[**Dog-vs-Cat-Classifier**](https://github.com/KNarendra99599/Dog-vs-Cat-CNN-Classifier-Dockerization_output.git)

**In Local working steps**

sudo apt install python3

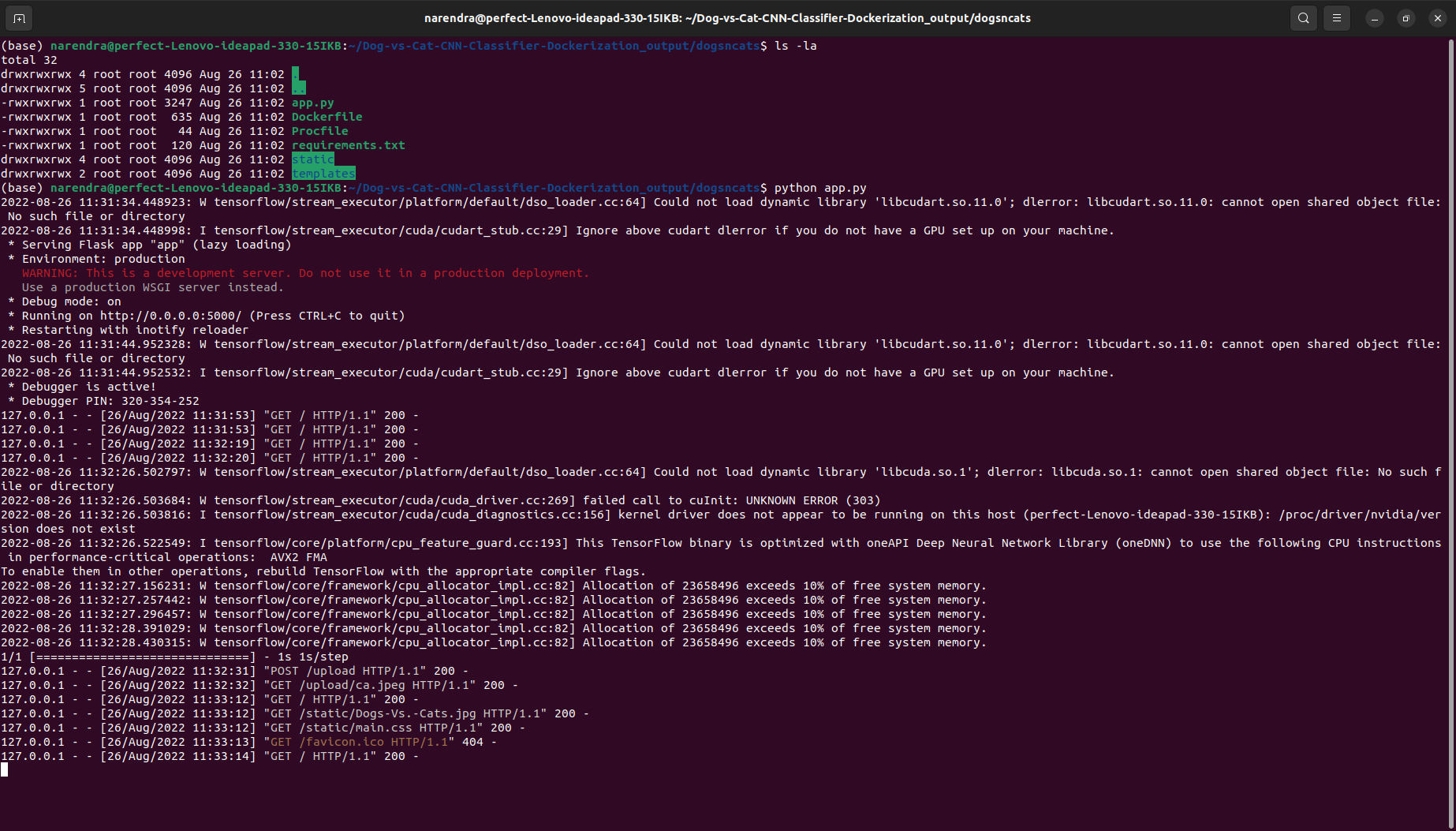
sudo pip install tensorflow

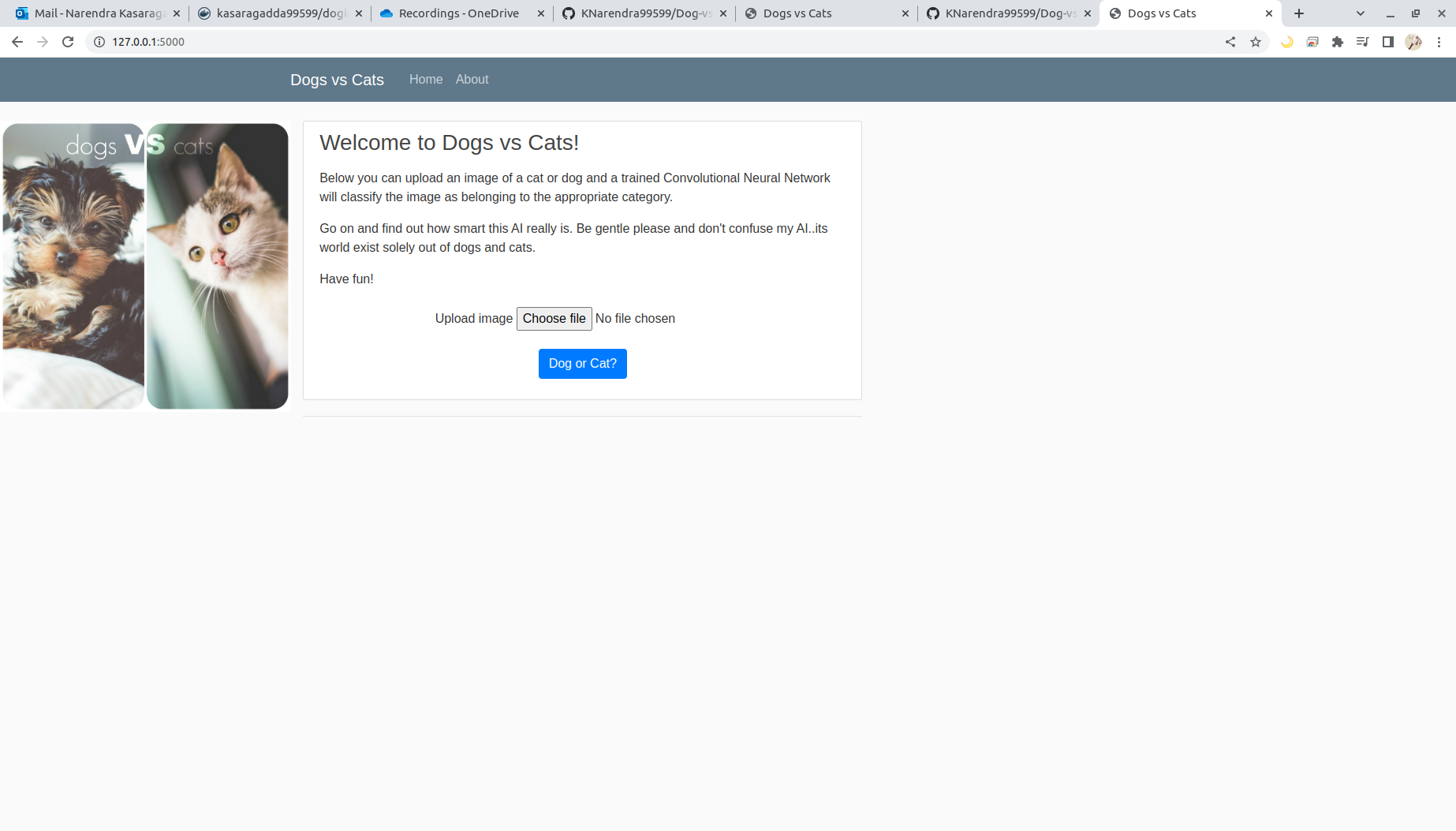
sudo git clone <https://github.com/KNarendra99599/Dog-vs-Cat-CNN-Classifier-Dockerization_output.git>

