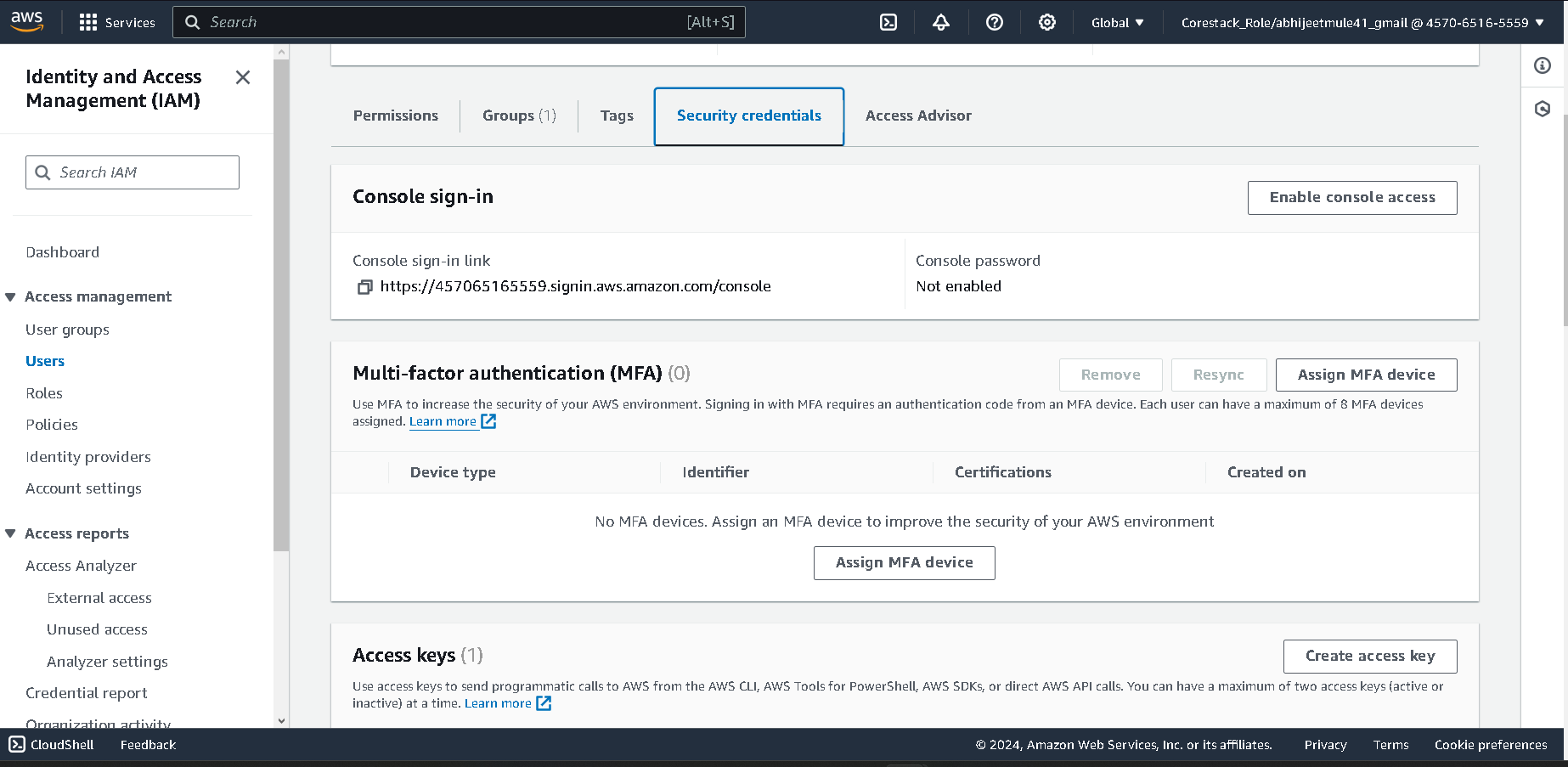
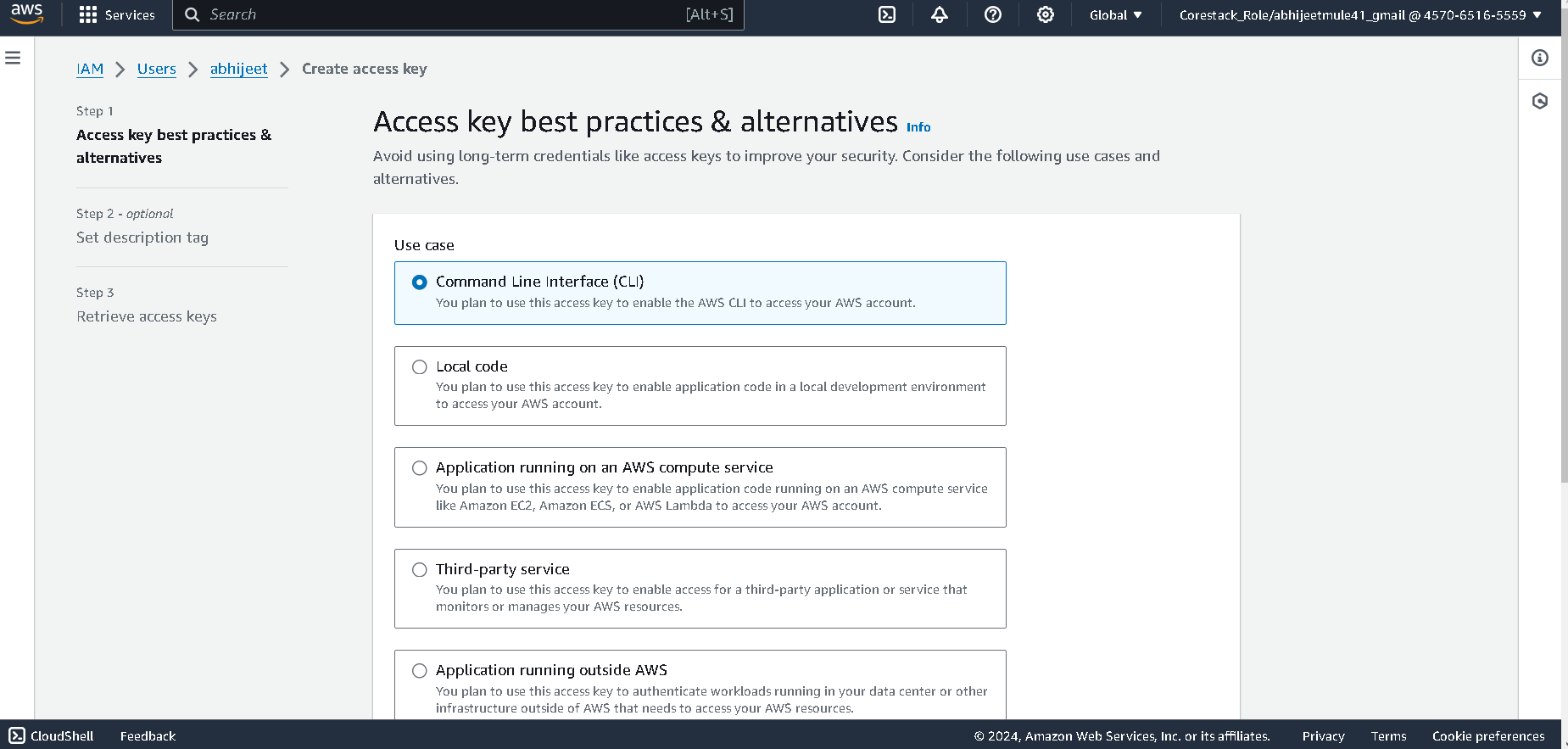


