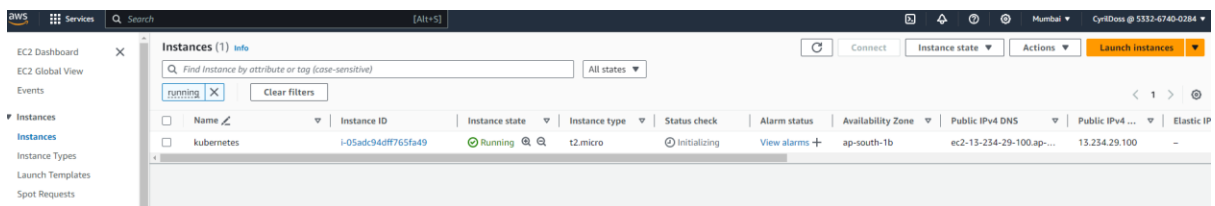
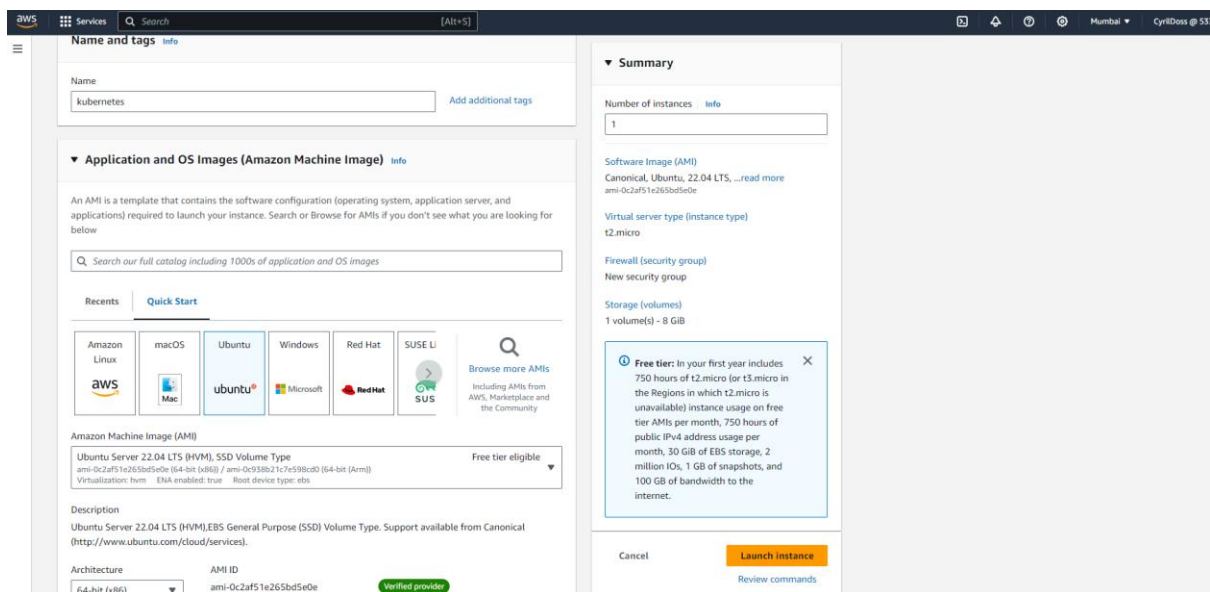


Task – 20

Pods and Services in Kubernetes Deployment

Launching ec2 instance:



Installing Aws Cli and configuring Aws cli:

```
aws | Services | Search | [A]
ubuntu@ip-172-31-8-11:~$ sudo apt install awscli -y
```

```
aws | Services | Search | [Alt+S]
update-alternatives: using /usr/bin/convert-im6.q16 to provide /usr/bin/convert (convert) in auto mode
update-alternatives: using /usr/bin/convert-im6.q16 to provide /usr/bin/convert-im6 (convert-im6) in auto mode
update-alternatives: using /usr/bin/composite-im6.q16 to provide /usr/bin/composite (composite) in auto mode
update-alternatives: using /usr/bin/composite-im6.q16 to provide /usr/bin/composite-im6 (composite-im6) in auto mode
update-alternatives: using /usr/bin/conjure-im6.q16 to provide /usr/bin/conjure (conjure) in auto mode
update-alternatives: using /usr/bin/conjure-im6.q16 to provide /usr/bin/conjure-im6 (conjure-im6) in auto mode
update-alternatives: using /usr/bin/import-im6.q16 to provide /usr/bin/import (import) in auto mode
update-alternatives: using /usr/bin/import-im6.q16 to provide /usr/bin/import-im6 (import-im6) in auto mode
update-alternatives: using /usr/bin/identify-im6.q16 to provide /usr/bin/identify (identify) in auto mode
update-alternatives: using /usr/bin/identify-im6.q16 to provide /usr/bin/identify-im6 (identify-im6) in auto mode
update-alternatives: using /usr/bin/stream-im6.q16 to provide /usr/bin/stream (stream) in auto mode
update-alternatives: using /usr/bin/stream-im6.q16 to provide /usr/bin/stream-im6 (stream-im6) in auto mode
update-alternatives: using /usr/bin/display-im6.q16 to provide /usr/bin/display (display) in auto mode
update-alternatives: using /usr/bin/display-im6.q16 to provide /usr/bin/display-im6 (display-im6) in auto mode
update-alternatives: using /usr/bin/montage-im6.q16 to provide /usr/bin/montage (montage) in auto mode
update-alternatives: using /usr/bin/montage-im6.q16 to provide /usr/bin/montage-im6 (montage-im6) in auto mode
update-alternatives: using /usr/bin/mogrify-im6.q16 to provide /usr/bin/mogrify (mogrify) in auto mode
update-alternatives: using /usr/bin/mogrify-im6.q16 to provide /usr/bin/mogrify-im6 (mogrify-im6) in auto mode
Setting up imagemagick (8:6.9.11.60+dfsg-1.3ubuntu0.22.04.5) ...
Processing triggers for install-info (6.8-4build1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.8) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for shared-mime-info (2.1-2) ...
Processing triggers for sgml-base (1.30) ...
Setting up docutils-common (0.17.1+dfsg-2) ...
Processing triggers for sgml-base (1.30) ...
Setting up python3-docutils (0.17.1+dfsg-2) ...
Setting up awscli (1.22.34-1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-8-11:~$
```

i-05adc94dff765fa49 (kubernetes)

```
ubuntu@ip-172-31-8-11:~$ aws configure
AWS Access Key ID [None]: AKIAXYKJWYZOK7N343ZO
AWS Secret Access Key [None]: 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
Default region name [None]: ap-south-1
Default output format [None]: json
ubuntu@ip-172-31-8-11:~$
```

i-05adc94dff765fa49 (kubernetes)

Installing kubectl and eksctl:

```
ubuntu@ip-172-31-8-11:~$ curl -LO "https://dl.k8s.io/release/${curl -L -s https://dl.k8s.io/release/stable.txt}/bin/linux/amd64/kubectl"
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100 138    100 138    0    0   397      0 --:--:-- --:--:-- --:--:--   397
100 49.0M 100 49.0M    0    0 60.7M      0 --:--:-- --:--:-- --:--:-- 141M
ubuntu@ip-172-31-8-11:~$ curl -LO "https://dl.k8s.io/release/${curl -L -s https://dl.k8s.io/release/stable.txt}/bin/linux/amd64/kubectl.sha256"
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100 138    100 138    0    0   400      0 --:--:-- --:--:-- --:--:--   401
100 64    100 64    0    0   154      0 --:--:-- --:--:-- --:--:--   154
ubuntu@ip-172-31-8-11:~$ echo "$(cat kubectl.sha256) kubectl" | sha256sum --check
kubectl: OK
ubuntu@ip-172-31-8-11:~$ sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
ubuntu@ip-172-31-8-11:~$ kubectl version --client
Client Version: v1.30.3
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
ubuntu@ip-172-31-8-11:~$

i-05adc94dff765fa49 (kubernetes)

ubuntu@ip-172-31-8-11:~$ curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_${uname -s}_amd64.tar.gz" | tar xz -C /tmp
ubuntu@ip-172-31-8-11:~$ mv /tmp/eksctl /usr/local/bin
ubuntu@ip-172-31-8-11:~$ eksctl version
0.187.0
```