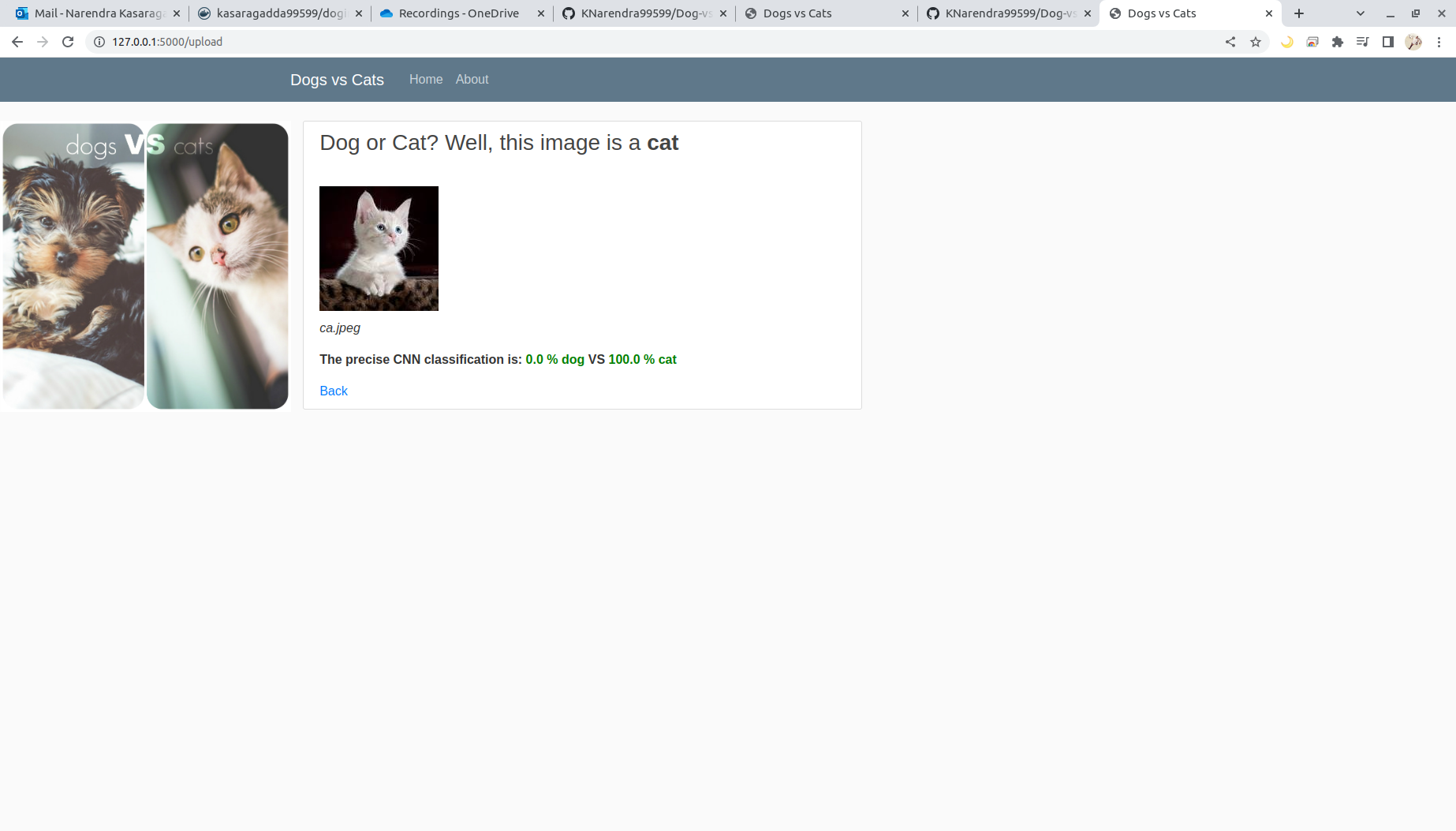
permission need to give folder

**cd** [Dog-vs-Cat-CNN-Classifier-Dockerization\_output](https://github.com/KNarendra99599/Dog-vs-Cat-CNN-Classifier-Dockerization_output.git)

**cd** dogsncats







**In Local - Containerization steps**

sudo curl <https://get.docker.com/> | bash

sudo git clone https://github.com/KNarendra99599/Dog-vs-Cat-CNN-Classifier-Dockerization\_output.git

**cd** [Dog-vs-Cat-CNN-Classifier-Dockerization\_output](https://github.com/KNarendra99599/Dog-vs-Cat-CNN-Classifier-Dockerization_output.git)

**cd** dogsncats

sudo docker build -t imagename:versionname .

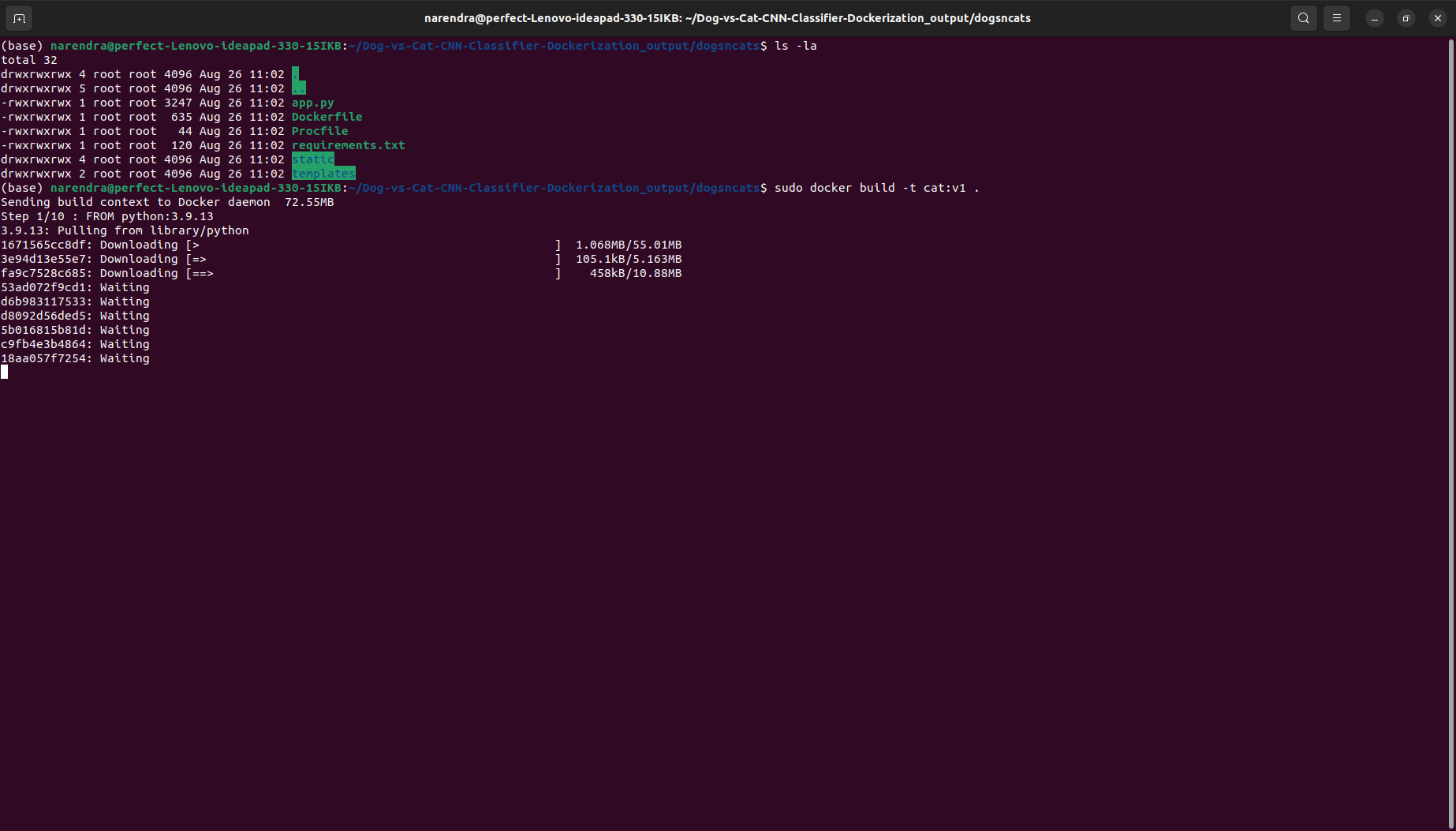
sudo docker images

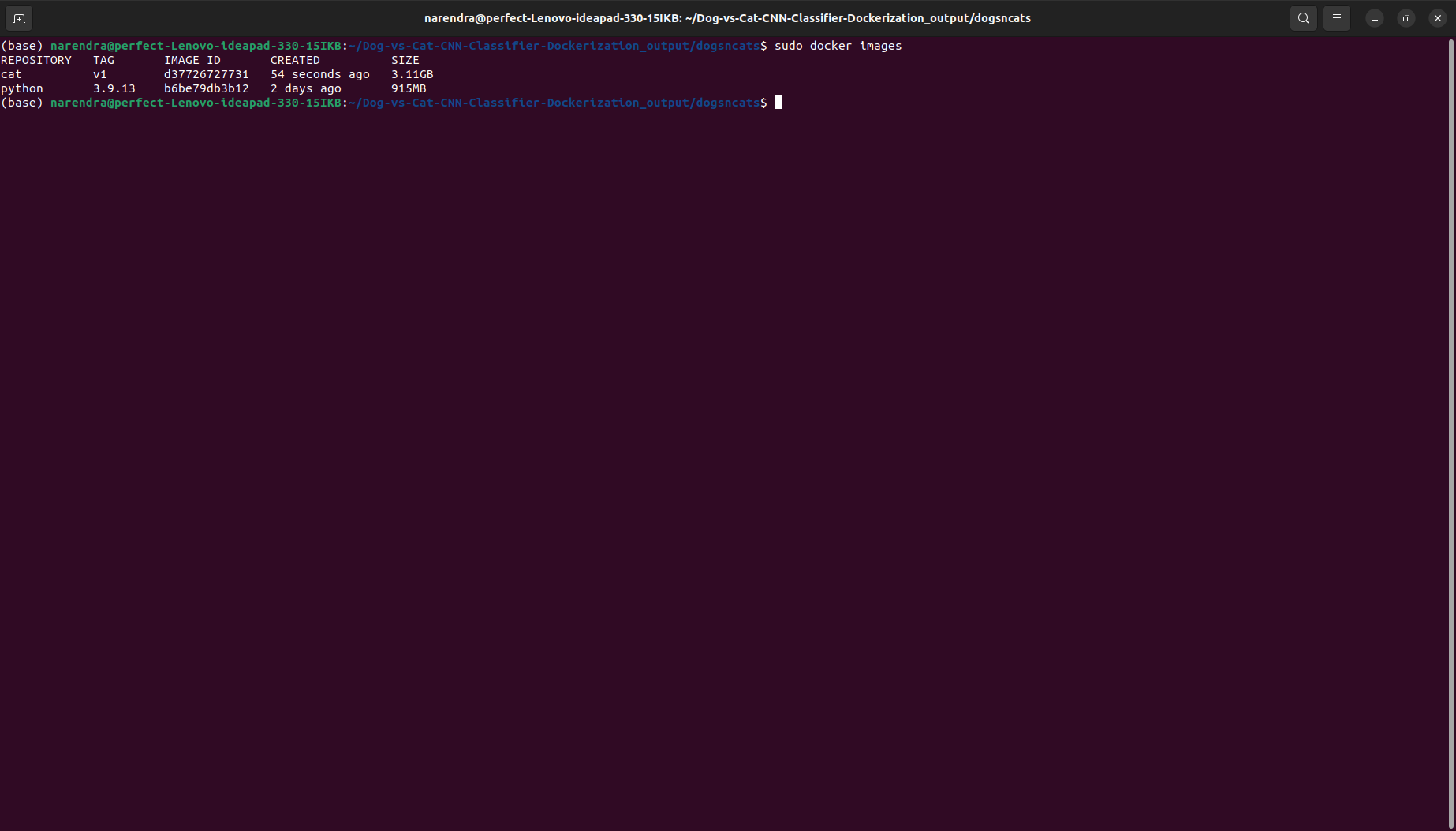
sudo docker run -dit --name anyname --hostname anyname

