$ ssh -i anubhavnv.pem ubuntu@44.206.247.105

The authenticity of host '44.206.247.105 (44.206.247.105)' can't be established.

ED25519 key fingerprint is SHA256:OAdo+XdknjPm4tVblLMLQtnN9niMagrXnbwVBZ8PGPk.

This host key is known by the following other names/addresses:

~/.ssh/known\_hosts:16: 13.221.22.92

~/.ssh/known\_hosts:23: 3.88.225.212

~/.ssh/known\_hosts:27: 54.86.136.112

~/.ssh/known\_hosts:28: 18.208.155.216

~/.ssh/known\_hosts:29: 3.86.188.19

~/.ssh/known\_hosts:30: 18.207.219.126

~/.ssh/known\_hosts:37: 54.167.45.200

~/.ssh/known\_hosts:41: 3.91.213.109

(2 additional names omitted)

Are you sure you want to continue connecting (yes/no/[fingerprint])? yes

Warning: Permanently added '44.206.247.105' (ED25519) to the list of known hosts.

Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1036-aws x86\_64)

\* Documentation: https://help.ubuntu.com

\* Management: https://landscape.canonical.com

\* Support: https://ubuntu.com/advantage

System information as of Thu Aug 7 13:46:18 UTC 2025

System load: 0.87

Usage of /: 88.9% of 28.89GB

Memory usage: 9%

Swap usage: 0%

Processes: 156

Users logged in: 0

IPv4 address for br-d52074ed2e28: 172.18.0.1

IPv4 address for docker0: 172.17.0.1

IPv4 address for eth0: 172.31.94.180

IPv4 address for weave: 10.44.0.0

=> / is using 88.9% of 28.89GB

\* Ubuntu Pro delivers the most comprehensive open source security and

compliance features.

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Infrastructure is not enabled.

0 updates can be applied immediately.

43 additional security updates can be applied with ESM Infra.

Learn more about enabling ESM Infra service for Ubuntu 20.04 at

https://ubuntu.com/20-04

New release '22.04.5 LTS' available.

Run 'do-release-upgrade' to upgrade to it.

Last login: Thu Aug 7 11:37:03 2025 from 223.181.15.72

ubuntu@kmaster:~$ sudo su

sudo: unable to resolve host kmaster: Name or service not known

root@kmaster:/home/ubuntu# ls

'# Add the official HashiCorp repository' **m1**

Dockerfile pod.yaml

**deployment-manifest** replicaset.yaml

**docker-k8s** **snap**

**llm-api** **terraform.zip**

**llm-inference**

root@kmaster:/home/ubuntu# cd llm-inference/

root@kmaster:/home/ubuntu/llm-inference# ls

Dockerfile deployment.yaml.save service.yaml

app.py **eksctl\_Linux\_amd64.tar.gz**

deployment.yaml requirements.txt

root@kmaster:/home/ubuntu/llm-inference# nano deployment.yaml

root@kmaster:/home/ubuntu/llm-inference# docker build -t anubhav1404/llm-infi:latest .

[+] Building 0.3s (12/12) FINISHED docker:default

=> [internal] load build definition from Dockerfile 0.0s

=> => transferring dockerfile: 333B 0.0s

=> [internal] load metadata for docker.io/library/python:3. 0.2s

=> [auth] library/python:pull token for registry-1.docker.i 0.0s

=> [internal] load .dockerignore 0.0s

=> => transferring context: 2B 0.0s

=> [1/6] FROM docker.io/library/python:3.9-slim@sha256:4b82 0.0s

=> [internal] load build context 0.0s

=> => transferring context: 247B 0.0s

=> CACHED [2/6] WORKDIR /app 0.0s

=> CACHED [3/6] COPY requirements.txt . 0.0s

=> CACHED [4/6] RUN pip install --no-cache-dir -r requireme 0.0s

=> CACHED [5/6] COPY . . 0.0s

=> CACHED [6/6] RUN python -c "from transformers import pip 0.0s

=> exporting to image 0.0s

=> => exporting layers 0.0s

=> => writing image sha256:8236404a070512a29f36340757b88721 0.0s

=> => naming to docker.io/anubhav1404/llm-infi:latest 0.0s

root@kmaster:/home/ubuntu/llm-inference# aws configure

AWS Access Key ID [\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*7WNY]:

AWS Secret Access Key [\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*r/qv]:

Default region name [ap-south-1]: us-east-2

Default output format [json]:

root@kmaster:/home/ubuntu/llm-inference#

root@kmaster:/home/ubuntu/llm-inference# eksctl create cluster \

> --name anubhav-llm \

> --version 1.32 \

> --region ap-south-1 \

> --nodegroup-name llm-nodes \

> --node-type t3.medium \

> --nodes 2 \

> --nodes-min 1 \

> --nodes-max 3 \

> --managed

2025-08-07 13:49:11 [ℹ] eksctl version 0.212.0

2025-08-07 13:49:11 [ℹ] using region ap-south-1

2025-08-07 13:49:12 [ℹ] setting availability zones to [ap-south-1c ap-south-1a ap-south-1b]

2025-08-07 13:49:12 [ℹ] subnets for ap-south-1c - public:192.168.0.0/19 private:192.168.96.0/19

2025-08-07 13:49:12 [ℹ] subnets for ap-south-1a - public:192.168.32.0/19 private:192.168.128.0/19

2025-08-07 13:49:12 [ℹ] subnets for ap-south-1b - public:192.168.64.0/19 private:192.168.160.0/19

2025-08-07 13:49:12 [ℹ] nodegroup "llm-nodes" will use "" [AmazonLinux2023/1.32]

2025-08-07 13:49:12 [ℹ] using Kubernetes version 1.32

2025-08-07 13:49:12 [ℹ] creating EKS cluster "anubhav-llm" in "ap-south-1" region with managed nodes

2025-08-07 13:49:12 [ℹ] will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup

2025-08-07 13:49:12 [ℹ] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=ap-south-1 --cluster=anubhav-llm'

2025-08-07 13:49:12 [ℹ] Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "anubhav-llm" in "ap-south-1"

2025-08-07 13:49:12 [ℹ] CloudWatch logging will not be enabled for cluster "anubhav-llm" in "ap-south-1"

2025-08-07 13:49:12 [ℹ] you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=ap-south-1 --cluster=anubhav-llm'

2025-08-07 13:49:12 [ℹ] default addons kube-proxy, coredns, metrics-server, vpc-cni were not specified, will install them as EKS addons

2025-08-07 13:49:12 [ℹ]

2 sequential tasks: { create cluster control plane "anubhav-llm",

2 sequential sub-tasks: {

2 sequential sub-tasks: {

1 task: { create addons },

wait for control plane to become ready,

},

create managed nodegroup "llm-nodes",

}

}

2025-08-07 13:49:12 [ℹ] building cluster stack "eksctl-anubhav-llm-cluster"

2025-08-07 13:49:14 [ℹ] deploying stack "eksctl-anubhav-llm-cluster"

2025-08-07 13:49:44 [ℹ] waiting for CloudFormation stack "eksctl-anubhav-llm-cluster"

2025-08-07 13:49:45 [✖] unexpected status "ROLLBACK\_IN\_PROGRESS" while waiting for CloudFormation stack "eksctl-anubhav-llm-cluster"

2025-08-07 13:49:45 [✖] unexpected status "ROLLBACK\_IN\_PROGRESS" while waiting for CloudFormation stack "eksctl-anubhav-llm-cluster"

2025-08-07 13:49:45 [ℹ] fetching stack events in attempt to troubleshoot the root cause of the failure

2025-08-07 13:49:45 [!] AWS::EC2::EIP/NATIP: DELETE\_IN\_PROGRESS

2025-08-07 13:49:45 [!] AWS::IAM::Role/ServiceRole: DELETE\_IN\_PROGRESS

2025-08-07 13:49:45 [✖] AWS::IAM::Role/ServiceRole: CREATE\_FAILED – "Resource creation cancelled"

2025-08-07 13:49:45 [✖] AWS::EC2::EIP/NATIP: CREATE\_FAILED – "Resource creation cancelled"

2025-08-07 13:49:45 [✖] AWS::EC2::InternetGateway/InternetGateway: CREATE\_FAILED – "Resource handler returned message: \"The maximum number of internet gateways has been reached. (Service: Ec2, Status Code: 400, Request ID: 67291c90-d98b-444e-b399-84591459c777) (SDK Attempt Count: 1)\" (RequestToken: a3f2d586-32ff-acf1-1bda-be0cd70d26e4, HandlerErrorCode: ServiceLimitExceeded)"

