2025

Ashish Kumar Patel

Executive Post-Graduate Certificate in Cloud Computing and DevOps - Sept'24

1/3/2025

DevOps Project



Contents

[Task 1: Setup infrastructure(infra) using Terraform (tf) 2](#_Toc187004757)

[Subtask 1.1: Setup Terraform 2](#_Toc187004758)

[Subtask 1.2: Create VPC Network 4](#_Toc187004759)

[Subtask 1.3: Create 3 EC2 instances (bastion, Jenkins, app), SecurityGroups 5](#_Toc187004760)

[Teffaform Code Repo 9](#_Toc187004761)

[Task 2 : Setup Config Management for hosts using Ansible & CI pipeline using Jenkins 11](#_Toc187004762)

[Subtask 2.1 Ansible setup 11](#_Toc187004763)

[Ansible Code Repo 13](#_Toc187004764)

[Subtask 2.2: Jenkins setup 13](#_Toc187004765)

[Subtask 2.3: Create an ALB 14](#_Toc187004766)

[Subtask 2.4: Create an ECR repository to store the docker images 15](#_Toc187004767)

[Task 3 : Dockerize sample Node.js app & write Jenkinsfile 17](#_Toc187004768)

[Subtask 3.1 Prepare repository in GitHub 17](#_Toc187004769)

[Subtask 3.2 Setup Jenkins CI Job 17](#_Toc187004770)

[Subtask 3.3 Setup Jenkins CD Job 17](#_Toc187004771)

[Docker Images and running status 18](#_Toc187004772)

[Accessing App - Vote 19](#_Toc187004773)

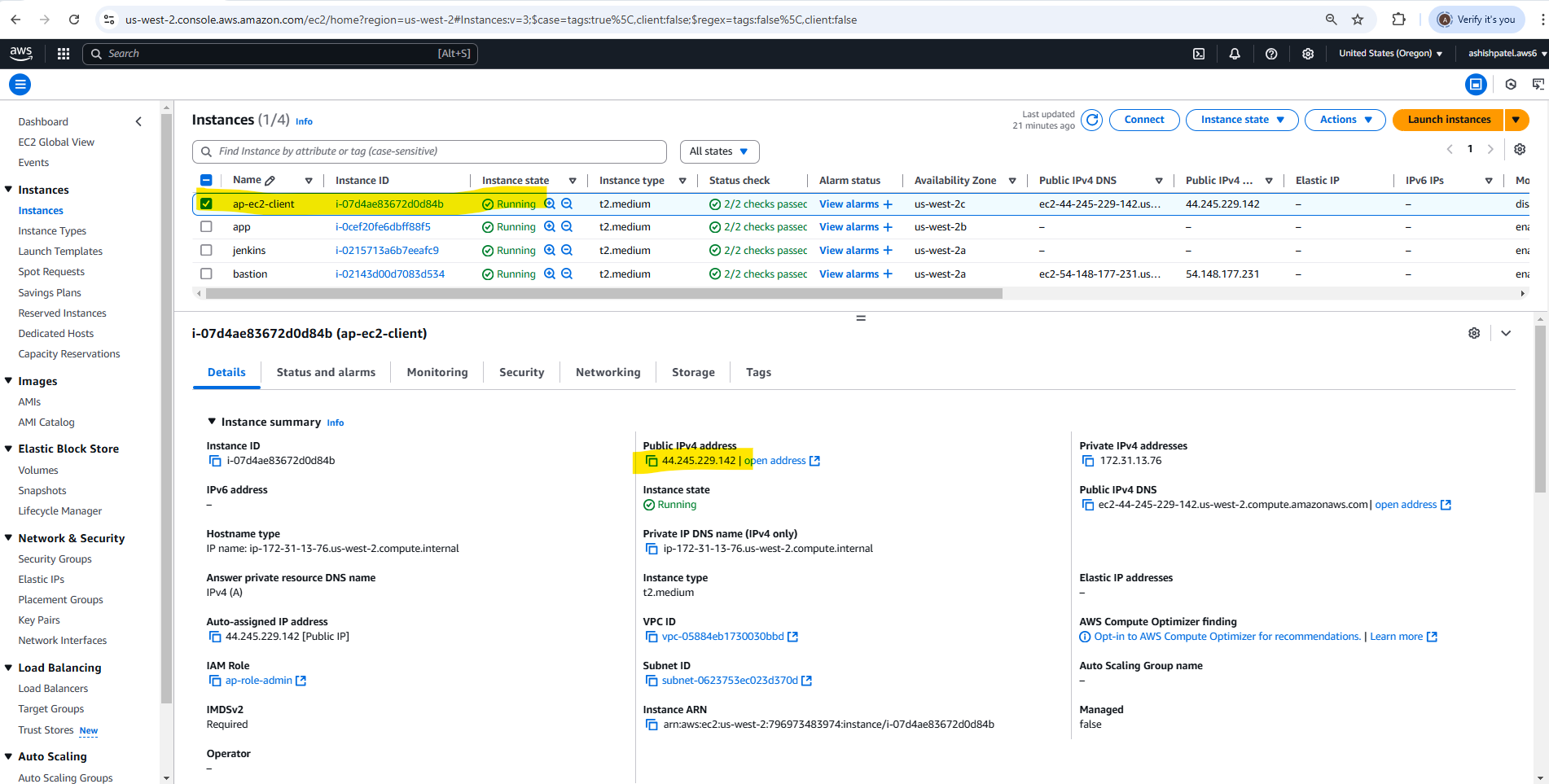
## Task 1: Setup infrastructure(infra) using Terraform (tf)

### Subtask 1.1: Setup Terraform

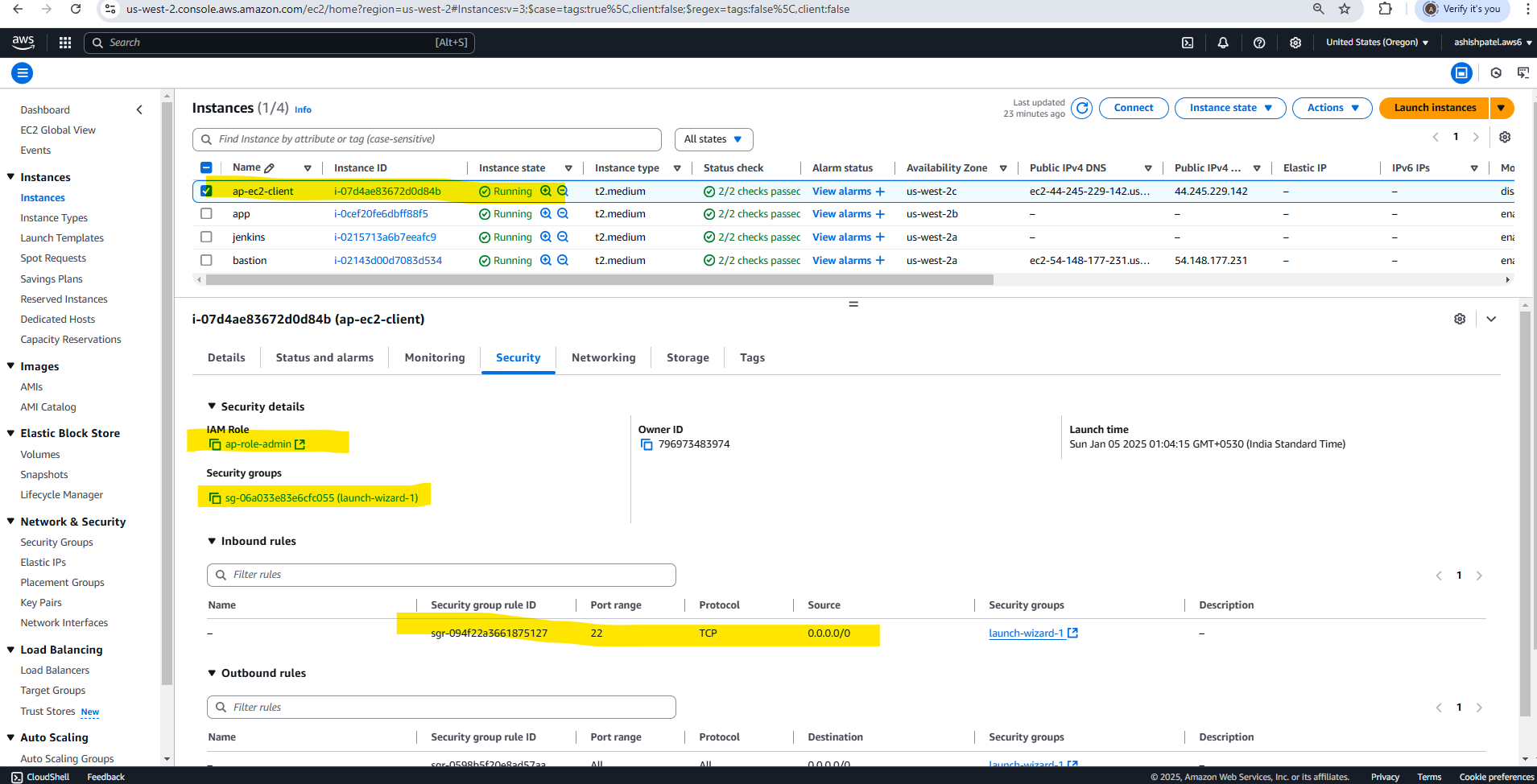
Ensure AWS CLI is installed and configured with full access in your Linux machine.

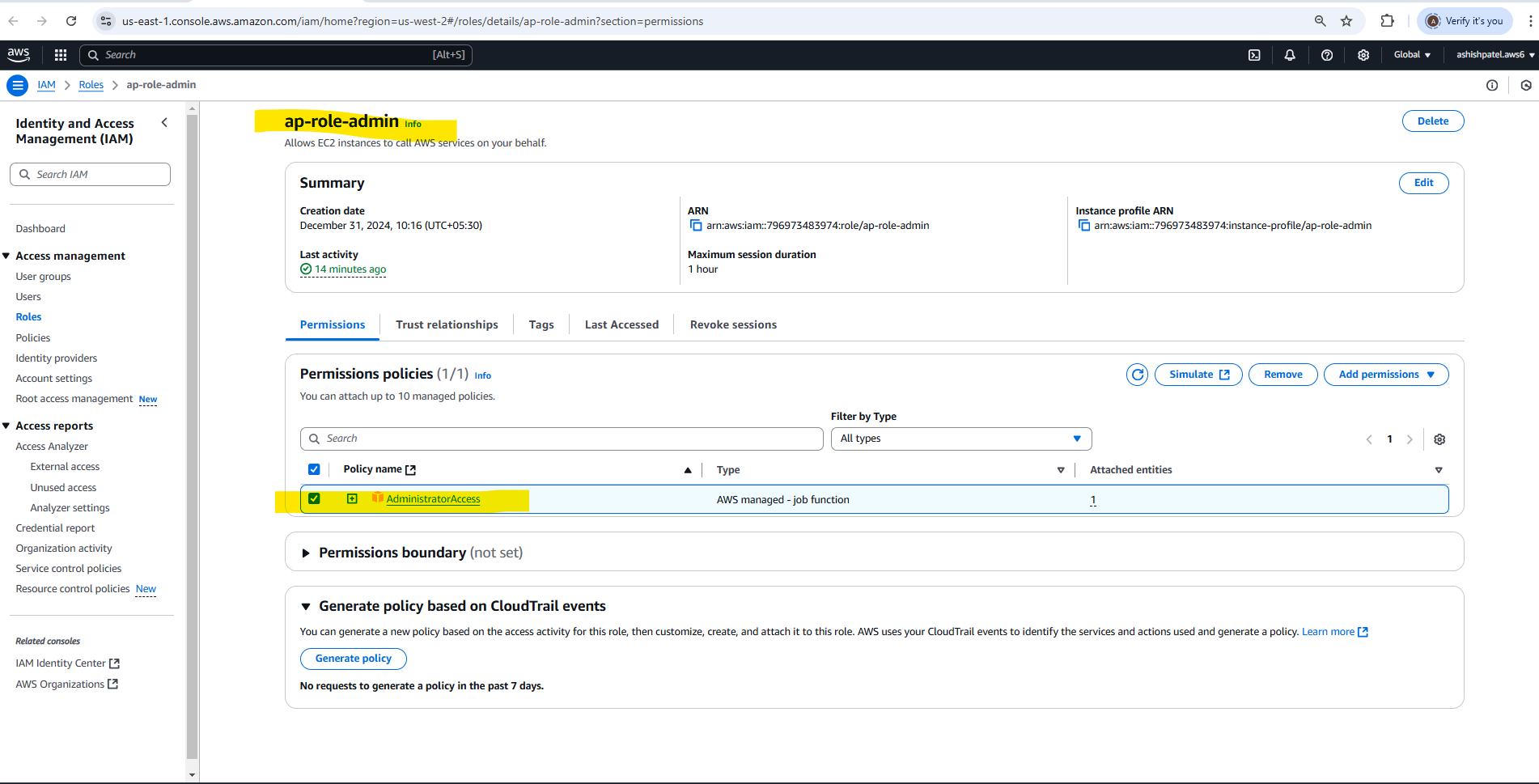
Initialize a bucket in s3 for the backend state store using Terraform. This bucket will be used later in this project to use the state files of Terraform.