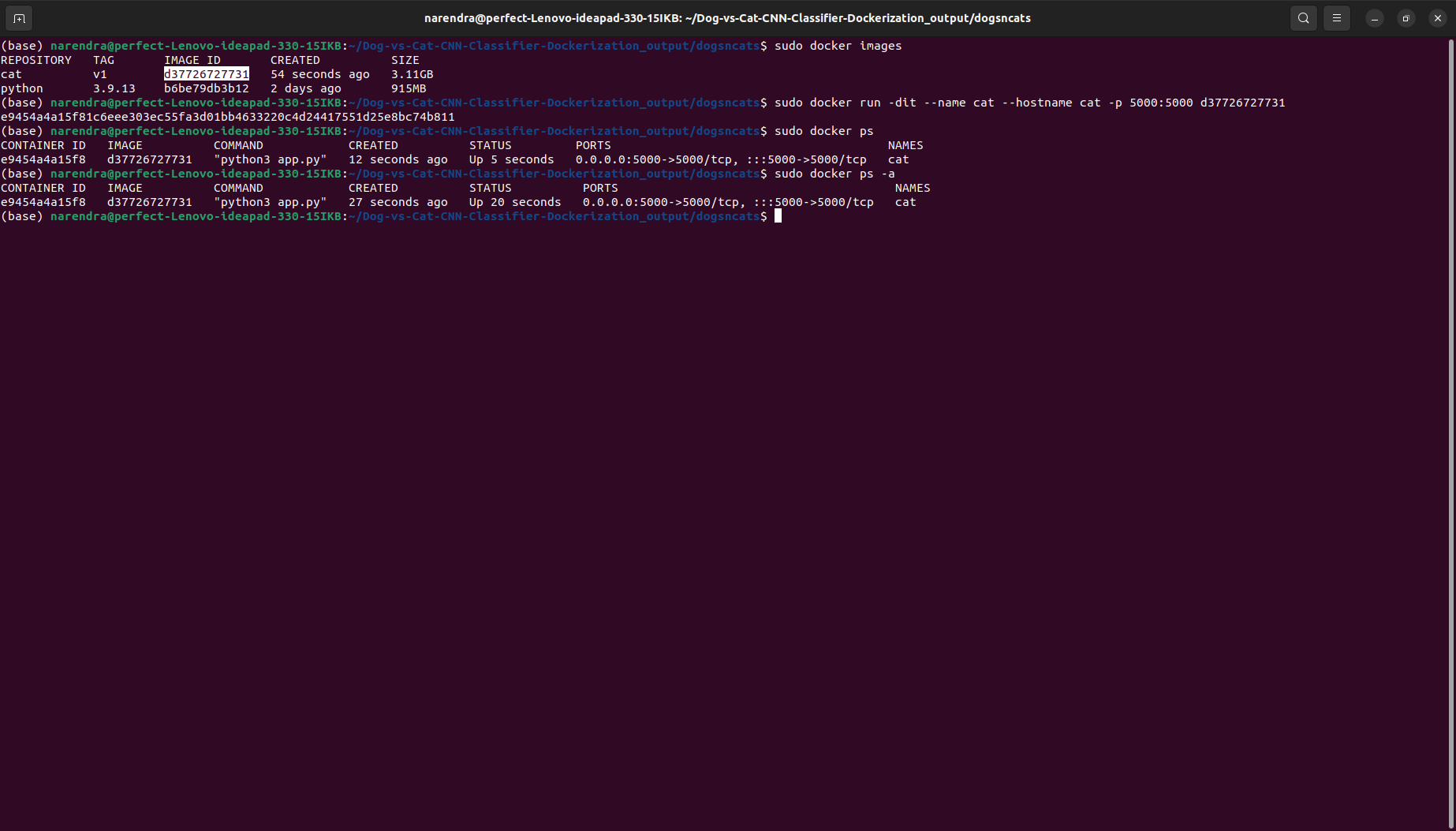
p 5000:500 imageID

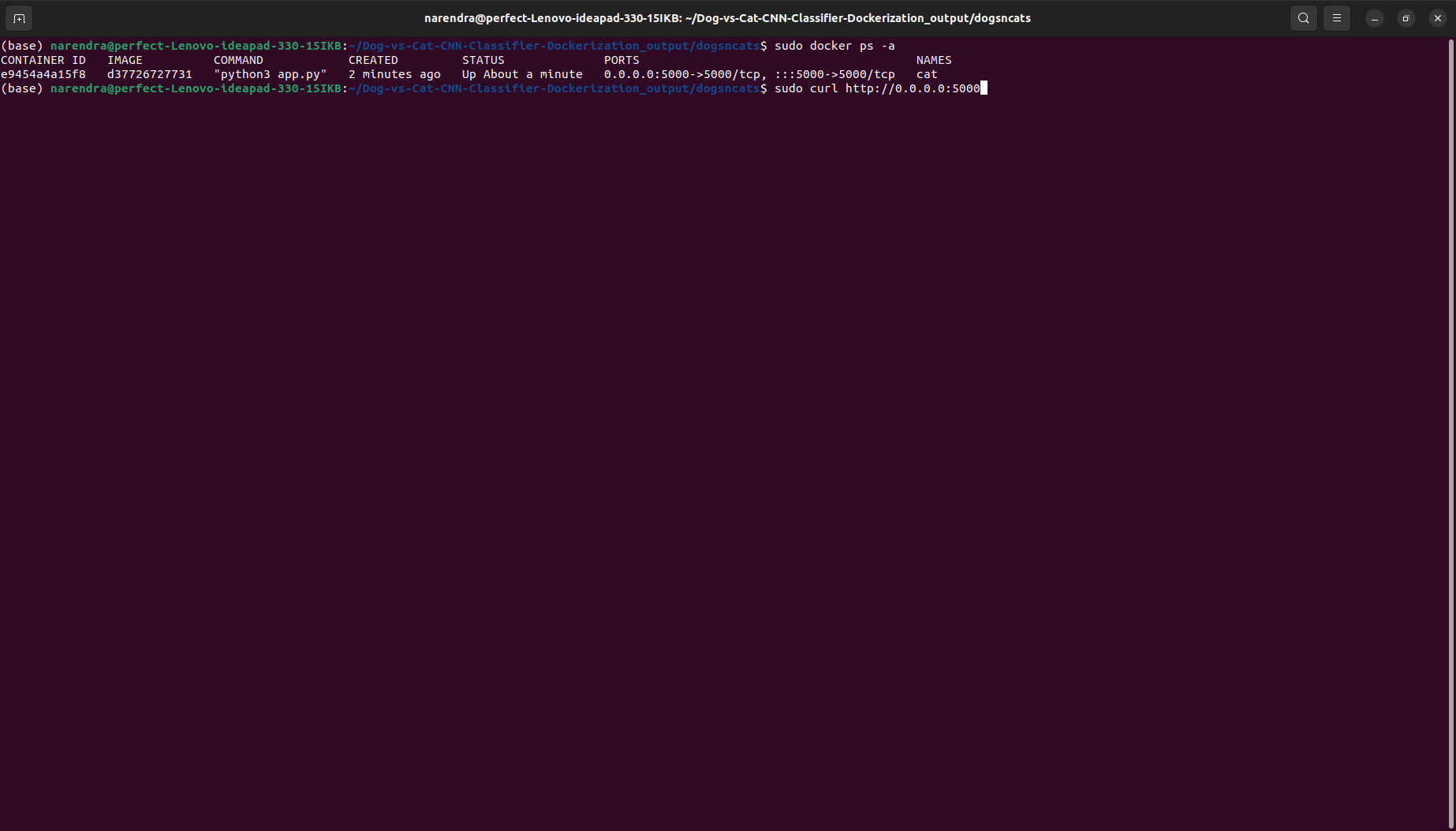
sudo docker ps

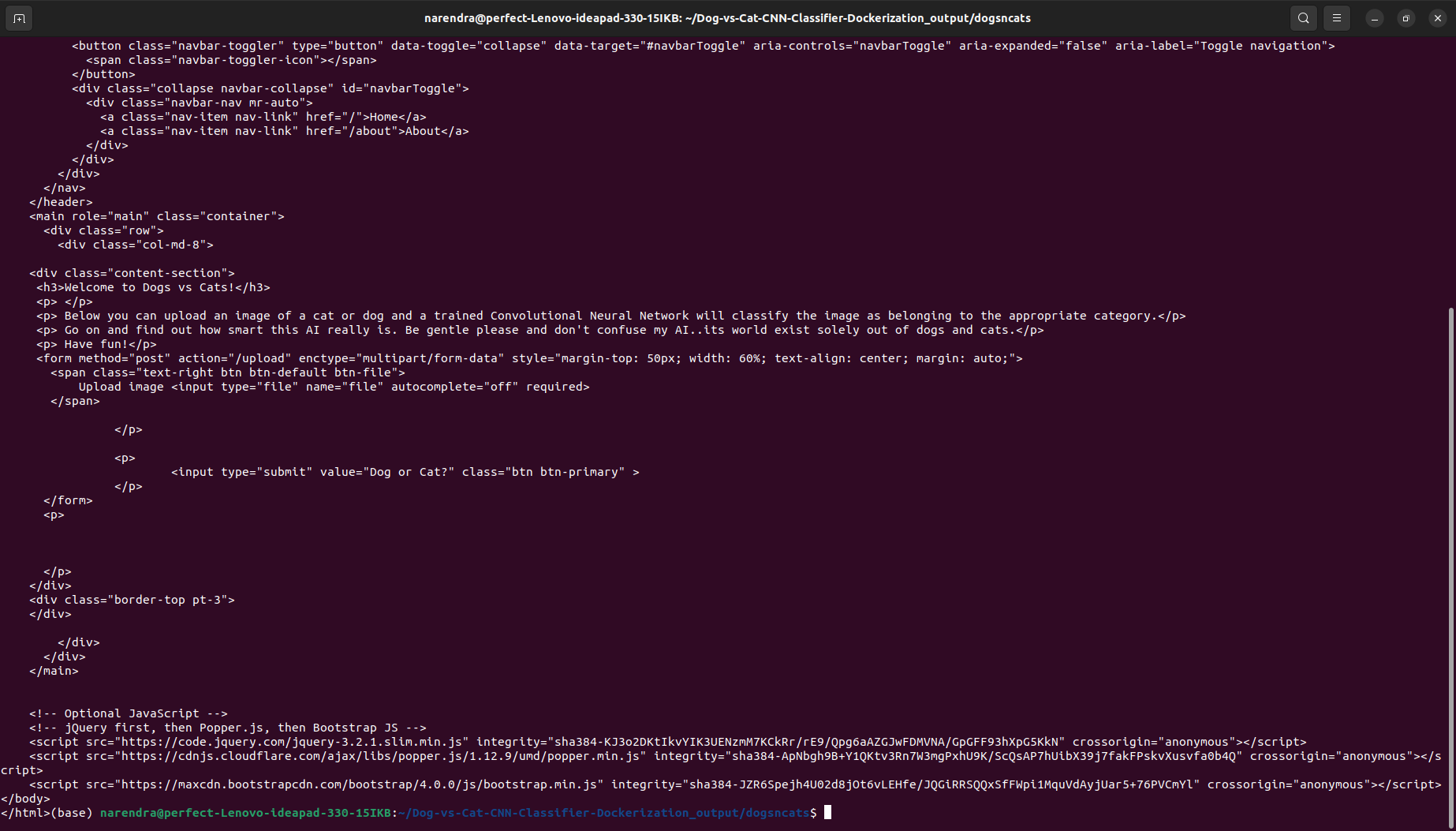
sudo docker ps -a

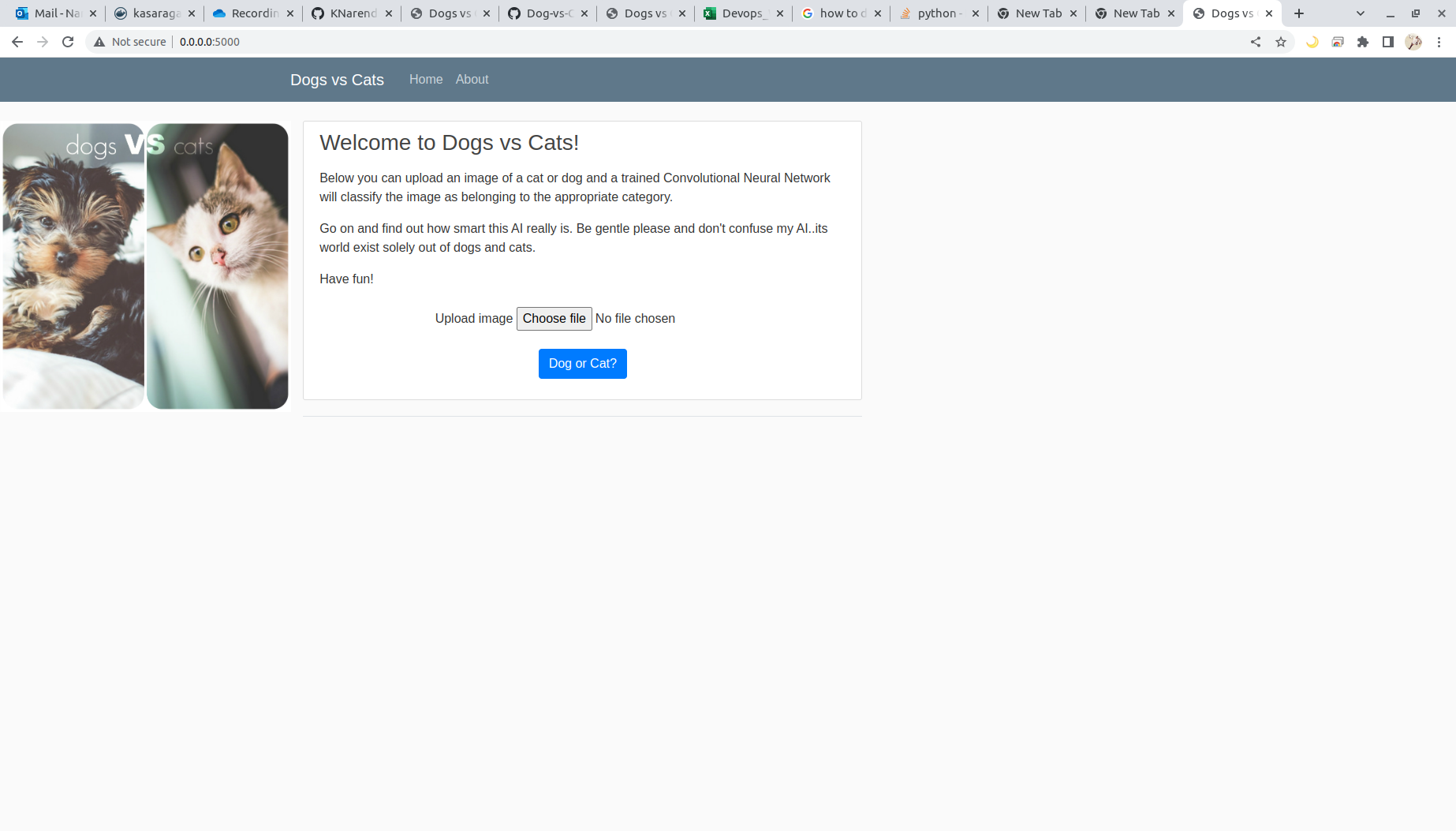


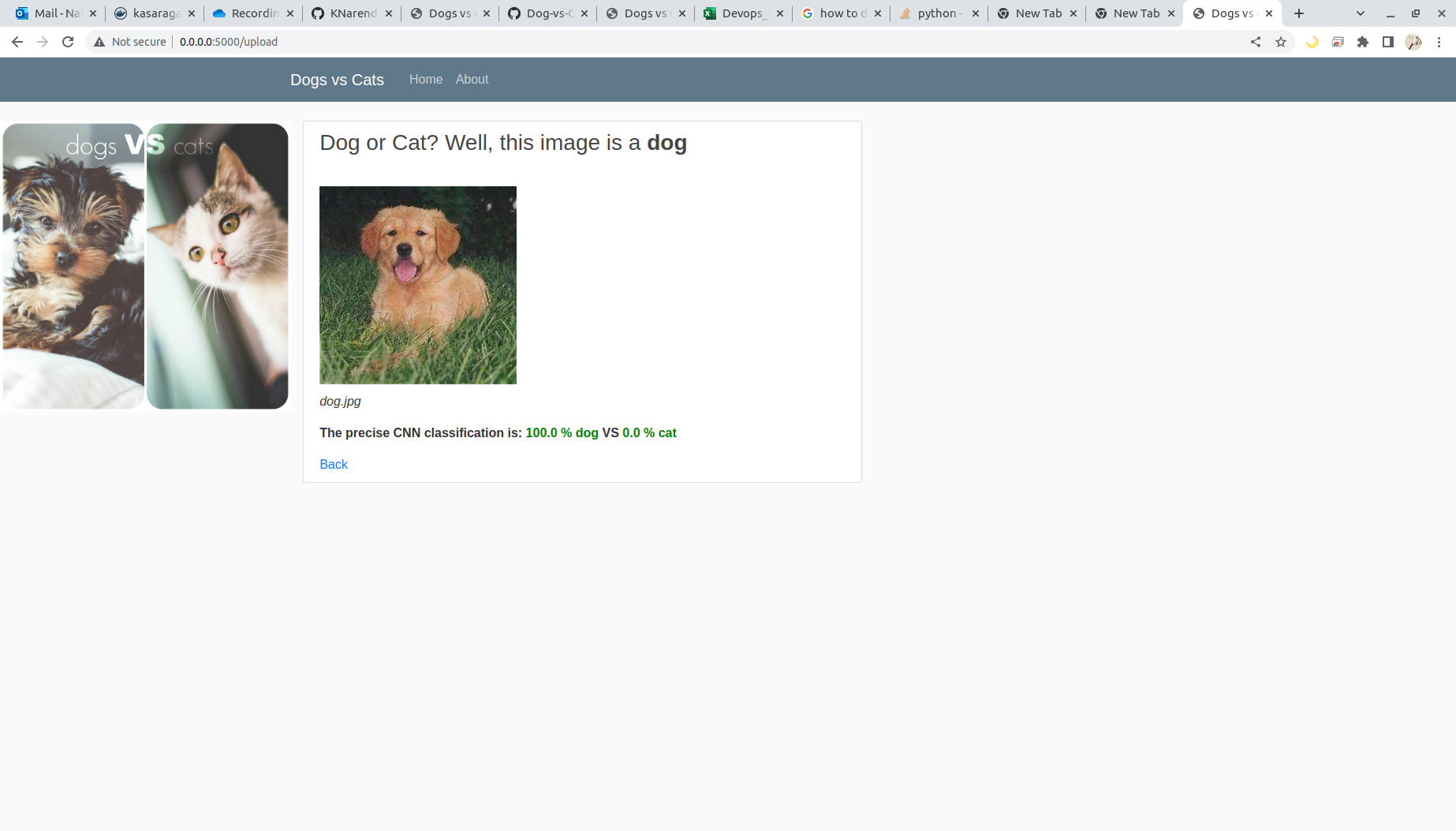












**AWS Cloud**

EC2 Instance – ubuntu 18.04 or 20.04 – instance type t2 medium



**Volume 30 gb**

**Steps/pre-requisities need to be perform inside the EC2 Instance**

**Jenkins Installation**

#!/bin/bash

sudo add-apt-repository ppa:openjdk-r/ppa -y

sudo apt-get update

sudo apt install openjdk-8-jdk -y

sudo update-alternatives --config java

java -version

sudo wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -

sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'

sudo apt-get update

sudo apt-get install jenkins -y

sudo systemctl start jenkins.service

sudo systemctl enable jenkins.service

sudo apt install maven -y

sudo iptables -A PREROUTING -t nat -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 8080

sudo sh -c "iptables-save > /etc/iptables.rules"

on jenkins master

sudo usermod -aG docker jenkins

if above doesn’t work:

usermod -aG root jenkins

chmod 664 /var/run/docker.sock

chmod 777 /var/run/docker.sock

sudo service jenkins restart

cat /var/lib/jenkins/secrets/initialAdminPassword

sudo visudo -f /etc/sudoers

jenkins ALL=(ALL) NOPASSWD:ALL

esc :wq!