2025-08-07 13:49:45 [✖] AWS::EC2::VPC/VPC: CREATE\_FAILED – "Resource handler returned message: \"The maximum number of VPCs has been reached. (Service: Ec2, Status Code: 400, Request ID: b7068329-a069-4560-9e41-84b6aff7c50b) (SDK Attempt Count: 1)\" (RequestToken: caf17c67-eb8d-ce22-36b6-d6c0b72ead53, HandlerErrorCode: GeneralServiceException)"

2025-08-07 13:49:45 [!] 1 error(s) occurred and cluster hasn't been created properly, you may wish to check CloudFormation console

2025-08-07 13:49:45 [ℹ] to cleanup resources, run 'eksctl delete cluster --region=ap-south-1 --name=anubhav-llm'

2025-08-07 13:49:45 [✖] ResourceNotReady: failed waiting for successful resource state

Error: failed to create cluster "anubhav-llm"

root@kmaster:/home/ubuntu/llm-inference# aws configure

AWS Access Key ID [\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*7WNY]: AKIAT4L7NELAAU5A4CLT

AWS Secret Access Key [\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*r/qv]: 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

Default region name [us-east-2]:

Default output format [json]:

root@kmaster:/home/ubuntu/llm-inference# eksctl create cluster --name anubhav-llm --version 1.32 --region us-east-2 --nodegroup-name llm-nodes --node-type t3.medium --nodes 2 --nodes-min 1 --nodes-max 3 --managed

2025-08-07 13:53:01 [ℹ] eksctl version 0.212.0

2025-08-07 13:53:01 [ℹ] using region us-east-2

2025-08-07 13:53:01 [ℹ] setting availability zones to [us-east-2a us-east-2c us-east-2b]

2025-08-07 13:53:01 [ℹ] subnets for us-east-2a - public:192.168.0.0/19 private:192.168.96.0/19

2025-08-07 13:53:01 [ℹ] subnets for us-east-2c - public:192.168.32.0/19 private:192.168.128.0/19

2025-08-07 13:53:01 [ℹ] subnets for us-east-2b - public:192.168.64.0/19 private:192.168.160.0/19

2025-08-07 13:53:01 [ℹ] nodegroup "llm-nodes" will use "" [AmazonLinux2023/1.32]

2025-08-07 13:53:01 [ℹ] using Kubernetes version 1.32

2025-08-07 13:53:01 [ℹ] creating EKS cluster "anubhav-llm" in "us-east-2" region with managed nodes

2025-08-07 13:53:01 [ℹ] will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup

2025-08-07 13:53:01 [ℹ] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=us-east-2 --cluster=anubhav-llm'

2025-08-07 13:53:01 [ℹ] Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "anubhav-llm" in "us-east-2"

2025-08-07 13:53:01 [ℹ] CloudWatch logging will not be enabled for cluster "anubhav-llm" in "us-east-2"

2025-08-07 13:53:01 [ℹ] you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=us-east-2 --cluster=anubhav-llm'

2025-08-07 13:53:01 [ℹ] default addons metrics-server, vpc-cni, kube-proxy, coredns were not specified, will install them as EKS addons

2025-08-07 13:53:01 [ℹ]

2 sequential tasks: { create cluster control plane "anubhav-llm",

2 sequential sub-tasks: {

2 sequential sub-tasks: {

1 task: { create addons },

wait for control plane to become ready,

},

create managed nodegroup "llm-nodes",

}

}

2025-08-07 13:53:01 [ℹ] building cluster stack "eksctl-anubhav-llm-cluster"

2025-08-07 13:53:01 [ℹ] deploying stack "eksctl-anubhav-llm-cluster"

2025-08-07 13:53:31 [ℹ] waiting for CloudFormation stack "eksctl-anubhav-llm-cluster"

2025-08-07 13:54:01 [ℹ] waiting for CloudFormation stack "eksctl-anubhav-llm-cluster"

2025-08-07 13:55:01 [ℹ] waiting for CloudFormation stack "eksctl-anubhav-llm-cluster"

2025-08-07 13:56:01 [ℹ] waiting for CloudFormation stack "eksctl-anubhav-llm-cluster"

2025-08-07 13:57:02 [ℹ] waiting for CloudFormation stack "eksctl-anubhav-llm-cluster"

2025-08-07 13:58:02 [ℹ] waiting for CloudFormation stack "eksctl-anubhav-llm-cluster"

2025-08-07 13:59:02 [ℹ] waiting for CloudFormation stack "eksctl-anubhav-llm-cluster"

2025-08-07 14:00:02 [ℹ] waiting for CloudFormation stack "eksctl-anubhav-llm-cluster"

2025-08-07 14:01:02 [ℹ] waiting for CloudFormation stack "eksctl-anubhav-llm-cluster"

2025-08-07 14:01:03 [ℹ] creating addon: metrics-server

2025-08-07 14:01:03 [ℹ] successfully created addon: metrics-server

2025-08-07 14:01:04 [!] recommended policies were found for "vpc-cni" addon, but since OIDC is disabled on the cluster, eksctl cannot configure the requested permissions; the recommended way to provide IAM permissions for "vpc-cni" addon is via pod identity associations; after addon creation is completed, add all recommended policies to the config file, under `addon.PodIdentityAssociations`, and run `eksctl update addon`

2025-08-07 14:01:04 [ℹ] creating addon: vpc-cni

2025-08-07 14:01:04 [ℹ] successfully created addon: vpc-cni

2025-08-07 14:01:05 [ℹ] creating addon: kube-proxy

2025-08-07 14:01:05 [ℹ] successfully created addon: kube-proxy

2025-08-07 14:01:05 [ℹ] creating addon: coredns

2025-08-07 14:01:05 [ℹ] successfully created addon: coredns

2025-08-07 14:03:06 [ℹ] building managed nodegroup stack "eksctl-anubhav-llm-nodegroup-llm-nodes"

2025-08-07 14:03:06 [ℹ] deploying stack "eksctl-anubhav-llm-nodegroup-llm-nodes"

2025-08-07 14:03:06 [ℹ] waiting for CloudFormation stack "eksctl-anubhav-llm-nodegroup-llm-nodes"

2025-08-07 14:03:36 [ℹ] waiting for CloudFormation stack "eksctl-anubhav-llm-nodegroup-llm-nodes"

2025-08-07 14:04:08 [ℹ] waiting for CloudFormation stack "eksctl-anubhav-llm-nodegroup-llm-nodes"

2025-08-07 14:05:42 [ℹ] waiting for CloudFormation stack "eksctl-anubhav-llm-nodegroup-llm-nodes"

2025-08-07 14:05:42 [ℹ] waiting for the control plane to become ready

2025-08-07 14:05:42 [✔] saved kubeconfig as "/root/.kube/config"

2025-08-07 14:05:42 [ℹ] no tasks

2025-08-07 14:05:42 [✔] all EKS cluster resources for "anubhav-llm" have been created

2025-08-07 14:05:42 [ℹ] nodegroup "llm-nodes" has 2 node(s)

2025-08-07 14:05:42 [ℹ] node "ip-192-168-26-129.us-east-2.compute.internal" is ready

2025-08-07 14:05:42 [ℹ] node "ip-192-168-90-19.us-east-2.compute.internal" is ready

2025-08-07 14:05:42 [ℹ] waiting for at least 1 node(s) to become ready in "llm-nodes"

2025-08-07 14:05:42 [ℹ] nodegroup "llm-nodes" has 2 node(s)

2025-08-07 14:05:42 [ℹ] node "ip-192-168-26-129.us-east-2.compute.internal" is ready

2025-08-07 14:05:42 [ℹ] node "ip-192-168-90-19.us-east-2.compute.internal" is ready

2025-08-07 14:05:42 [✔] created 1 managed nodegroup(s) in cluster "anubhav-llm"

2025-08-07 14:05:44 [ℹ] kubectl command should work with "/root/.kube/config", try 'kubectl get nodes'

2025-08-07 14:05:44 [✔] EKS cluster "anubhav-llm" in "us-east-2" region is ready

root@kmaster:/home/ubuntu/llm-inference# docker push anubhav1404/llm-infi:latest

The push refers to repository [docker.io/anubhav1404/llm-infi]

