**EKS Cluster Micro services Deployment - Proof of Concept (POC)**

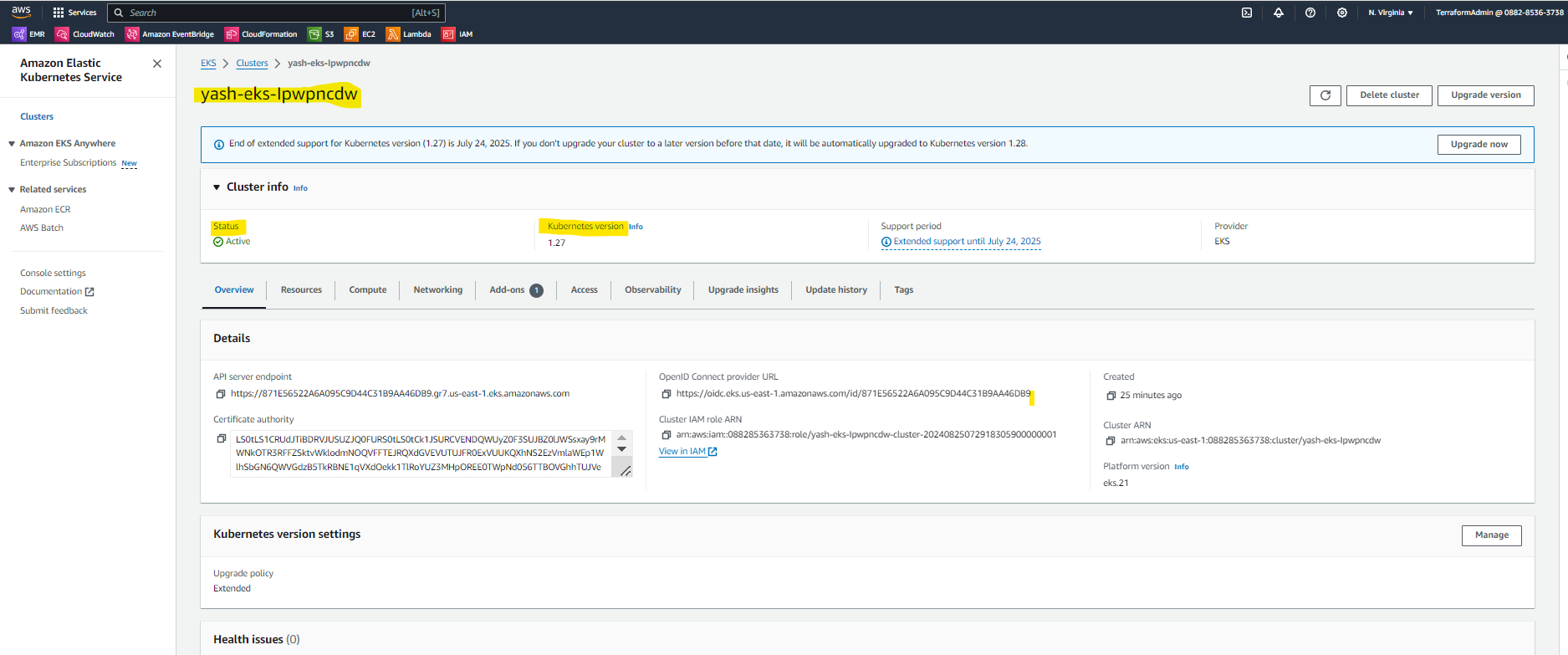
**Objective:**

To demonstrate the deployment and management of microservices on an Amazon EKS (Elastic Kubernetes Service) cluster using Kubernetes manifests and kubectl.

**Prerequisites:**

* An EKS cluster deployed using Terraform.
* kubectl installed on your local machine.
* Access to your EKS cluster and proper permissions.

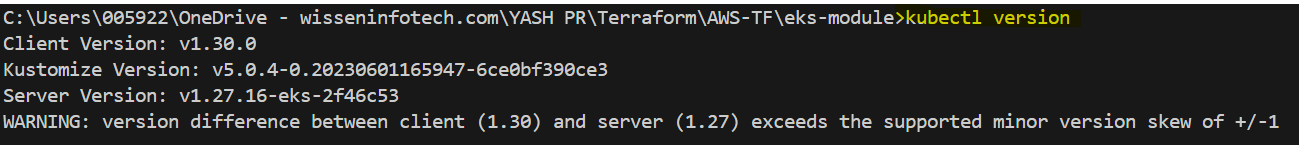
Deployed EKS Cluster with terraform



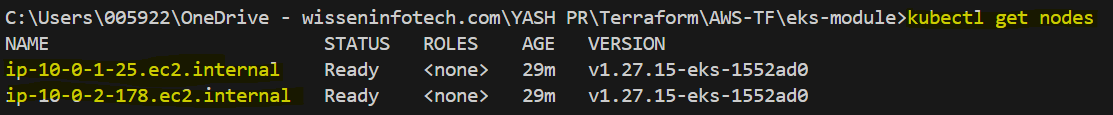
To access and interact to this cluster through **kubectl.**

Install the kubectl in cmd and check the version.