**Install the AWS CLI version 2 on Linux | How to Install the AWS CLI version 2 on Linux**

sudo curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"

sudo apt install unzip

sudo unzip awscliv2.zip

sudo ./aws/install

OR

sudo apt install awscli

aws –version

**Download and extract the latest release of eksctl with the following command**

sudo curl --silent --location

"https://github.com/weaveworks/eksctl/releases/latest/download/eksctl\_$(uname -

s)\_amd64.tar.gz" | tar xz -C /tmp

sudo mv /tmp/eksctl /usr/local/bin

eksctl version

**Install kubectl on Ubuntu Instance | How to install kubectl in Ubuntu | Install kubectl on Linux Instance**

sudo curl --silent --location -o /usr/local/bin/kubectl <https://s3.us-west-1.amazonaws.com/amazon-eks/1.22.6/2022-03-09/bin/linux/amd64/kubectl>

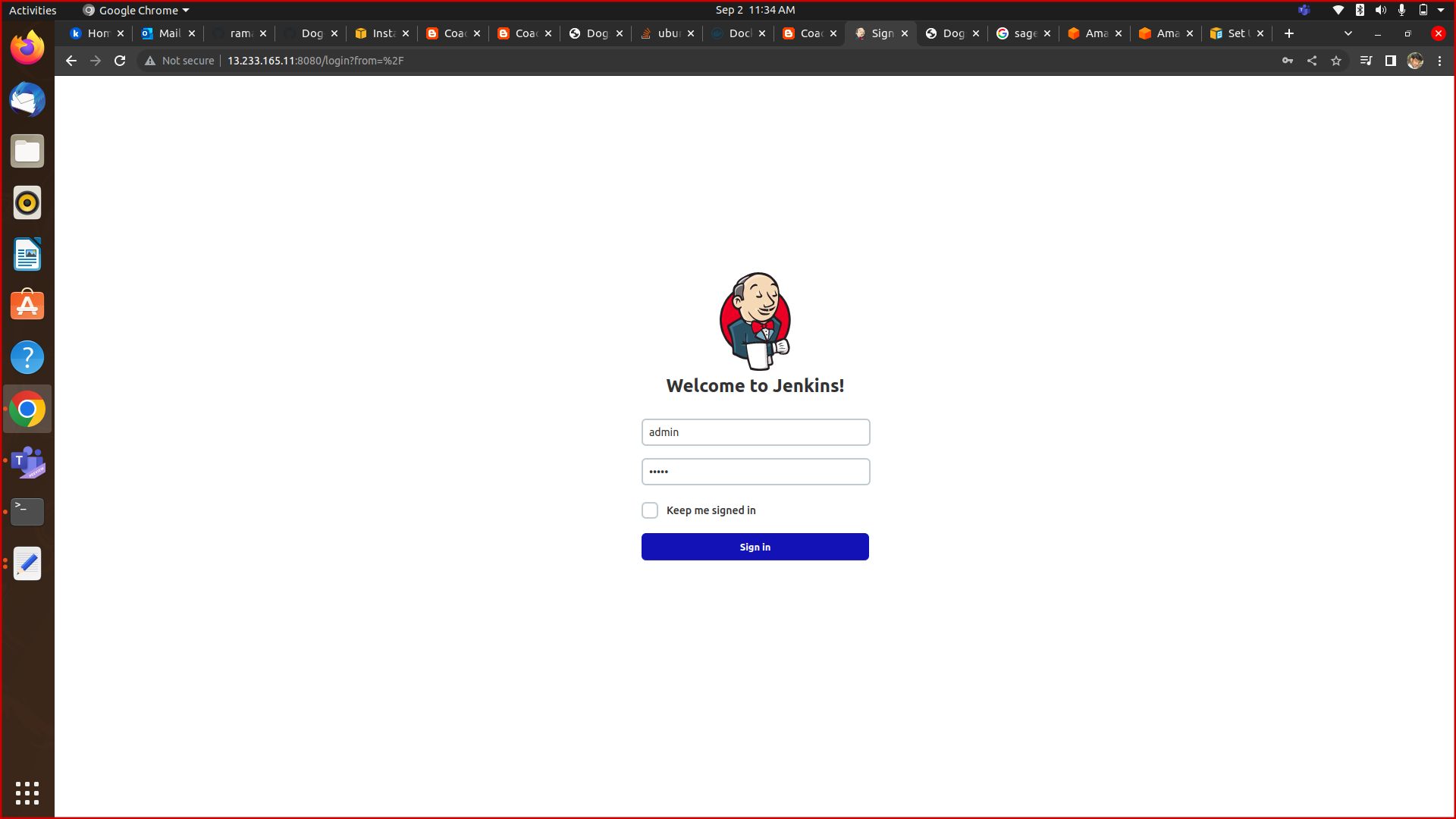
sudo chmod +x /usr/local/bin/kubectl

kubectl version --short –client

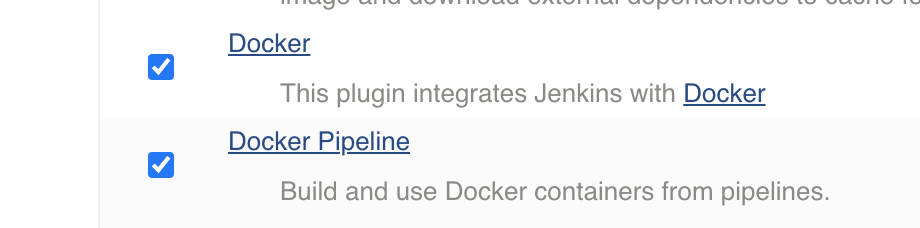
**Jenkins dashboad**

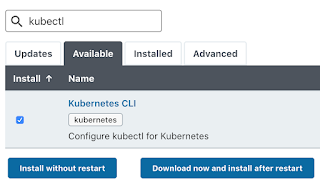
To Acess -> Public IP of EC2 Instance : port number of Jenkins

