

Microservices Kubernetes

Deployment Assessment

(SKILL TEST-2)

1. Creating AWS EC2 instance:

- Ubuntu 24.04
- t2.medium or larger (2 CPU, 4GB RAM minimum)
- Allow inbound traffic so using existing SG named Aks-SecurityGroup.

The screenshot shows the AWS EC2 Instances page with one instance listed. The instance is named 'Akshay-ST2-Assig...', has an ID of 'i-0182e38f007a25502', is in a 'Running' state, and is a 't2.medium' type. It has 2/2 checks passed and is associated with the availability zone 'us-east-1a'. The public IPv4 is '54.166.167.'. The instance was launched on 'Sun Feb 23 2025 14:54:39 GMT+0530 (India Standard Time)' (19 minutes ago). The AMI name is 'ubuntu/images/hvm-ssd/gp3/ubuntu-noble-24.04-amd64-server-20250115'. The instance has no termination protection, is located in 'amazon/ubuntu/images/hvm-ssd/gp3/ubuntu-noble-24.04-amd64-server-20250115', and has no stop-hibernate behavior. The state transition reason and message are both '-'.

Attribute	Value
AMI name	ubuntu/images/hvm-ssd/gp3/ubuntu-noble-24.04-amd64-server-20250115
Launch time	Sun Feb 23 2025 14:54:39 GMT+0530 (India Standard Time) (19 minutes)
Kernel ID	-
RAM disk ID	-
Boot mode	uefi-preferred
Use RBN as guest OS hostname	Disabled
Owner	975050024946
Current instance boot mode	legacy-bios
Answer RBN DNS hostname IPv4	Disabled

2. Login and install dependencies like Docker/Kubectl/Minikube, etc.

```
ssh -i "EC2-AMI-Aks-HV.pem" ubuntu@54.166.167.139
```

- Update server

```
sudo apt-get update
```

```
sudo apt-get upgrade -y
```

- Install Docker

- `sudo apt-get install docker.io -y`
- `sudo usermod -aG docker ubuntu`
- `sudo systemctl start docker`
- `sudo systemctl enable docker`
- `sudo systemctl status docker`

```
root@ip-10-0-0-136:/home/ubuntu# sudo systemctl status docker
● docker.service - Docker Application Container Engine
  Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
  Active: active (running) since Sun 2025-02-23 09:50:44 UTC; 32s ago
TriggeredBy: ● docker.socket
  Docs: https://docs.docker.com
 Main PID: 9574 (dockerd)
    Tasks: 9
   Memory: 28.7M (peak: 28.9M)
     CPU: 260ms
    CGroup: /system.slice/docker.service
            └─9574 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Feb 23 09:50:44 ip-10-0-0-136 systemd[1]: Starting docker.service - Docker Application Container Engine...
Feb 23 09:50:44 ip-10-0-0-136 dockerd[9574]: time="2025-02-23T09:50:44.199806509Z" level=info msg="Starting up"
Feb 23 09:50:44 ip-10-0-0-136 dockerd[9574]: time="2025-02-23T09:50:44.201611883Z" level=info msg="detected 127.0.0.5"
Feb 23 09:50:44 ip-10-0-0-136 dockerd[9574]: time="2025-02-23T09:50:44.548406079Z" level=info msg="Loading containers"
Feb 23 09:50:44 ip-10-0-0-136 dockerd[9574]: time="2025-02-23T09:50:44.776618242Z" level=info msg="Loading containers"
Feb 23 09:50:44 ip-10-0-0-136 dockerd[9574]: time="2025-02-23T09:50:44.797934311Z" level=info msg="Docker daemon" com
Feb 23 09:50:44 ip-10-0-0-136 dockerd[9574]: time="2025-02-23T09:50:44.798051515Z" level=info msg="Daemon has complet
Feb 23 09:50:44 ip-10-0-0-136 dockerd[9574]: time="2025-02-23T09:50:44.853721168Z" level=info msg="API listen on /run
Feb 23 09:50:44 ip-10-0-0-136 systemd[1]: Started docker.service - Docker Application Container Engine.
root@ip-10-0-0-136:/home/ubuntu#
```

- Install kubectl

- sudo curl -LO "https://dl.k8s.io/release/\$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
- sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl

- Install Minikube

- curl -LO
<https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64>
- sudo install minikube-linux-amd64 /usr/local/bin/minikube

```
root@ip-10-0-0-136:/home/ubuntu# sudo 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  Current
          Dload Upload   Total Spent   Left  Speed
100 138  100 138    0     0  2156  0 ---:--- ---:--- 2198
100 54.6M  100 54.6M  0     0  94.3M  0 ---:--- ---:--- 94.3M
root@ip-10-0-0-136:/home/ubuntu# sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
root@ip-10-0-0-136:/home/ubuntu#
root@ip-10-0-0-136:/home/ubuntu# curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
% Total    % Received % Xferd  Average Speed   Time   Time  Current
          Dload Upload   Total Spent   Left  Speed
100 119M  100 119M    0     0  87.0M  0 0:00:01 0:00:01 87.0M
root@ip-10-0-0-136:/home/ubuntu# sudo install minikube-linux-amd64 /usr/local/bin/minikube
root@ip-10-0-0-136:/home/ubuntu#
```

