**AWS EC2 /ALB/ASG : POC**

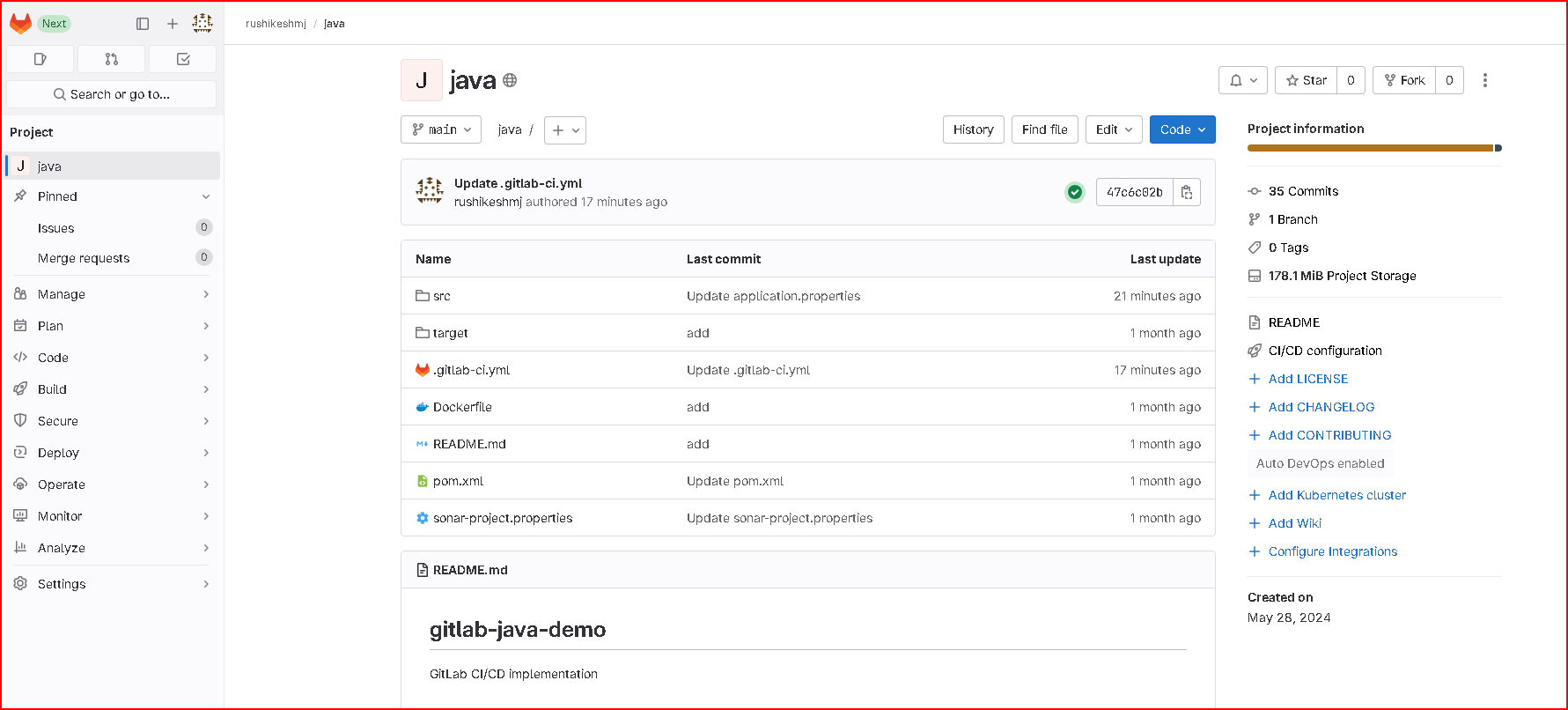
**Task 1 : Perform CI/CD Steps & Push Image to AWS ECR**

Setup CI/CD Pipeline:

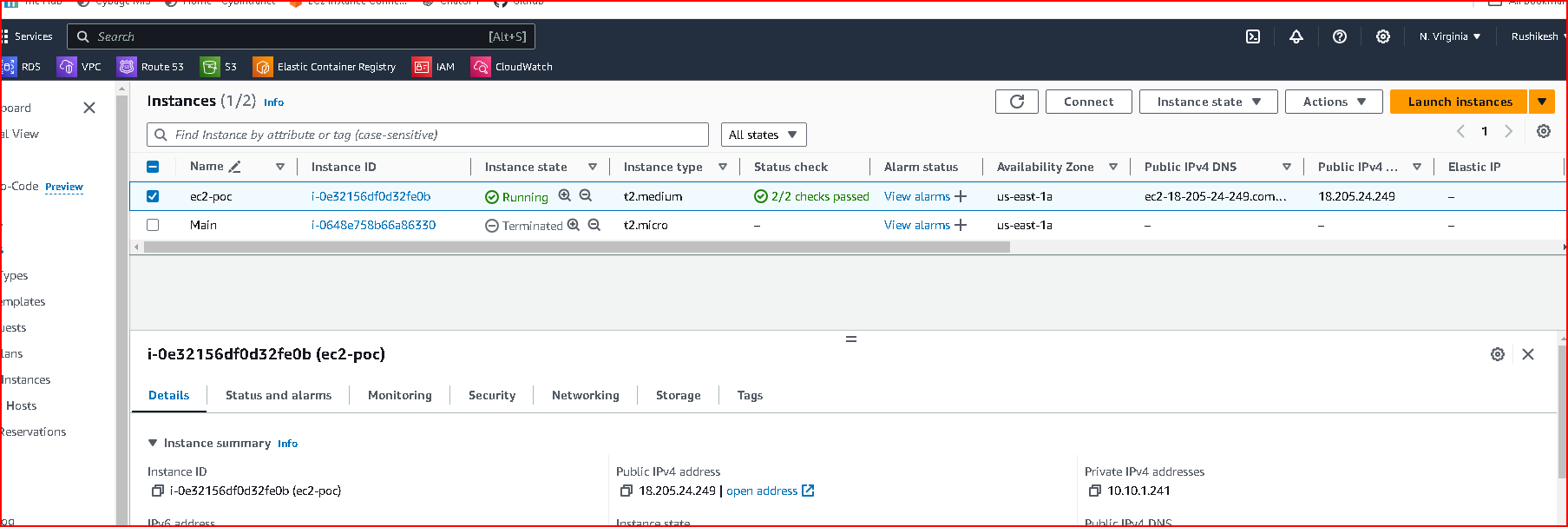
Choose your CI/CD tool - GitLab

Configure your pipeline to build your application, run tests, and build a Docker image.

Repo : <https://gitlab.com/rushikeshmj/java.git>



Launch EC2 instance



Run few command on EC2 and configure the runner

sudo apt-get update

sudo apt-get install docker.io

# Download the binary for your system

sudo curl -L --output /usr/local/bin/gitlab-runner https://gitlab-runner-downloads.s3.amazonaws.com/latest/binaries/gitlab-runner-linux-amd64

# Give it permission to execute

sudo chmod +x /usr/local/bin/gitlab-runner

# Create a GitLab Runner user

sudo useradd --comment 'GitLab Runner' --create-home gitlab-runner --shell /bin/bash

# Install and run as a service

sudo gitlab-runner install --user=gitlab-runner --working-directory=/home/gitlab-runner

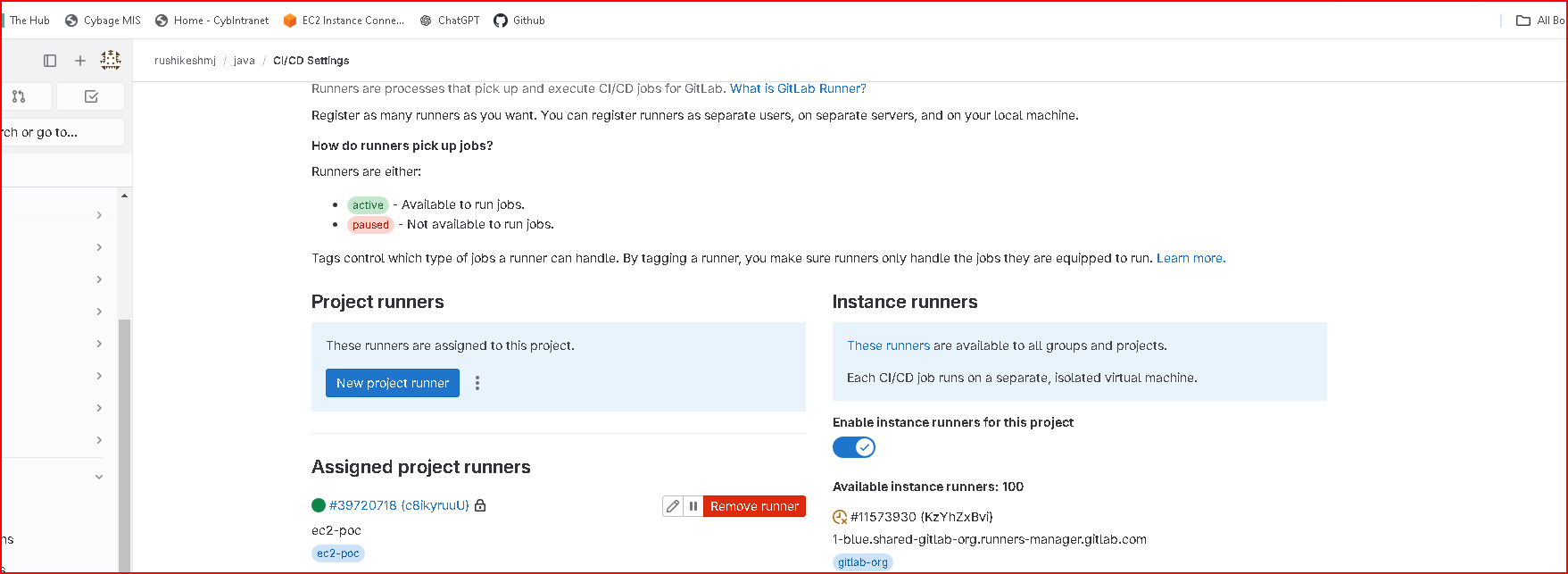
sudo gitlab-runner start

sudo gitlab-runner register --url https://gitlab.com/ --registration-token GR1348941WyG1vEqykTtxMytRzSdi