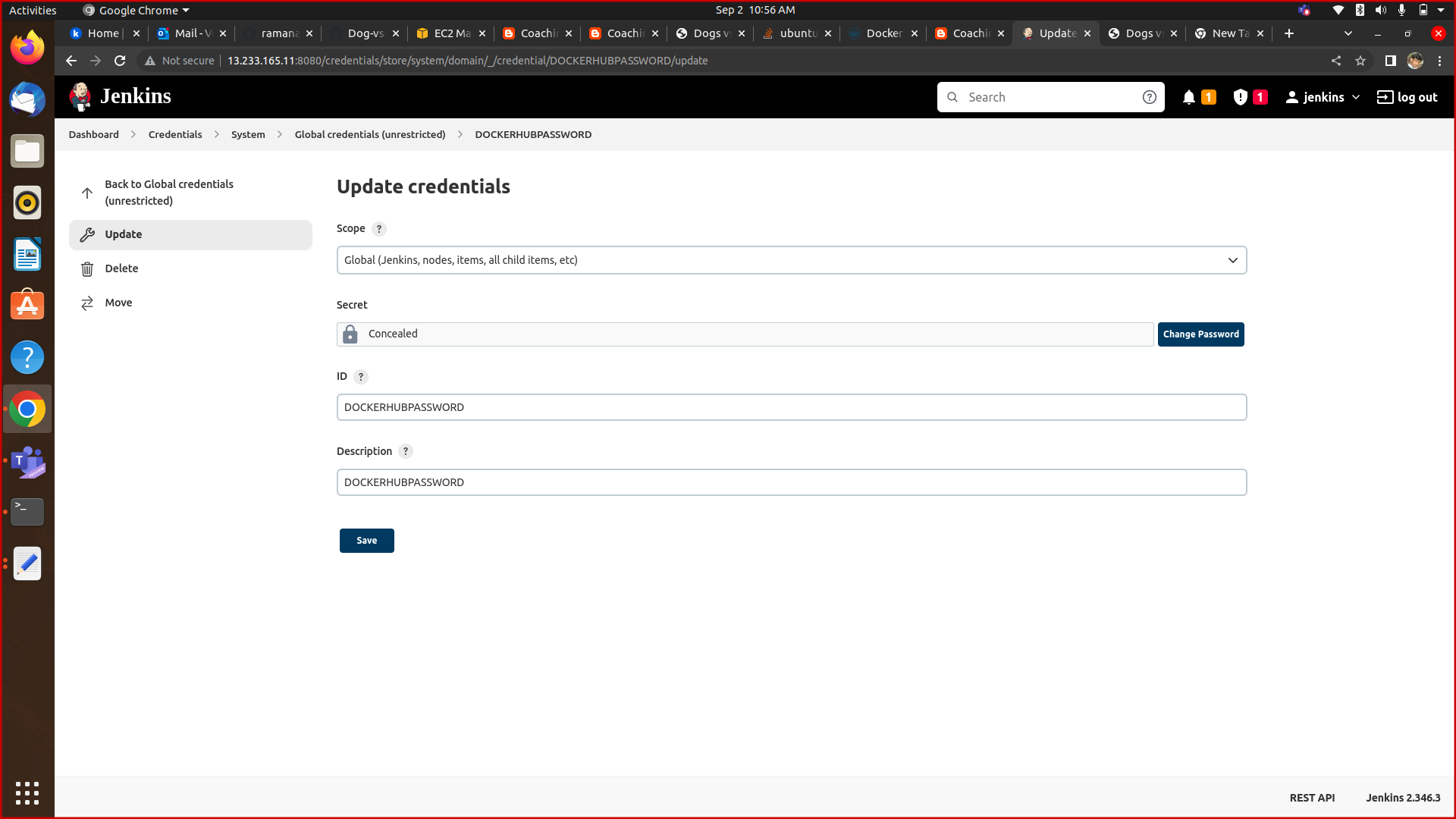
Login to Jenkins dashboad



**Docker, Docker pipeline and Kubernetes CLI plug-ins are installed in Jenkins**



****



**Assign role on jenkins mechine - AdministratorAccess**

**Switch to Jenkins user**

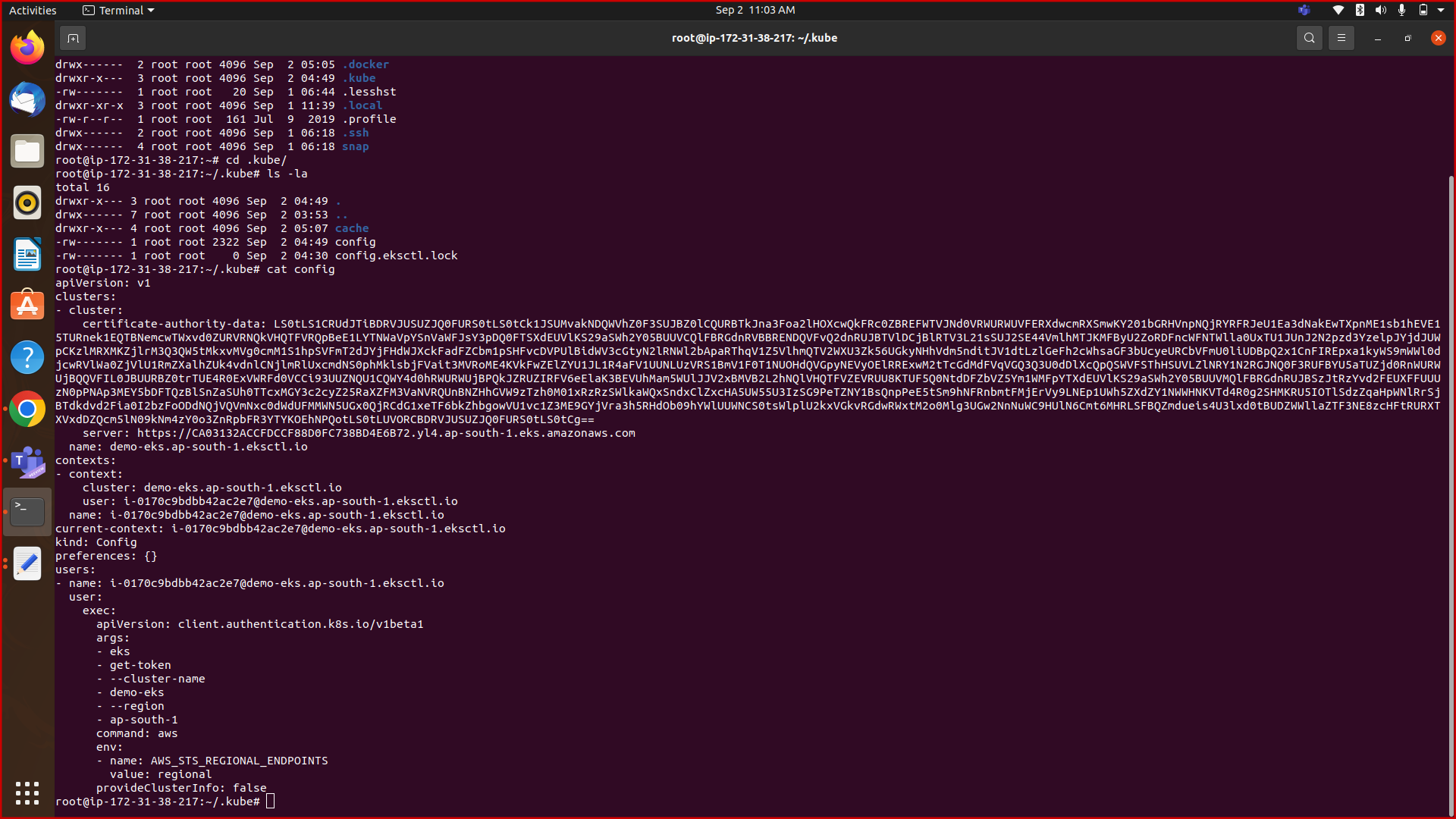
sudo su – jenkins

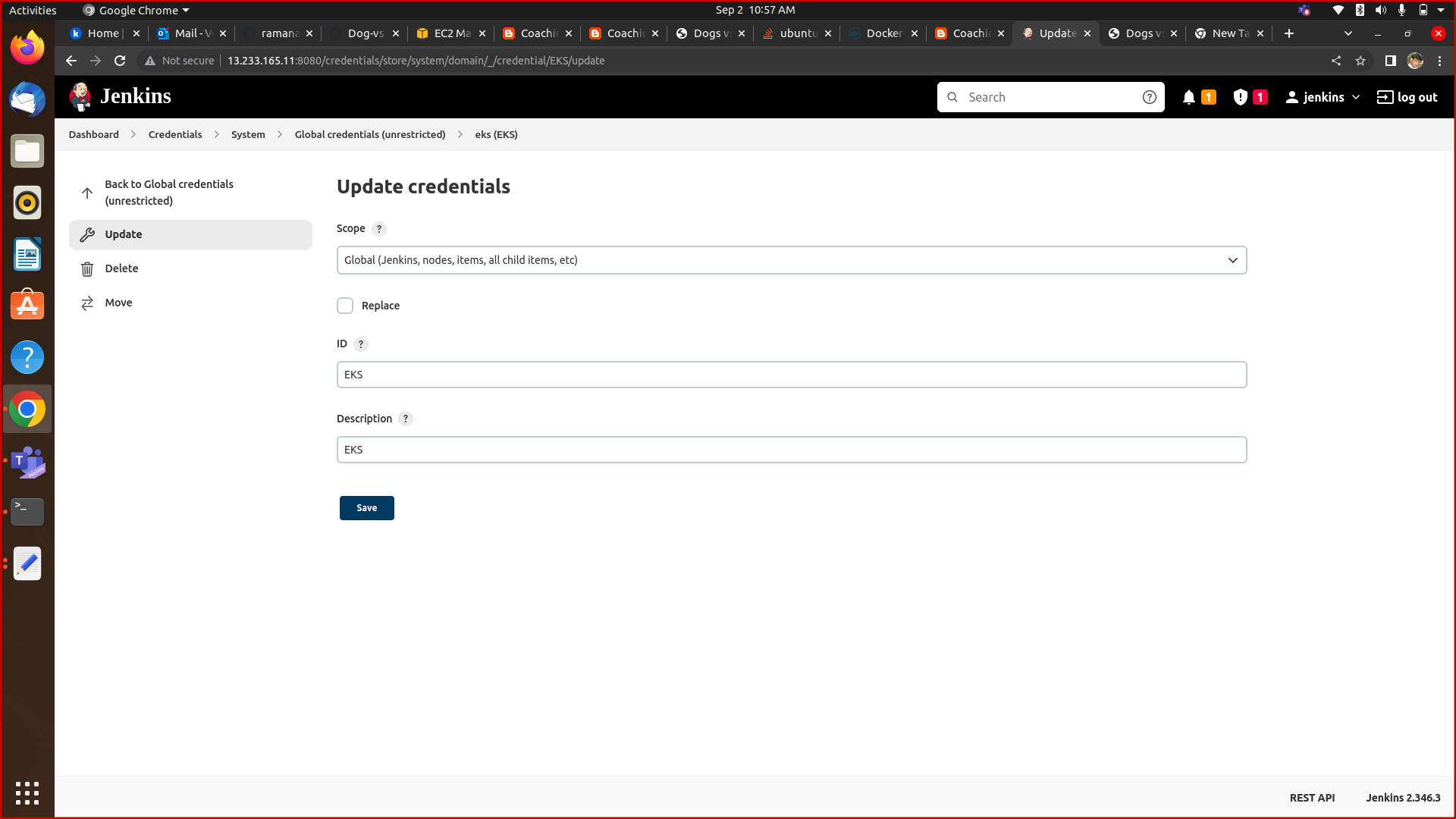
**Create EKS Cluster with two worker nodes using eksctl**

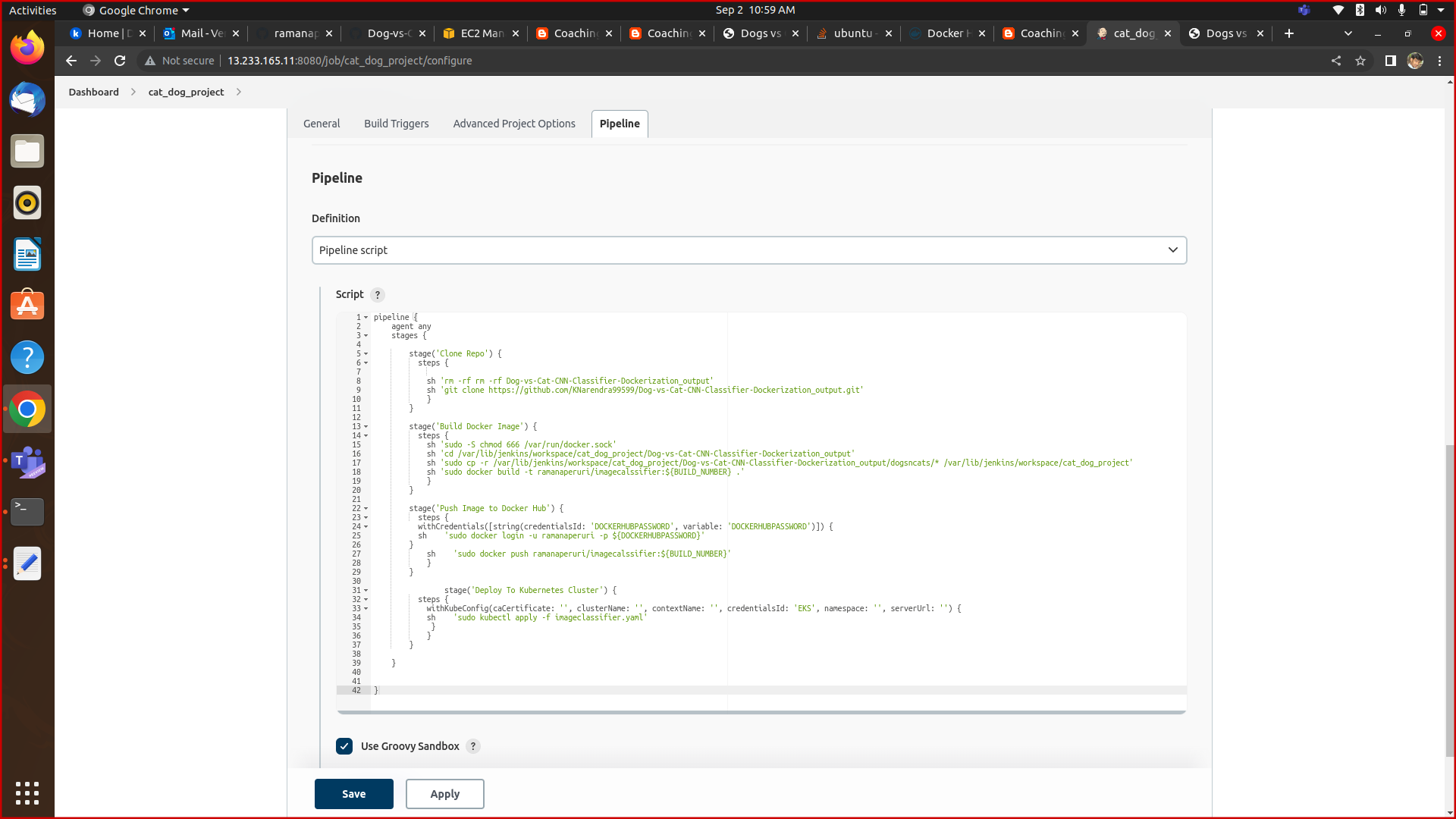
sudo eksctl create cluster --name demo-eks --region us-east-1 --nodegroup-name my-nodes --node-type t3.small --managed --nodes 1