32510385fd02: Pushed

15b256406b3c: Pushed

4c124ef1e813: Mounted from anubhav1404/llm-inference

626f17f9904c: Mounted from anubhav1404/llm-inference

aeb343c146ab: Mounted from anubhav1404/llm-inference

c201c43fc3ab: Mounted from anubhav1404/llm-inference

55bb45c46b59: Mounted from anubhav1404/llm-inference

0072d143794a: Mounted from anubhav1404/llm-inference

7cc7fe68eff6: Mounted from anubhav1404/llm-inference

latest: digest: sha256:23f17d9f3ae87d694230a9848a8b6c4a70e21602a76b6335ebc443cb5299e2bb size: 2211

root@kmaster:/home/ubuntu/llm-inference# kubectl apply -f deployment.yaml

deployment.apps/llm-deployment created

root@kmaster:/home/ubuntu/llm-inference# kubectl get pod

NAME READY STATUS RESTARTS AGE

llm-deployment-69d6d7cfd4-d6g7h 0/1 ContainerCreating 0 7s

llm-deployment-69d6d7cfd4-v8r25 0/1 ContainerCreating 0 7s

root@kmaster:/home/ubuntu/llm-inference# kubectl apply -f service.yaml

service/llm-service created

root@kmaster:/home/ubuntu/llm-inference# kubectl get svc

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kubernetes ClusterIP 10.100.0.1 <none> 443/TCP 10m

llm-service LoadBalancer 10.100.77.64 a1f7f213986234c3094e053d55d43fcc-488074274.us-east-2.elb.amazonaws.com 80:32732/TCP 12s

root@kmaster:/home/ubuntu/llm-inference# kubectl get pod

NAME READY STATUS RESTARTS AGE

llm-deployment-69d6d7cfd4-d6g7h 0/1 ContainerCreating 0 44s

llm-deployment-69d6d7cfd4-v8r25 0/1 ContainerCreating 0 44s

root@kmaster:/home/ubuntu/llm-inference# kubectl get svc

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kubernetes ClusterIP 10.100.0.1 <none> 443/TCP 10m

llm-service LoadBalancer 10.100.77.64 a1f7f213986234c3094e053d55d43fcc-488074274.us-east-2.elb.amazonaws.com 80:32732/TCP 30s

root@kmaster:/home/ubuntu/llm-inference# curl http://a1f7f213986234c3094e053d55d43fcc-488074274.us-east-2.elb.amazonaws.com

curl: (52) Empty reply from server

root@kmaster:/home/ubuntu/llm-inference# kubectl get pod

NAME READY STATUS RESTARTS AGE

llm-deployment-69d6d7cfd4-d6g7h 0/1 ContainerCreating 0 85s

llm-deployment-69d6d7cfd4-v8r25 0/1 ContainerCreating 0 85s

root@kmaster:/home/ubuntu/llm-inference# kubectl get pod -o wide

NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

llm-deployment-69d6d7cfd4-d6g7h 0/1 ContainerCreating 0 96s <none> ip-192-168-90-19.us-east-2.compute.internal <none> <none>

llm-deployment-69d6d7cfd4-v8r25 0/1 ContainerCreating 0 96s <none> ip-192-168-26-129.us-east-2.compute.internal <none> <none>

root@kmaster:/home/ubuntu/llm-inference# kubectl watch po llm-deployment-69d6d7cfd4-d6g7h

error: unknown command "watch" for "kubectl"

Did you mean this?

patch

root@kmaster:/home/ubuntu/llm-inference# kubectl get pod

NAME READY STATUS RESTARTS AGE

llm-deployment-69d6d7cfd4-d6g7h 0/1 ContainerCreating 0 2m3s

llm-deployment-69d6d7cfd4-v8r25 0/1 ContainerCreating 0 2m3s

root@kmaster:/home/ubuntu/llm-inference# kubectl describe pod llm-deployment-69d6d7cfd4-d6g7h

Name: llm-deployment-69d6d7cfd4-d6g7h

Namespace: default

Priority: 0

Service Account: default

Node: ip-192-168-90-19.us-east-2.compute.internal/192.168.90.19

Start Time: Thu, 07 Aug 2025 14:07:59 +0000

Labels: app=llm

pod-template-hash=69d6d7cfd4

Annotations: <none>

Status: Pending

IP:

IPs: <none>

Controlled By: ReplicaSet/llm-deployment-69d6d7cfd4

Containers:

llm-container:

Container ID:

Image: anubhav1404/llm-inference:latest

Image ID:

Port: 5000/TCP

Host Port: 0/TCP

State: Waiting

Reason: ContainerCreating

Ready: False

Restart Count: 0

Environment: <none>

Mounts:

/var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-l45h6 (ro)

Conditions:

Type Status

PodReadyToStartContainers False

Initialized True

Ready False

ContainersReady False

PodScheduled True

Volumes:

kube-api-access-l45h6:

Type: Projected (a volume that contains injected data from multiple sources)

TokenExpirationSeconds: 3607

ConfigMapName: kube-root-ca.crt

Optional: false

DownwardAPI: true

QoS Class: BestEffort

Node-Selectors: <none>

Tolerations: node.kubernetes.io/not-ready:NoExecute op=Exists for 300s

node.kubernetes.io/unreachable:NoExecute op=Exists for 300s

Events:

Type Reason Age From Message

---- ------ ---- ---- -------

Normal Scheduled 2m33s default-scheduler Successfully assigned default/llm-deployment-69d6d7cfd4-d6g7h to ip-192-168-90-19.us-east-2.compute.internal

Normal Pulling 2m32s kubelet Pulling image "anubhav1404/llm-inference:latest"

root@kmaster:/home/ubuntu/llm-inference#

root@kmaster:/home/ubuntu/llm-inference# kubectl get pod

NAME READY STATUS RESTARTS AGE

llm-deployment-69d6d7cfd4-d6g7h 1/1 Running 0 3m11s

llm-deployment-69d6d7cfd4-v8r25 0/1 ContainerCreating 0 3m11s

root@kmaster:/home/ubuntu/llm-inference# kubectl get pod

NAME READY STATUS RESTARTS AGE

llm-deployment-69d6d7cfd4-d6g7h 1/1 Running 0 3m58s

llm-deployment-69d6d7cfd4-v8r25 1/1 Running 0 3m58s

root@kmaster:/home/ubuntu/llm-inference# kubectl scale deployment llm-deployment --replicas=4

deployment.apps/llm-deployment scaled

root@kmaster:/home/ubuntu/llm-inference# kubectl scale deployment llm-deployment --replicas=2

deployment.apps/llm-deployment scaled

root@kmaster:/home/ubuntu/llm-inference# kubectl get pod

NAME READY STATUS RESTARTS AGE

llm-deployment-69d6d7cfd4-2qmt2 1/1 Terminating 0 14s

llm-deployment-69d6d7cfd4-8nlgt 1/1 Terminating 0 14s

llm-deployment-69d6d7cfd4-d6g7h 1/1 Running 0 4m16s

llm-deployment-69d6d7cfd4-v8r25 1/1 Running 0 4m16s

root@kmaster:/home/ubuntu/llm-inference# kubectl get svc

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kubernetes ClusterIP 10.100.0.1 <none> 443/TCP 14m

llm-service LoadBalancer 10.100.77.64 a1f7f213986234c3094e053d55d43fcc-488074274.us-east-2.elb.amazonaws.com 80:32732/TCP 4m7s

root@kmaster:/home/ubuntu/llm-inference# curl http://a1f7f213986234c3094e053d55d43fcc-488074274.us-east-2.elb.amazonaws.com

{"status":"LLM Inference Running"}

root@kmaster:/home/ubuntu/llm-inference# curl -X POST http://localhost:5000/generate \

> -H "Content-Type: application/json" \

> -d '{"prompt": "In the future,"}'

curl: (7) Failed to connect to localhost port 5000: Connection refused

root@kmaster:/home/ubuntu/llm-inference# kubectl port-forward deployment/llm-deployment 5000:5000

Forwarding from 127.0.0.1:5000 -> 5000

Forwarding from [::1]:5000 -> 5000

^Croot@kmaster:/home/ubuntu/llm-inferencecurl -X POST http://localhost:5000/generate -H "Content-Type: application/json" -d '{"prompt": "In the future,"}'

curl: (7) Failed to connect to localhost port 5000: Connection refused

root@kmaster:/home/ubuntu/llm-inference# kubectl port-forward deployment/llm-deployment 5000:5000

Forwarding from 127.0.0.1:5000 -> 5000

Forwarding from [::1]:5000 -> 5000

^Croot@kmaster:/home/ubuntu/llm-inferencekubectl get pod

NAME READY STATUS RESTARTS AGE

llm-deployment-69d6d7cfd4-d6g7h 1/1 Running 0 7m26s

llm-deployment-69d6d7cfd4-v8r25 1/1 Running 0 7m26s

root@kmaster:/home/ubuntu/llm-inference# kubectl get po -o wide

NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

llm-deployment-69d6d7cfd4-d6g7h 1/1 Running 0 7m39s 192.168.68.112 ip-192-168-90-19.us-east-2.compute.internal <none> <none>

llm-deployment-69d6d7cfd4-v8r25 1/1 Running 0 7m39s 192.168.2.248 ip-192-168-26-129.us-east-2.compute.internal <none> <none>