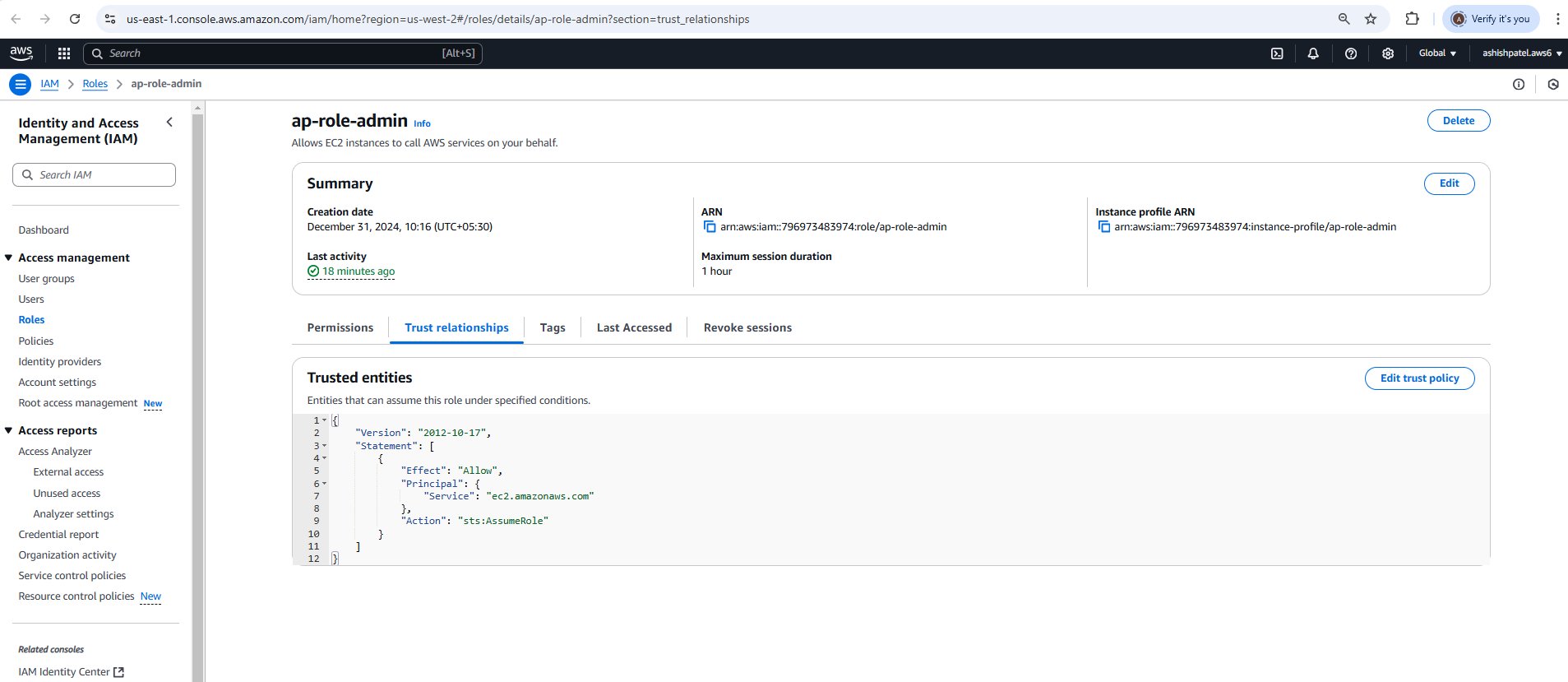
#### Create Client host EC2 Ubuntu



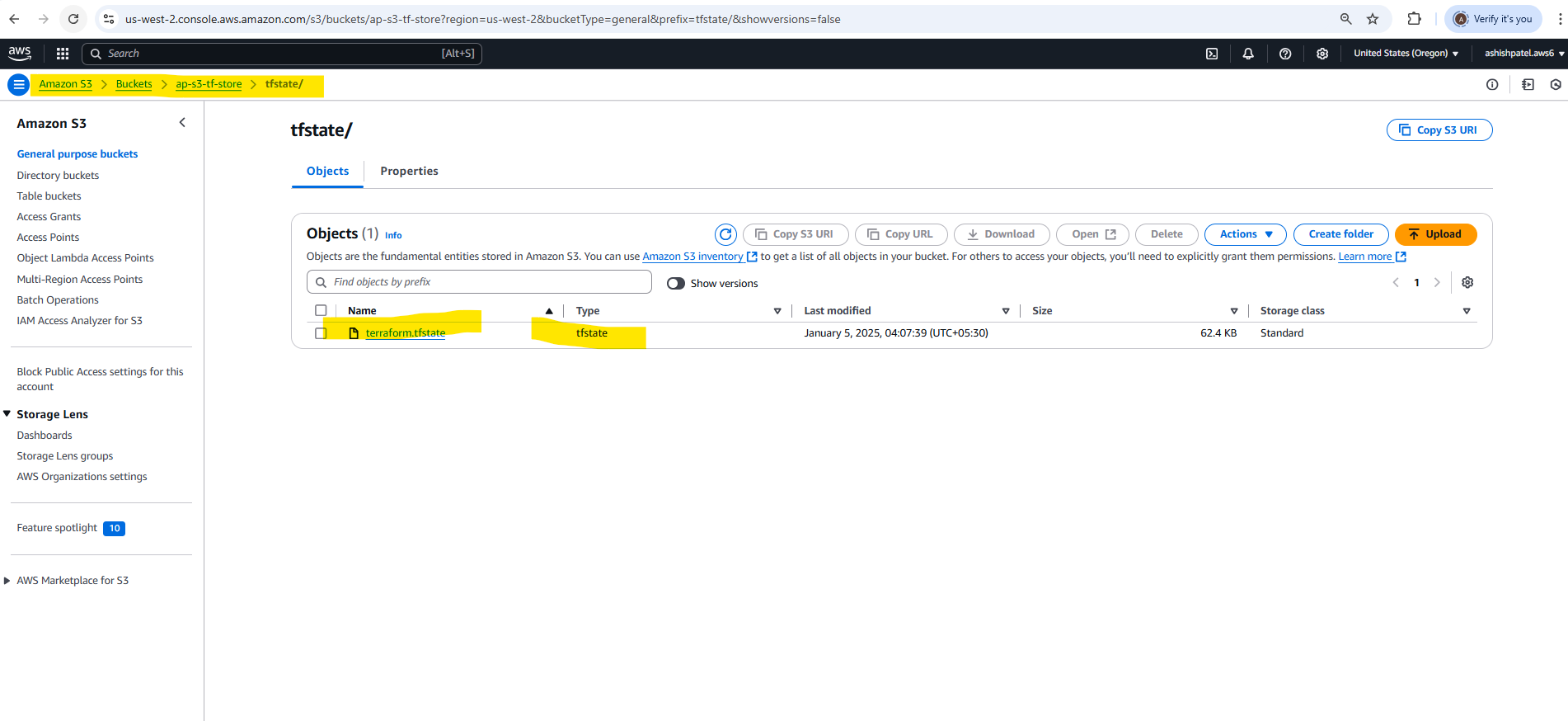
#### Create a IAM Role ap-role-admin for EC2 with Role [AdministratorAccess](https://us-east-1.console.aws.amazon.com/iam/home?region=us-west-2#/policies/details/arn%3Aaws%3Aiam%3A%3Aaws%3Apolicy%2FAdministratorAccess)







#### Setup S3 bucket for Terraform backend store



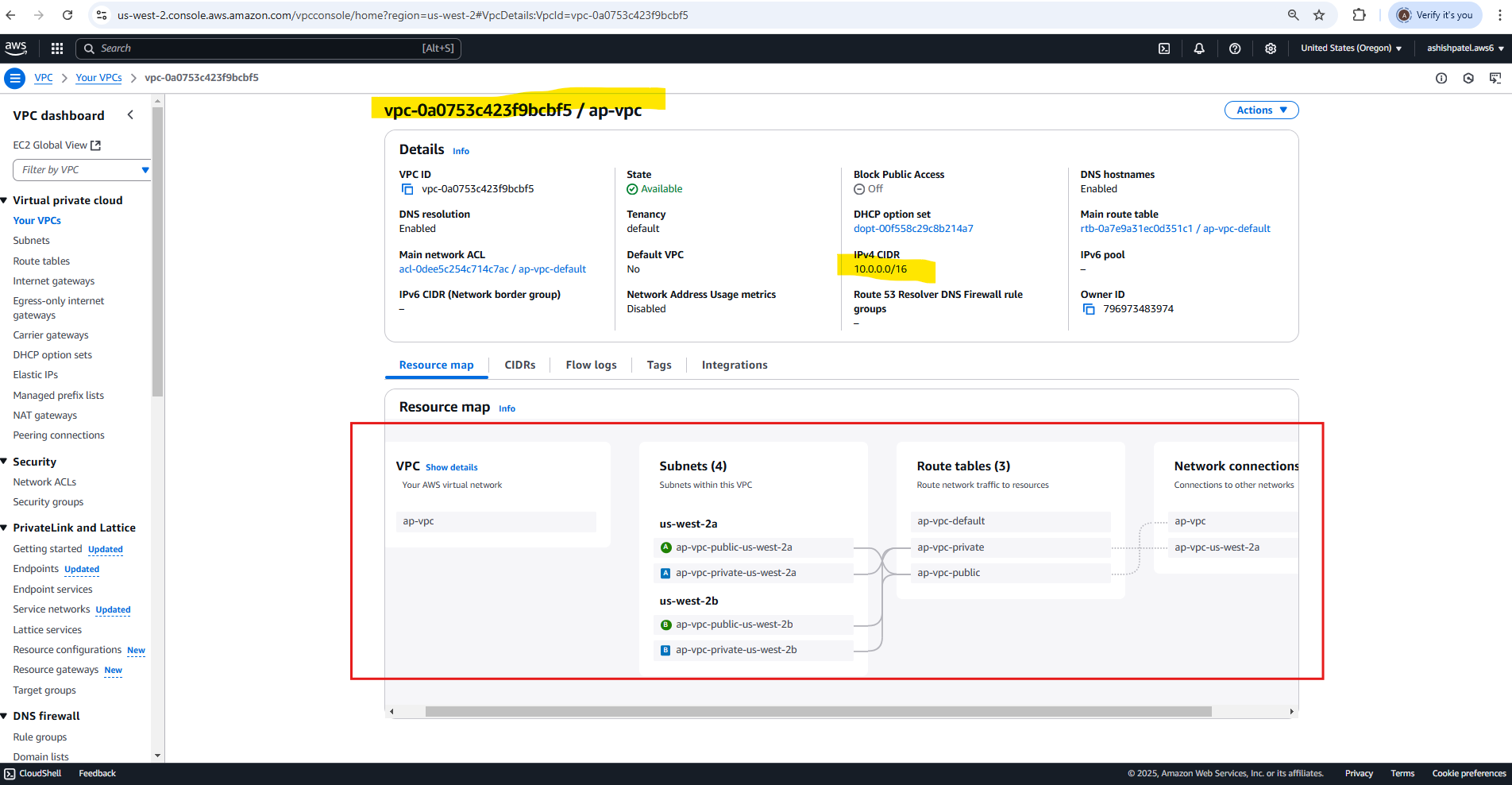
#### Install AWS CLI & Terraform to EC2 Client Host



### Subtask 1.2: Create VPC Network

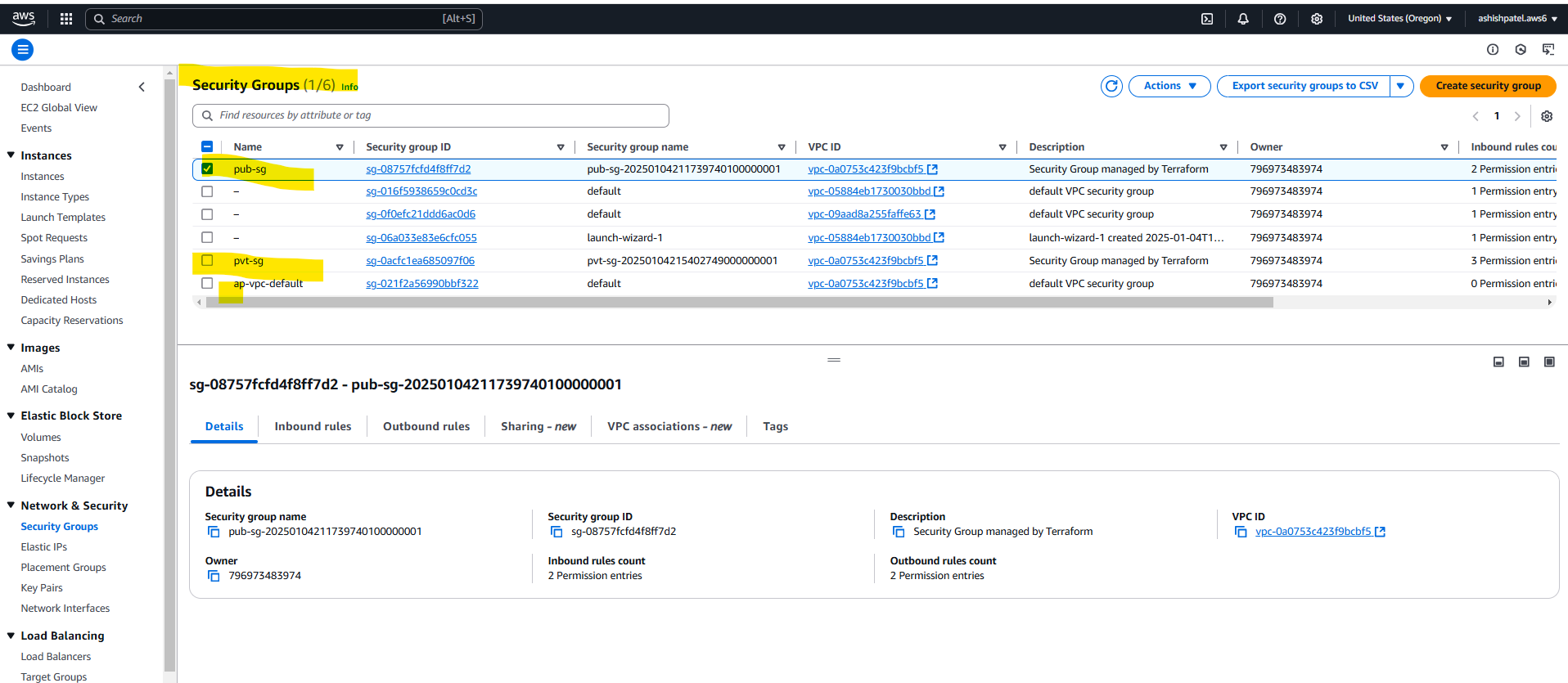
* AWS VPC
* 1 IGW
* 1 NAT-GW in AZ-a,
* Allow 0.0.0.0/0
* Subnets ( 2 private, 1 each in AZ-a &b)
* Route Tables for both subnets
* Choose a /16 CIDR for VPC and /24 CIDRs for other subnets