eksctl get cluster --name demo-eks --region us-east-1

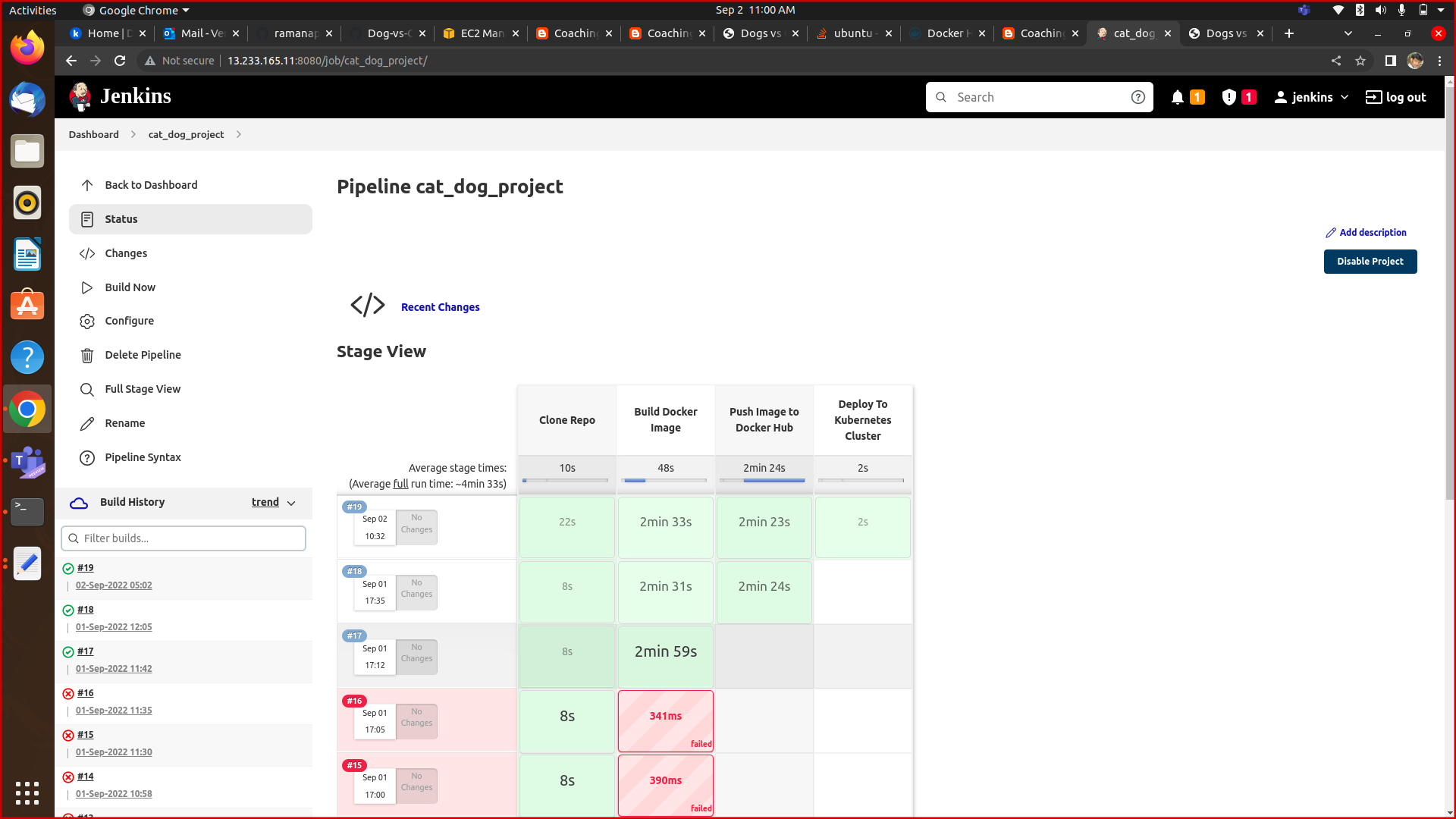
**kubeconfig file be updated under /var/lib/jenkins/.kube folder**

cat /var/lib/jenkins/.kube/config





**Jenkins dashboad-Run pipeline**



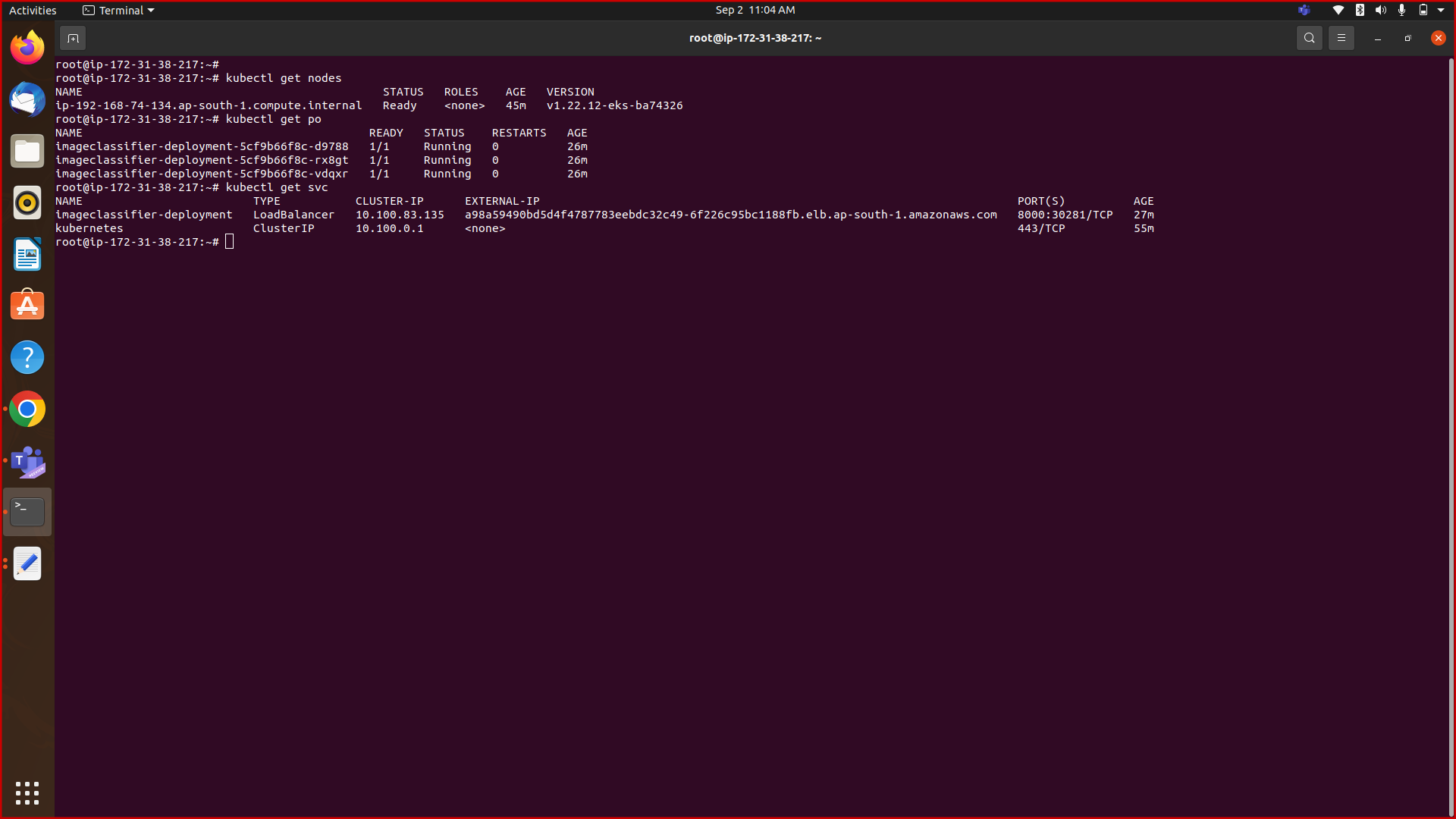
**Connect to EKS cluster using kubectl commands**