Now go to security credential then click on create access key button and download .svc file





**Note this 2 keys**

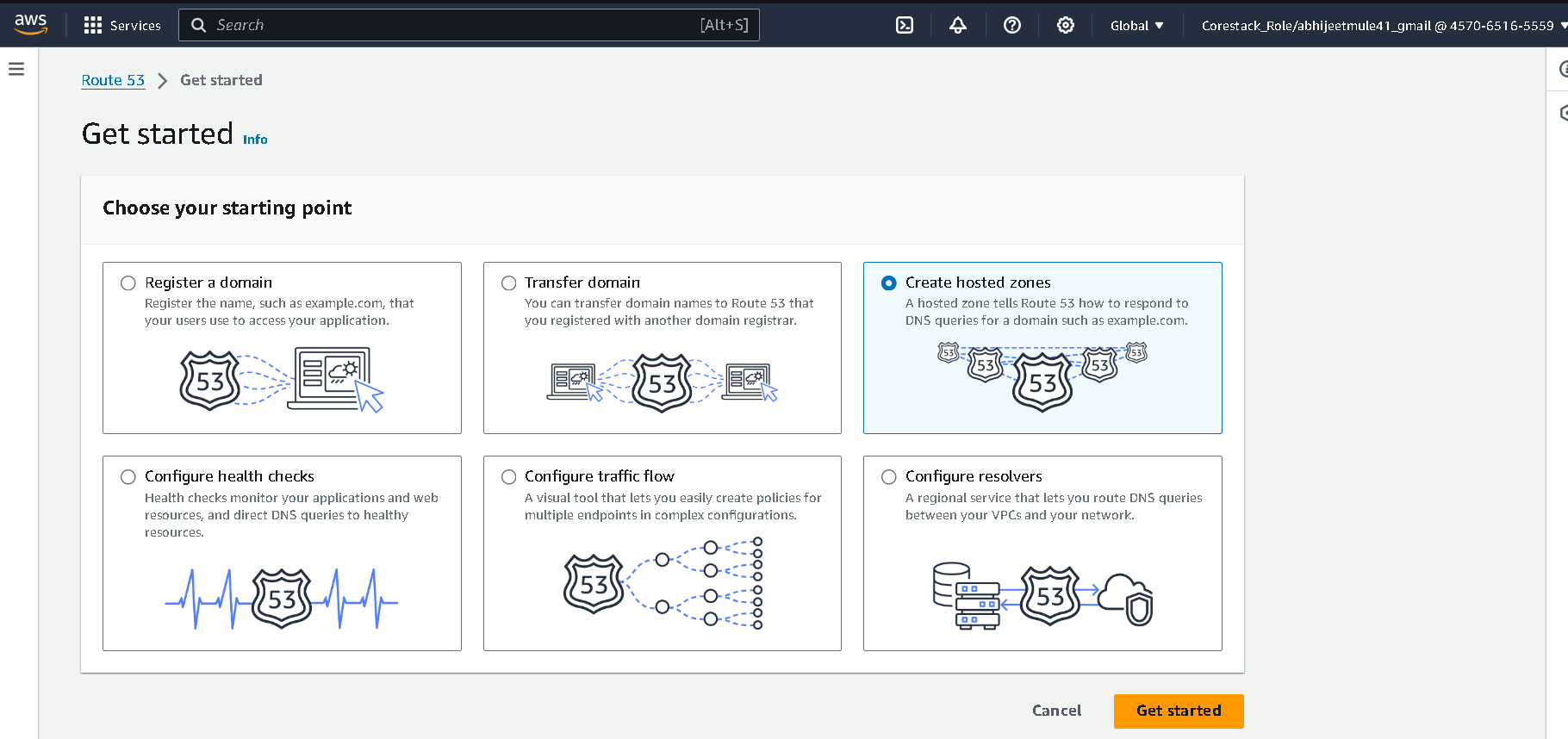
Access key ID

AKIAWU2ZU6L3X3I4ON53

Secret access key"