root@kmaster:/home/ubuntu/llm-inference# kubectl get deplo

error: the server doesn't have a resource type "deplo"

root@kmaster:/home/ubuntu/llm-inference# kubectl get deployment

NAME READY UP-TO-DATE AVAILABLE AGE

llm-deployment 2/2 2 2 7m54s

root@kmaster:/home/ubuntu/llm-inference# kubectl get replicaset

NAME DESIRED CURRENT READY AGE

llm-deployment-69d6d7cfd4 2 2 2 8m6s

root@kmaster:/home/ubuntu/llm-inference# nano deployment.yaml

root@kmaster:/home/ubuntu/llm-inference# nano deployment.yaml

root@kmaster:/home/ubuntu/llm-inference# kubectl scale deployment llm-deployment --replicas=4

deployment.apps/llm-deployment scaled

root@kmaster:/home/ubuntu/llm-inference# kubectl get po

NAME READY STATUS RESTARTS AGE

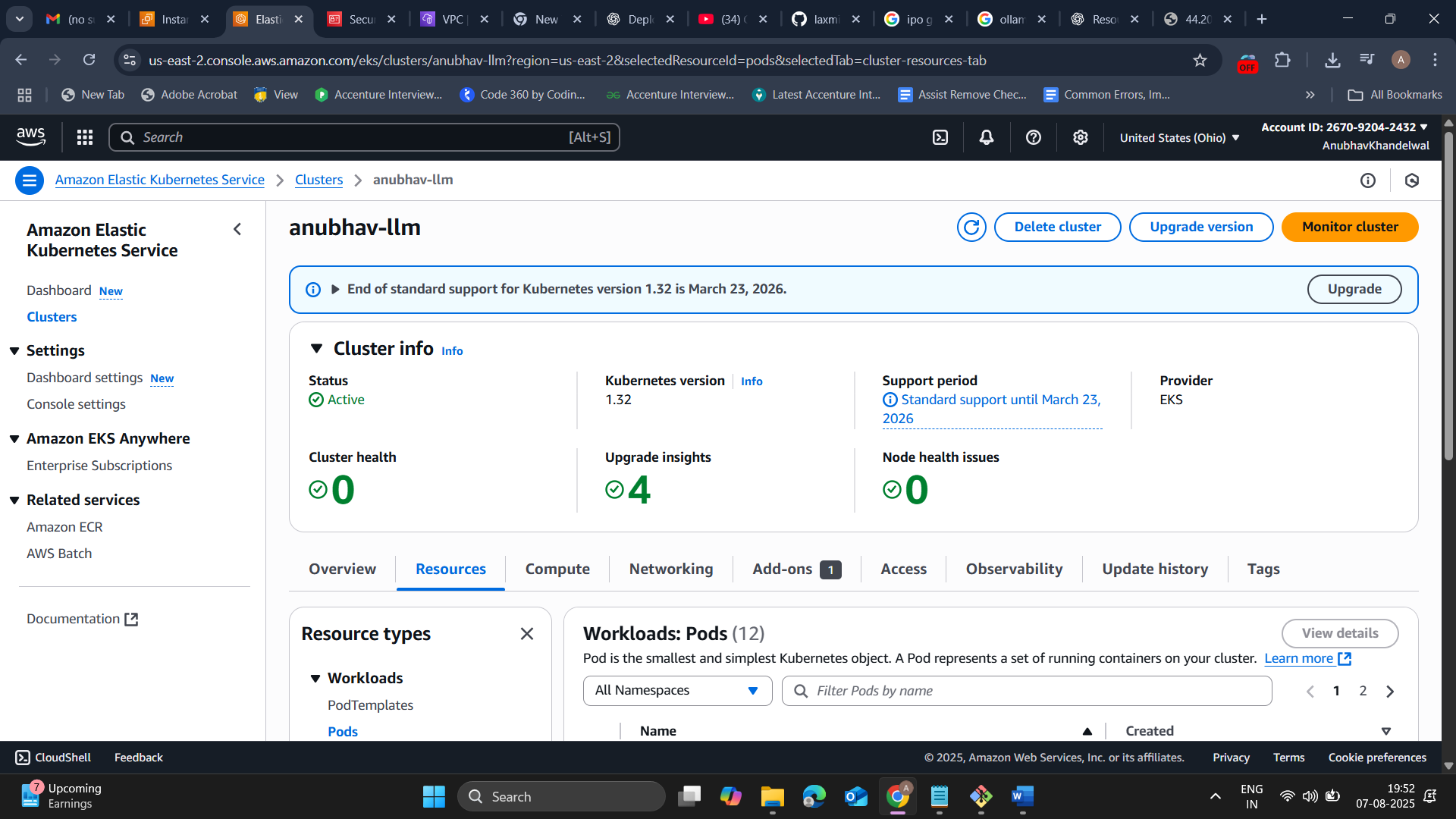
llm-deployment-69d6d7cfd4-d6g7h 1/1 Running 0 11m

llm-deployment-69d6d7cfd4-h4w82 1/1 Running 0 5s

llm-deployment-69d6d7cfd4-v8r25 1/1 Running 0 11m

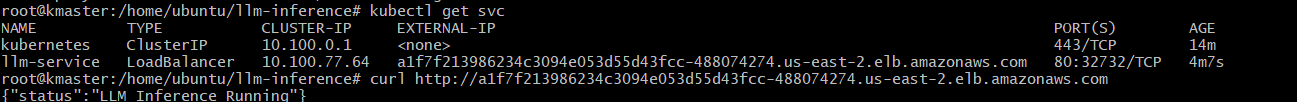
llm-deployment-69d6d7cfd4-vln5p 1/1 Running 0 5s

root@kmaster:/home/ubuntu/llm-inference#



A screenshot of a computer

AI-generated content may be incorrect.



root@kmaster:/home/ubuntu/llm-inference# aws ecr get-login-password --region us-east-2 | docker login --username AWS --password-stdin 267092042432.dkr.ecr.us-east-2.amazonaws.com

WARNING! Your credentials are stored unencrypted in '/root/.docker/config.json'.

Configure a credential helper to remove this warning. See

https://docs.docker.com/go/credential-store/

Login Succeeded

root@kmaster:/home/ubuntu/llm-inference# docker build -t anubhav1404/llm-infi .

[+] Building 33.8s (12/12) FINISHED docker:default

=> [internal] load build definition from Dockerfile 0.0s

=> => transferring dockerfile: 333B 0.0s

=> [internal] load metadata for docker.io/library/python:3.9-slim 0.1s

=> [auth] library/python:pull token for registry-1.docker.io 0.0s

=> [internal] load .dockerignore 0.0s

=> => transferring context: 2B 0.0s

=> [1/6] FROM docker.io/library/python:3.9-slim@sha256:4b826e2ca2191c24a03bdbf0c342a73c6ee4ce929eac5ebc47fef78f66b839d6 0.0s

=> [internal] load build context 0.0s

=> => transferring context: 598B 0.0s

=> CACHED [2/6] WORKDIR /app 0.0s

=> CACHED [3/6] COPY requirements.txt . 0.0s

=> CACHED [4/6] RUN pip install --no-cache-dir -r requirements.txt 0.0s

=> [5/6] COPY . . 0.8s

=> [6/6] RUN python -c "from transformers import pipeline; pipeline('text-generation', model='distilgpt2')" 28.9s

=> exporting to image 3.7s

=> => exporting layers 3.7s

=> => writing image sha256:3f43b44706f2b81843138eff4a16d5eb6538fac5163d5ea1b2608a5466297f05 0.0s

=> => naming to docker.io/anubhav1404/llm-infi 0.0s

root@kmaster:/home/ubuntu/llm-inference# docker tag anubhav1404/llm-infi:latest 267092042432.dkr.ecr.us-east-2.amazonaws.com/anubhav1404/llm-infi:latest

root@kmaster:/home/ubuntu/llm-inference# docker push 267092042432.dkr.ecr.us-east-2.amazonaws.com/anubhav1404/llm-infi:latest

The push refers to repository [267092042432.dkr.ecr.us-east-2.amazonaws.com/anubhav1404/llm-infi]

20a3103817f7: Pushed

25fbbcb42322: Pushed

4c124ef1e813: Pushed

626f17f9904c: Pushed

aeb343c146ab: Pushed

c201c43fc3ab: Pushed

55bb45c46b59: Pushed

0072d143794a: Pushed

7cc7fe68eff6: Pushed

latest: digest: sha256:ab7fb2643ebac2349f3c5a6561bf90a3bc5624f1b532649052124d0bd264c6c0 size: 2211

root@kmaster:/home/ubuntu/llm-inference#