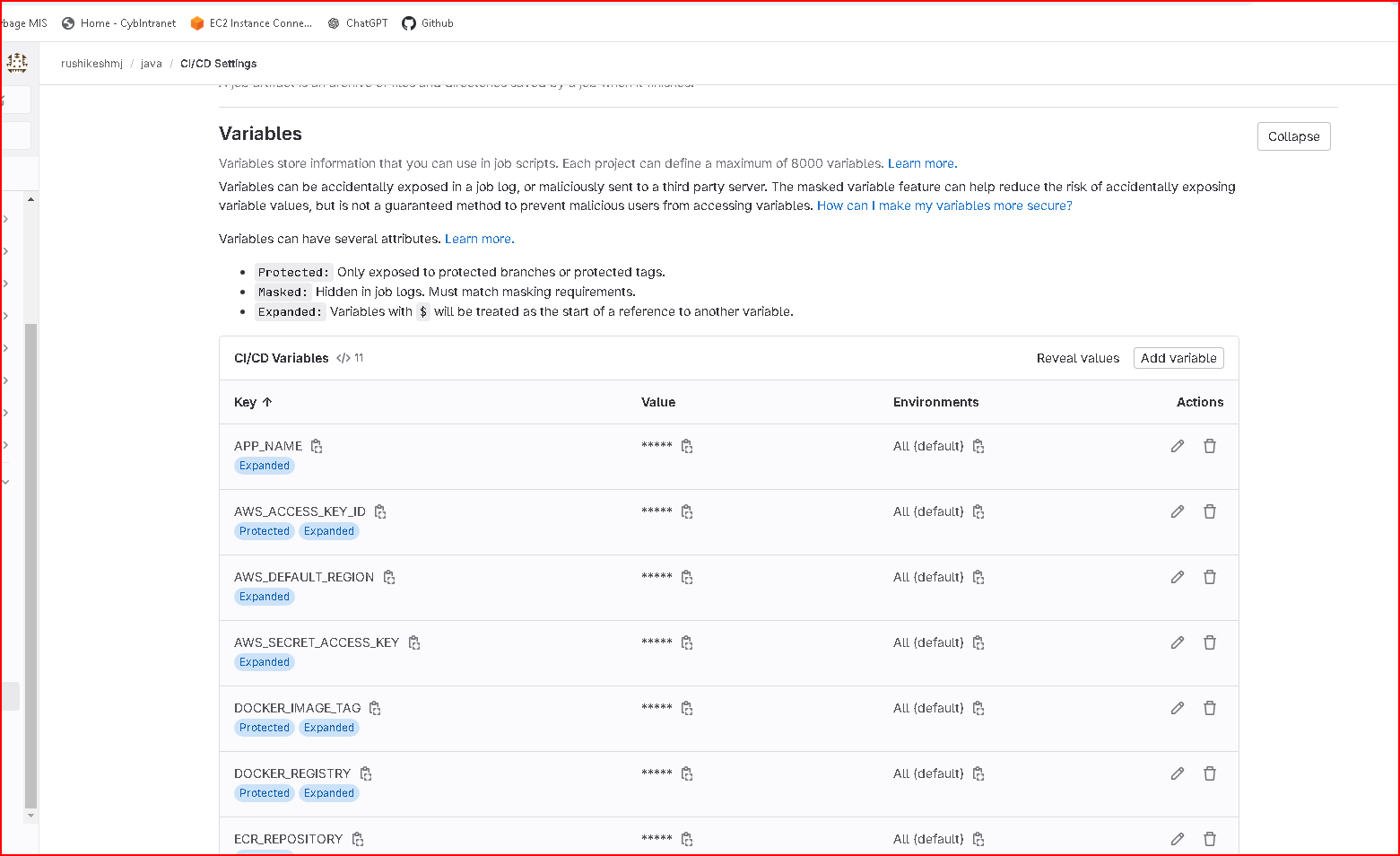
vi /etc/gitlab-runner/config.toml

sudo gitlab-runner restart

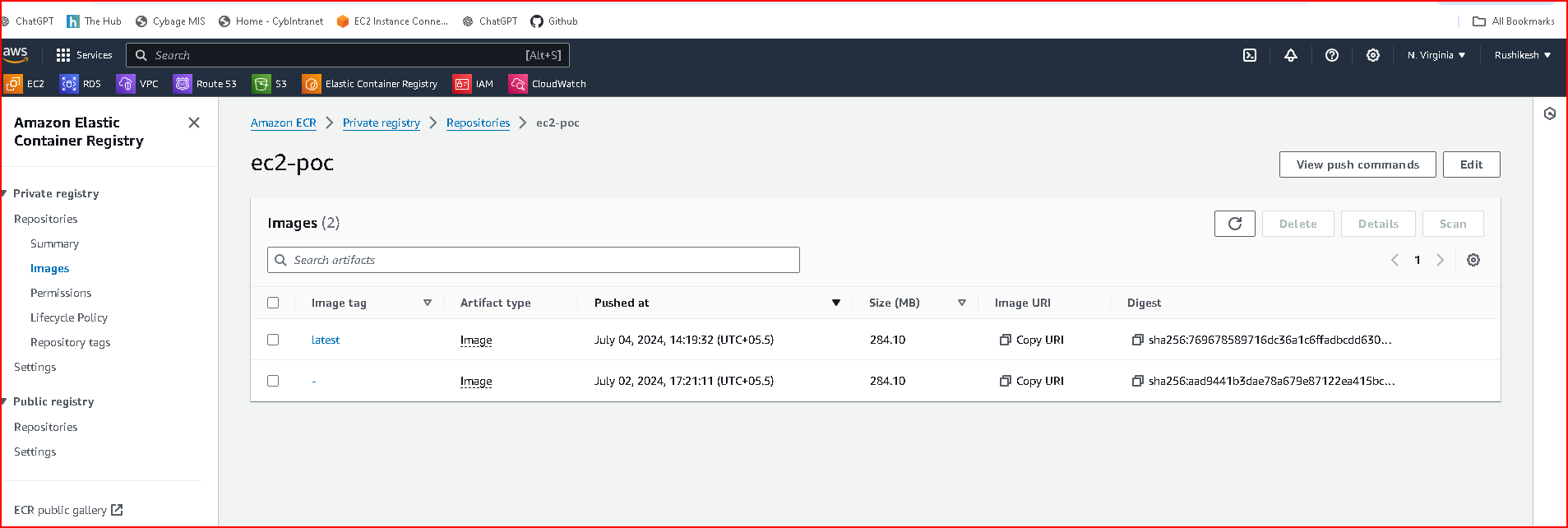
sudo gitlab-runner status



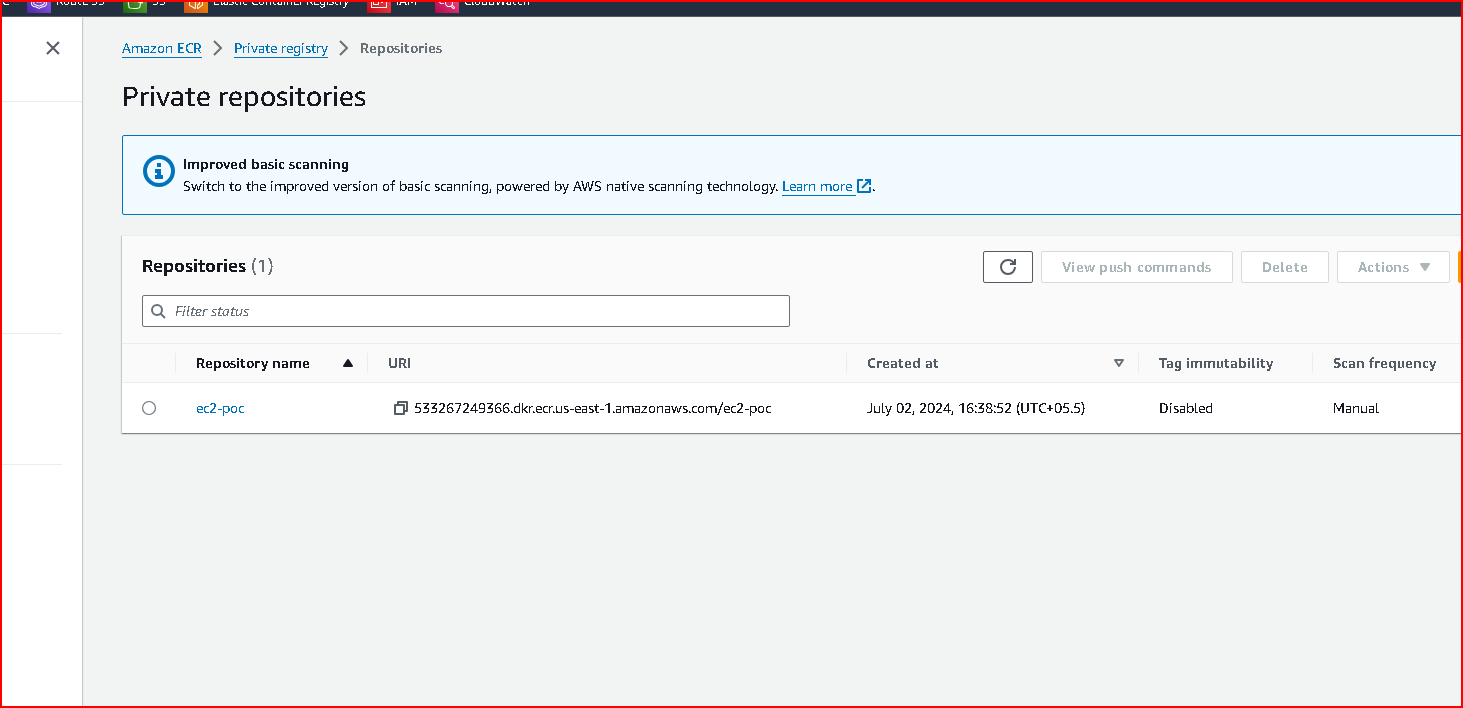
Save the ENV variables in CI/CD – Variable section