Install conntrack

sudo apt install conntrack -y

```
root@ip-10-0-0-136:/home/ubuntu#
root@ip-10-0-0-136:/home/ubuntu# sudo apt install conntrack -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  conntrack
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 37.9 kB of archives.
After this operation, 119 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 conntrack amd64 1:1.4.8-1ubuntu1 [37.9 kB]
Fetched 37.9 kB in 0s (1777 kB/s)
Selecting previously unselected package conntrack.
(Reading database ... 70923 files and directories currently installed.)
Preparing to unpack .../conntrack_1%3a1.4.8-1ubuntu1_amd64.deb ...
Unpacking conntrack (1:1.4.8-1ubuntu1) ...
Setting up conntrack (1:1.4.8-1ubuntu1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning candidates...
Scanning linux images...

Running kernel seems to be up-to-date.

Restarting services...

Service restarts being deferred:
/etc/needrestart/restart.d/dbus.service
systemctl restart getty@tty1.service
systemctl restart networkd-dispatcher.service
systemctl restart serial-getty@ttyS0.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service

No containers need to be restarted.

User sessions running outdated binaries:
  ubuntu @ session #3: sshd[1170,1284], su[1332]
  ubuntu @ user manager service: systemd[1175]

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-10-0-0-136:/home/ubuntu#
root@ip-10-0-0-136:/home/ubuntu# which conntrack
/usr/sbin/conntrack
```

Start Minikube:

```
minikube start --driver=docker --force
```

```
root@ip-10-0-0-136:/home/ubuntu# minikube start --driver=docker
🌟 minikube v1.35.0 on Ubuntu 24.04 (xen/amd64)
💡 Using the docker driver based on user configuration
⚠️ The "docker" driver should not be used with root privileges. If you wish to continue as root, use --force.
💡 If you are running minikube within a VM, consider using --driver=none:
    https://minikube.sigs.k8s.io/docs/reference/drivers/none/

✖ Exiting due to DRV_AS_ROOT: The "docker" driver should not be used with root privileges.

root@ip-10-0-0-136:/home/ubuntu# minikube start --driver=docker --force
🌟 minikube v1.35.0 on Ubuntu 24.04 (xen/amd64)
❗ minikube skips various validations when --force is supplied; this may lead to unexpected behavior
💡 Using the docker driver based on user configuration
⚠️ The "docker" driver should not be used with root privileges. If you wish to continue as root, use --force.
💡 If you are running minikube within a VM, consider using --driver=none:
    https://minikube.sigs.k8s.io/docs/reference/drivers/none/
💡 Using Docker driver with root privileges
👍 Starting "minikube" primary control-plane node in "minikube" cluster
Pulling base image v0.0.46 ...
💻 Downloading Kubernetes v1.32.0 preload ...
> preloaded-images-k8s-v18-v1...: 333.57 MiB / 333.57 MiB 100.00% 62.14 M
> gcr.io/k8s-minikube/kicbase...: 500.31 MiB / 500.31 MiB 100.00% 68.75 M
🔥 Creating docker container (CPUs=2, Memory=2200MB) ...
🌐 Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
    ■ Generating certificates and keys ...
    ■ Booting up control plane ...
    ■ Configuring RBAC rules ...
🔗 Configuring bridge CNI (Container Networking Interface) ...
🌐 Verifying Kubernetes components...
    ■ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟 Enabled addons: storage-provisioner, default-storageclass
🌟 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
root@ip-10-0-0-136:/home/ubuntu#
```

Enable ingress addon on K8S:

```
minikube addons enable ingress
```

```
root@ip-10-0-0-136:/home/ubuntu#
root@ip-10-0-0-136:/home/ubuntu# minikube addons enable ingress
💡 ingress is an addon maintained by Kubernetes. For any concerns contact minikube on GitHub.
You can view the list of minikube maintainers at: https://github.com/kubernetes/minikube/blob/master/OWNERS
    ■ Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.4
    ■ Using image registry.k8s.io/ingress-nginx/controller:v1.11.3
    ■ Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.4
🌐 Verifying ingress addon...
🌟 The 'ingress' addon is enabled
root@ip-10-0-0-136:/home/ubuntu#
root@ip-10-0-0-136:/home/ubuntu#
```

3. Clone git repository and work on manifest files:

- Fork repo <https://github.com/mohanDevOps-arch/Microservices-Task>
- Forked github repo <https://github.com/akshaybhu/Microservices-Task>

