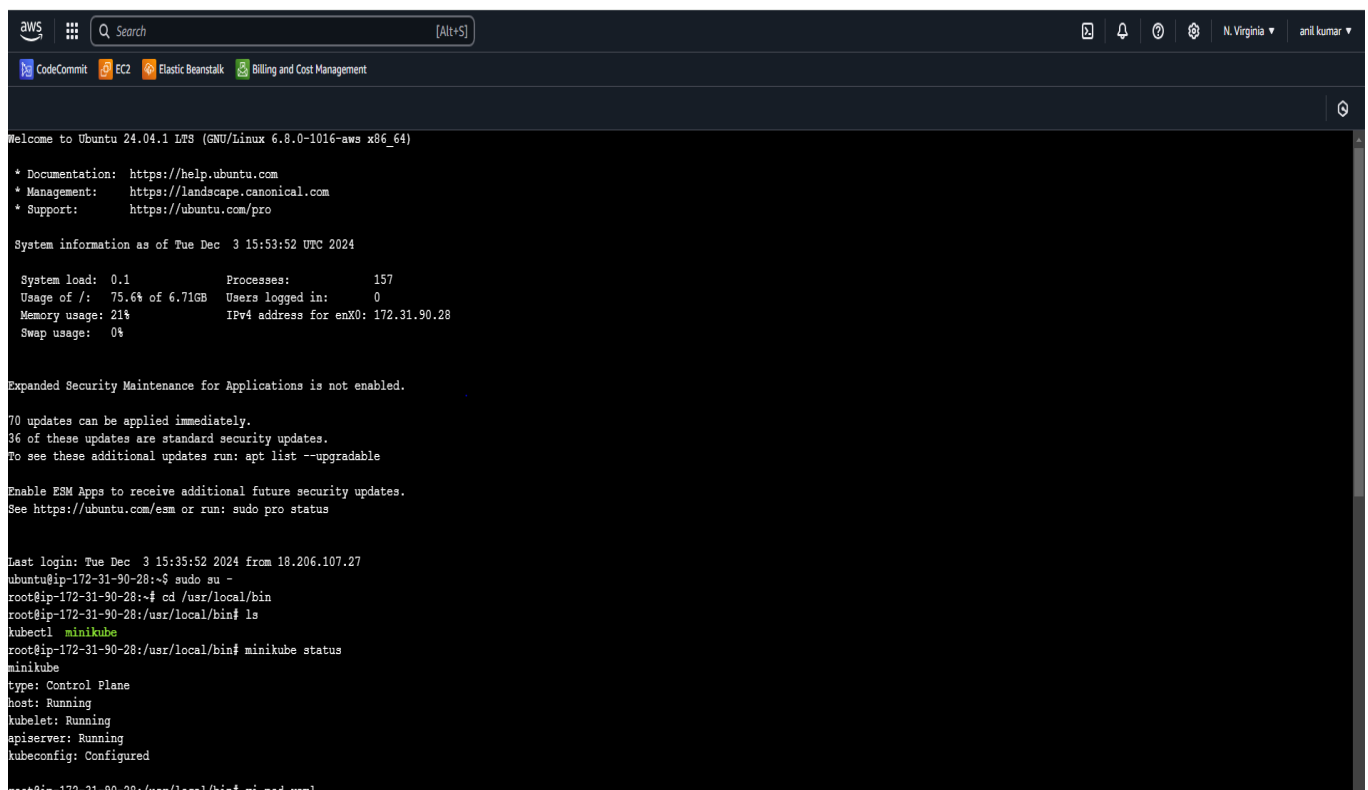


KUBERNETES TASK

Steps to create 5 containers in single pod with manifest file

Step 1:

1. Create the server with ubuntu flavor
2. Install the docker packages
3. Install the minikube packages
4. Install the kubectl packages



```
aws
Search [Alt+S]
CodeCommit EC2 Elastic Beanstalk Billing and Cost Management
N. Virginia anil.kumar

Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1016-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Tue Dec 3 15:53:52 UTC 2024

System load: 0.1          Processes: 157
Usage of /: 75.6% of 6.71GB Users logged in: 0
Memory usage: 21%        IPv4 address for enx0: 172.31.90.28
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

70 updates can be applied immediately.
36 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Tue Dec 3 15:35:52 2024 from 18.206.107.27
ubuntu@ip-172-31-90-28:~$ sudo su -
root@ip-172-31-90-28:~# cd /usr/local/bin
root@ip-172-31-90-28:/usr/local/bin# ls
kubectl minikube
root@ip-172-31-90-28:/usr/local/bin# minikube status
minikube
type: Control Plane
host: Running
kubernetes: Running
apiserver: Running
kubeconfig: Configured
root@ip-172-31-90-28:/usr/local/bin#
```

Step 2:

1. Create the manifest file to create two containers in one pod as shown in below figure:

```

apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  labels:
    app: nginx
spec:
  containers:
  - name: nginx
    image: nginx:1.14.2
    ports:
    - containerPort: 80
---
apiVersion: v1
kind: Pod
metadata:
  name: httpd-pod
  labels:
    app: httpd
spec:
  containers:
  - name: httpd
    image: httpd:2.4
    ports:
    - containerPort: 80
---
apiVersion: v1
kind: Pod
metadata:
  name: redis-pod
  labels:
    app: redis
spec:
  containers:
  - name: redis
    image: redis:6.2.6
    ports:
    - containerPort: 6379
---
apiVersion: v1
kind: Pod

```


```

---
apiVersion: v1
kind: Pod
metadata:
  name: ubuntu-pod
  labels:
    app: ubuntu
spec:
  containers:
  - name: ubuntu
    image: ubuntu:20.04
    command: ["sleep", "3600"]

```

i-0e162732afacda0f9 (Kubernetes)

PublicIPs: 54.162.100.29 PrivateIPs: 172.31.90.28

 CloudShell [Feedback](#)

- Applies the specified pod.yaml file to the Kubernetes cluster

The command is :

kubectl apply -f pod.yaml

- Give command **kubectl get pods** to see all the pods as shown in

below figure

```
root@ip-172-31-90-28:/usr/local/bin# kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
httpd-pod     1/1     Running   0           20s
nginx-pod     1/1     Running   0           20s
pod1          1/1     Running   0           16m
postgres-pod  1/1     Running   0           20s
redis-pod     1/1     Running   0           20s
ubuntu-pod    1/1     Running   0           20s
```