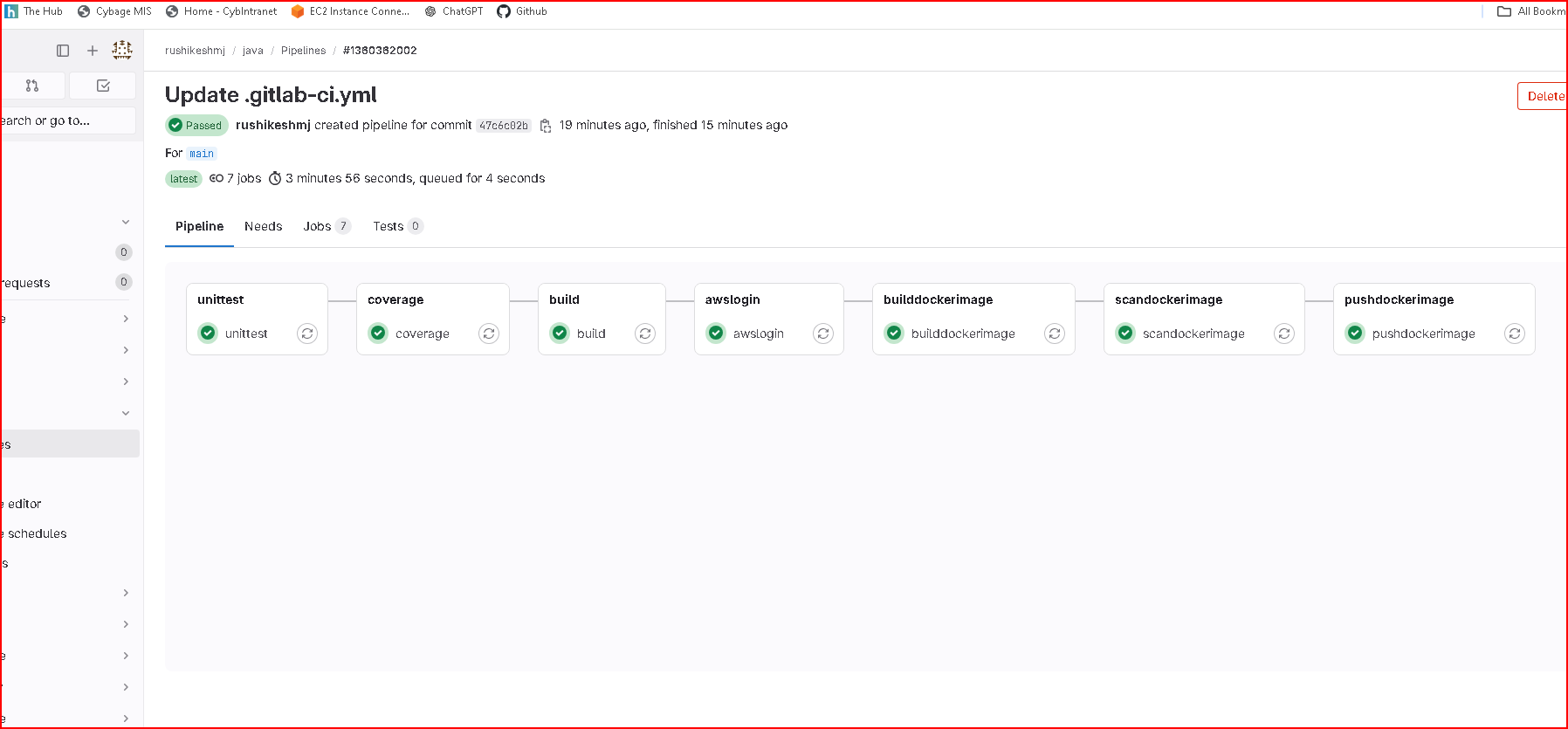
Push Image to AWS ECR:



Create an AWS ECR repository.



Configure your CI/CD pipeline to push the Docker image to ECR after a successful build.



TASK 2 : Perform EC2 Instance Creation Steps

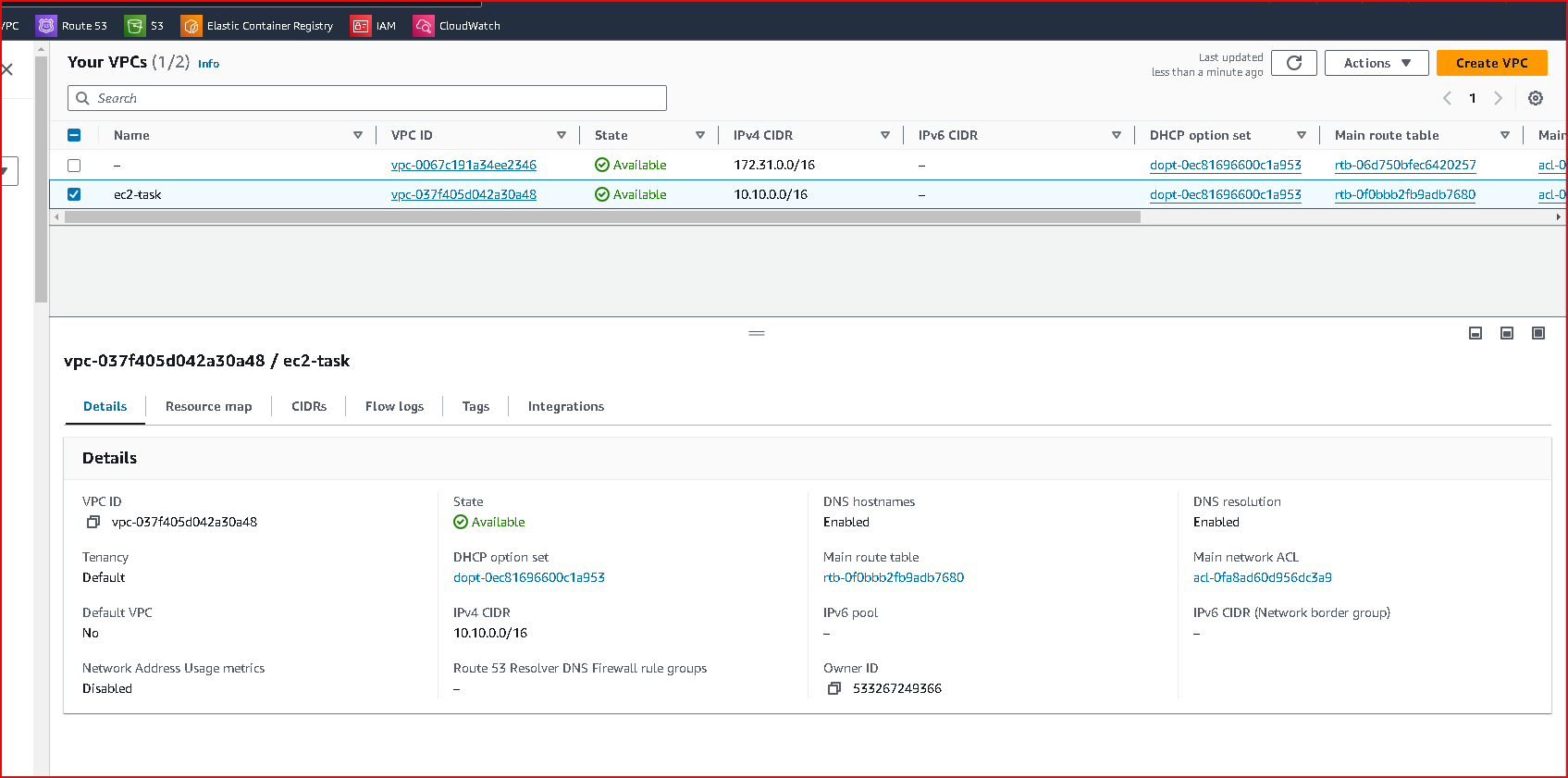
Create VPC and Subnets

VPC Creation:

Navigate to AWS Management Console > VPC > Create VPC.

Define the CIDR block

Create the VPC.

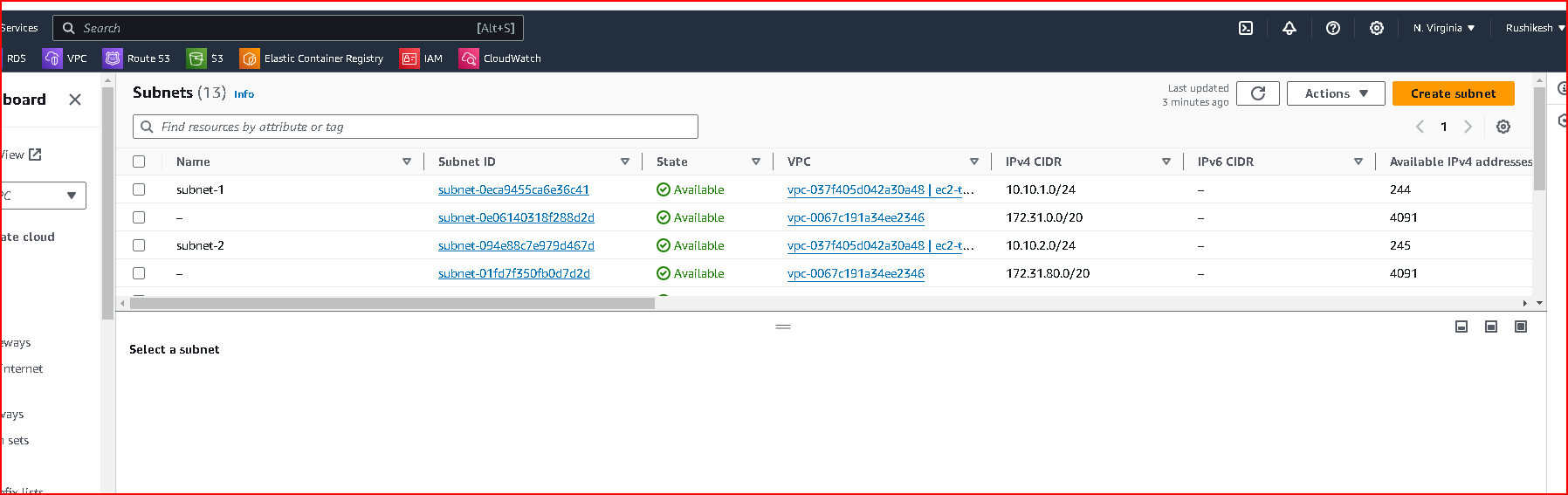


Subnets Creation:

Inside the VPC dashboard, navigate to Subnets > Create Subnet.

Define subnet CIDR blocks

Create at least two subnets in different AZs for high availability.



Security Groups:

Create security groups to control inbound and outbound traffic to your EC2 instances.

Launch EC2 Instances:

Navigate to EC2 Dashboard > Instances > Launch Instance.

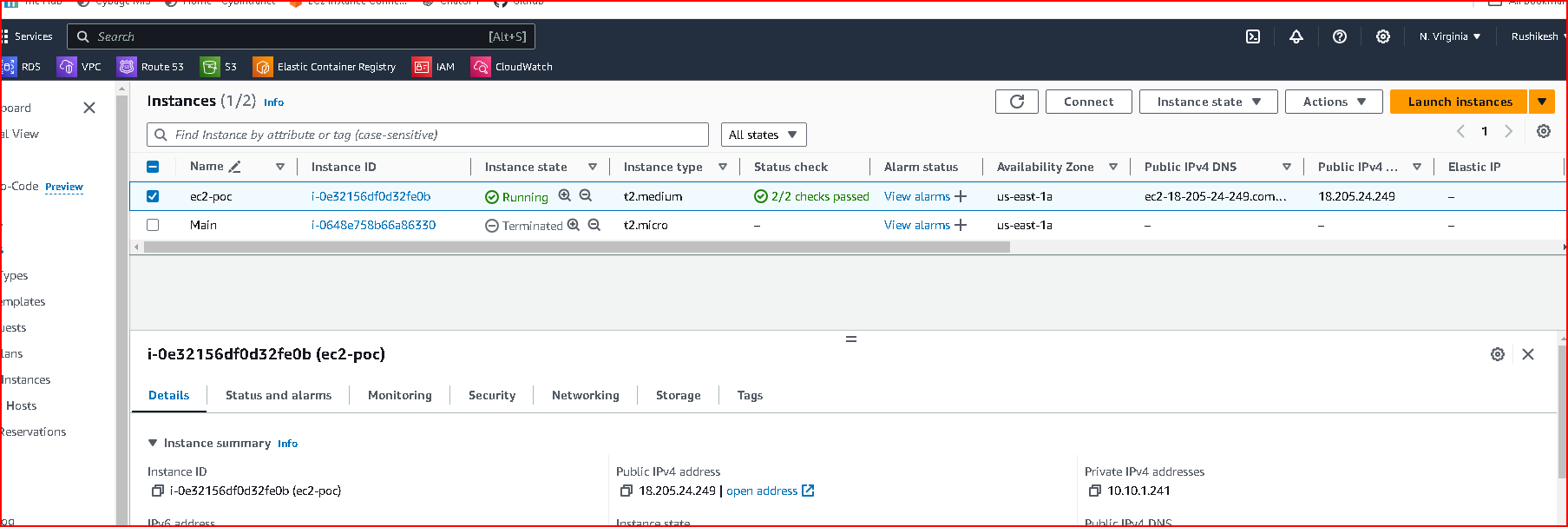
Choose an Amazon Machine Image (AMI).