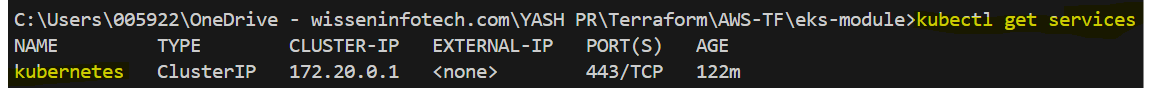
$kubectl version



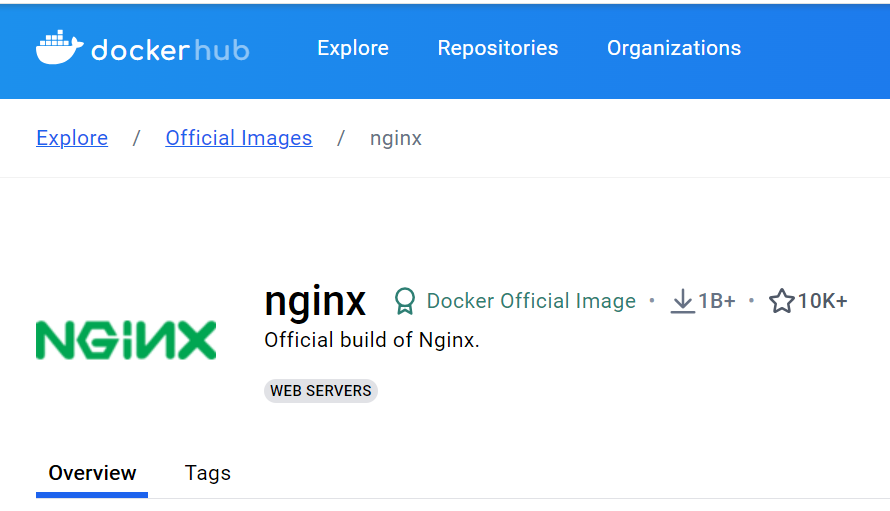
To check running nodes $kubectl get nodes



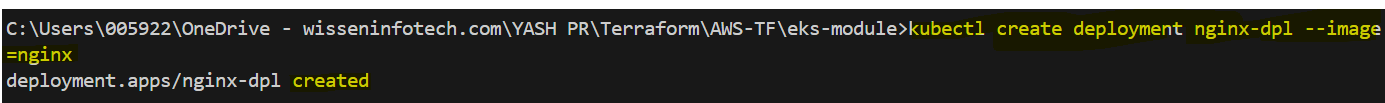
To check services $kubectl get services



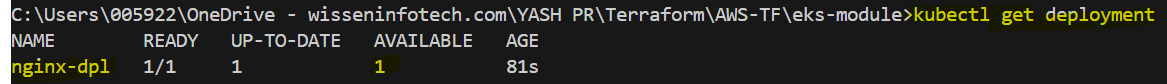
To install micro applications: pod: **create deployment** in eks by pulling the **image** from **docker hub.**

****

$kubectl create deployment nginx-dpl --image=nginx

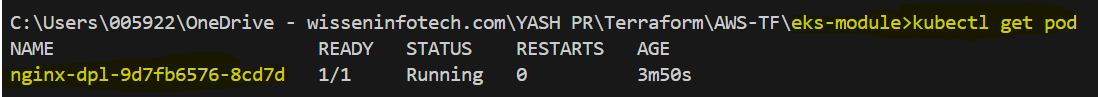


To check deployment $kubectl get deployment



If want to check the pod status $kubectl describe pod nginx-dpl

To check pods $kubectl get pod



Deploying Microservices

1. **Deploy Nginx**

**Create the Nginx Configuration File:** Save the following configuration as nginx.yaml:

apiVersion: v1

kind: ConfigMap

metadata:

name: nginx-project

data:

index.html: |

<HTML>

<HEAD><TITLE>K8s nginx Machine 1</TITLE></HEAD>

<BODY>

<H1>Welcome to machine 1</H1>

</BODY></HTML

---

apiVersion: v1

kind: Service

metadata:

name: nginx-project

spec:

type: LoadBalancer

ports:

- port: 80

- targetport: 8080

selector:

app: nginx-project

---

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-project

spec:

replicas: 4

selector:

matchLabels:

app: nginx-project

template:

metadata:

labels:

app: nginx-project

spec:

containers:

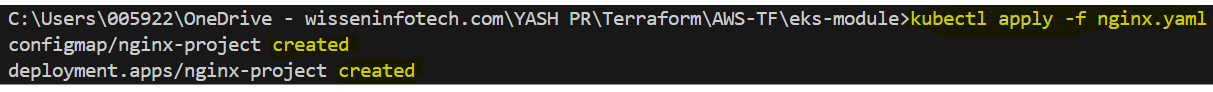
- name: nginx

image: nginx:1.17.3

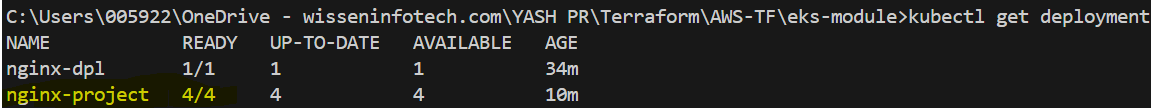
ports:

- containerPort: 80

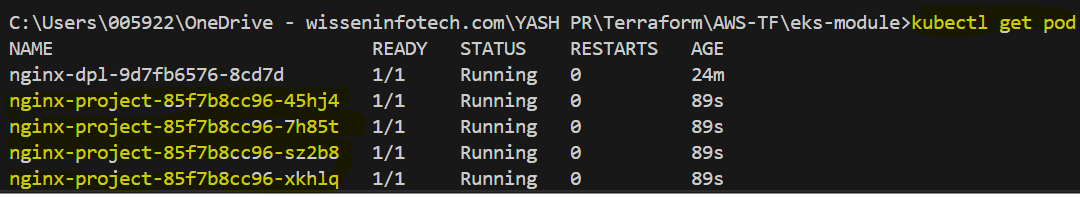
To deploy the nginx.yaml file $kubectl apply –f nginx.yaml



To check deployments $kubectl get deployment



Check pod for new nginx deployment $kubectl get pod



1. **Deploy MongoDB**

**Create the MongoDB Configuration File:** Save the following configuration as mongo.yaml:

apiVersion: apps/v1

kind: Deployment

metadata:

name: mongodb-deployment

labels:

app: mongodb

spec:

replicas: 1

selector:

matchLabels:

app: mongodb

template:

metadata:

labels:

app: mongodb

spec:

containers:

- name: mongodb

image: mongo

ports:

- containerPort: 27017

env:

- name: MONGO\_INITDB\_ROOT\_USERNAME

valueFrom:

secretKeyRef:

name: mongodb-secret

key: mongo-root-username

- name: MONGO\_INITDB\_ROOT\_PASSWORD

valueFrom:

secretKeyRef:

name: mongodb-secret

key: mongo-root-password

---

apiVersion: v1

kind: Service

metadata:

name: mongodb-service

spec:

selector:

app: mongodb

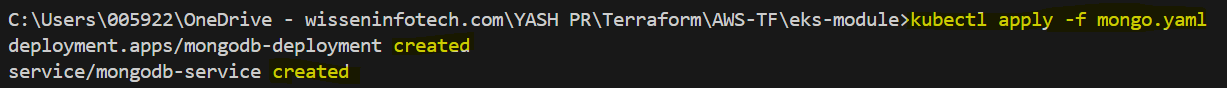
ports:

- protocol: TCP

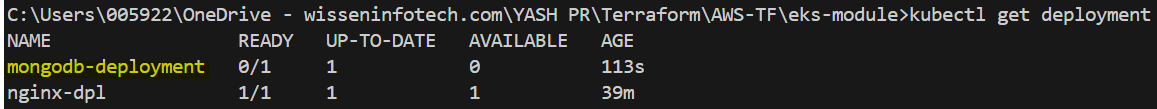
port: 27017

targetPort: 27017

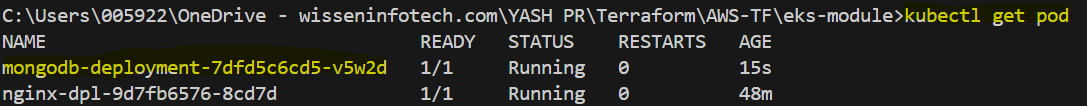
Deploy mongo: $kubectl apply –f mongo.yaml



Check deployment $kubectl get deployment



Check running pods $kubectl get pod



Delete deployment $kubectl delete deployment “Name”