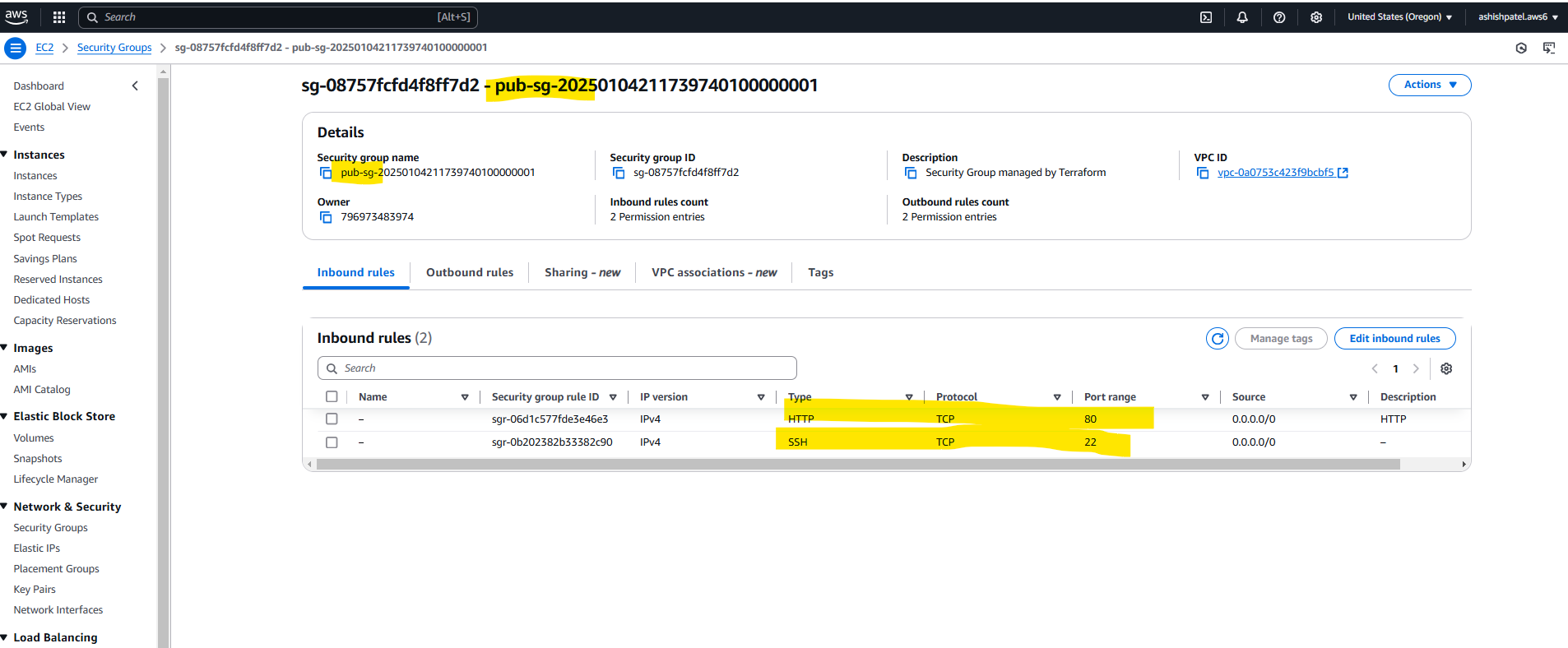
Using Terraform create VPC Network

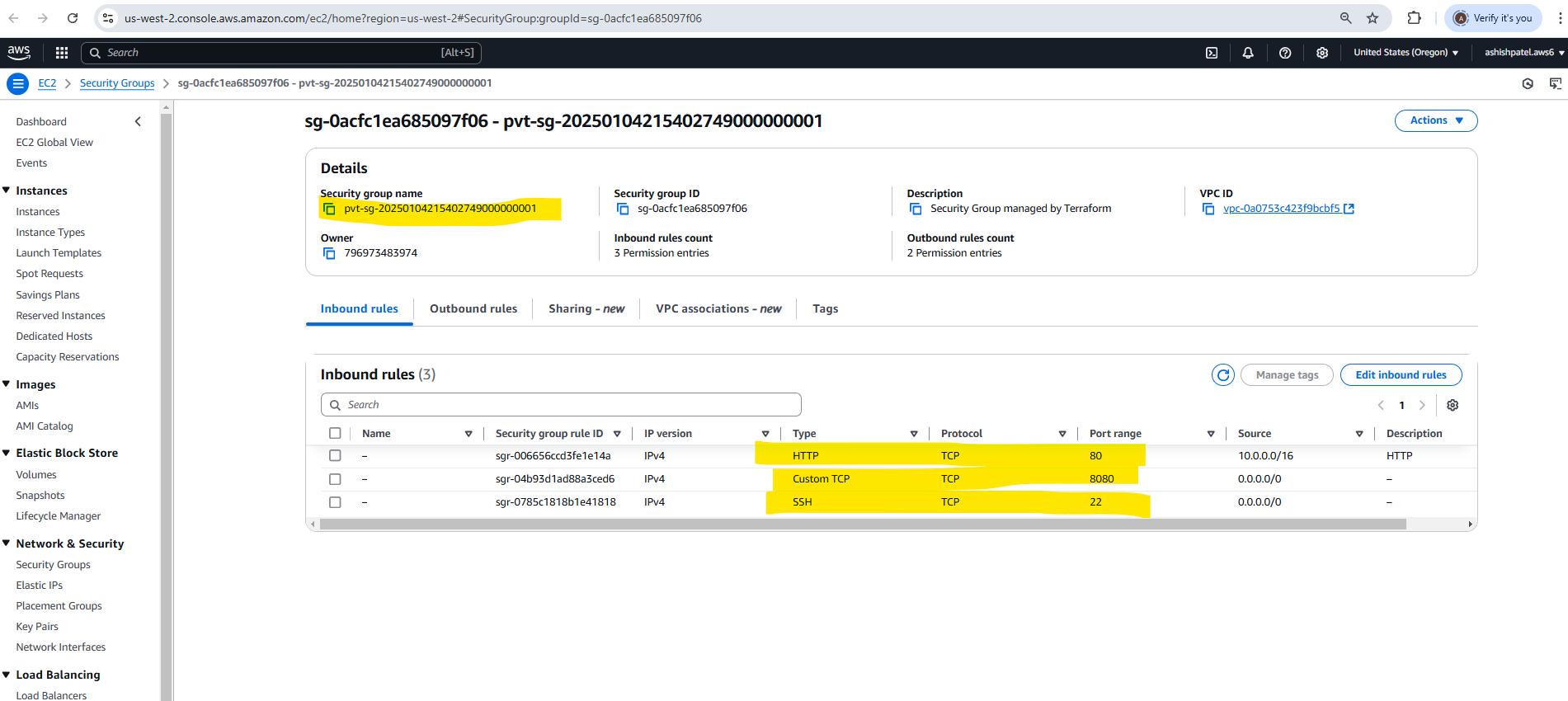


### Subtask 1.3: Create 3 EC2 instances (bastion, Jenkins, app), SecurityGroups

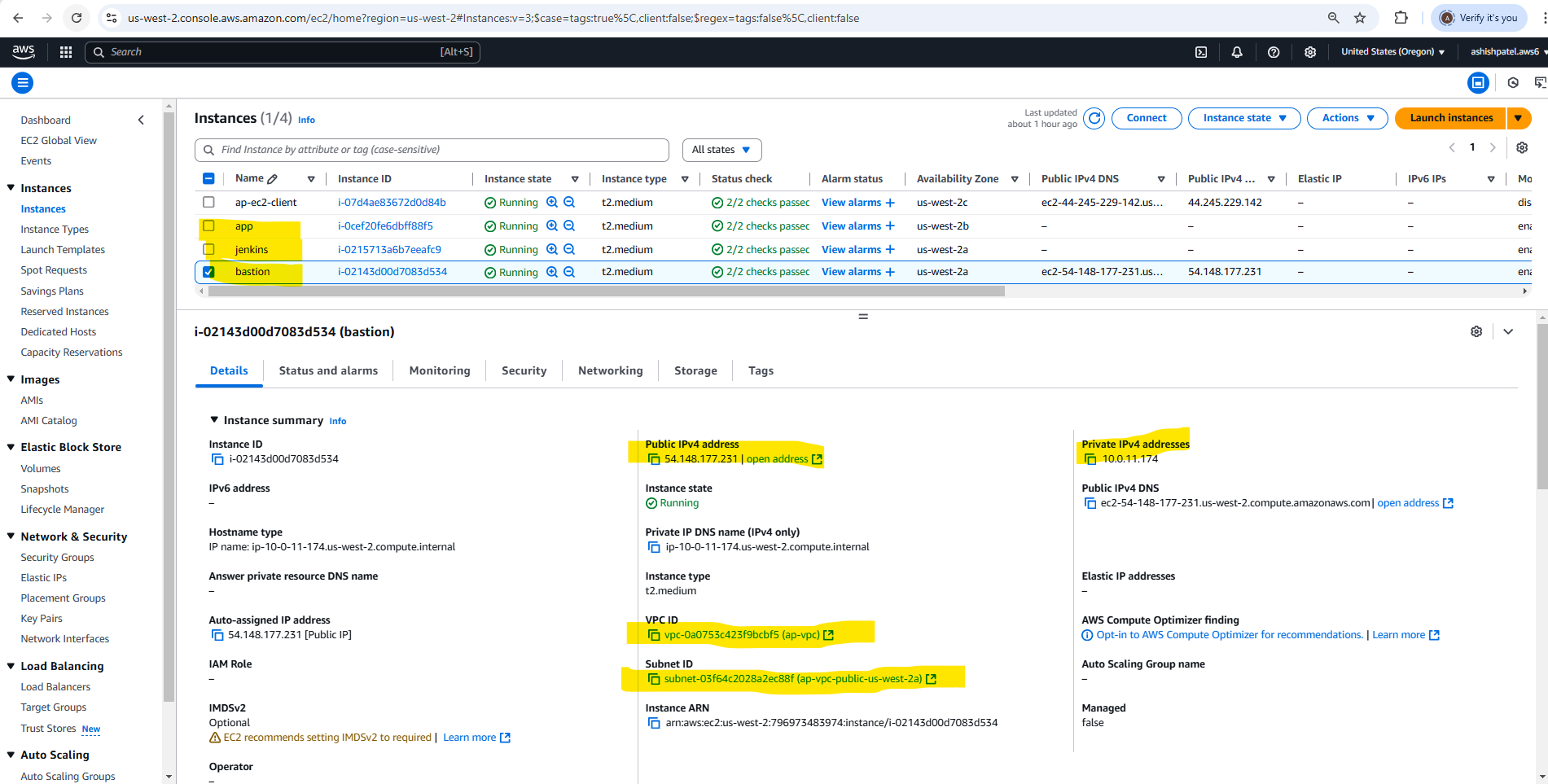
#### Security Groups [pub-sg, pvt-sg]

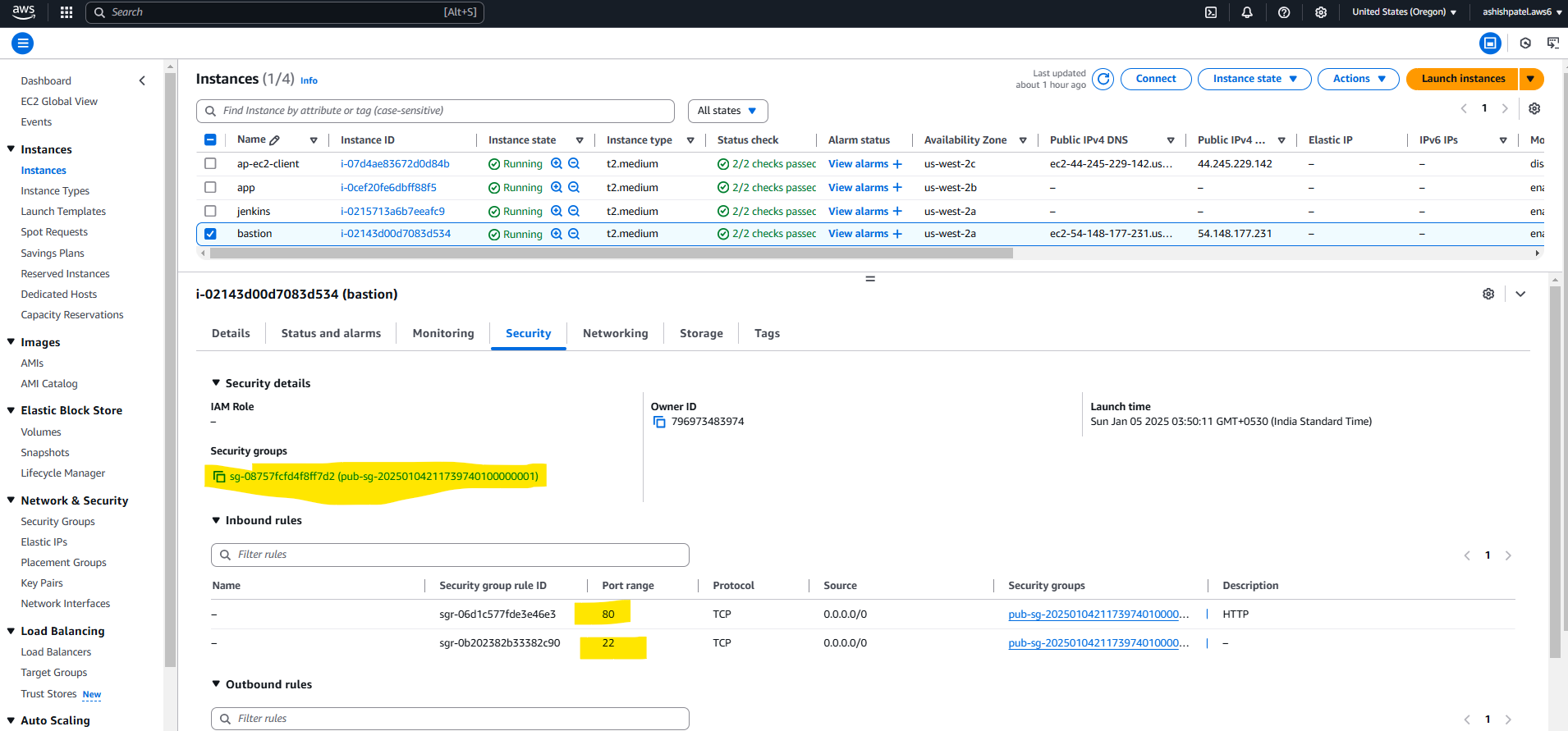




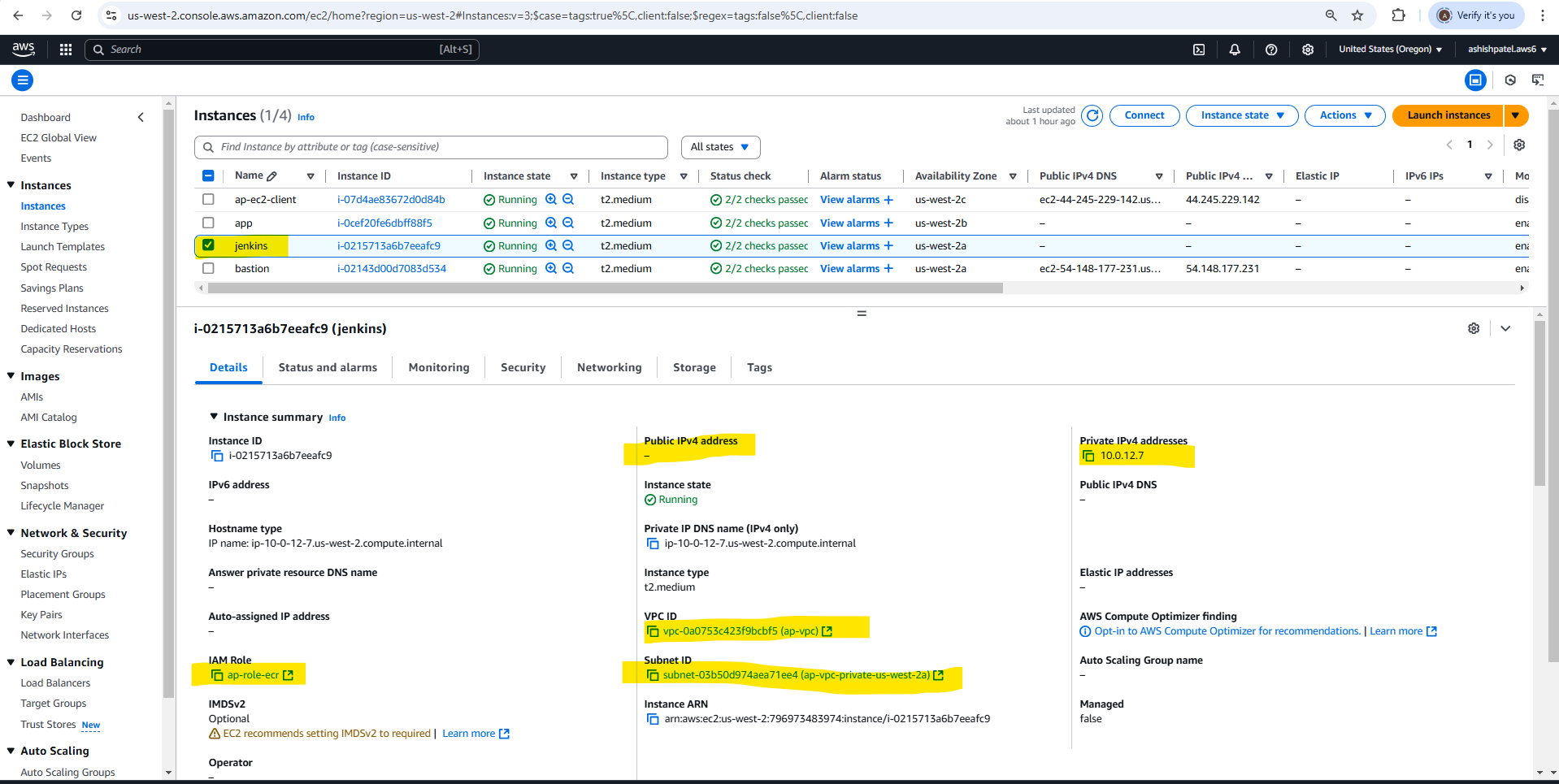


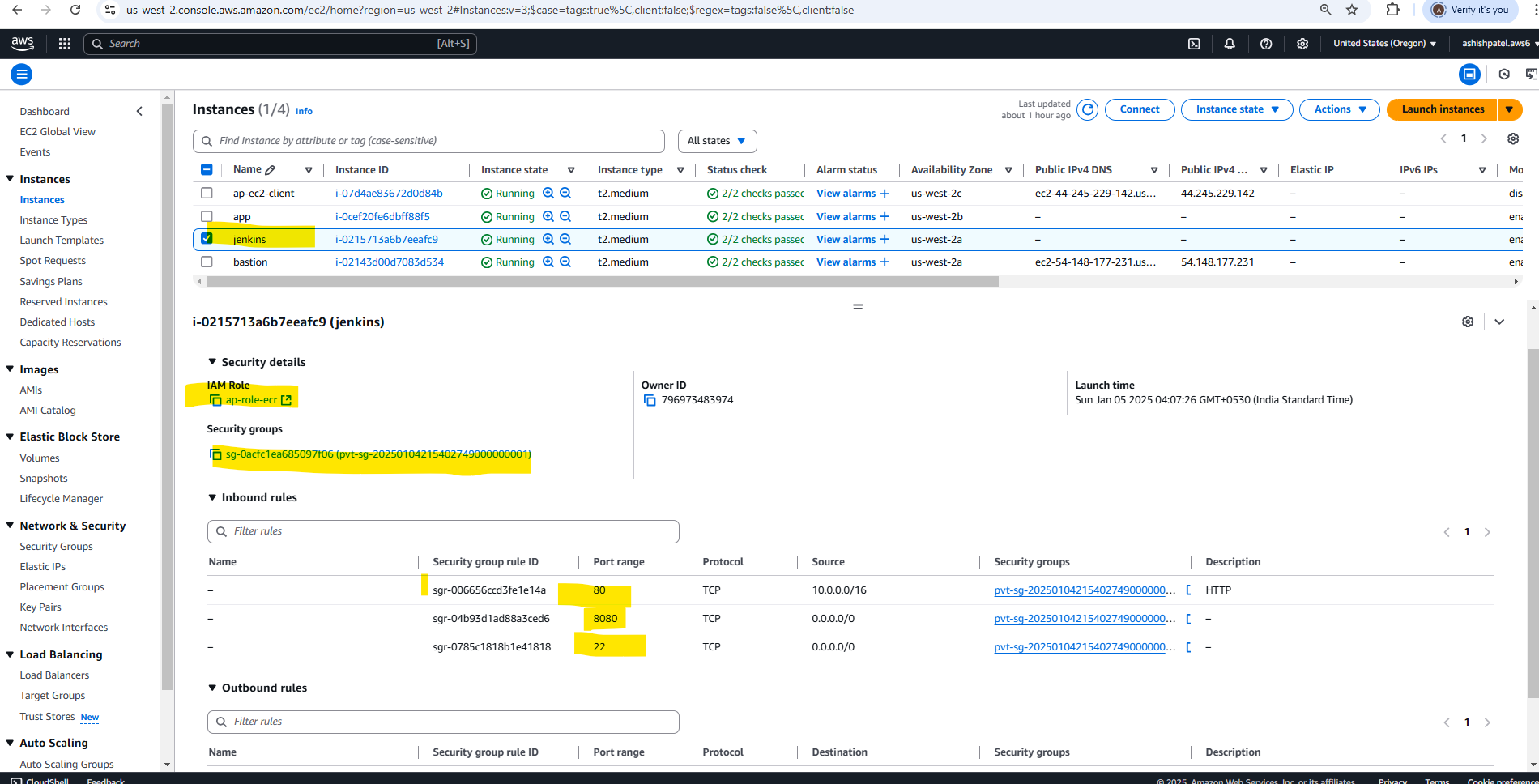
#### EC2 Bastion Instance



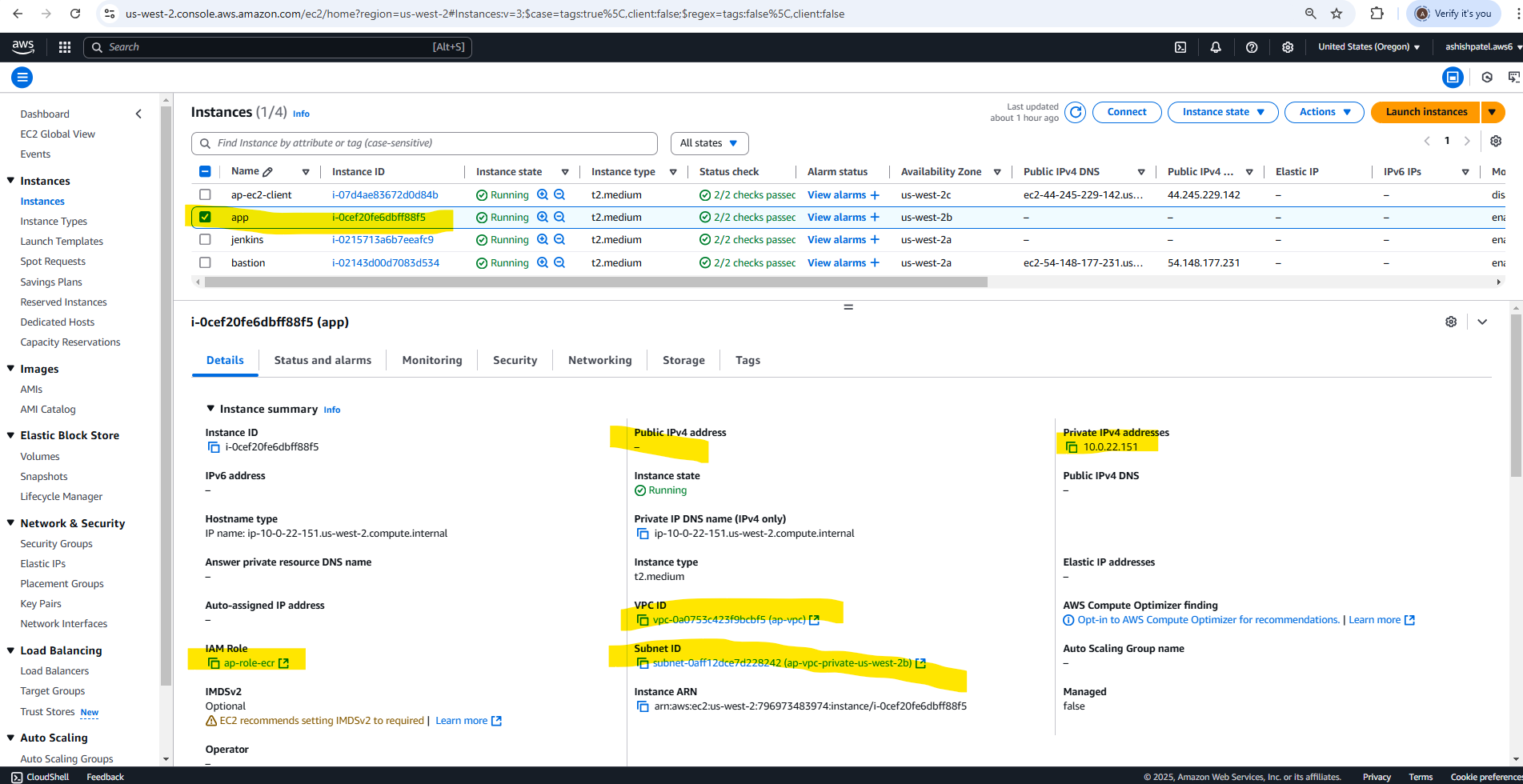


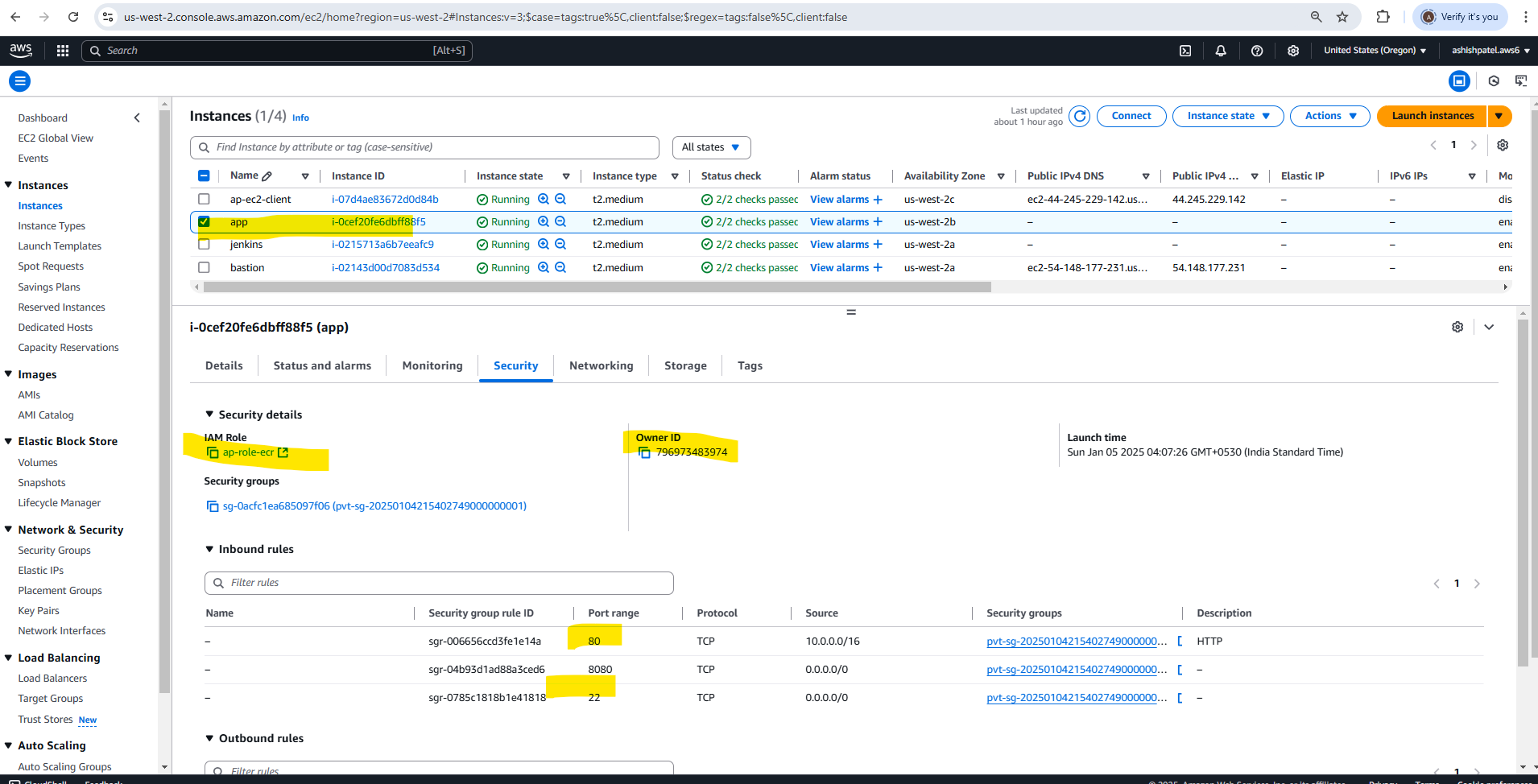
#### EC2 Jenkins Instance





#### EC2 App Instance

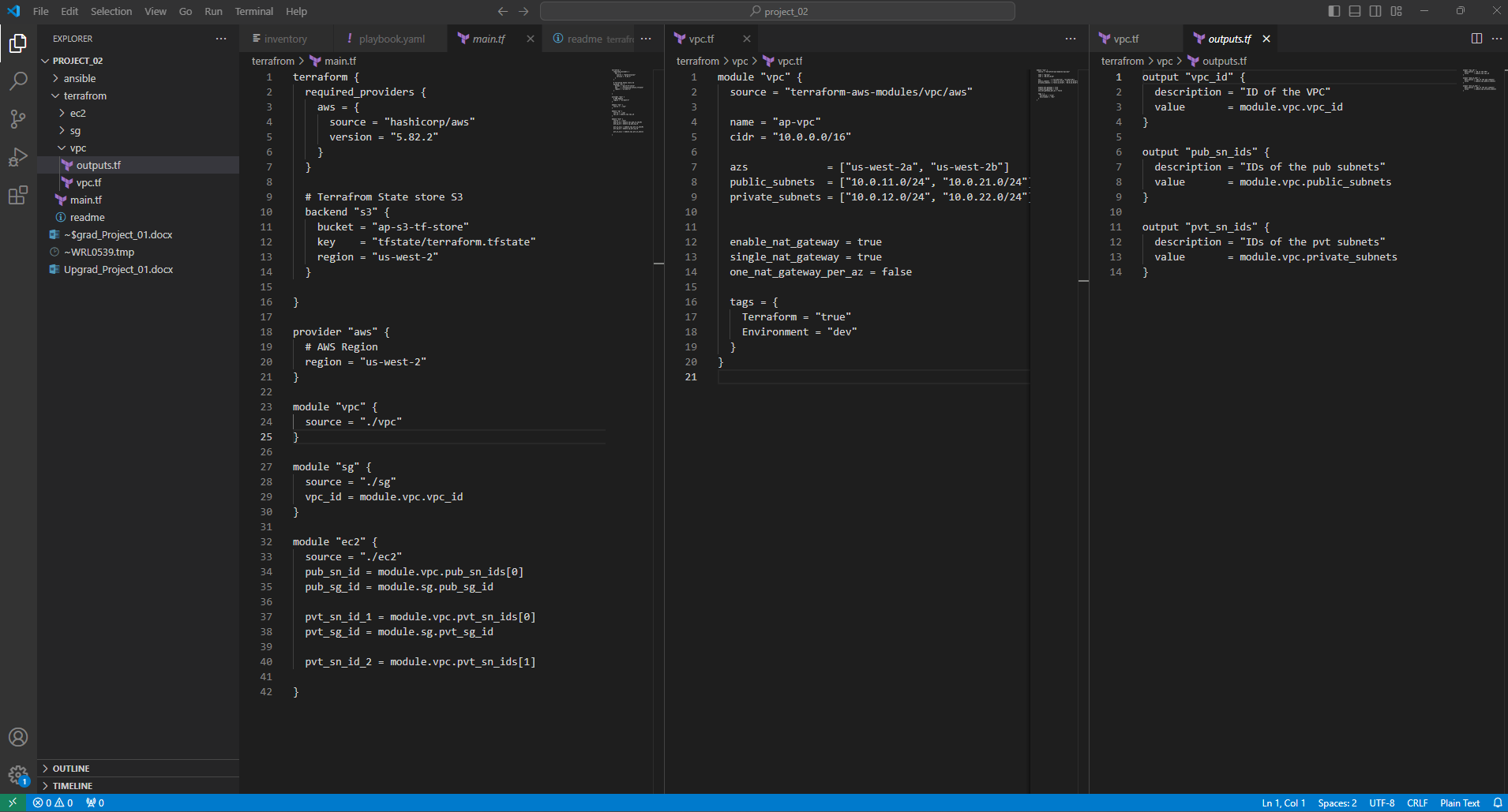




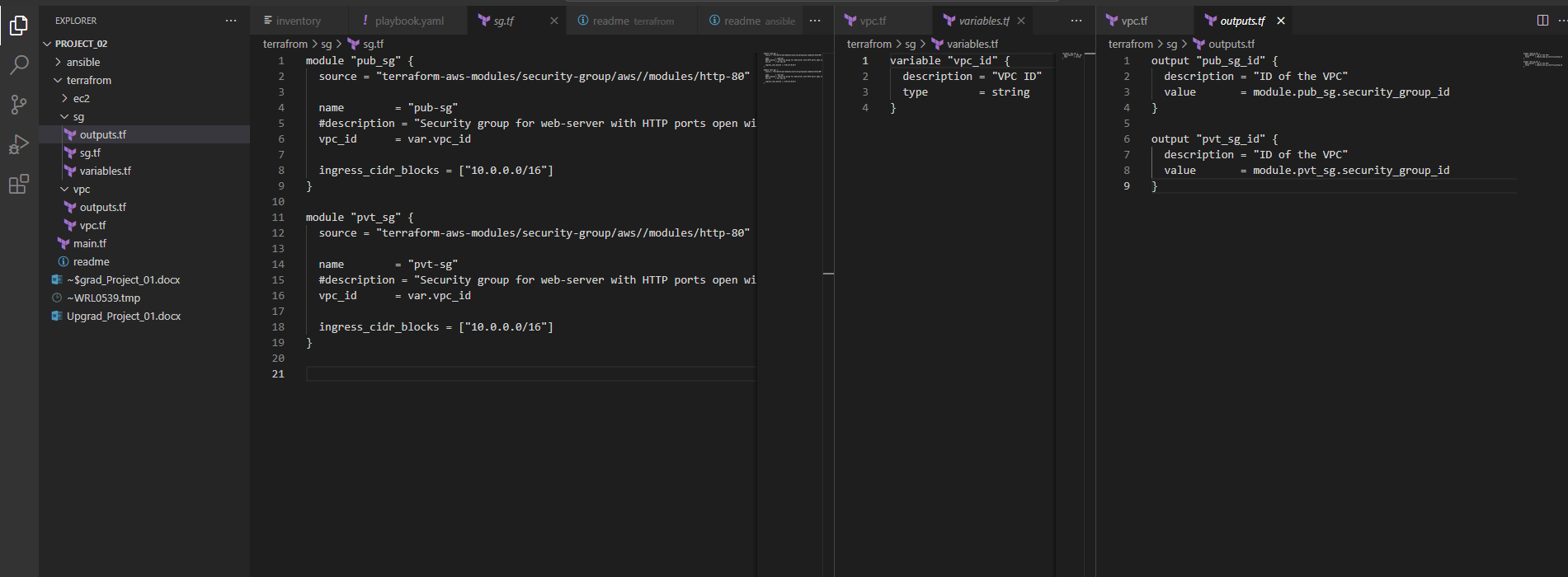
### Teffaform Code Repo

#### Main Module

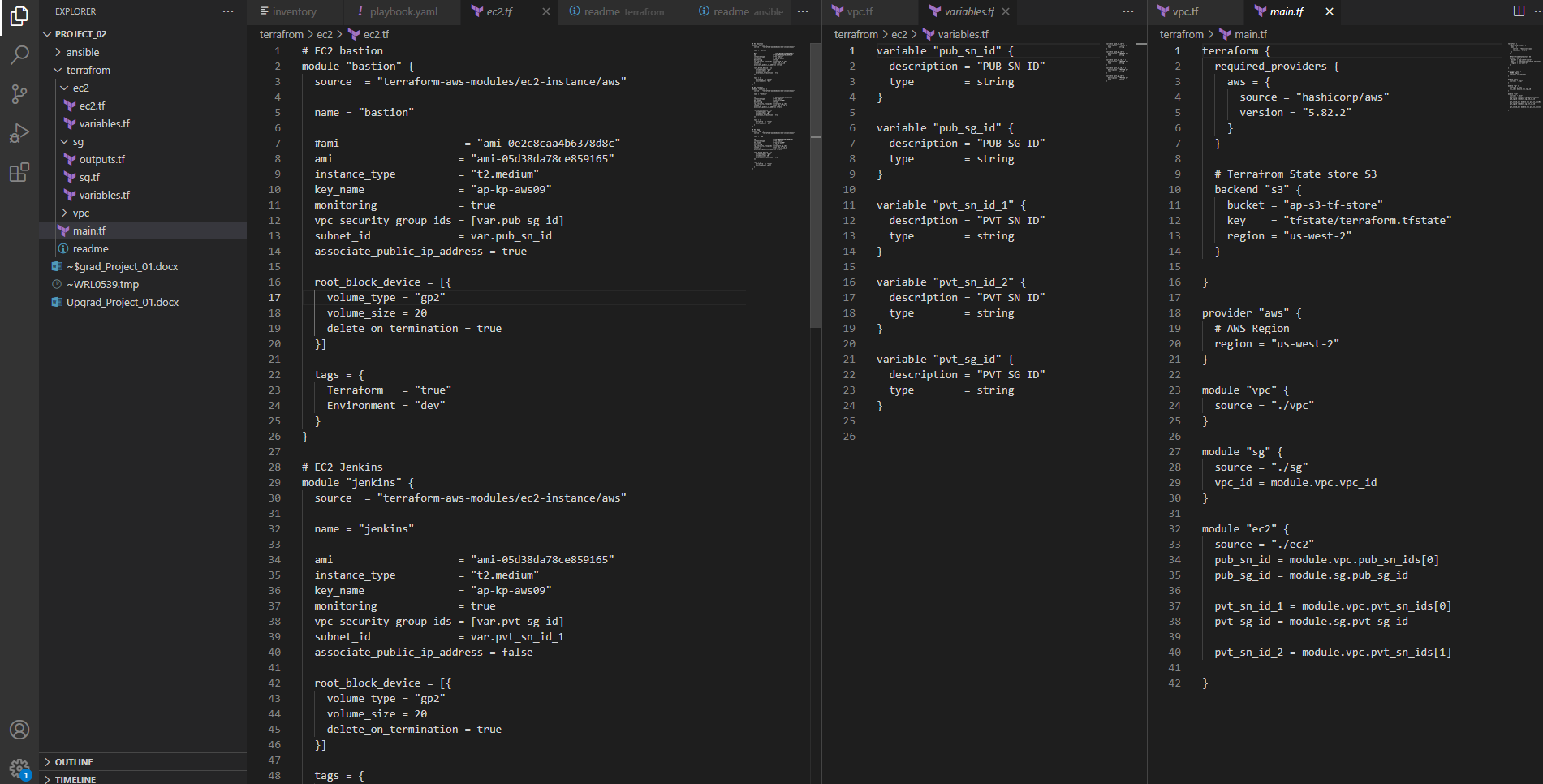
#### Module to create VPC



#### SG Module



#### EC2 Module

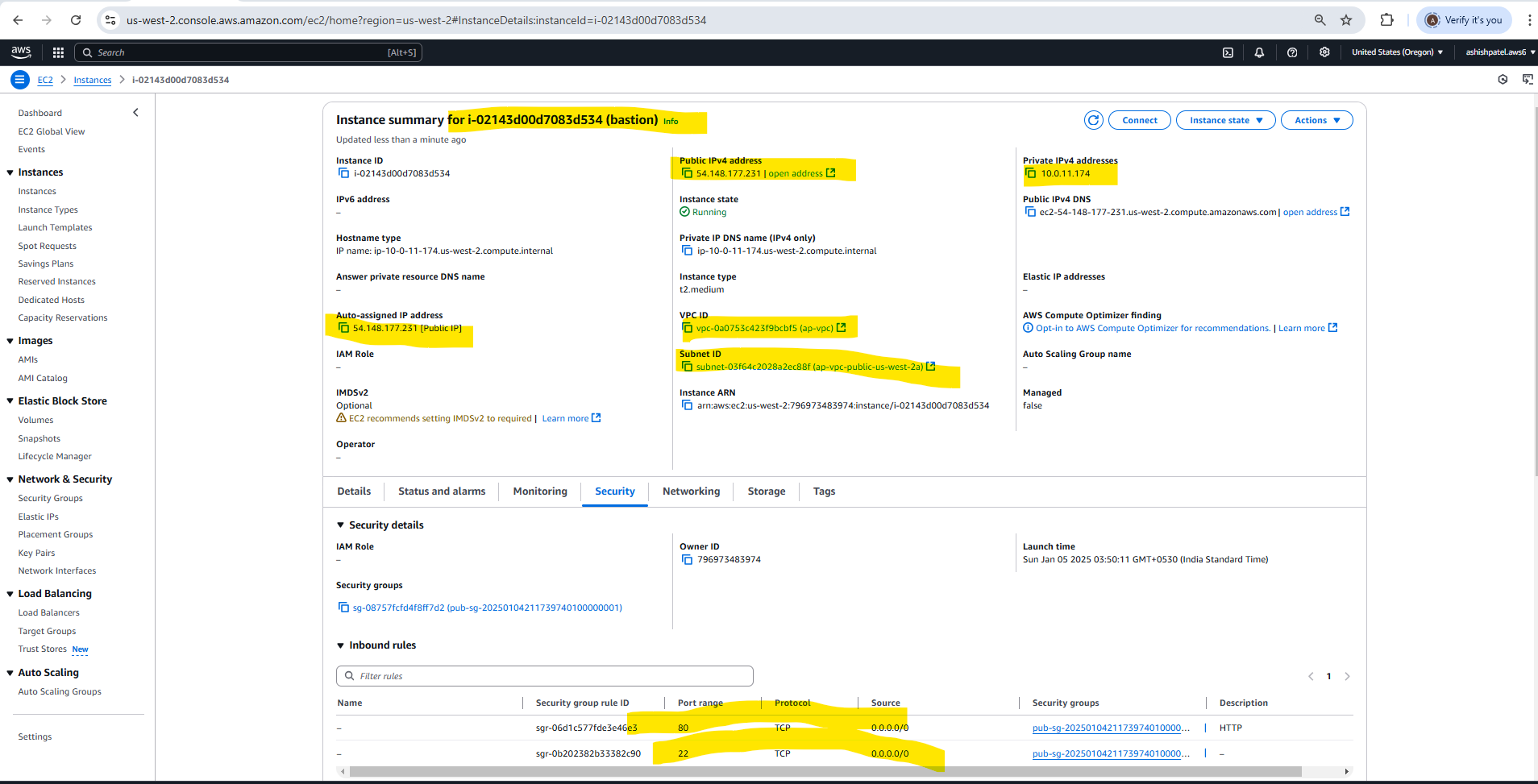


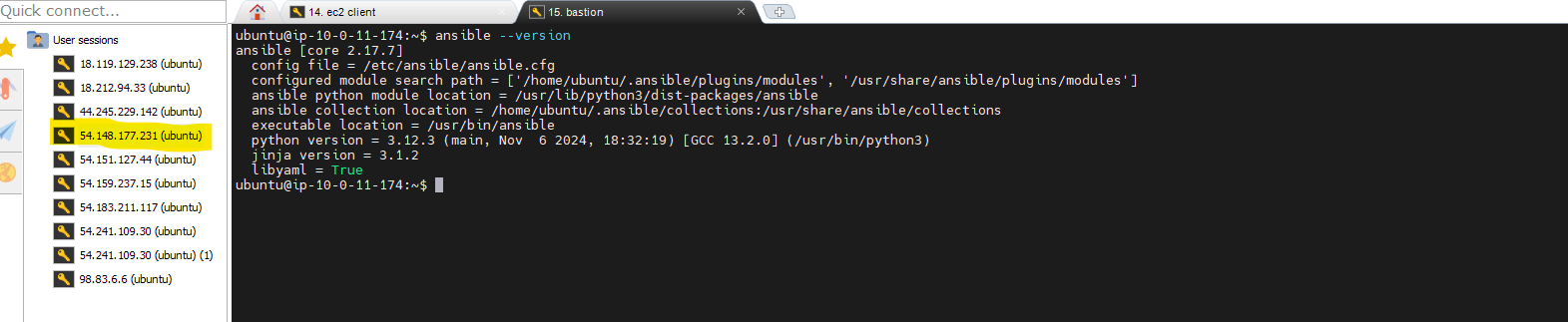
## Task 2 : Setup Config Management for hosts using Ansible & CI pipeline using Jenkins

### Subtask 2.1 Ansible setup

* Install ansible on Bastion host
* Bootstrap ansible codebase for managing Jenkins and app hosts.
* Write an inventory and playbook file.
* Install docker on Jenkins and App hosts using ansible.