Select instance type, configure instance details (subnet etc.).

Configure storage and tags as necessary.

Configure security group settings.

Review and launch the instance.



TASK 3 : Creating Load Balancer and Deploying Application

Create Application Load Balancer (ALB):

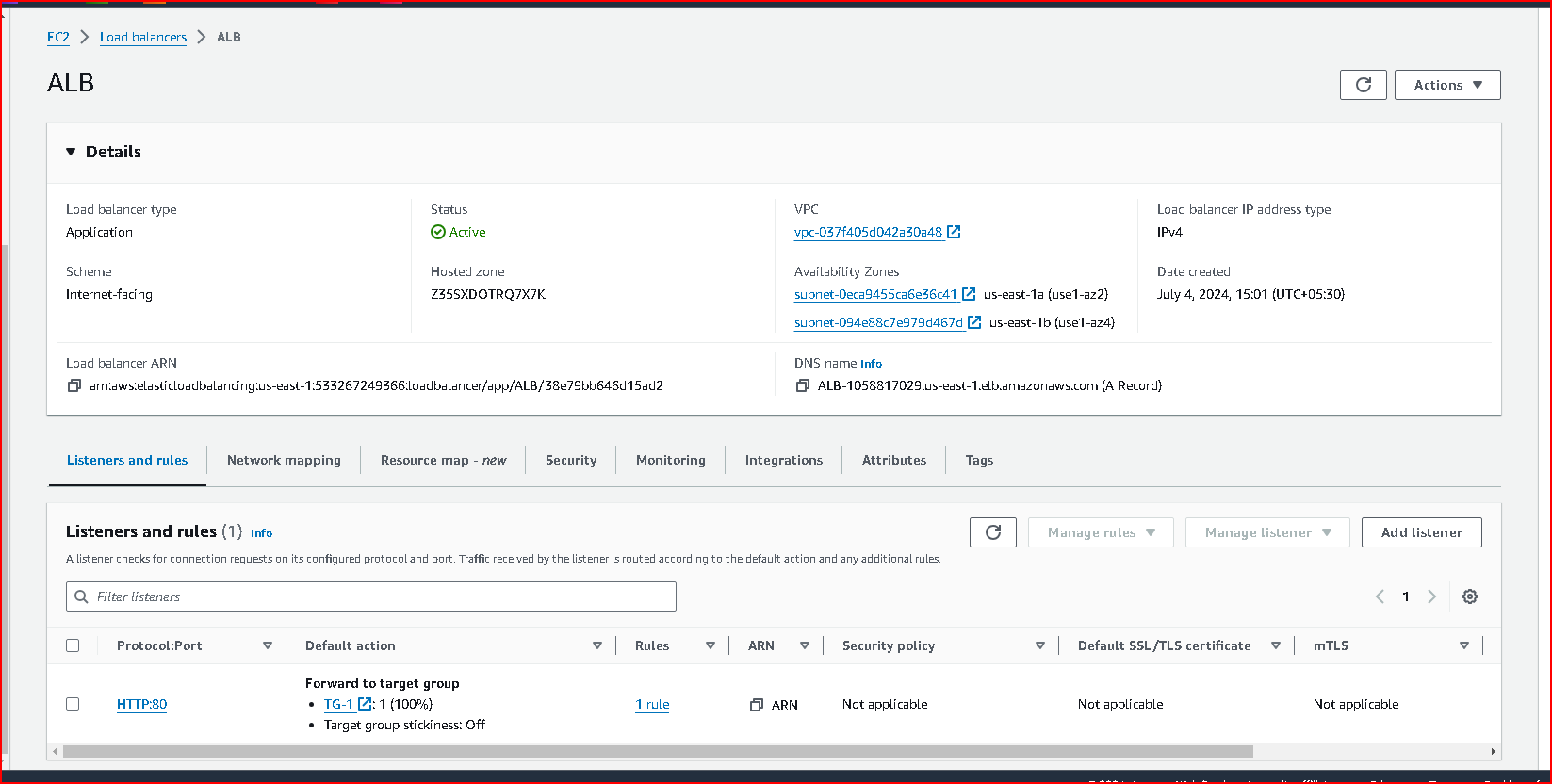
Navigate to EC2 Dashboard > Load Balancers > Create Load Balancer.

Choose Application Load Balancer.

Configure listeners (HTTP/HTTPS), configure security settings.

Select VPC and subnets.

Configure routing and health checks.