Creating a cluster:

```
ubuntu@ip-172-31-8-11:~/new$ eksctl create cluster --name clus1 --region ap-south-1 --node-type t2.micro --nodes-min 2 --nodes-max 10
2024-07-26 10:05:15 [i] eksctl version 0.187.0
2024-07-26 10:05:15 [i] using region ap-south-1
2024-07-26 10:05:15 [i] skipping ap-south-1c from selection because it doesn't support the following instance type(s): t2.micro
2024-07-26 10:05:15 [i] setting availability zones to [ap-south-1a ap-south-1b]
2024-07-26 10:05:15 [i] subnets for ap-south-1a - public:192.168.0.0/19 private:192.168.64.0/19
2024-07-26 10:05:15 [i] subnets for ap-south-1b - public:192.168.32.0/19 private:192.168.96.0/19
2024-07-26 10:05:15 [i] nodegroup "ng-ee2f870f" will use "" (AmazonLinux2/1.30)
2024-07-26 10:05:15 [i] using Kubernetes version 1.30
2024-07-26 10:05:15 [i] creating EKS cluster "clus1" in "ap-south-1" region with managed nodes
2024-07-26 10:05:15 [i] will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2024-07-26 10:05:15 [i] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=ap-south-1 --cluster=clus1'
2024-07-26 10:05:15 [i] Kubernetes API endpoint access will use default of [publicAccess=true, privateAccess=false] for cluster "clus1" in "ap-south-1"
2024-07-26 10:05:15 [i] CloudWatch logging will not be enabled for cluster "clus1" in "ap-south-1"
```

Creating pod:

```
aws | Services | Search
ubuntu@ip-172-31-8-11:~/new$ vi pod.yaml
```

```
aws | Services | Search
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  labels:
    app: web
spec:
  containers:
  - name: nginx-con
    image: nginx
```

```
ubuntu@ip-172-31-8-11:~/new$ kubectl create -f pod.yaml
pod/nginx-pod created
ubuntu@ip-172-31-8-11:~/new$
```

Creating deployment:

```
ubuntu@ip-172-31-8-11:~/new$ vi deployment.yaml
```

```
aws | Services | Search
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deploy
  labels:
    app: web
spec:
  replicas: 3
  selector:
    matchLabels:
      app: web
  template:
    metadata:
      labels:
        app: web
    spec:
      containers:
      - name: mycon
        image: nginx:1.7.9
        ports:
        - containerPort: 80
```

```
ubuntu@ip-172-31-8-11:~/new$ kubectl create -f deployment.yaml
deployment.apps/nginx-deploy created
ubuntu@ip-172-31-8-11:~/new$
```

Creating service:

```
aws | Services | Search
ubuntu@ip-172-31-8-11:~/new$ vi service.yaml
```

```
aws | Services | Search
apiVersion: v1
kind: Service
metadata:
  name: my-service1
spec:
  selector:
    app: web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
  type: LoadBalancer
```

```
ubuntu@ip-172-31-8-11:~/new$ kubectl create -f service.yaml
service/my-service1 created
ubuntu@ip-172-31-8-11:~/new$
```

```
ubuntu@ip-172-31-8-11:~/new$ kubectl get svc
NAME         TYPE          CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
kubernetes   ClusterIP     10.100.0.1       <none>           443/TCP          20m
my-service1  LoadBalancer 10.100.204.199   afe19c64688034fb6a08cb6200579cf8-1500587517.ap-south-1.elb.amazonaws.com 80:31505/TCP    40s
```

Output of the cluster:

