

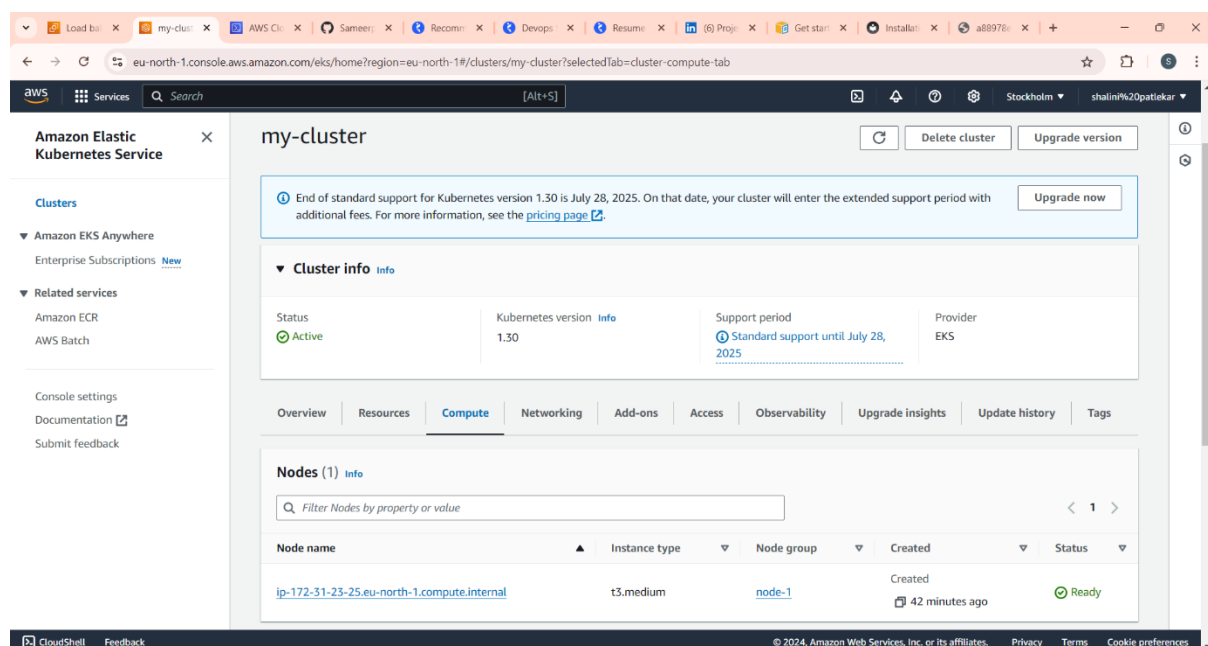
# Kubernetes Deployment of Flask & MongoDB on AWS EKS

## Description:

Deployed a Python Flask application interacting with a MongoDB database on AWS Elastic Kubernetes Service (EKS). The project involved setting up Kubernetes resources, implementing autoscaling, and securing the database connection. Additionally, used NGINX Ingress Controller to expose the Flask application externally.

## Steps to deploy flask k8S application

Step 1 :Create an EKS Cluster and Add a Node Group



Step 2: Connect to the EKS Cluster Using AWS CloudShell

Command:

```
aws eks update-kubeconfig --region <region-code> --name my-cluster
```

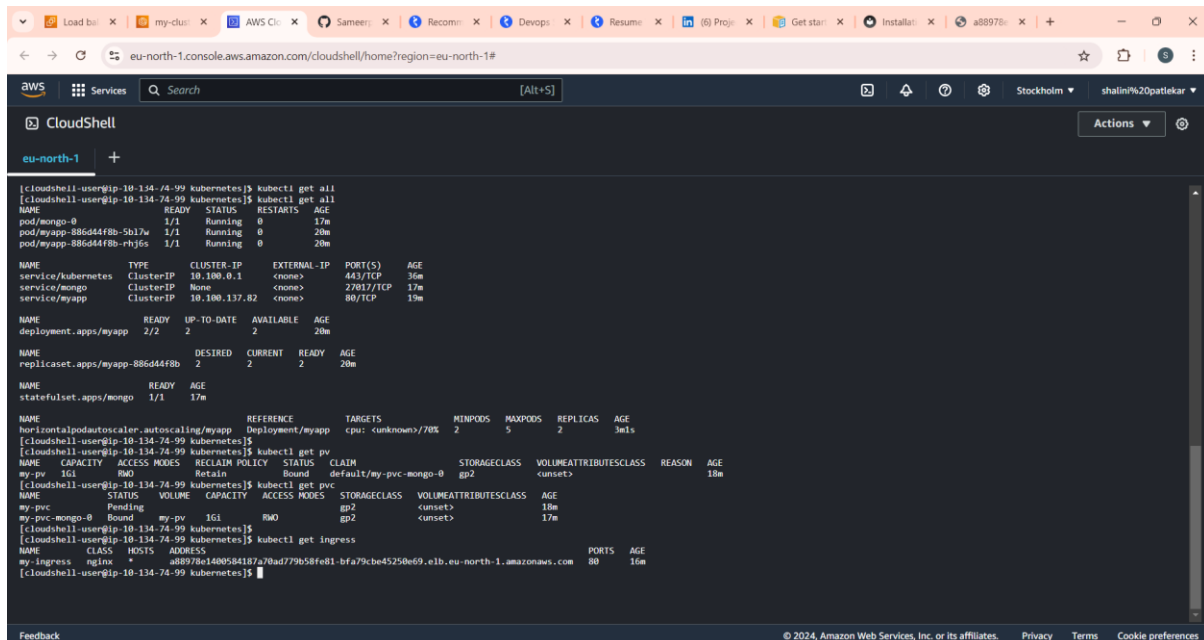
step 3:Clone the Git Repository

command:

```
git clone https://github.com/Sameerpatlekar/python-flask-kubernetes.git
cd python-flask-kubernetes/
```

step : 5 Apply Kubernetes Manifest Files

kubectl apply -f <manifest-file.yaml>



```
[cloudshell-user@ip-10-134-74-99 ~]$ kubectl get all
[cloudshell-user@ip-10-134-74-99 ~]$ kubectl get all
NAME                                READY    STATUS    RESTARTS   AGE
pod/mongo-0                         1/1     Running   0           17m
pod/myapp-886d44fb-b517w            1/1     Running   0           20m
pod/myapp-886d44fb-rhj6s            1/1     Running   0           20m

NAME                                TYPE                CLUSTER-IP      EXTERNAL-IP    PORT(S)          AGE
service/kubernetes                 ClusterIP           10.100.0.1      <none>         443/TCP          36m
service/mongo                      ClusterIP           None            <none>         27017/TCP        17m
service/myapp                      ClusterIP           10.100.137.82   <none>         80/TCP           19m

NAME                                READY    UP-TO-DATE    AVAILABLE    AGE
deployment.apps/myapp              2/2      2             2            20m

NAME                                DESIRED    CURRENT    READY    AGE
replicaset.apps/myapp-886d44fb     2          2          2        20m

NAME                                READY    AGE
statefulset.apps/mongo             1/1      17m

NAME                                REFERENCE    TARGETS    MINPODS    MAXPODS    REPLICAS    AGE
horizontalpodautoscaler.autoscaling/myapp  Deployment/myapp  cpu: <unknown>/70%  2          5          2            3m

[cloudshell-user@ip-10-134-74-99 ~]$ kubectl get pv
NAME     CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM   STORAGECLASS   VOLUMEATTRIBUTESCLASS   REASON   AGE
my-pv    1Gi        RWO            Retain           Bound   default/my-pvc-mongo-0  gp2           <unset>                18m

[cloudshell-user@ip-10-134-74-99 ~]$ kubectl get pvc
NAME     STATUS   VOLUME   CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
my-pvc   Pending  <none>   1Gi        RWO            gp2            <unset>                18m
my-pvc-mongo-0  Bound   my-pv    1Gi        RWO            gp2            <unset>                17m

[cloudshell-user@ip-10-134-74-99 ~]$ kubectl get ingress
NAME     CLASS    HOSTS                                ADDRESS     PORTS     AGE
my-ingress  nginx    a88978e1400584187a70ad779b58fe81-bfa79cbe45250e69.elb.eu-north-1.amazonaws.com  80        16m

[cloudshell-user@ip-10-134-74-99 ~]$
```

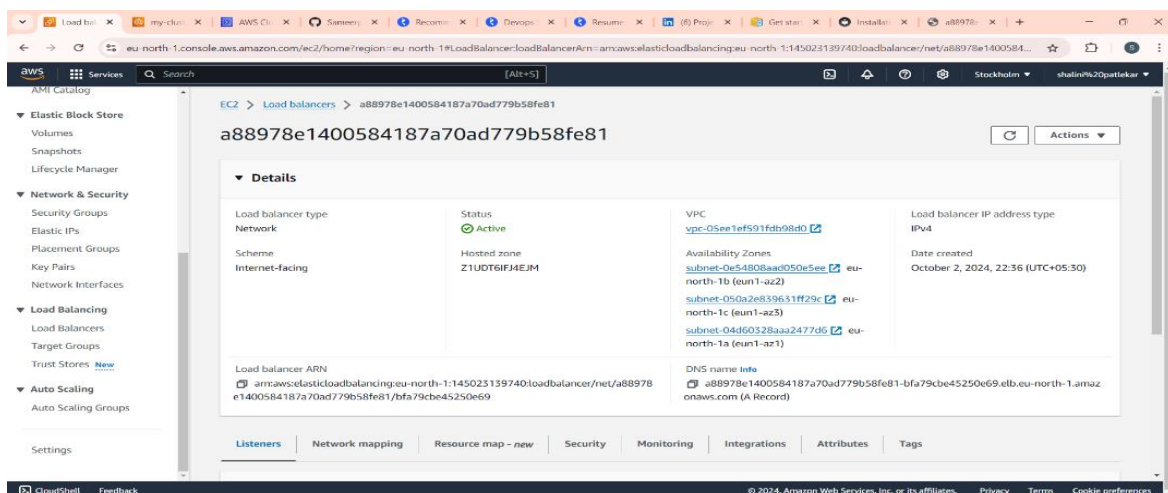
Step 5 :Configure NGINX Ingress Controller and Follow the official NGINX Ingress Controller documentation for AWS: [NGINX Ingress on AWS](https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.11.2/deploy/static/provider/aws/deploy.yaml).

kubectl apply -f <https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.11.2/deploy/static/provider/aws/deploy.yaml>

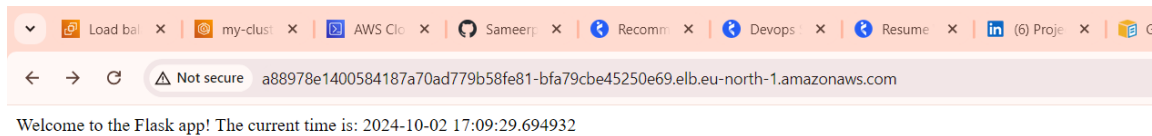
Verify that the NGINX Ingress Controller is running:

kubectl get pods -n ingress-nginx

step 6 : Once the NGINX Ingress Controller is installed and the Ingress resource applied, an NLB will be automatically created. You can check the status of the NLB in the **AWS Console**



Step 7: Confirm that the Ingress Controller is correctly forwarding traffic by visiting the DNS of the NLB.



Curl the NLB DNS

```
[cloudshell-user@ip-10-134-74-99 kubernetes]$ curl -X POST -H "Content-Type: application/json" -d '{"sampleKey":"sampleValue"}' http://a88978e1400584187a70ad779b58fe81-bfa79cbe45250e69.elb.eu-north-1.amazonaws.com/
{"status": "Data inserted"}
[cloudshell-user@ip-10-134-74-99 kubernetes]$ curl http://a88978e1400584187a70ad779b58fe81-bfa79cbe45250e69.elb.eu-north-1.amazonaws.com/
[{"sampleKey": "sampleValue"}]
[cloudshell-user@ip-10-134-74-99 kubernetes]$ curl http://a88978e1400584187a70ad779b58fe81-bfa79cbe45250e69.elb.eu-north-1.amazonaws.com
Welcome to the Flask app! The current time is: 2024-10-02 17:13:35.609786[cloudshell-user@ip-10-134-74-99 kubernetes]$
```

