1. Create the Project Folder

Folder is structured like this:

movieflix/

├── app.py

├── requirements.txt

├── Dockerfile

└── templates/

└── index.html

1. Create Kubernetes Cluster in AWS with default configurations (costs approx. Rs 20 per hour)

In Networking, change the API server endpoint access to **Public**

In Networking, click on Cluster Security Group and add **all traffic in Inbound Rules**

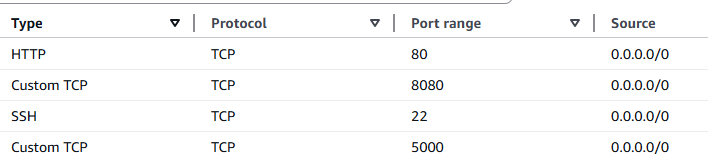
Go to VPC and Subnets. For all your subnets add the below **tags**:

kubernetes.io/cluster/clustername – owned

kubernetes.io/role/elb - 1

1. Create one EC2 Instance (t3.micro will work as it is free tier and if it doesn’t work, go for t3.medium which cost you approximately Rs 4/-)

Below should be configured as Inbound for EC2:



1. Login to EC2 using EC2 instance connect

Install Docker

Check if the docker package is available:

dnf list docker

install the docker

dnf install docker

start the docker

systemctl start docker

Enable the docker to start docker on every reboot

systemctl enable docker

1. Build the Docker Image

docker build -t movieflix-app .

docker image ls (check if image is created)

1. Run the Docker Container

docker run -d -p 5000:5000 movieflix-app

docker container ls (check if container is running)

1. Stop and Remove the Container (if needed)

docker stop containerid

docker rm containerid

1. Tag the docker image

docker tag movieflix-app bhanureddy/abctest:v1

docker login

docker push bhanureddy/abctest:v1

1. Create Jenkins using the below:  
   docker run -u 0 --privileged --name jenkins -it -d -p 8080:8080 -v /var/run/docker.sock:/var/run/docker.sock -v $(which docker):/usr/bin/docker -v /home/jenkins\_home:/var/jenkins\_home customjenkinsimagename

✅ "Run as the root user inside the container"

✅ "Also give this root user access to all host-level capabilities"

1. Access Jenkins url from <http://ip:8080> and get the initial admin password from /home/Jenkins\_home

Or you can loginto container and get from /var/Jenkins\_home

Docker exec -it containerid bash

1. Install Dockerpipeline, Pipline stage, Blueocean, aws credentials Plugin in Jenkins
2. Configure aws-eks-creds for AWS Credentials

Manage Jenkins -> Credentials -> System -> Global Credentials -> Add Credentials

Kind -> AWS Credentials and id should be **aws-eks-creds**

To generate Access Key id and secret access Key in AWS click on Profile -> Security Credentials -> Access Keys -> Create Access Keys (Download .csv file as it is available only for one time)

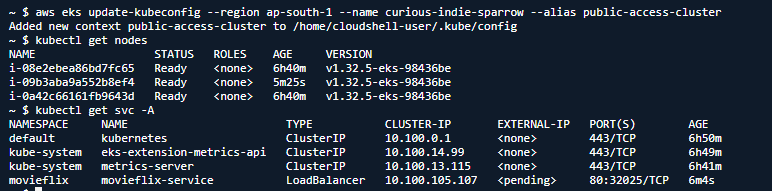
1. Create secret file eks-kubeconfig and upload the public access kubeconfig.yaml

aws eks update-kubeconfig --region ap-south-2 --name curious-dubstep-crab --alias public-access-cluster --kubeconfig kubeconfig.yaml

also execute the below command in cloudshell

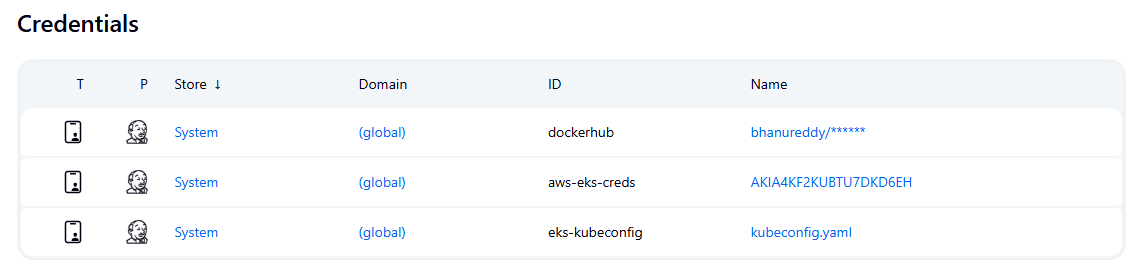
aws eks update-kubeconfig --region ap-south-1 --name curious-indie-sparrow --alias public-access-cluster

once deployment is done using the below command get the URL:



But here expternal-ip is showing as pending. This is because subnets were not tagged. If you follow the step2 and tag the subnets you can see the URL part under external-ip with which you can access the application

1. Create Docker credentials with id as dockerhub



Finally create the job using Jenkinsfile and build it

If any update happens to Github repo automatically the build and deployment should happen and for this in Poll SCM give \* \* \* \* \*