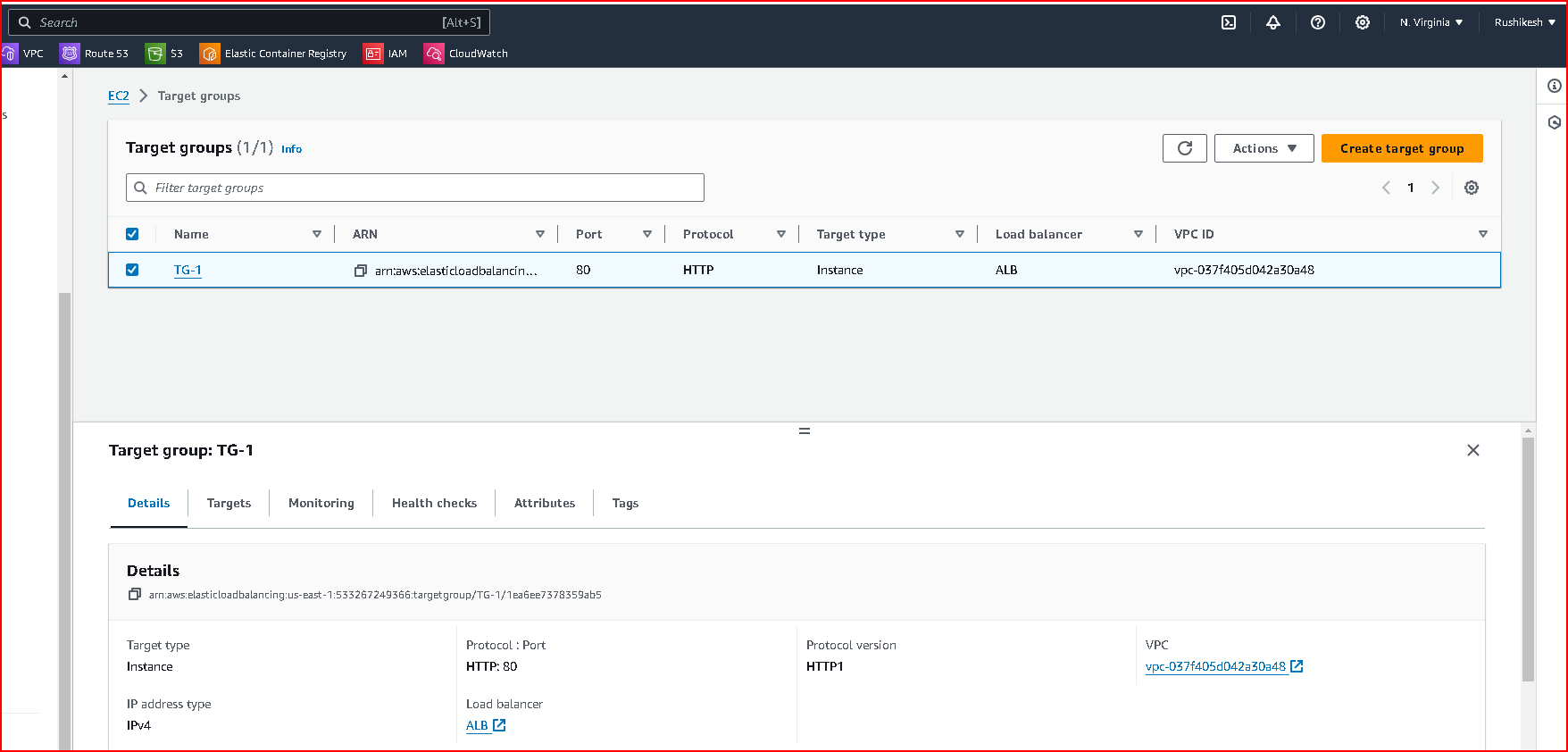


Set Up Target Groups:

Define target groups for your EC2 instances.

Specify instance protocol and port.

Register instances from the EC2 dashboard or via Auto Scaling.



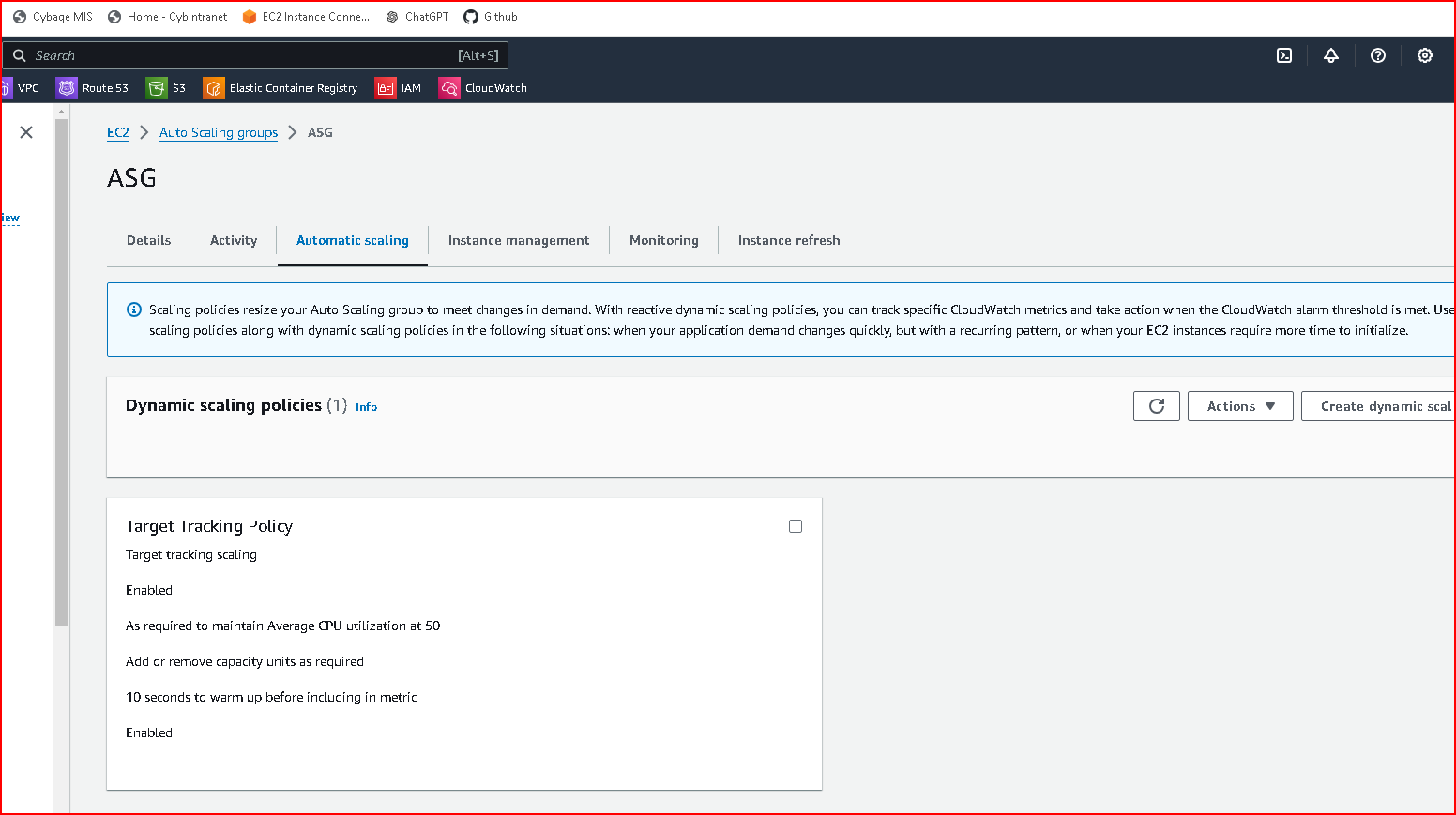
Auto Scaling Group

Navigate to EC2 Dashboard > Auto Scaling > Auto Scaling Groups.