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

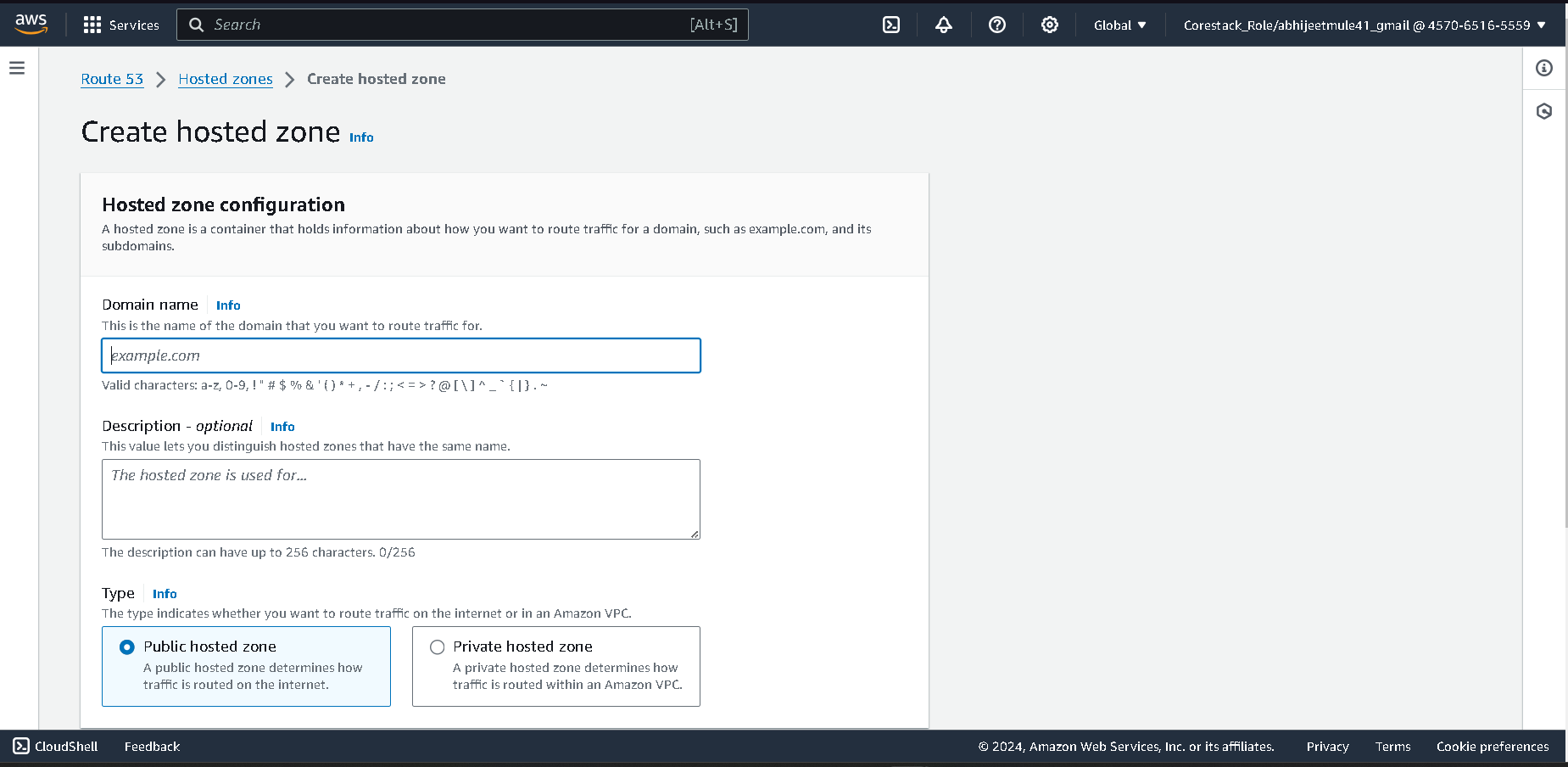
Now Open Aws Route 53 option



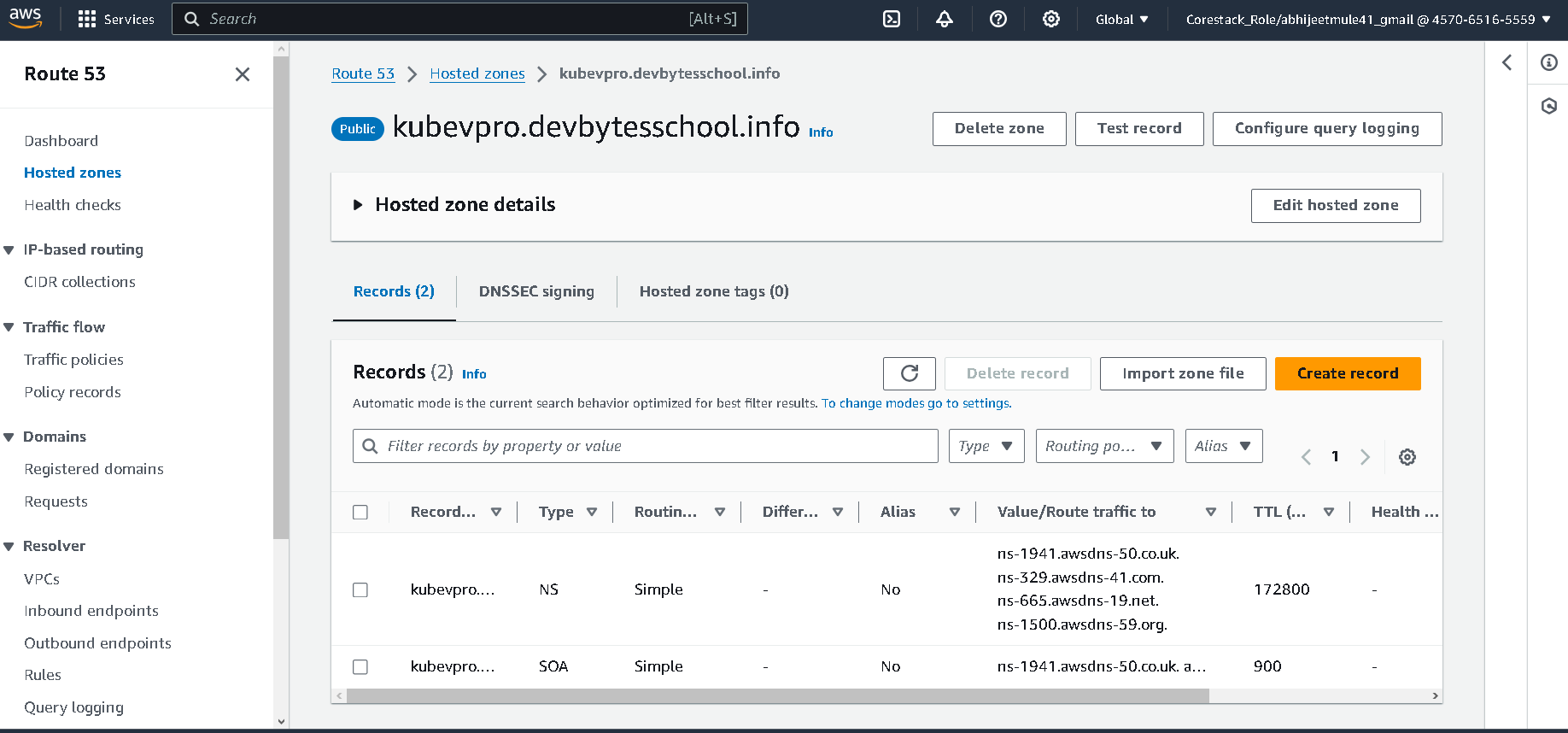
After then click on create hosted zones: It should be your project name along with

