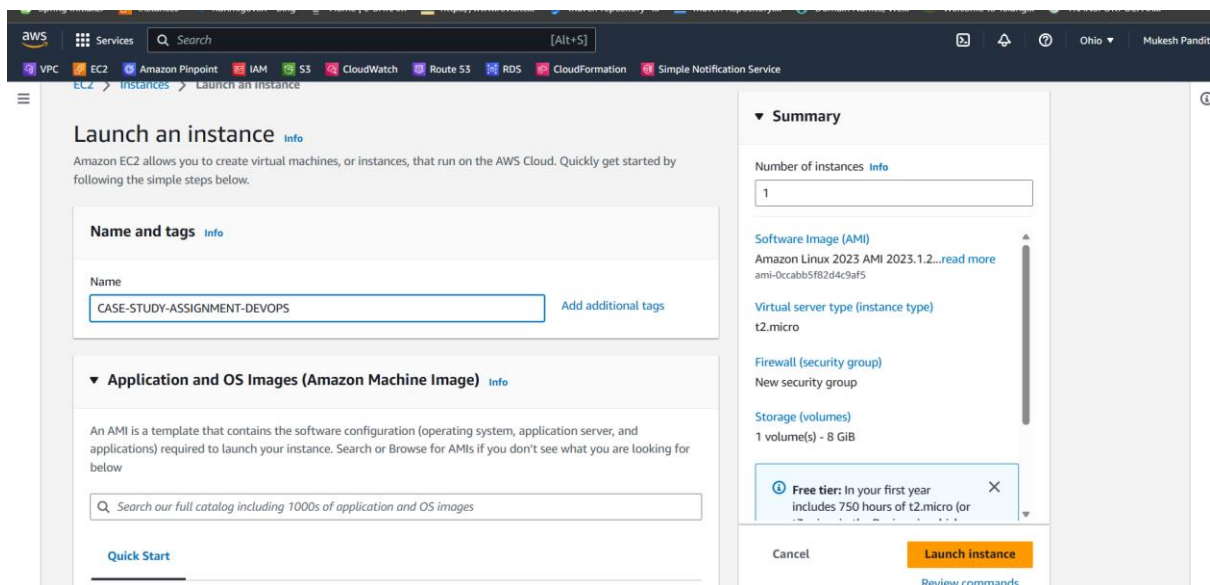# AWS -DEVOPS
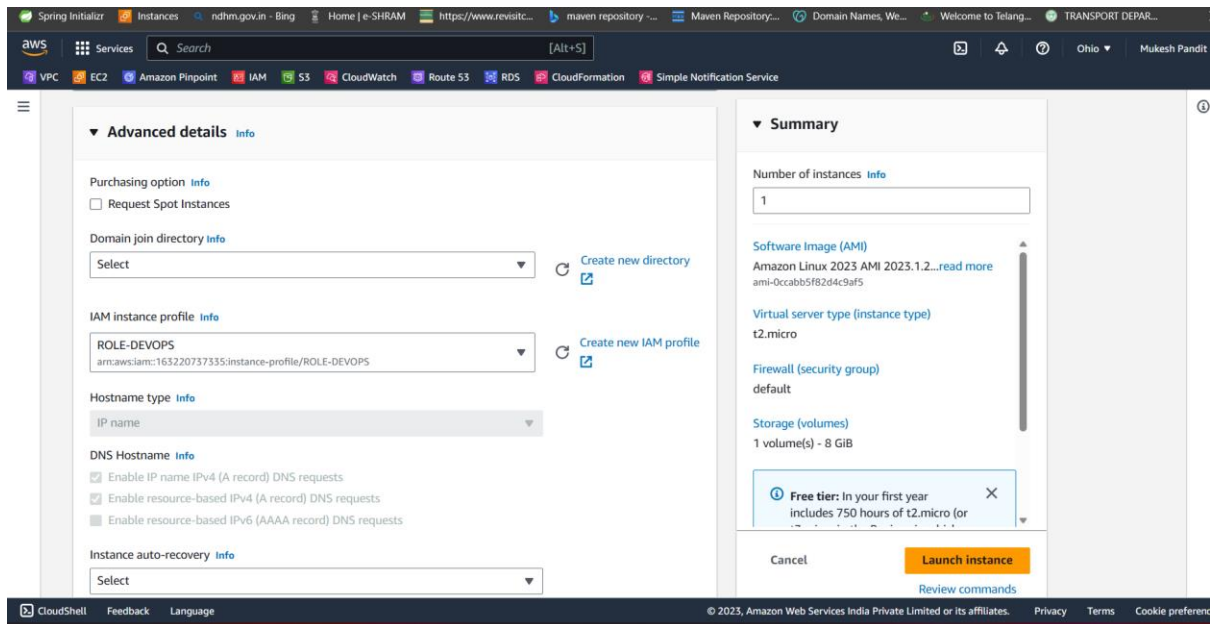
Case study assignment
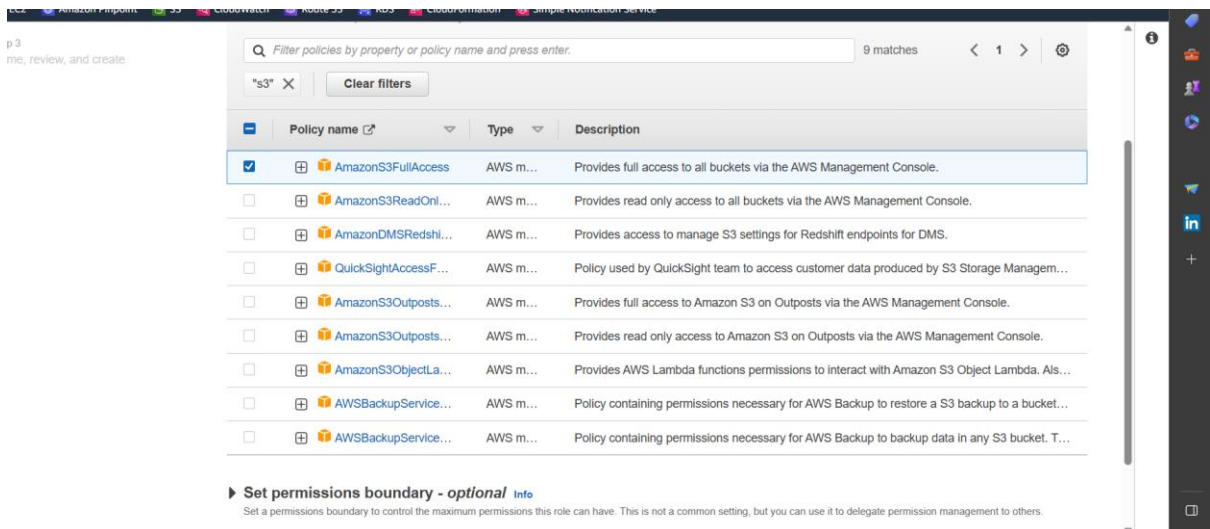
Tasks To Be Performed:

1. Create a website in any language of your choice and push the code into GitHub.
2. Migrate your GitHub repository into the AWS CodeCommit repository.
3. Create two CodeDeploy deployments (for the QA stage and the Production stage) with an EC2 deployment group into which you can push the code from the CodeCommit repository.
4. Using AWS CodePipeline, create a software development life cycle:

   a. The source is the CodeCommit repository
   b. The code will be pushed into the deployment created in CodeDeploy
   c. There should be two stages in deployment, the QA stage and the Production stage
   d. Only when the QA stage is successful, the Production stage should be executed

5. Create a third stage where the same website is pushed into an Elastic Beanstalk environment.

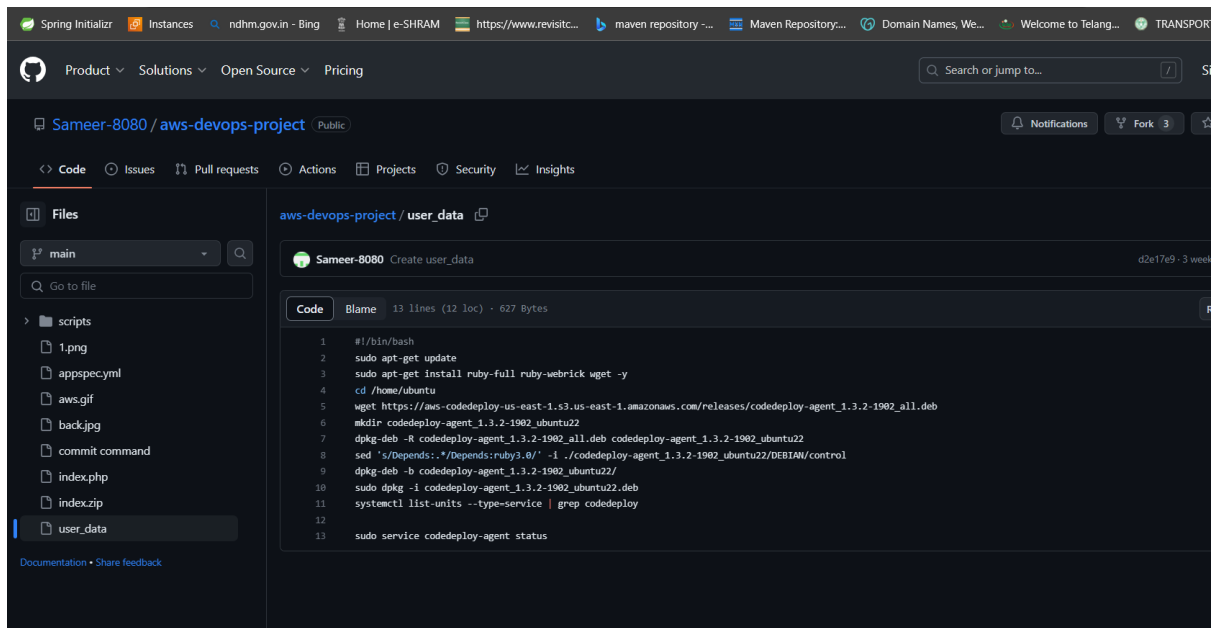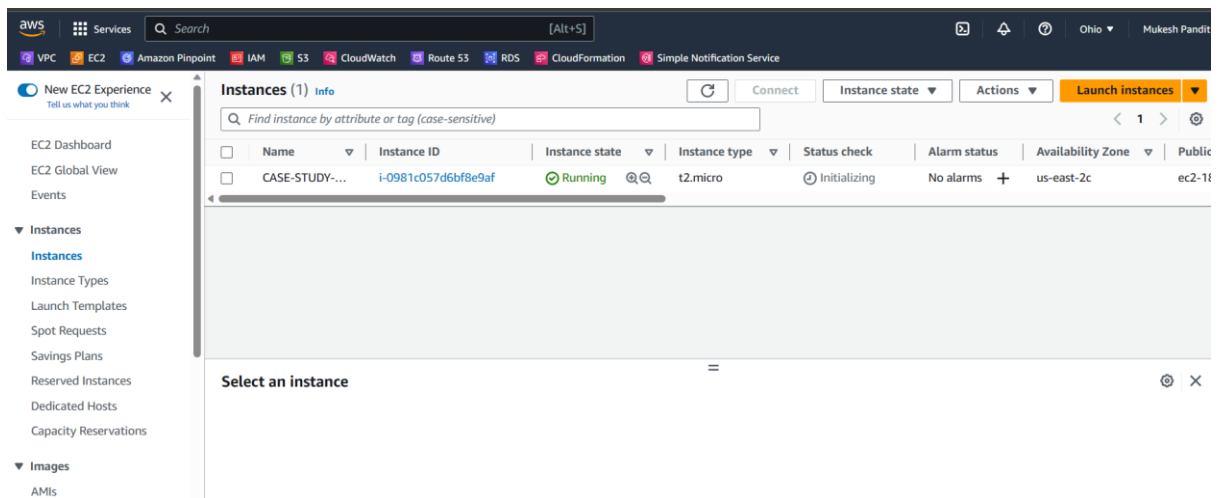Let start a new website to create github .

Let we create a EC2 instance

We need to copy the ode of git hub where given by it

Product ⌄   Solutions ⌄   Open Source ⌄   Pricing

Search or jump to...

☐ Sameer-8080 / aws-devops-project Public

🔔 Notifications   ⑂ Fork 3

<> Code   ⊙ Issues   ⑂ Pull requests   ⊙ Actions   ⊞ Projects   ⊙ Security   Insights

Files

main ⌄

Go to file

> 📁 scripts
📄 1.png
📄 appspec.yml
📄 aws.gif
📄 back.jpg
📄 commit command
📄 index.php
📄 index.zip
📄 user_data

Documentation • Share feedback

aws-devops-project / user_data

Sameer-8080   Create user_data

d2e17e9 · 3 week

Code   Blame   13 lines (12 loc) · 627 Bytes

```
1    #!/bin/bash
2    sudo apt-get update
3    sudo apt-get install ruby-full ruby-webrick wget -y
4    cd /home/ubuntu
5    wget https://aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com/releases/codedeploy-agent_1.3.2-1902_all.deb
6    mkdir codedeploy-agent_1.3.2-1902_ubuntu22
7    dpkg-deb -R codedeploy-agent_1.3.2-1902_all.deb codedeploy-agent_1.3.2-1902_ubuntu22
8    sed 's/Depends:.*/Depends:ruby3.0/' -i ./codedeploy-agent_1.3.2-1902_ubuntu22/DEBIAN/control
9    dpkg-deb -b codedeploy-agent_1.3.2-1902_ubuntu22/
10   sudo dpkg -i codedeploy-agent_1.3.2-1902_ubuntu22.deb
11   systemctl list-units --type=service | grep codedeploy
12
13   sudo service codedeploy-agent status
```

Then create instance

🔵 New EC2 Experience
    Tell us what you think   ✕

EC2 Dashboard
EC2 Global View
Events

▼ Instances
    Instances
    Instance Types
    Launch Templates
    Spot Requests
    Savings Plans
    Reserved Instances
    Dedicated Hosts
    Capacity Reservations

▼ Images
    AMIs

Instances (1) Info

🔄   Connect   Instance state ⌄   Actions ⌄   **Launch instances** ⌄

🔍 Find instance by attribute or tag (case-sensitive)

< 1 > ⚙

☐	Name ▽	Instance ID	Instance state ▽	Instance type ▽	Status check	Alarm status	Availability Zone ▽	Public
☐	CASE-STUDY-...	i-0981c057d6bf8e9af	⊘ Running ⊕⊖	t2.micro	⊙ Initializing	No alarms ✚	us-east-2c	ec2-18

=

Select an instance   ⚙ ✕

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.

ROLE-DEVOPS

Maximum 64 characters. Use alphanumeric and '+=, @-_' characters.

Description
Add a short explanation for this role.

Allows EC2 instances TO

Maximum 1000 characters. Use alphanumeric and '+=, @-_' characters.

Step 1: Select trusted entities

Edit

```
1  {
2      "Version": "2012-10-17",
3      "Statement": [
4          {
```

Edit

Permissions policy summary

Policy name	Type	Attached as
AdministratorAccess-AWSElasticBeanstalk	AWS managed	Permissions policy
AmazonS3FullAccess	AWS managed	Permissions policy
AWSCodeDeployFullAccess	AWS managed	Permissions policy

Tags

Add tags - *optional* Info
Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

Add tag

You can add up to 50 more tags.

Cancel   Previous   Create role

IF WE CONNECT TO EC2 INSTACE THEN WE ARE SUCCESSFUL TO NEXT SETP
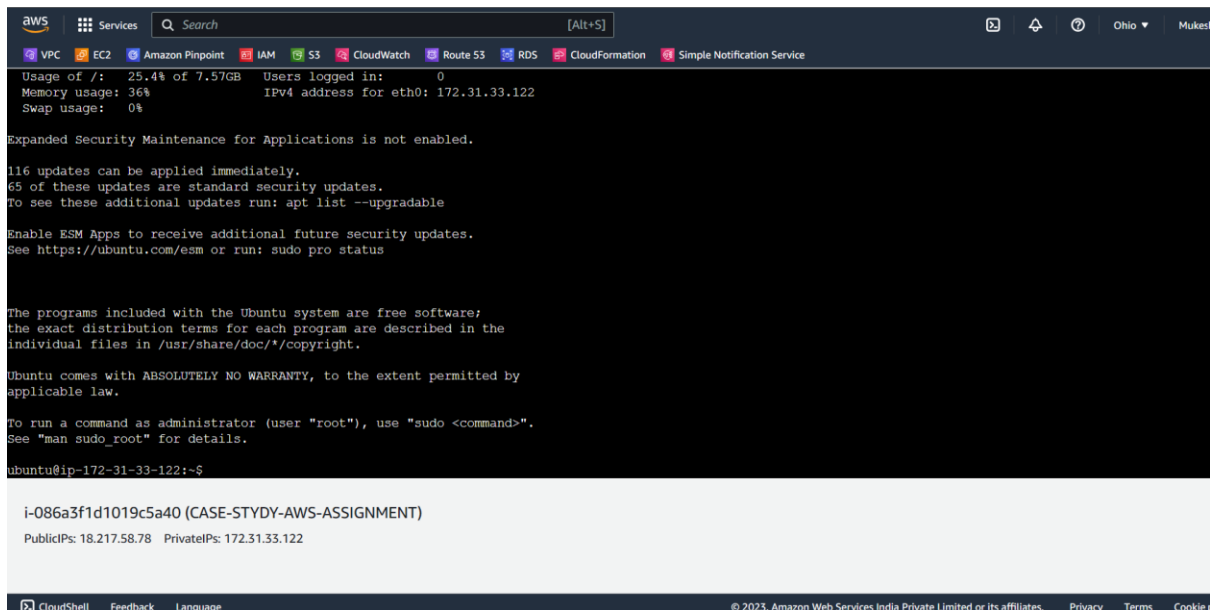
PUSH THE GIT TO MAIN

GIT PUSH HTTPS:// codecommit::us-east-2://devops

ASKING PASSWORD: WE GET THE PASSWORD AT I AM ROLE GENERATED PASSWORD AND USERNAME
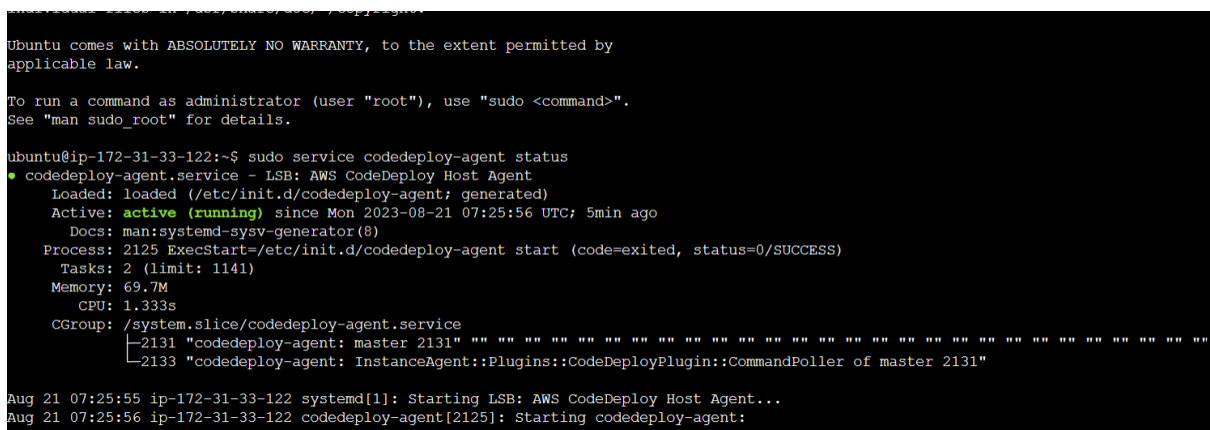
Create a repostion to instance .

Create a environment then create a pipeline to connect with instance
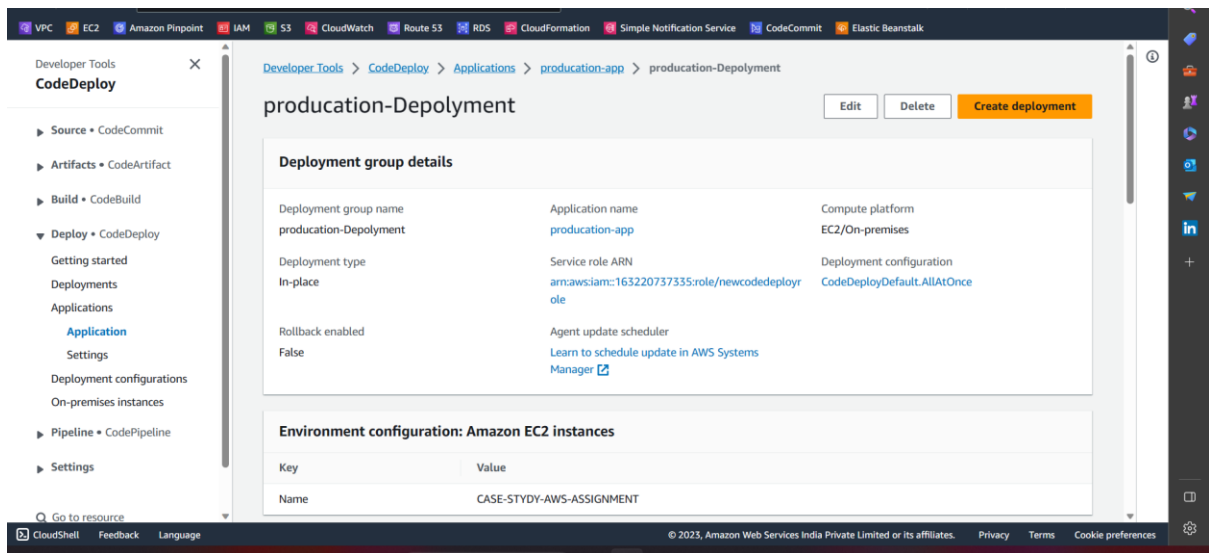
ENVIORNMENT IS READY TO CONNECT LET CHECK IT

REFRESH AGAIN U GET THEN PAGE

. COMPLETED ASSIGNMENT