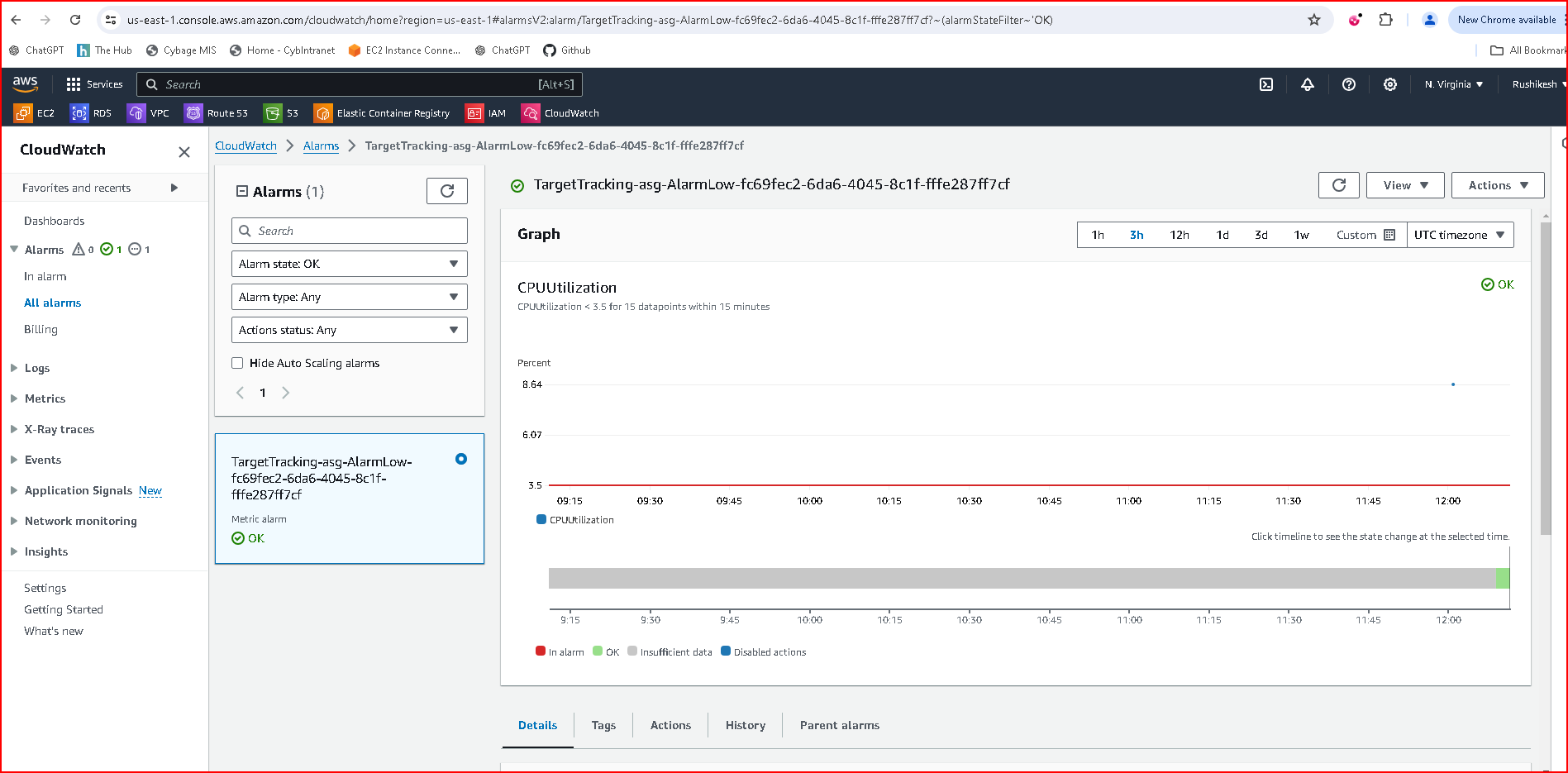
Create a new Auto Scaling group.

Configure launch configuration (AMI, instance type, key pair, etc.).

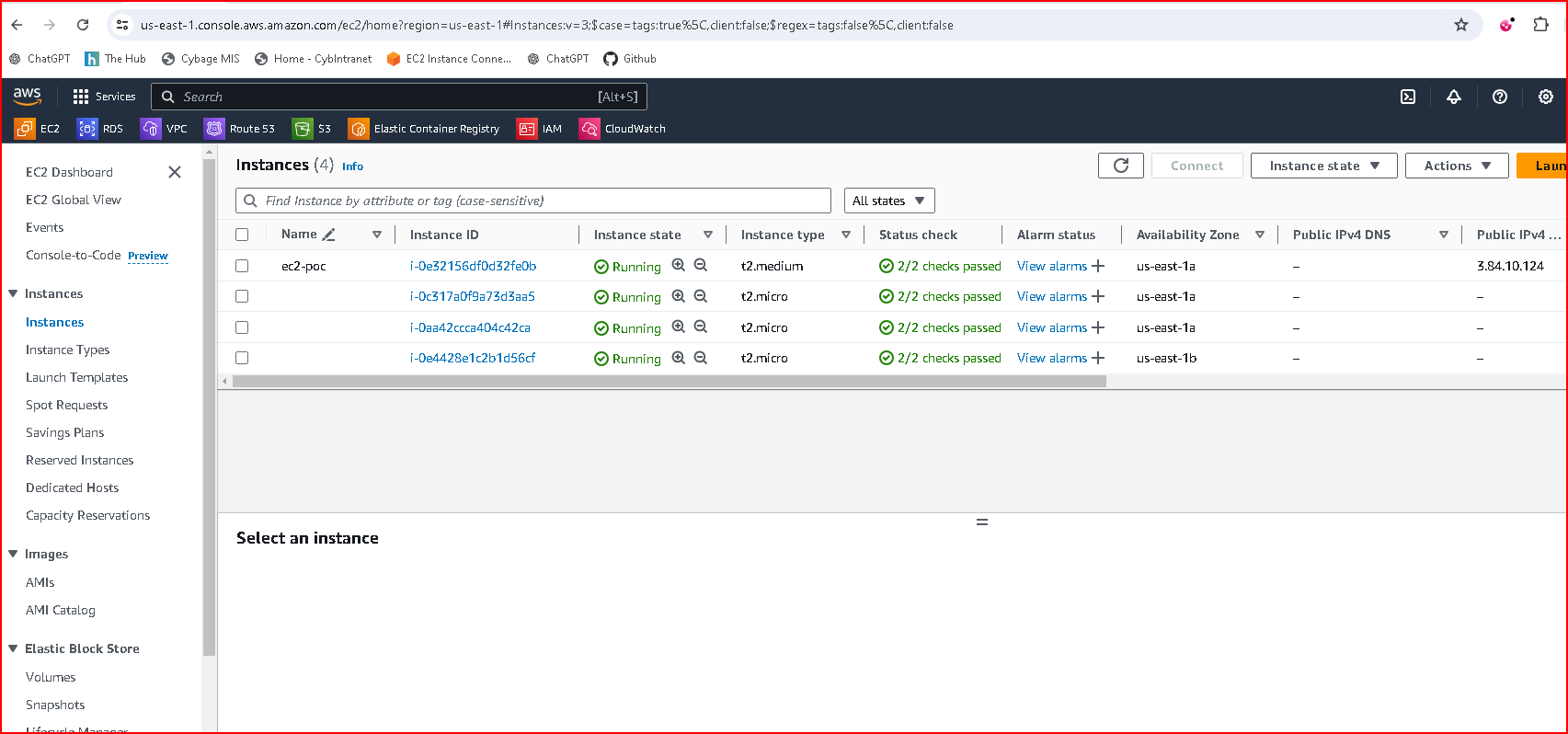
Define scaling policies based on metrics like CPU utilization or request count.



**Cloudwatch Alarms**

****

After increase load on instance Autoscaling launches instances



After decrease load on instance Autoscaling terminates instances