Domain name like this kubevpro.devbytesschool.info

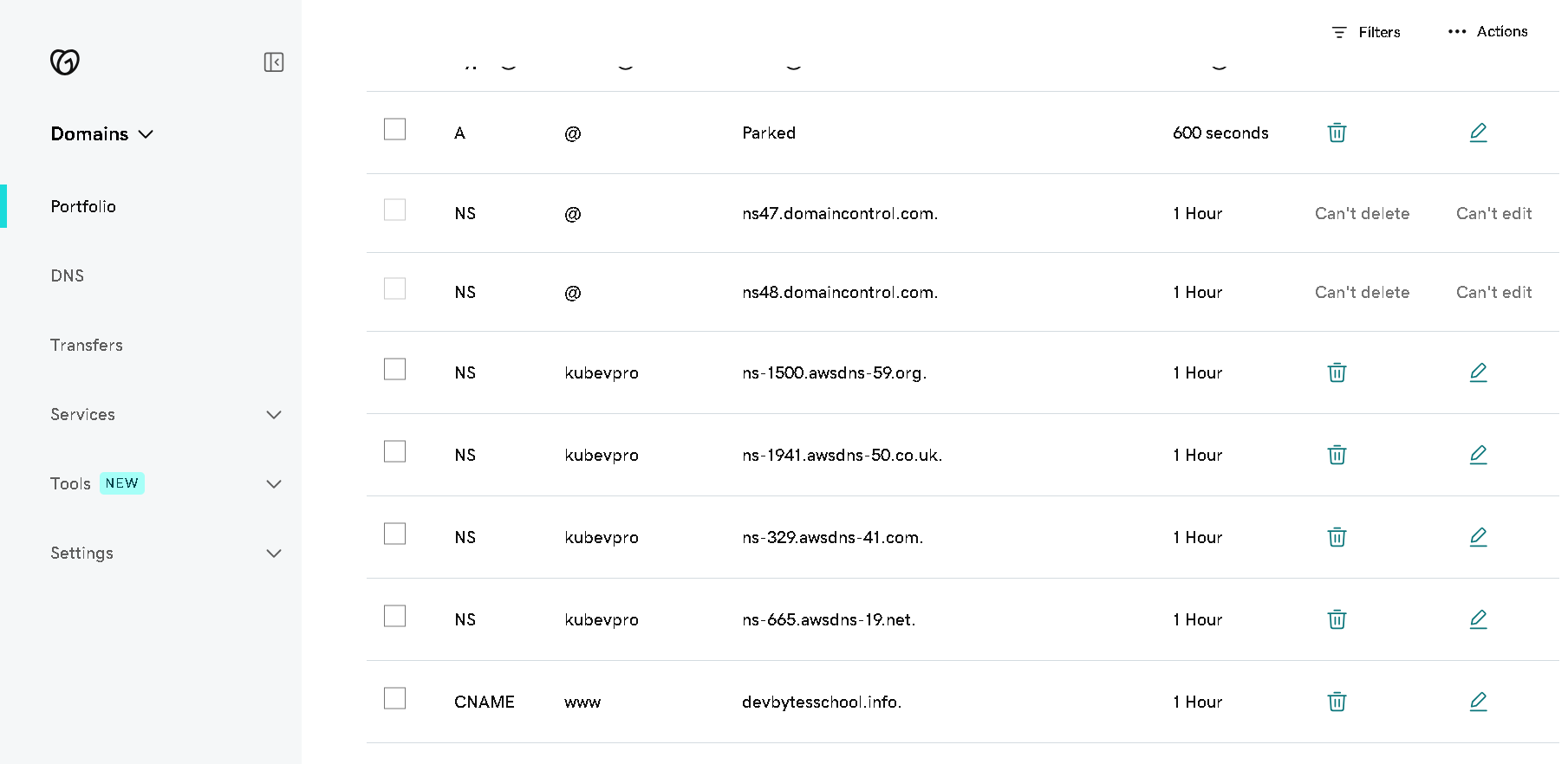
ssh-keygen



Now note down this name servers which will be required to specify in Godaddy subdomain

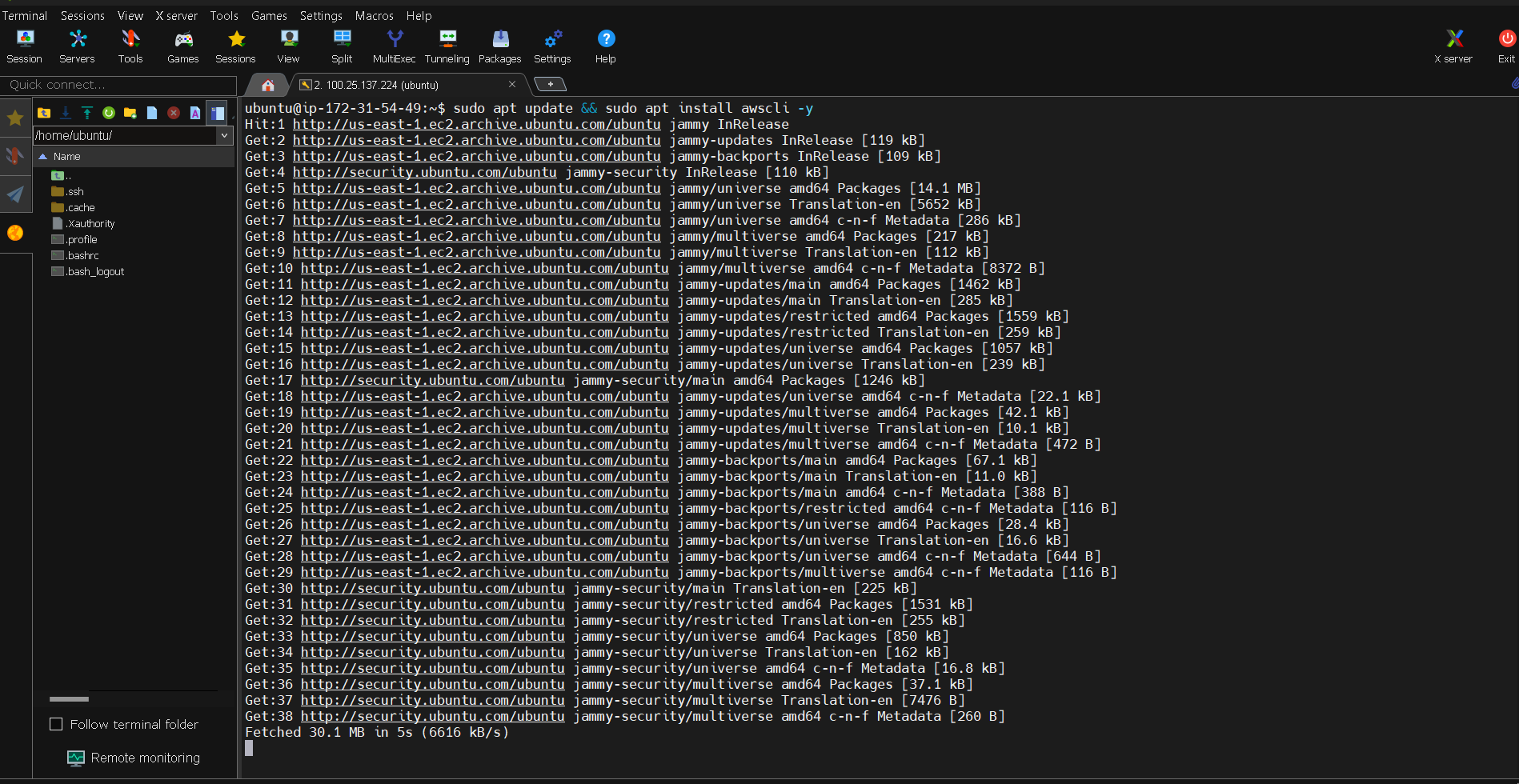


Add this one by one in godaddy dns and click on add new records



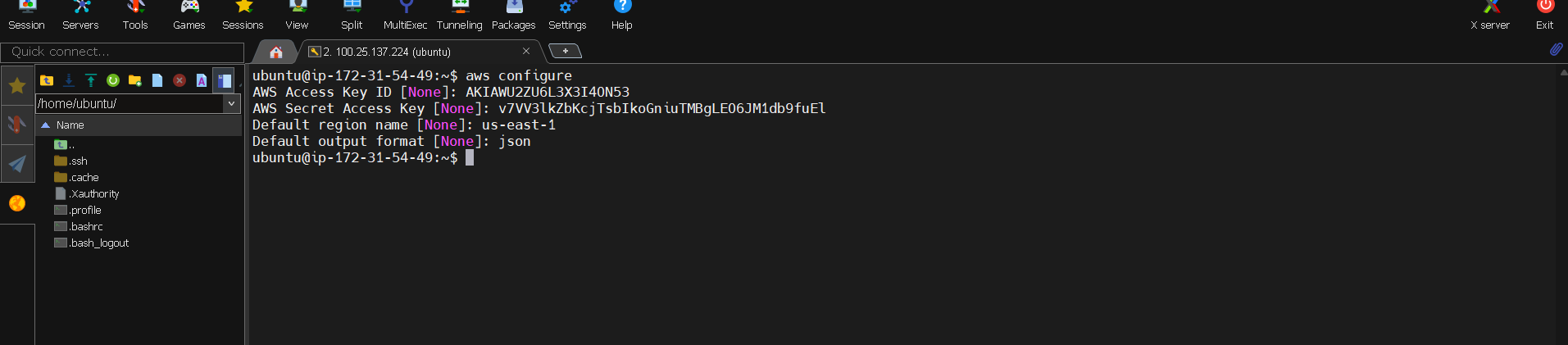
Now S3,Route 53 and I am User configuration part is done now login to ec2 instance and setup aws cli

sudo apt update && sudo apt install awscli -y



Now enter this command aws configure

And add access key ,secret key, region and output format



Now Aws CLI Configured now we need kubectl and kops referred below documentation link