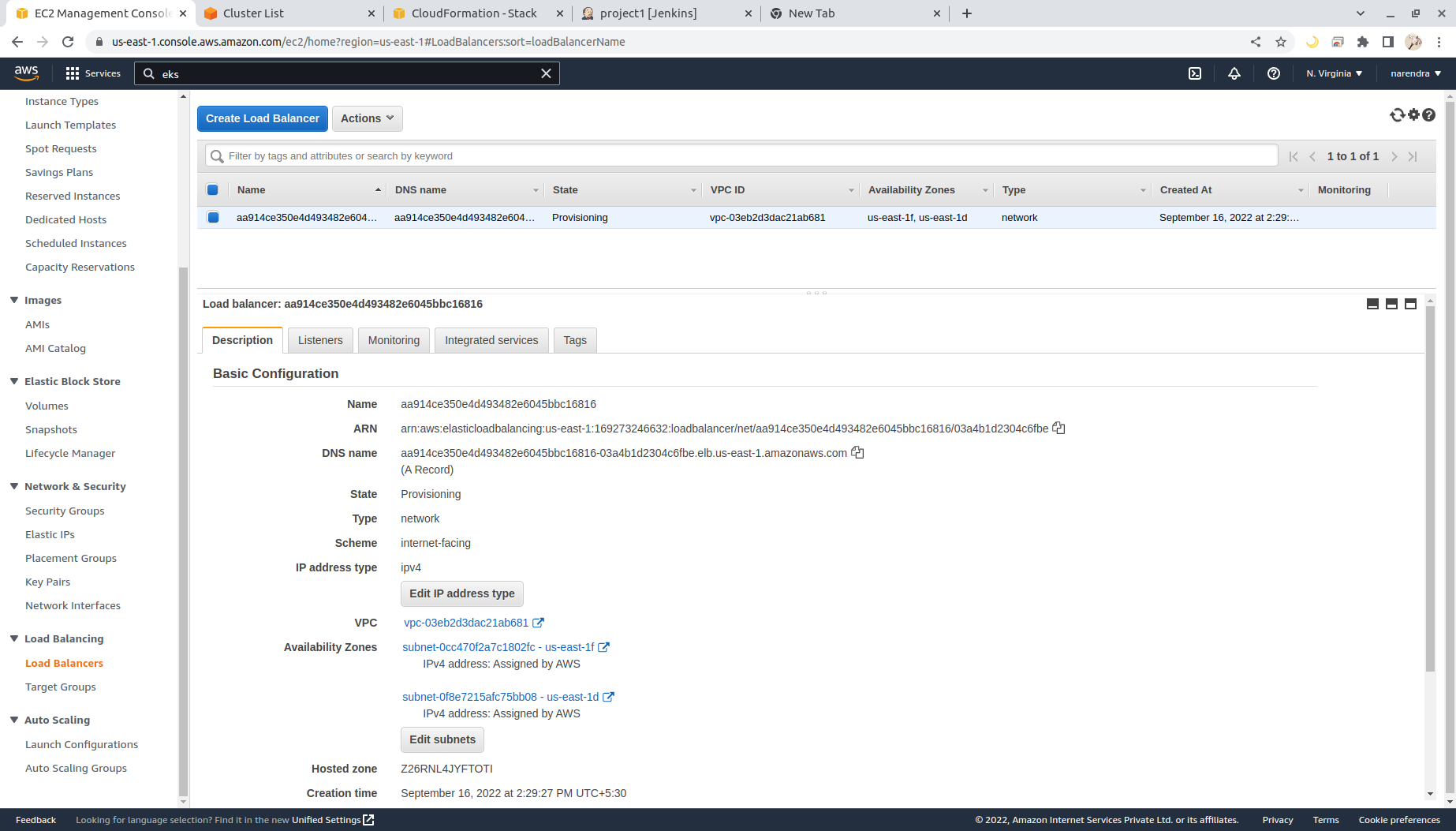
sudo kubectl get nodes

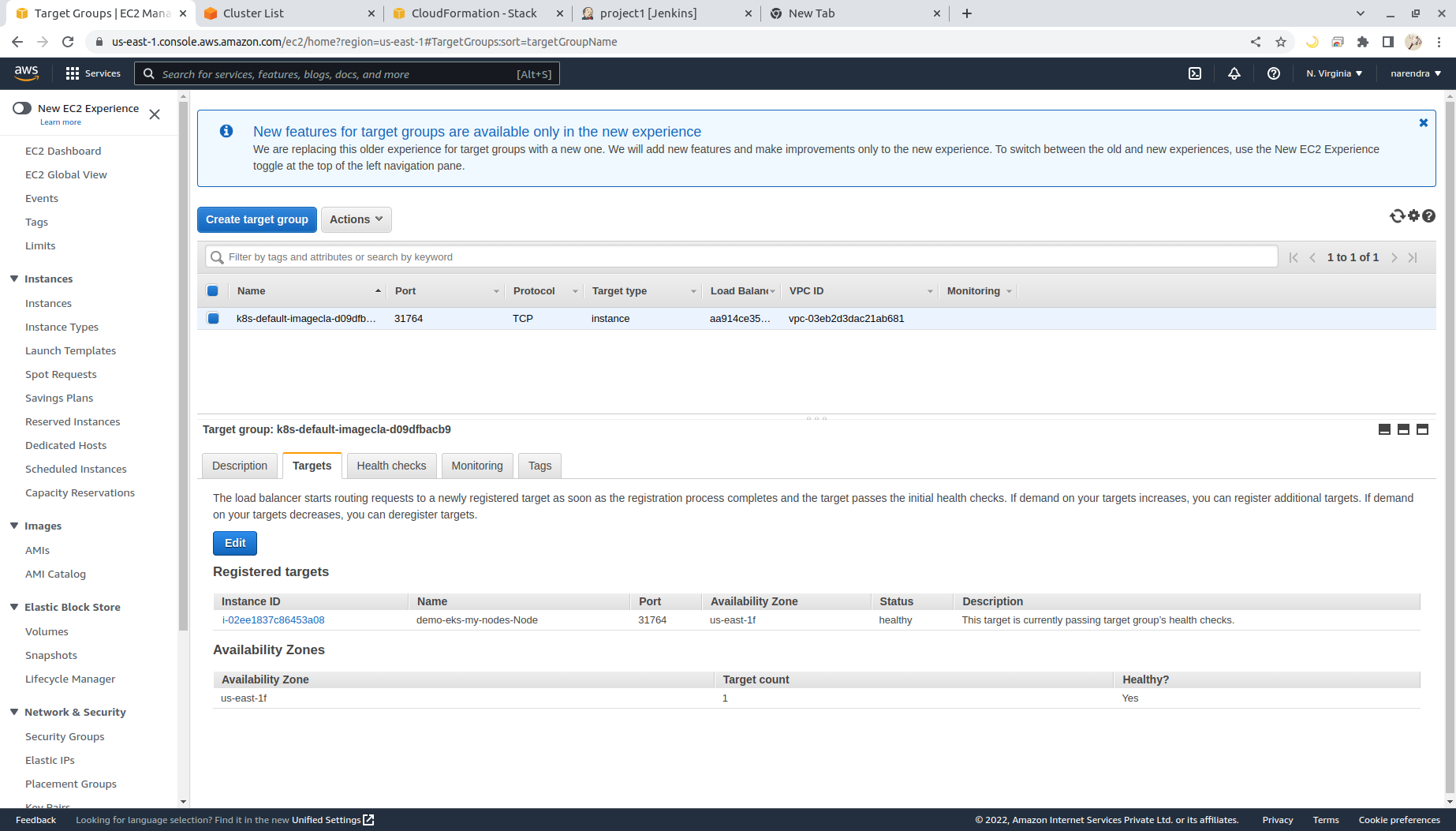
sudo kubectl get ns

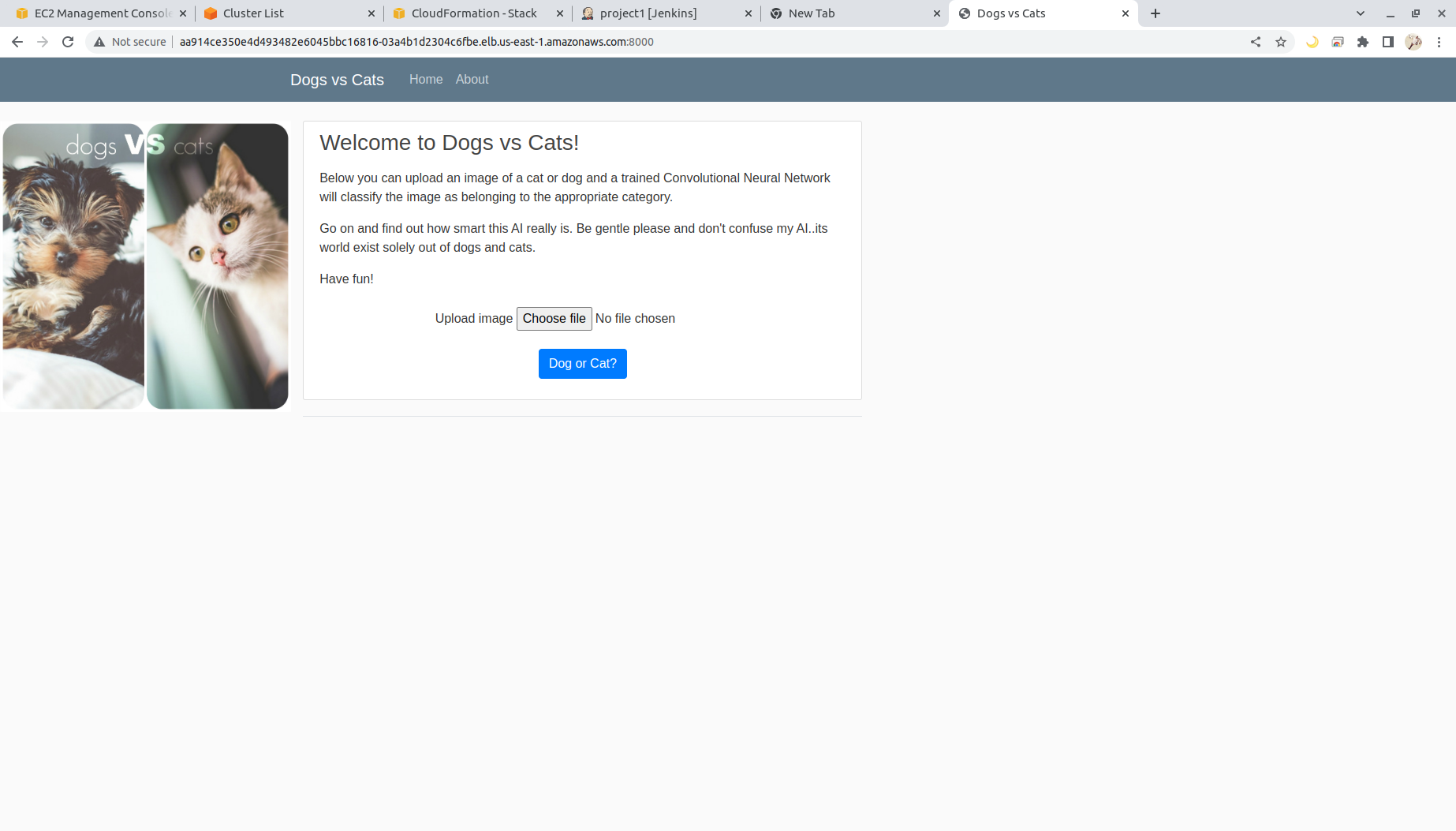
sudo kubectl get po

sudo kubectl get svc





****







**Delete EKS Cluster using eksctl**

eksctl delete cluster --name demo-eks --region us-east-1