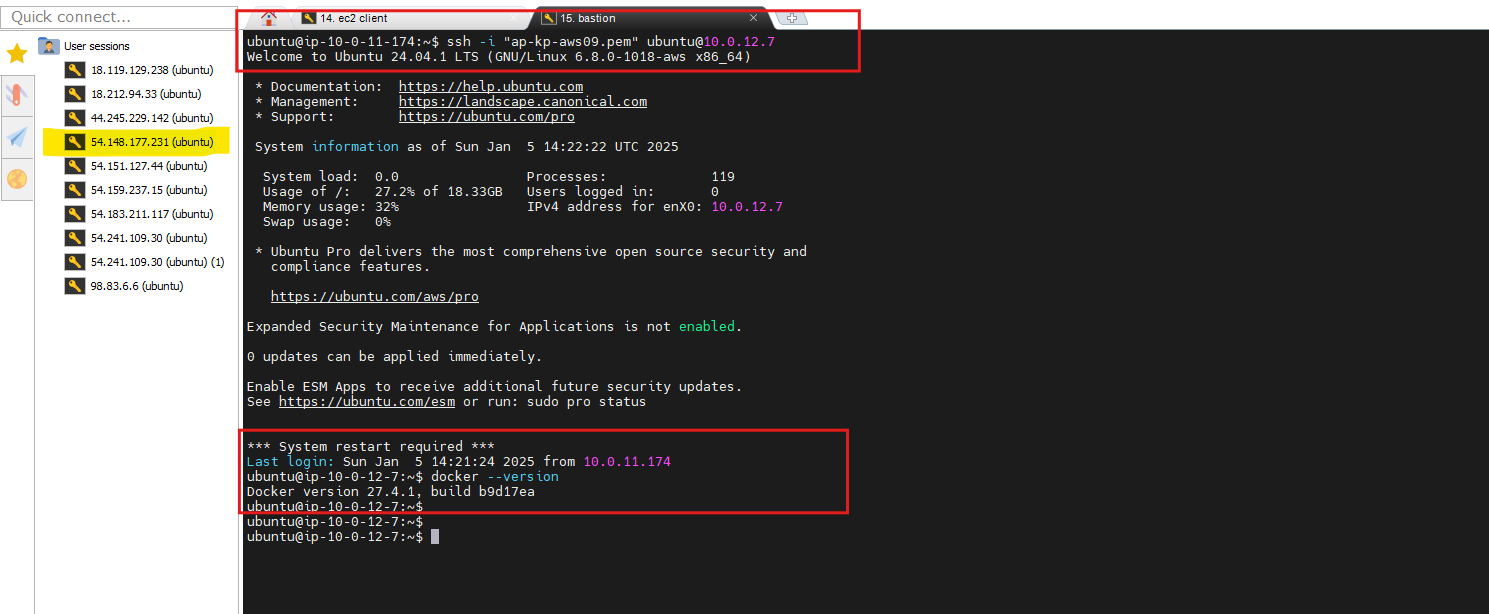
#### Bastion Host



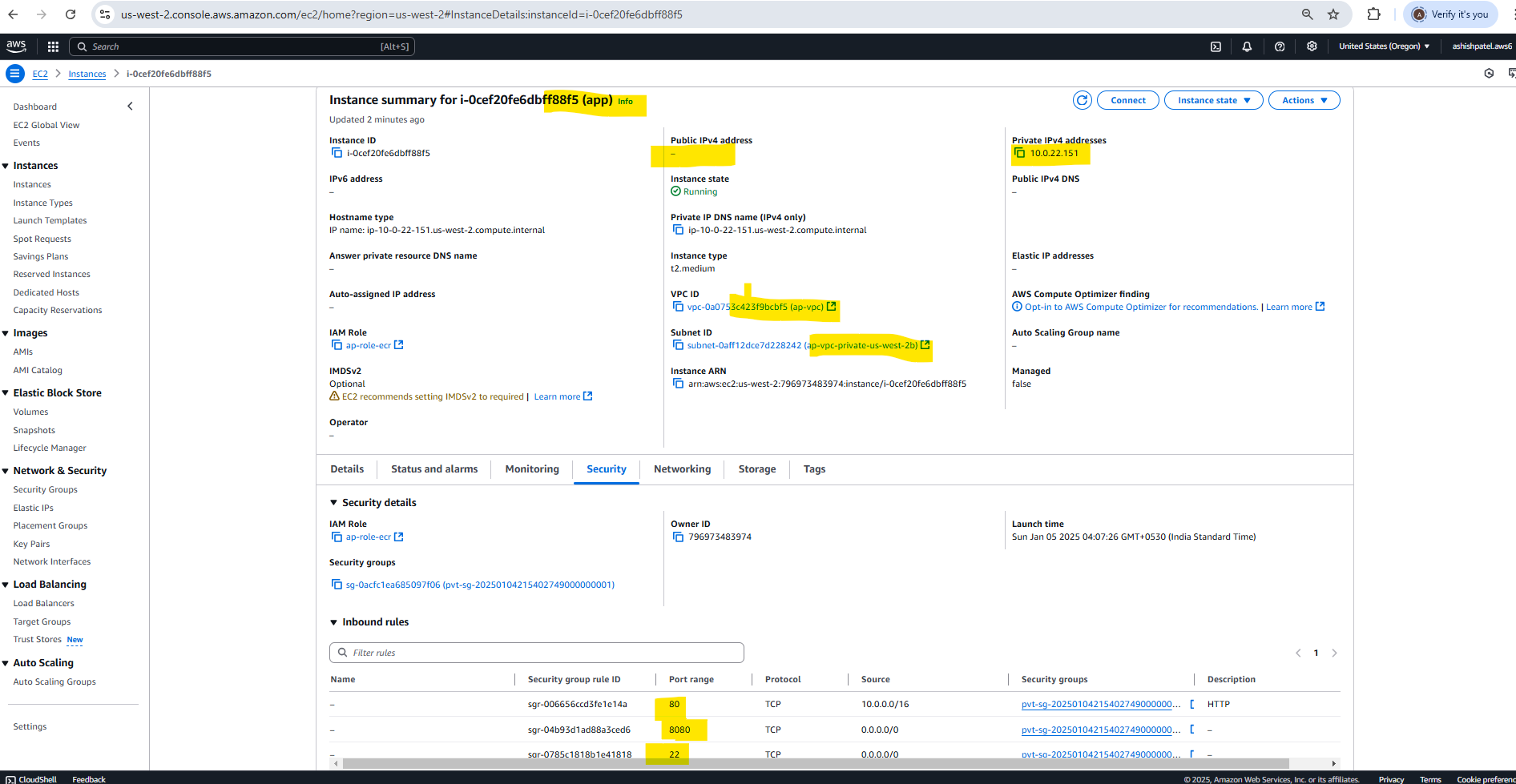


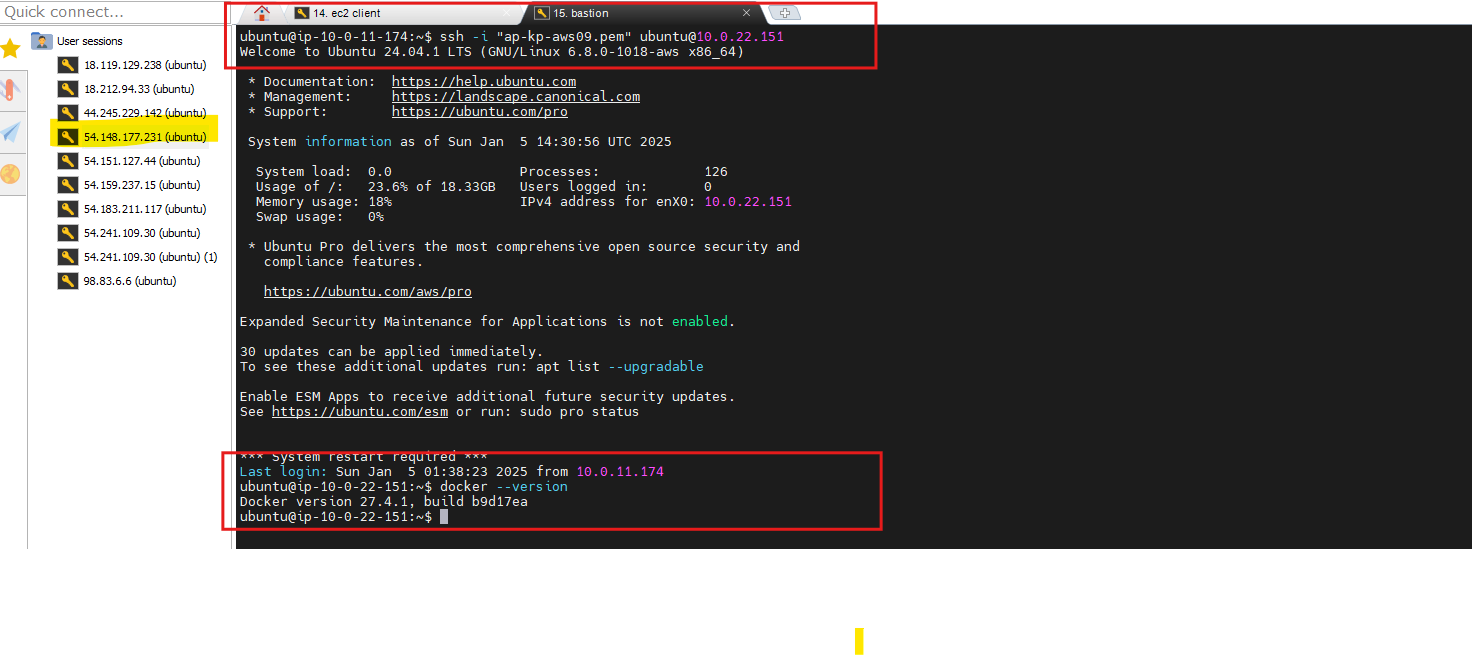
#### Jenkins Host



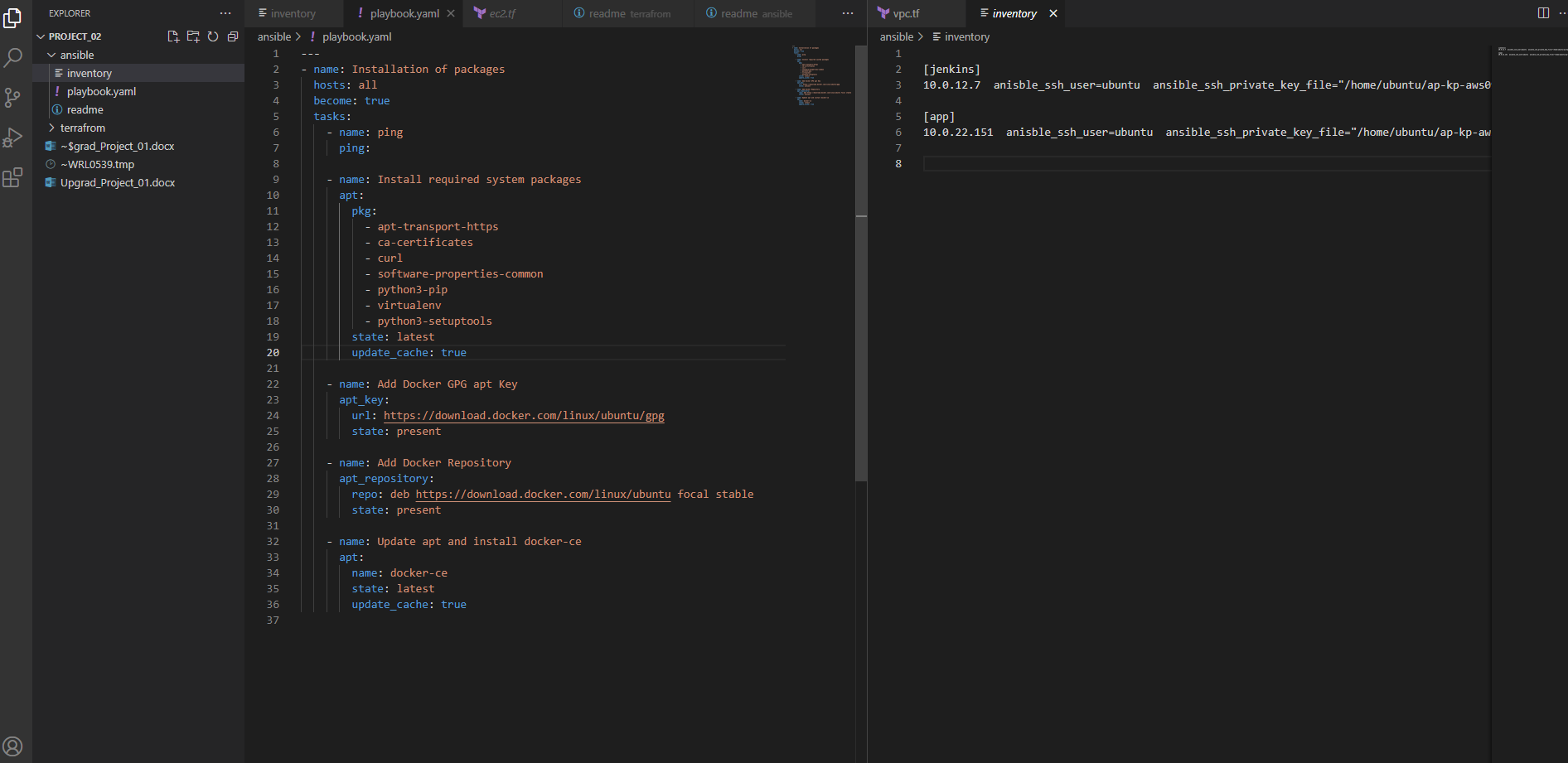


#### App Host



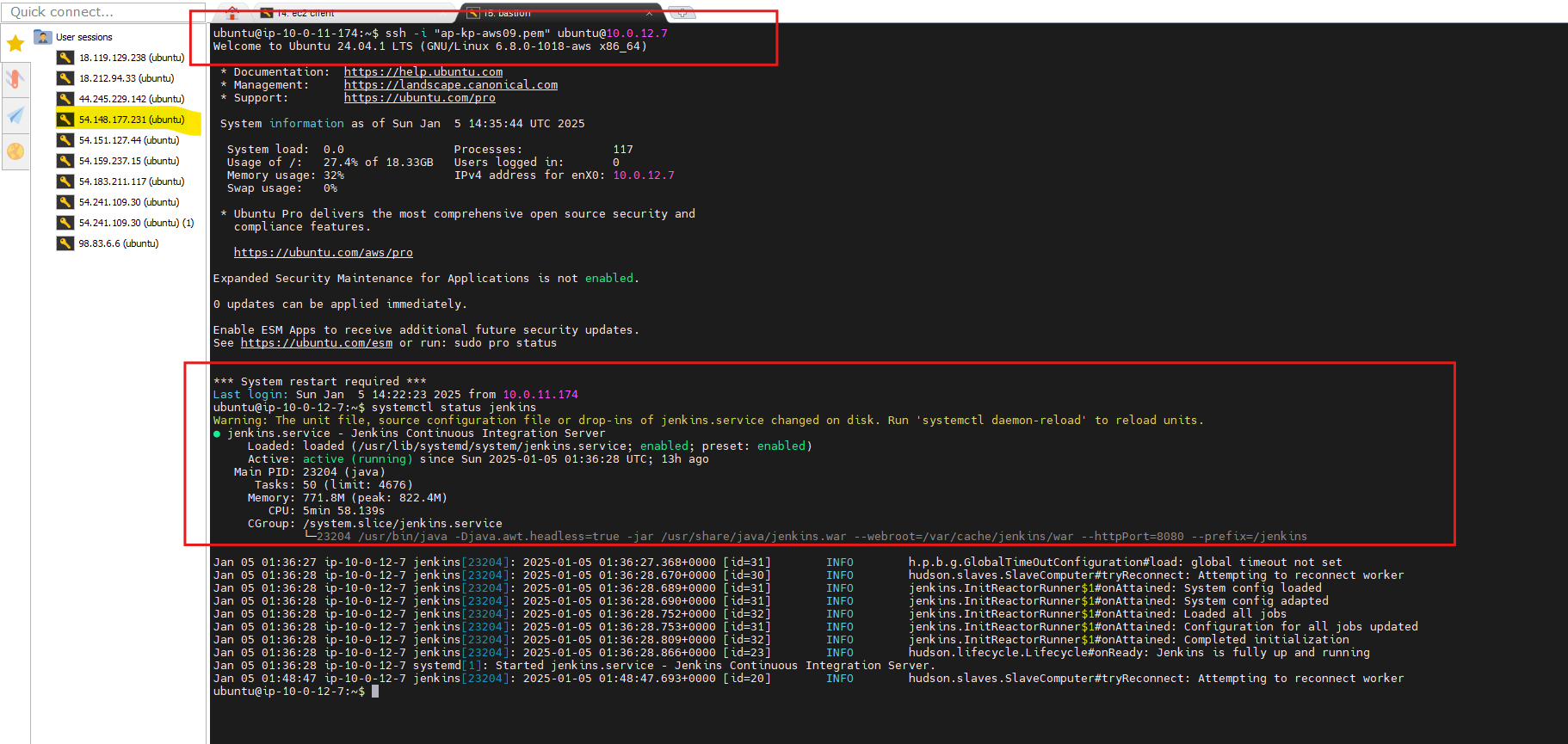


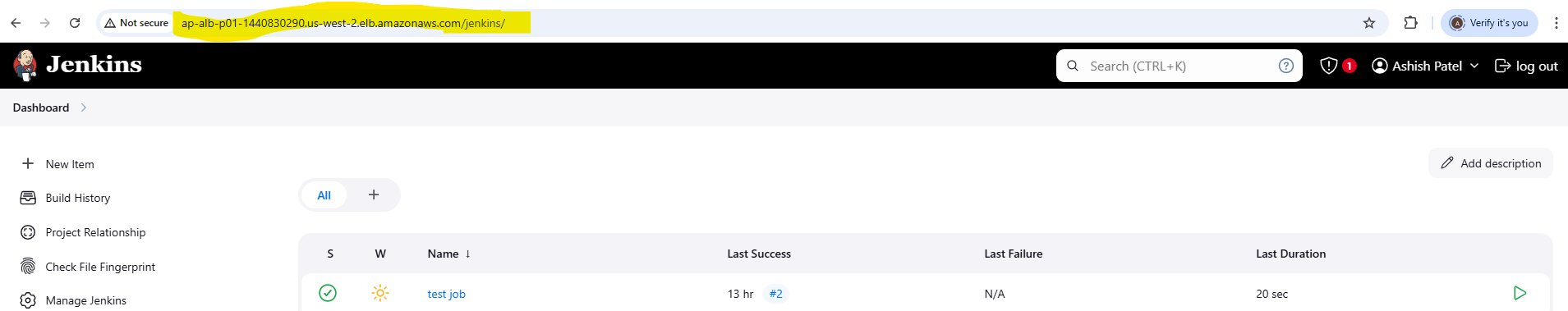
### Ansible Code Repo



### Subtask 2.2: Jenkins setup

* Manually install Jenkins on the Jenkins EC2 instance.
* Access it via ALB endpoint to configure further

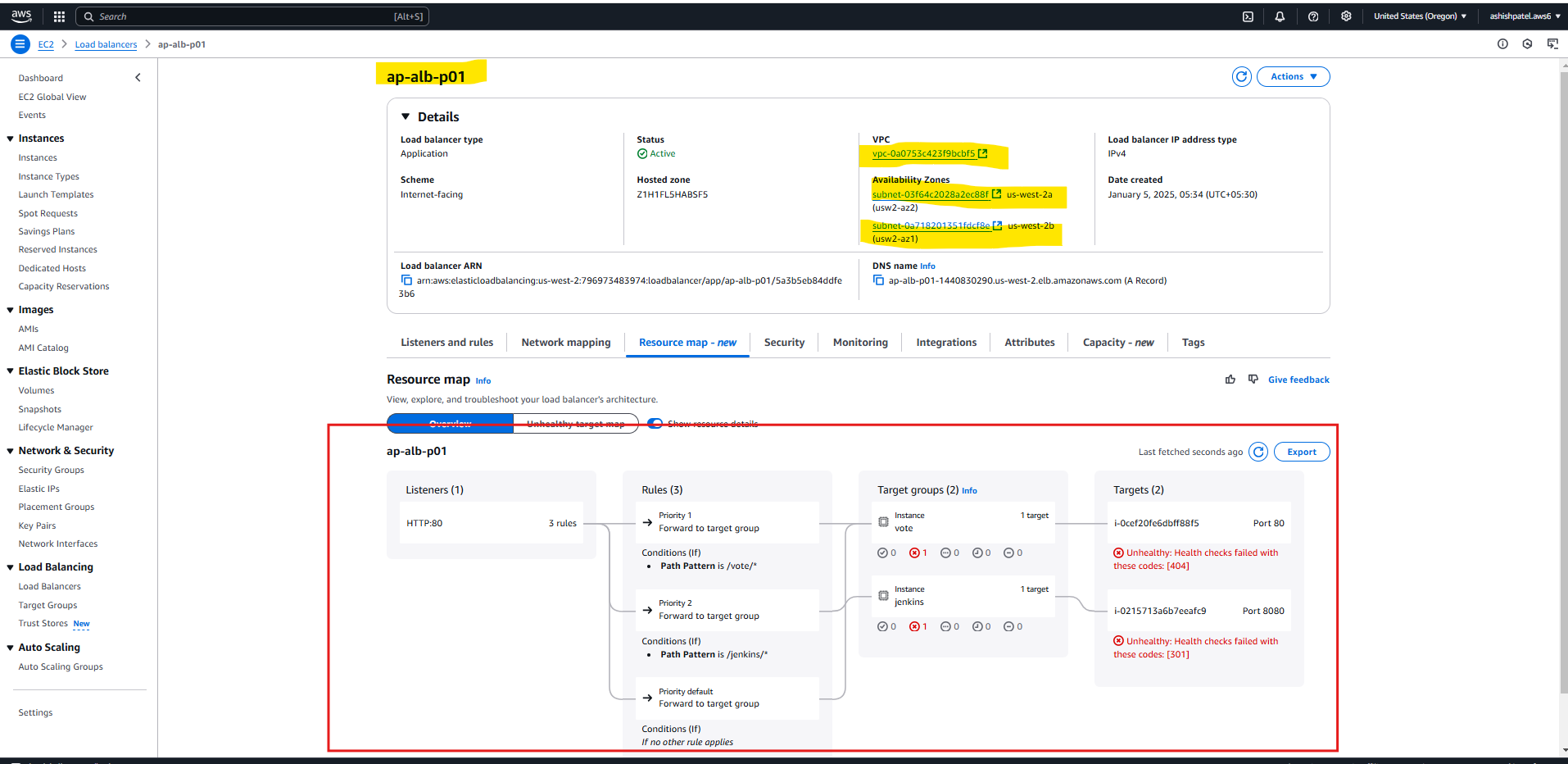




### Subtask 2.3: Create an ALB

* ALB listens on port 80 and should be internet-facing.
* ALB forwards /jenkins, /jenkins/\* to a Target Group having Jenkins host (port 8080) as backend

ALB setup created manually

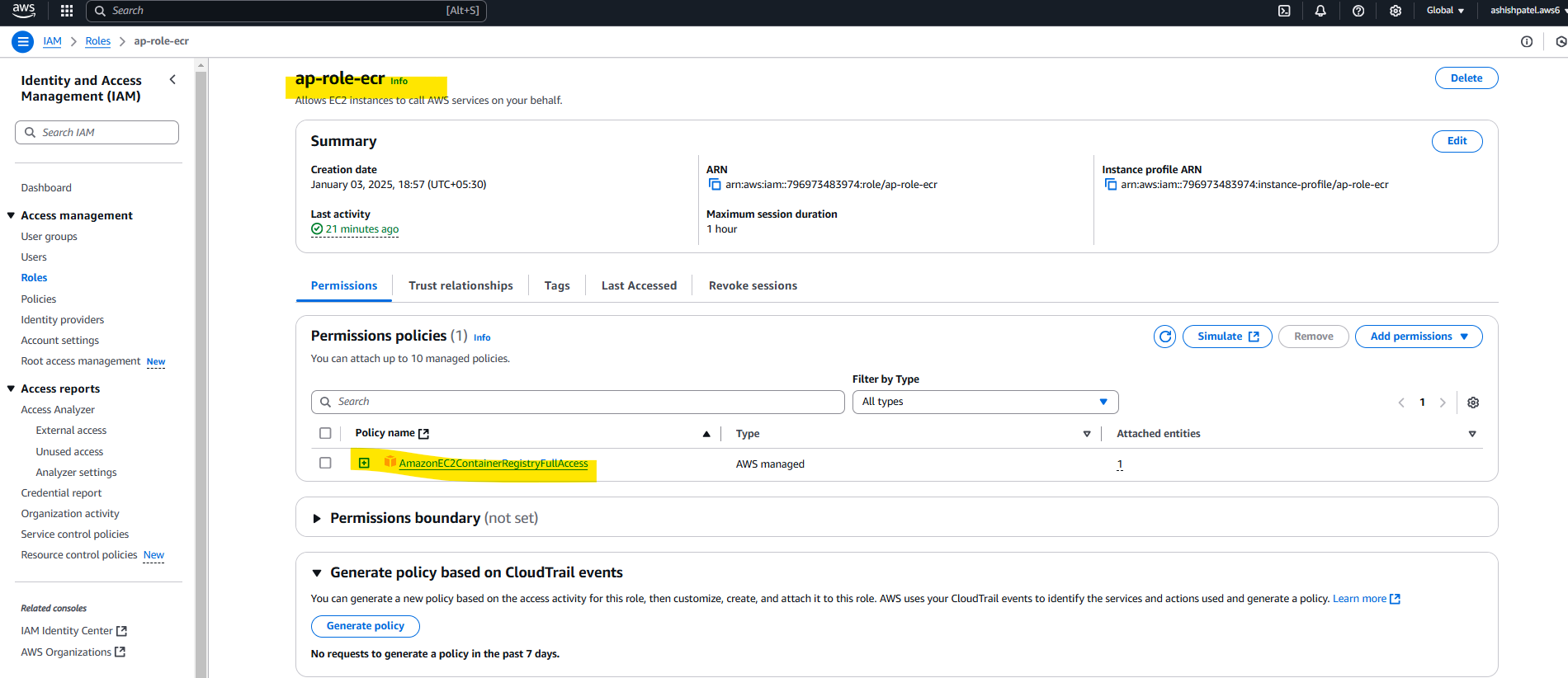


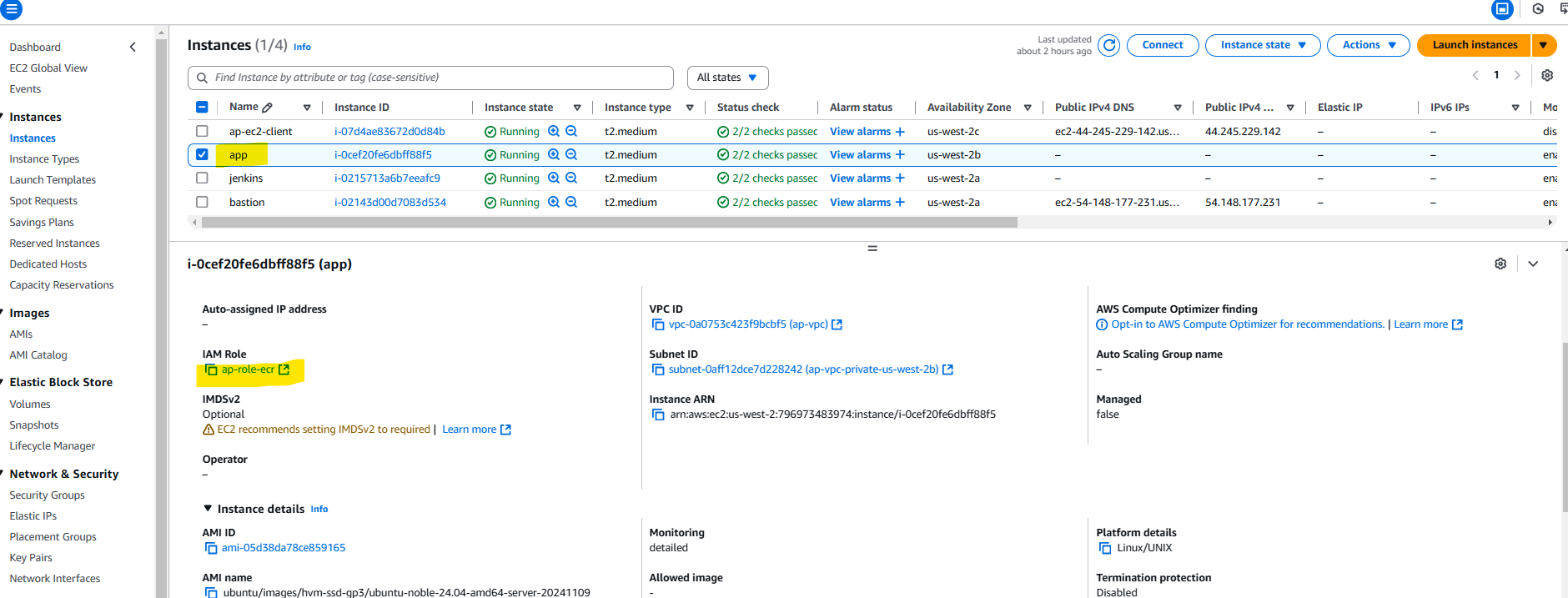
### Subtask 2.4: Create an ECR repository to store the docker images

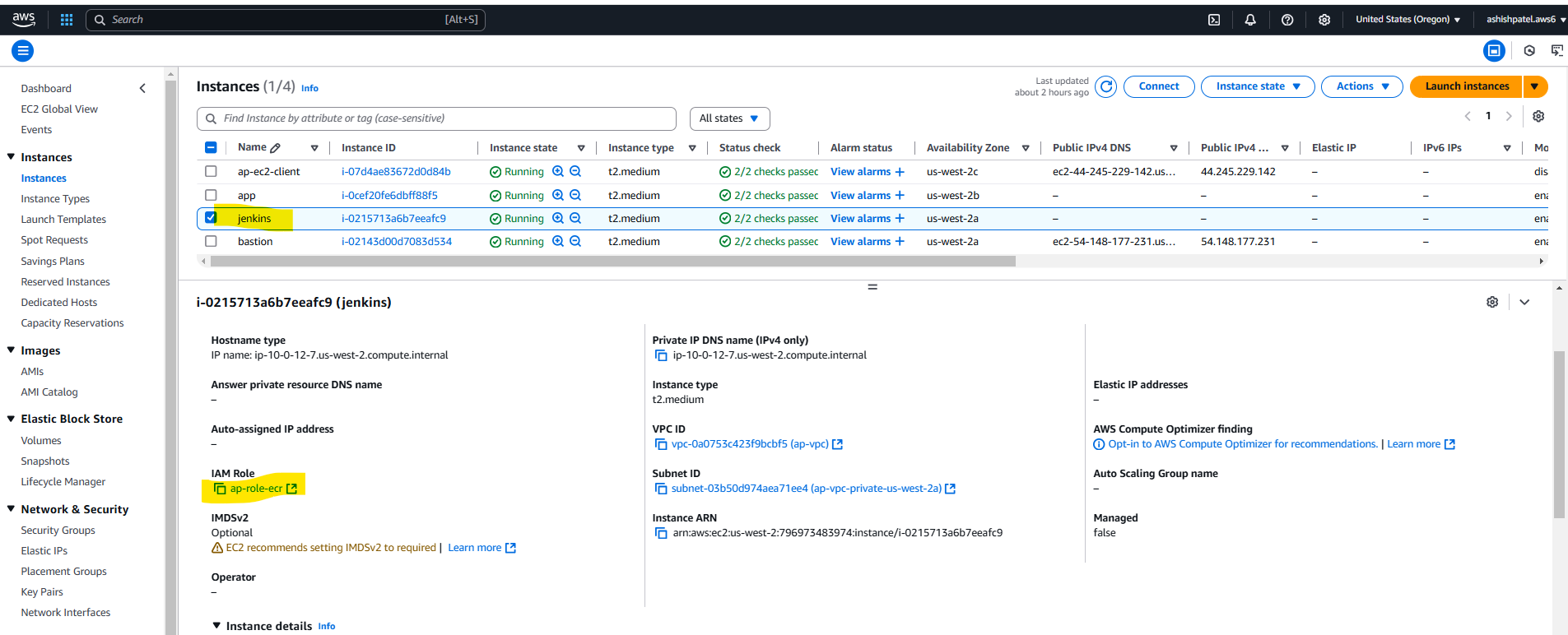
* Create an ECR repository to store the docker image of the node application.
* Create & attach IAM role to provide ECR access to ‘jenkins’ and ‘app’ hosts.
* Ensure that ‘jenkins’ and ‘app’ hosts are authenticated to use the ECR repository.
* Ensure a sample Jenkins job is able to ssh to the app host and run simple commands. This is needed so that Jenkins can ssh to the ‘app’ instance, to pull docker images and spin up containers.

#### Create an ECR repository to store the docker image of the node application

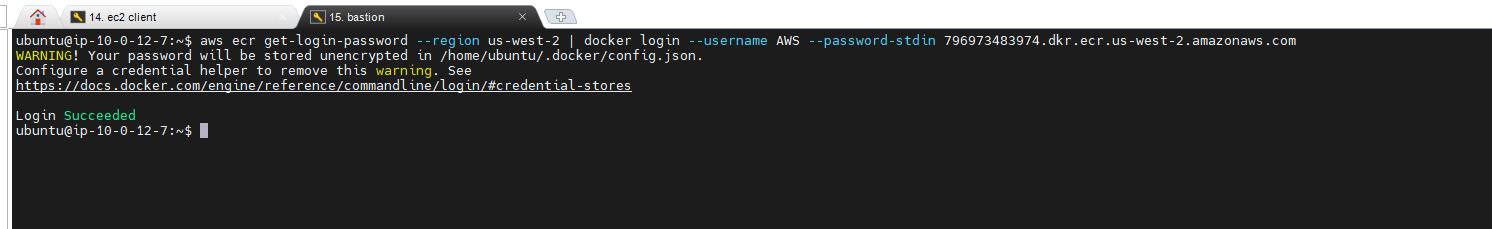
#### Create & attach IAM role to provide ECR access to ‘jenkins’ and ‘app’ hosts.