<https://kubernetes.io/docs/tasks/tools/install-kubectl-linux/>

step1: download kubectl

curl -LO "https://dl.k8s.io/release/**$(**curl -L -s https://dl.k8s.io/release/stable.txt**)**/bin/linux/amd64/kubectl"

step2: give executable permission

chmod +x ./kubectl

step3: now move folder in bin

sudo mv kubectl /usr/local/bin

Now Install Kops using this below url documentation

<https://kops.sigs.k8s.io/getting_started/install/>

Download Kops from here

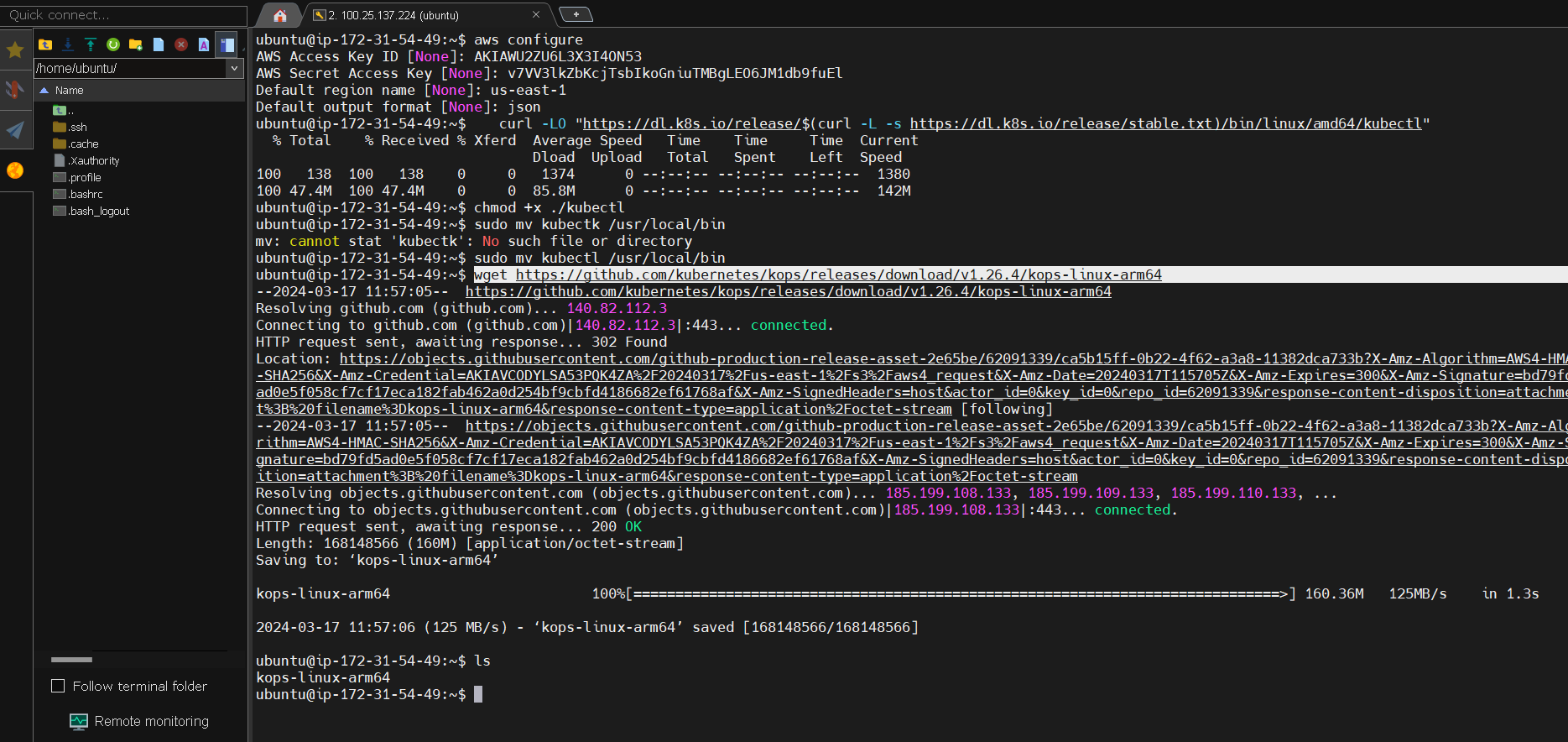
<https://github.com/kubernetes/kops/releases/tag/v1.26.4>

this is the link for download file most stable one

<https://github.com/kubernetes/kops/releases/download/v1.26.4/kops-linux-amd64>

using this command download file

wget <https://github.com/kubernetes/kops/releases/download/v1.26.4/kops-linux-amd64>



Now make this file executable using chmod command

chmod +x kops-linux-amd64

move to bin kops folder

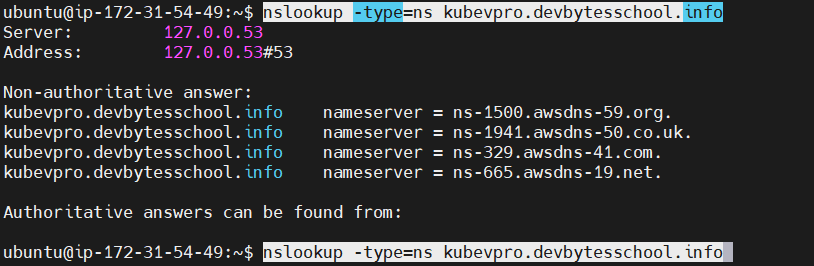
sudo mv kops-linux-amd64 /usr/local/bin/kops

// so now we can enter kops command from any where

Before creating cluster verify domain in properly setup

nslookup -type=ns kubevpro.devbytesschool.info

O/P



//-------------kops command for run kubernates cluster---------------

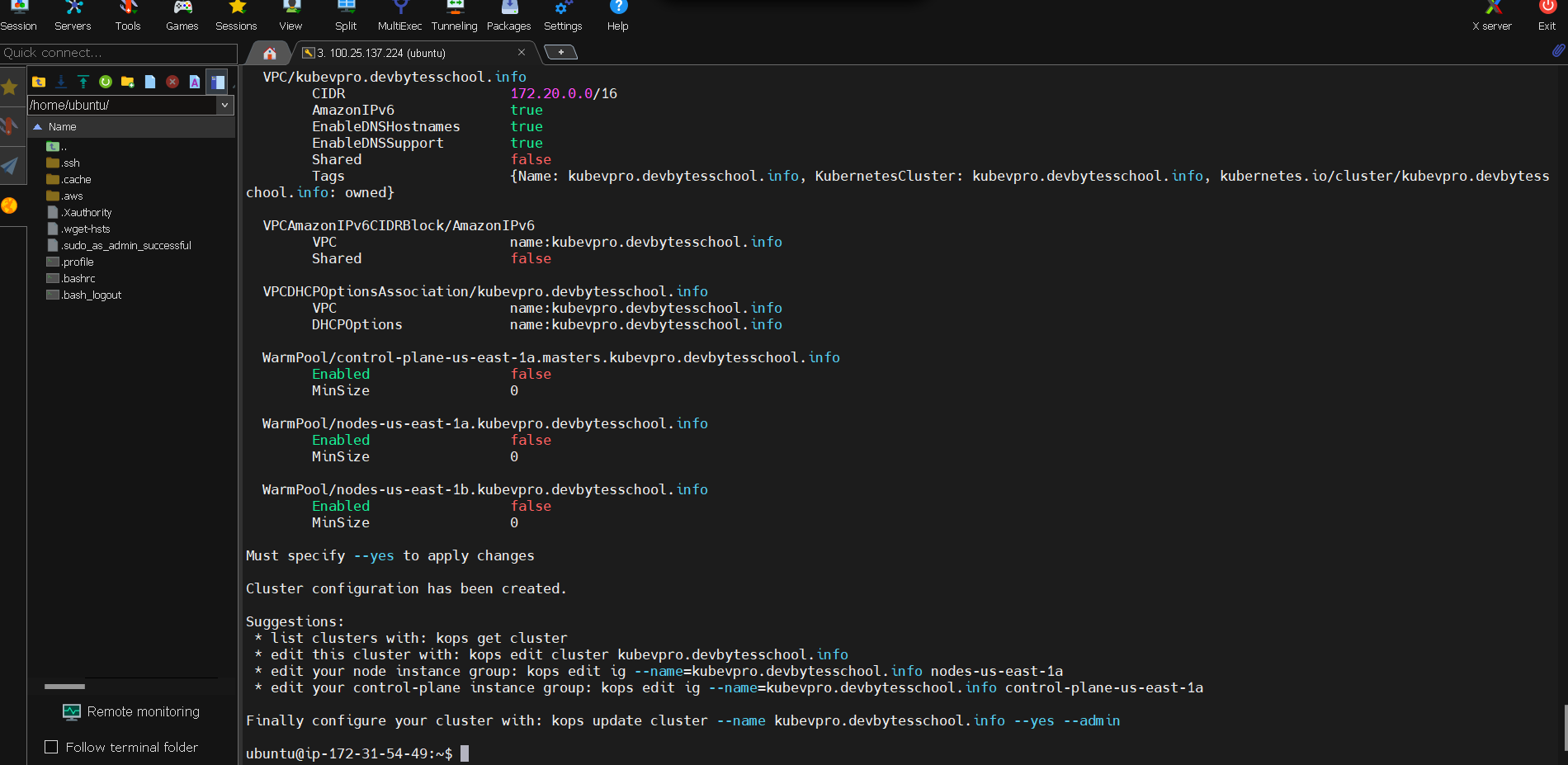
kops create cluster --name=kubevpro.devbytesschool.info \

--state=s3://vprof1le-bucket-kops --zones=us-east-1a,us-east-1b \

--node-count=2 --node-size=t2.medium --master-size=t2.medium --dns-zone=kubevpro.devbytesschool.info \

--node-volume-size=8 --master-volume-size=8

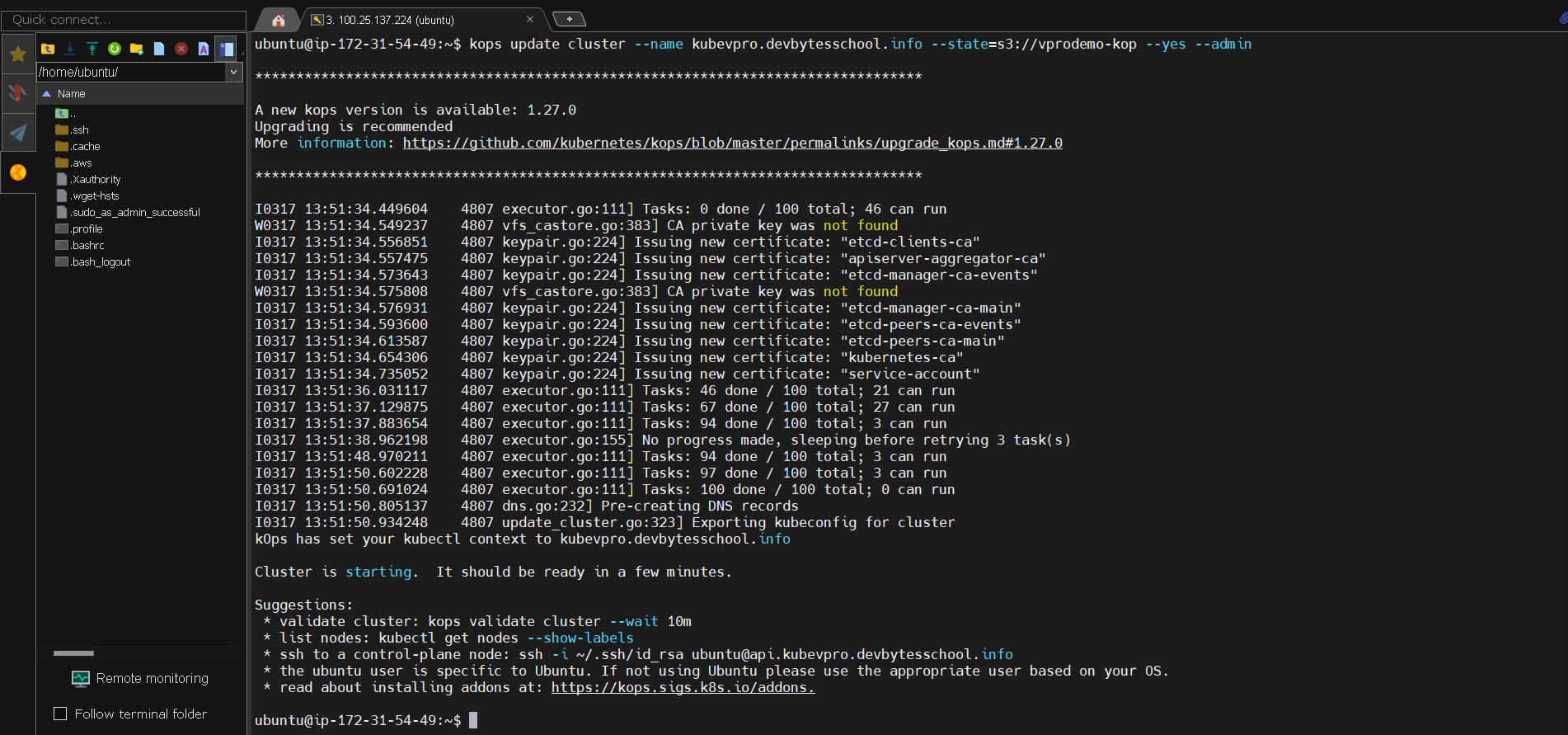
O/p



After then need to enter this command

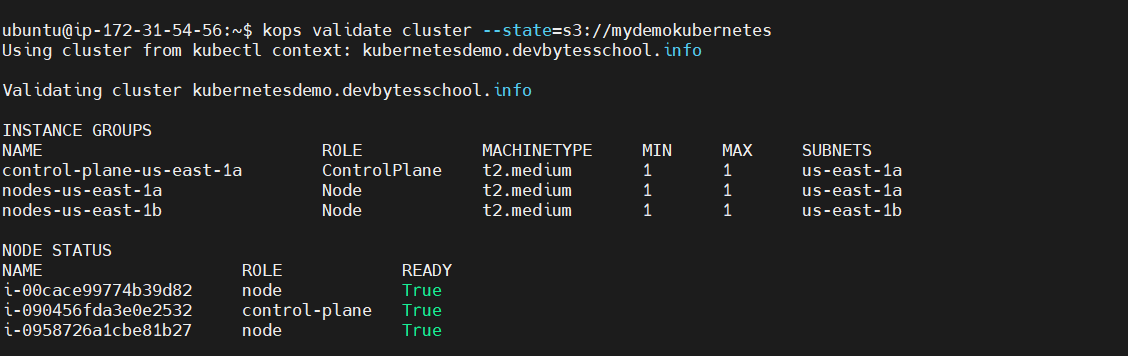
kops update cluster --name kubevpro.devbytesschool.info --state=s3://vprodemo-kop --yes --admin