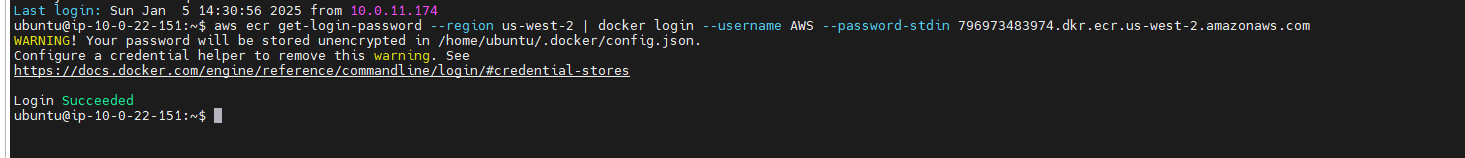






#### Ensure that ‘jenkins’ and ‘app’ hosts are authenticated to use the ECR repository.



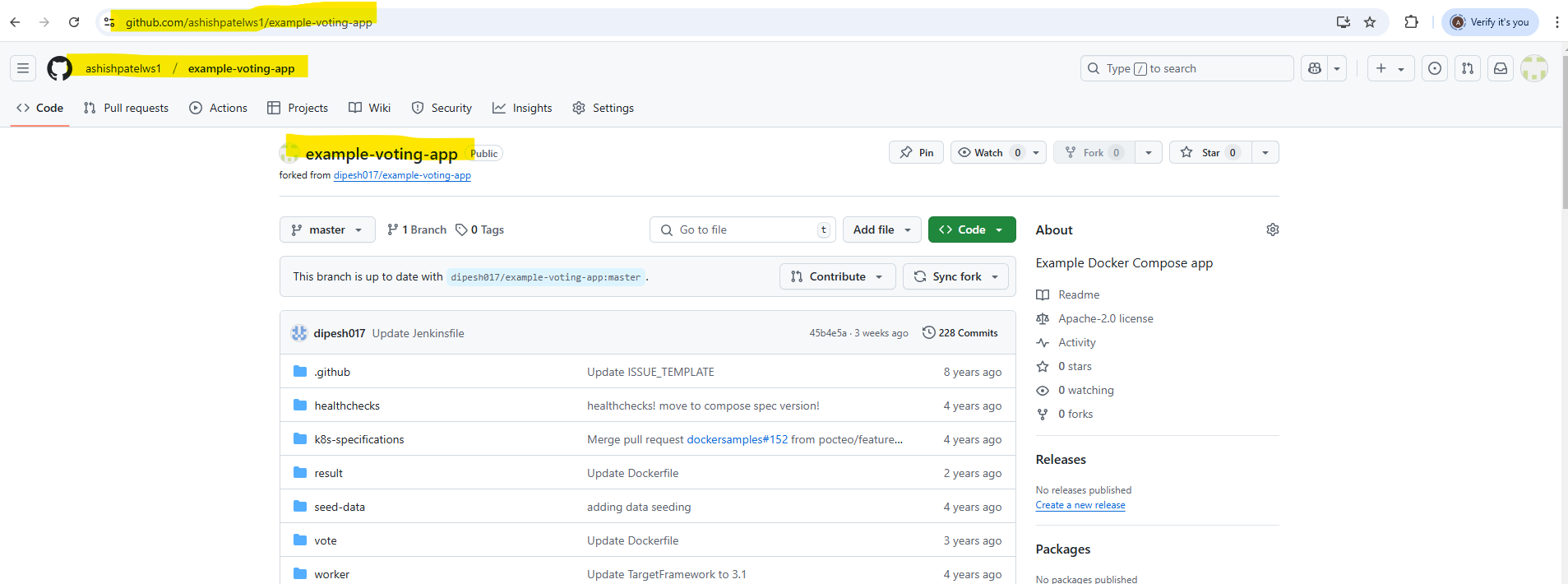


#### <TBD> Ensure a sample Jenkins job is able to ssh to the app host and run simple commands. -

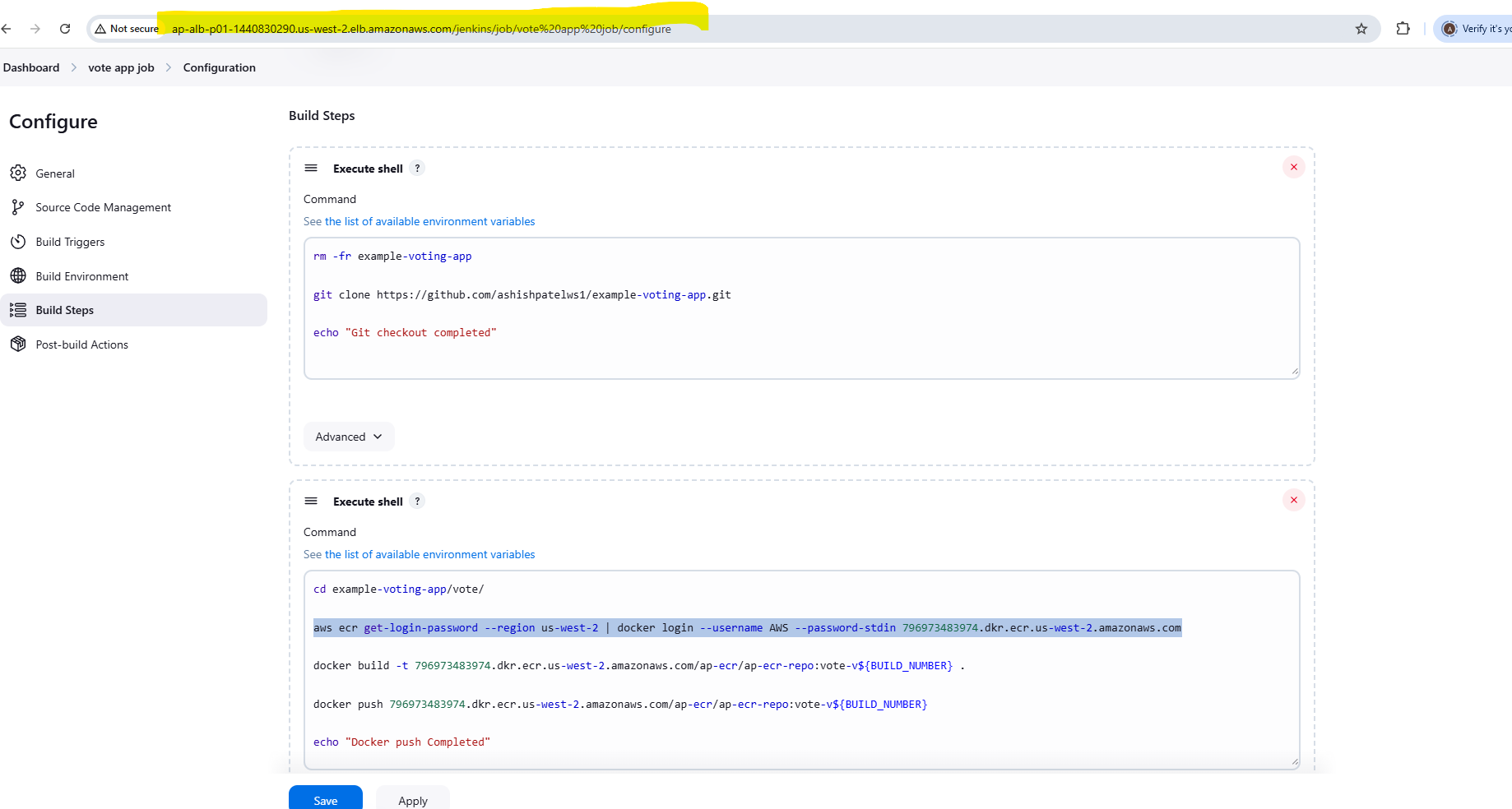
## Task 3 : Dockerize sample Node.js app & write Jenkinsfile

### Subtask 3.1 Prepare repository in GitHub

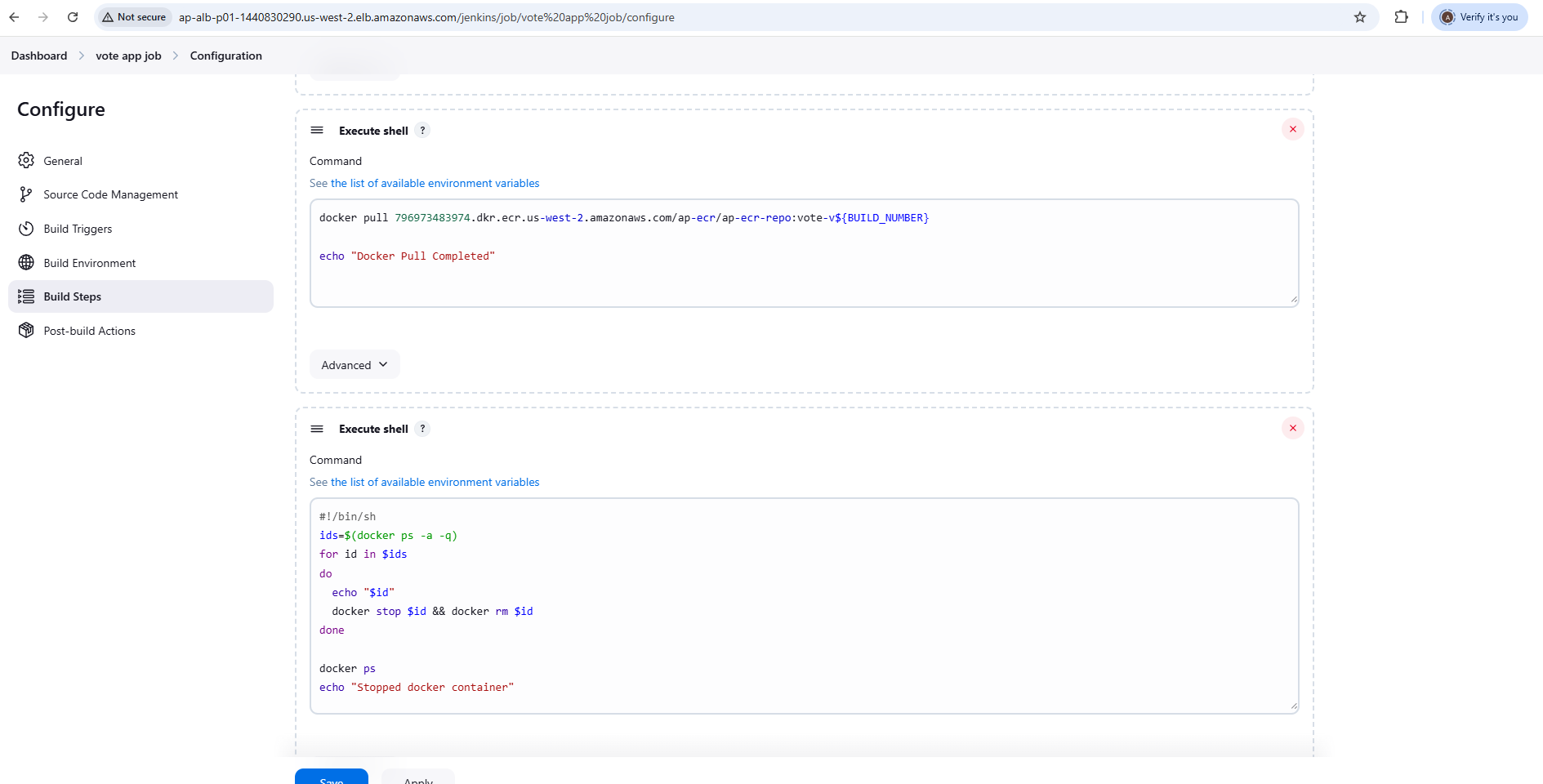
* Create a GitHub repository.
* Write Dockerfile to dockerise the node application. This file should be able to install dependencies for the application.
* Test docker build and run on your local system
* Push the docker file of the node application in the same GitHub repository.

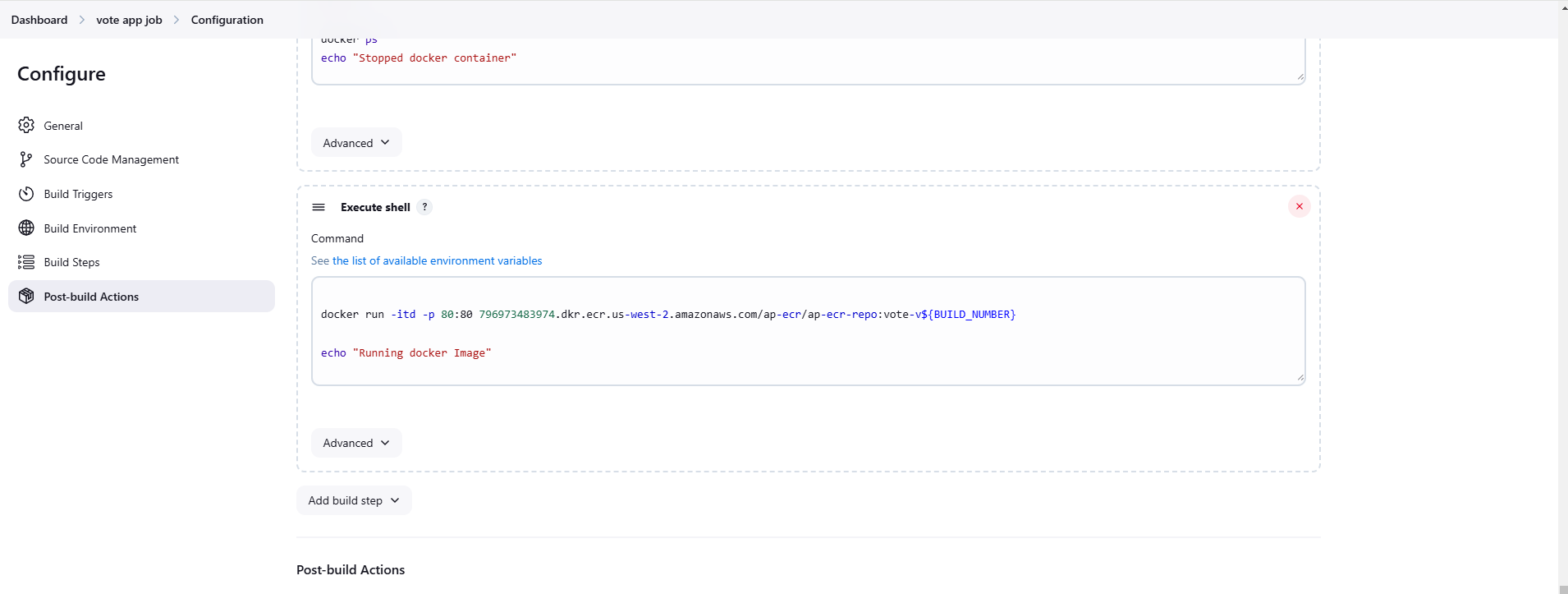


### Subtask 3.2 Setup Jenkins CI Job

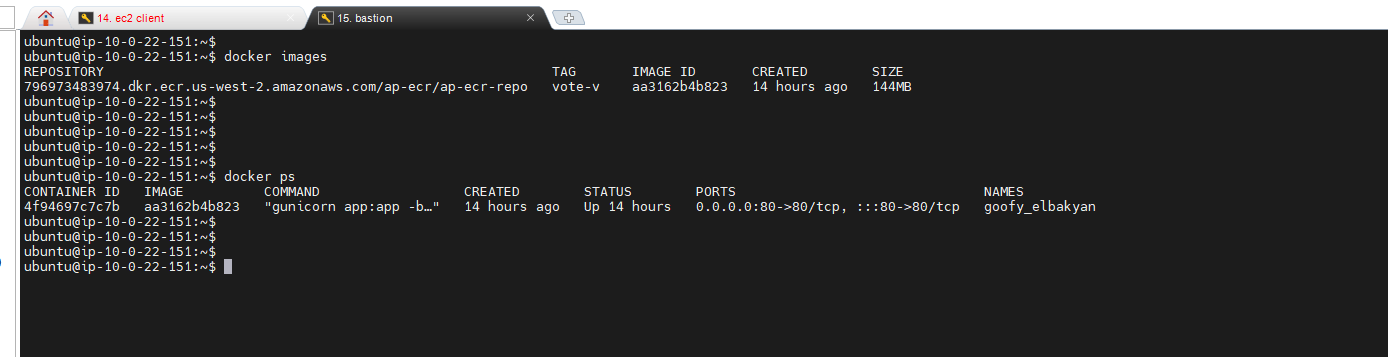


### Subtask 3.3 Setup Jenkins CD Job





### Docker Images and running status



### Accessing App - Vote