O/p



After this wait for 15 min validate cluster using belo command

kops validate cluster --state=s3://vprof1le-bucket-kops

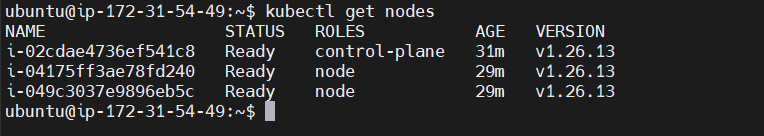


This command to read kubectl file to connect to the kluster

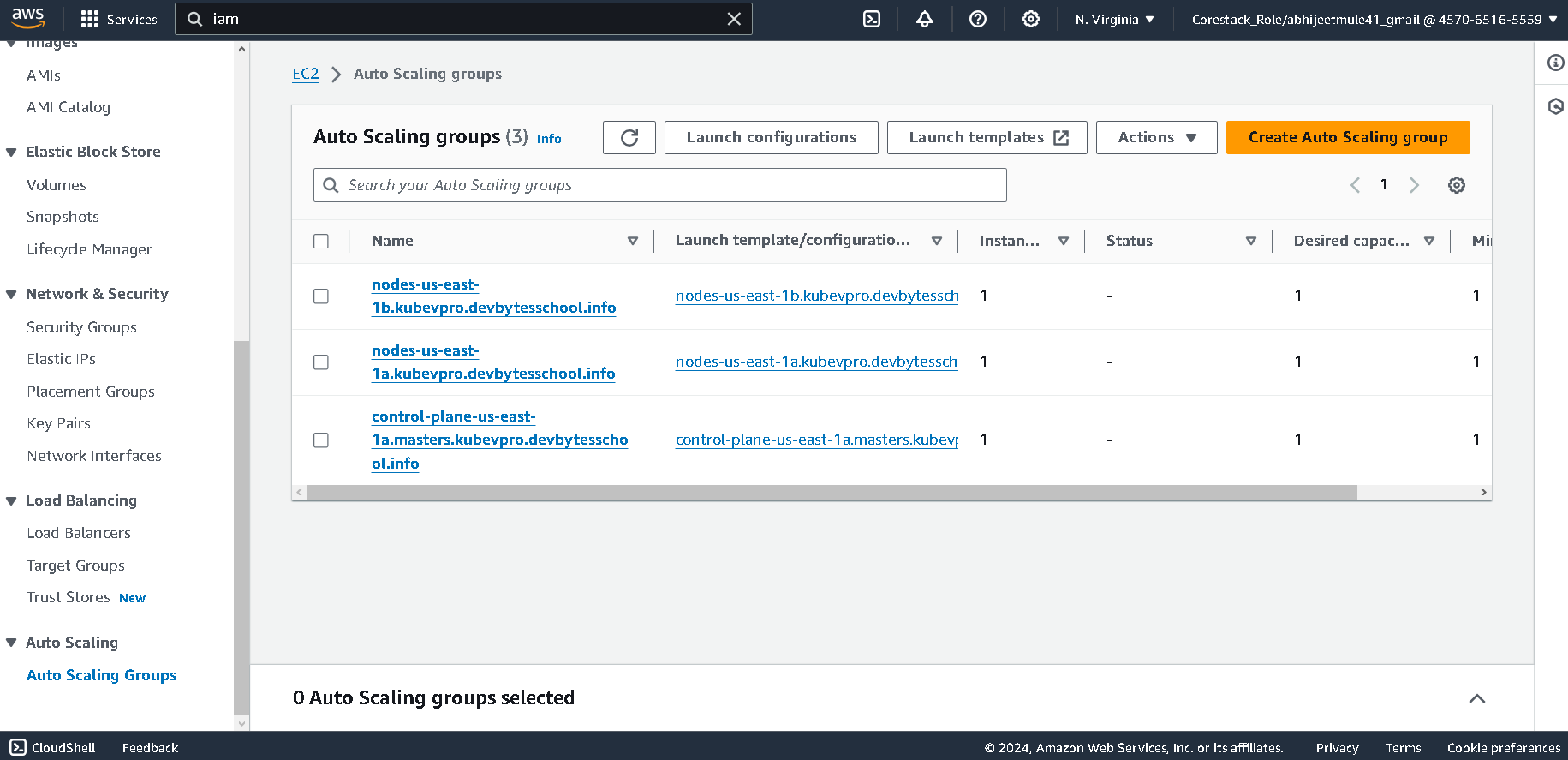
cat /.kube/config

after then check master nodes are created or not using below command

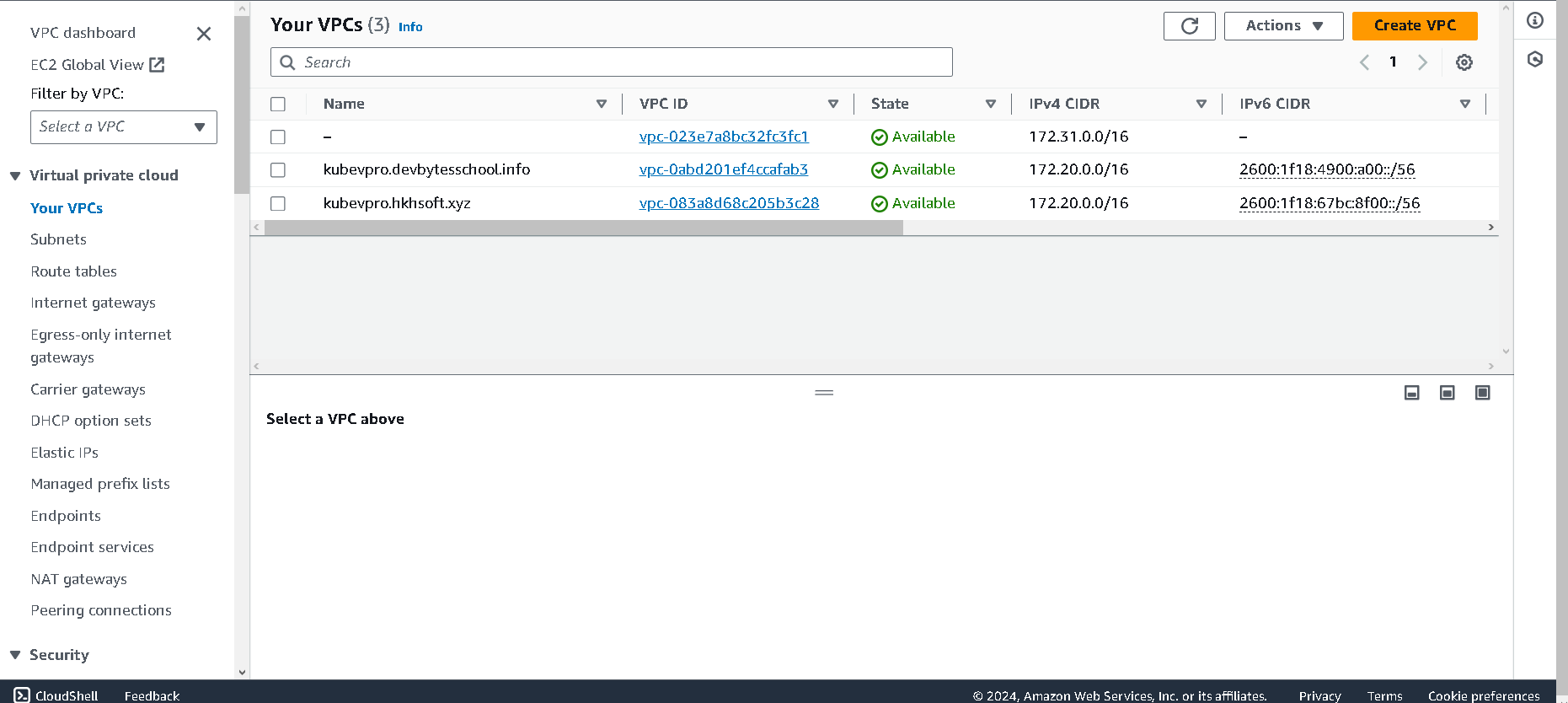
kubectl get nodes



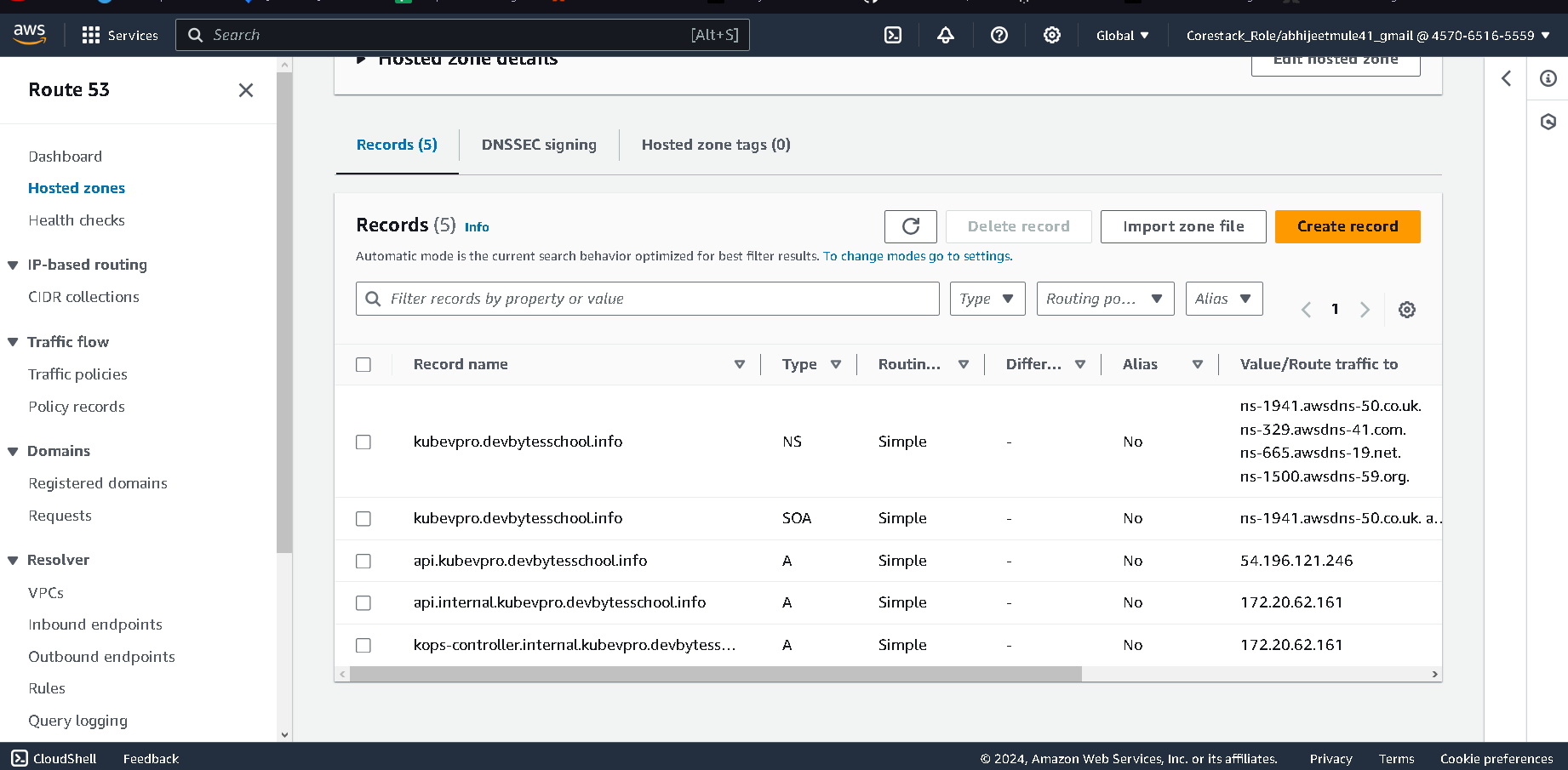
Also check auto scaling group



Also check vpc



Also in Route 53 able to see in hosted zone new record like api.kubevpro….



**If you want to Delete the Cluster the use below commad**

kops delete cluster –name=kubevpro.devbytesschool.info –state=s3://vprofile-kops-state --yes

Install Aws CLi in ubuntu

It seems there might have been a typo or an issue with the command execution. Let's go through the installation process again to ensure everything is correct.

Here’s a step-by-step guide:

1. **Update the package list and install prerequisites:**

sudo apt update

sudo apt install -y curl unzip

1. **Download the AWS CLI installation script:**

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

1. **Unzip the installation script:**

unzip awscliv2.zip

1. **Run the AWS CLI installer:**

Change to the unzipped directory and run the install program:

sudo ./aws/install

Note that you need to use sudo ./aws/install instead of sudo .install.

1. **Verify the installation:**

aws --version