Create ssh key and add to your github repository

```
root@ip-10-0-0-136:/home/ubuntu# ssh-keygen -t ed25519 -C "Akshay.thebest@yahoo.co.in"
Generating public/private key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_ed25519
Your public key has been saved in /root/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:cMyV39Lqqx0CuLIq0rZXvZKPPkYgD5V5ojDVO6aIoI8 Akshay.thebest@yahoo.co.in
The key's randomart image is:
++-[ED25519 256]--+
| ... o ...
|o * .o ..
| o o +. + . o
|. + = + o o
|+. * o..S o
|+ . . o... .
| + .o... ....
|E + .o=.. o..
|..oo+.oo+. ..o.
+---[SHA256]---+
root@ip-10-0-0-136:/home/ubuntu# cat /root/.ssh/id_ed25519.pub
```

4. Created docker images for 4 services

```
docker build -t akshayap2901/gateway-service:latest .
docker build -t akshayap2901/user-service:latest .
docker build -t akshayap2901/product-service:latest .
docker build -t akshayap2901/order-service:latest .
```

```
root@ip-10-0-0-136:/home/ubuntu/Microservices-Task/Microservices# docker images
REPOSITORY          TAG      IMAGE ID      CREATED     SIZE
akshayap2901/order-service    latest   9c19e3754892  3 minutes ago  125MB
akshayap2901/product-service  latest   49b0040c4650  3 minutes ago  125MB
akshayap2901/user-service     latest   76aa9eeff8b7d  4 minutes ago  125MB
akshayap2901/gateway-service  latest   3603b5d99b9a  5 minutes ago  174MB
node                  23-alpine3.20  376057dfd4ca  9 days ago   164MB
gcr.io/k8s-minikube/kicbase  v0.0.46   e72c4cbe9b29  5 weeks ago   1.31GB
node                  16-alpine    2573171e0124  18 months ago  118MB
root@ip-10-0-0-136:/home/ubuntu/Microservices-Task/Microservices#
```

Push images to docker Hub:

The screenshot shows the Docker Hub interface with the user's repositories listed. The repositories are:

Name	Last Pushed	Contains	Visibility	Scout
akshayap2901/user-service	about 1 hour ago	IMAGE	Public	Inactive
akshayap2901/order-service	about 1 hour ago	IMAGE	Public	Inactive
akshayap2901/product-service	about 1 hour ago	IMAGE	Public	Inactive
akshayap2901/gateway-service	about 1 hour ago	IMAGE	Public	Inactive
akshayap2901/flask-aks	2 months ago	IMAGE	Public	Inactive
akshayap2901/test1	2 months ago		Private	Inactive

```
[root@ip-10-0-0-136:/home/ubuntu/Microservices-Task/Microservices# docker push akshayap2901/gateway-service:latest
The push refers to repository [docker.io/akshayap2901/gateway-service]
39d8e4e52188: Pushed
e8210234ed73: Pushed
f03fe3b98272: Pushed
8ad5d243031c: Pushed
e88bf9aa3d9f: Mounted from library/node
49c82db8b337d: Mounted from library/node
9e4187464845: Mounted from library/node
994456c4fd7b: Mounted from library/node
latest: digest: sha256:f37d113d54e9c829f54ddb6e47846daa366f6d578a45bd7b4256a2fc2e0c885f size: 1990
root@ip-10-0-0-136:/home/ubuntu/Microservices-Task/Microservices#
root@ip-10-0-0-136:/home/ubuntu/Microservices-Task/Microservices#
root@ip-10-0-0-136:/home/ubuntu/Microservices-Task/Microservices# docker push akshayap2901/product-service:latest
The push refers to repository [docker.io/akshayap2901/product-service]
f4207646f85a: Pushed
fed8e3d1082e: Pushed
9a9dec96257a: Pushed
0d1e55ea2ef4: Pushed
365cccd918307: Mounted from library/node
1bba629c69e9: Mounted from library/node
139c1270acf1: Mounted from library/node
4693057ce236: Mounted from library/node
latest: digest: sha256:47aa86b80ce48a8aad6a9d27b583544e445c1e791c51194e03bed4f5915e409 size: 1989
root@ip-10-0-0-136:/home/ubuntu/Microservices-Task/Microservices#
root@ip-10-0-0-136:/home/ubuntu/Microservices-Task/Microservices#
root@ip-10-0-0-136:/home/ubuntu/Microservices-Task/Microservices# docker push akshayap2901/order-service:latest
The push refers to repository [docker.io/akshayap2901/order-service]
a68a2d707ccb: Pushed
fed8e3d1082e: Mounted from akshayap2901/product-service
9a9dec96257a: Mounted from akshayap2901/product-service
0d1e55ea2ef4: Mounted from akshayap2901/product-service
365cccd918307: Mounted from akshayap2901/product-service
1bba629c69e9: Mounted from akshayap2901/product-service
139c1270acf1: Mounted from akshayap2901/product-service
4693057ce236: Mounted from akshayap2901/product-service
latest: digest: sha256:8ef6de747d08086979c6743906fa7fe8da75e4897208179dde9bf1ea6aa35e987 size: 1990
root@ip-10-0-0-136:/home/ubuntu/Microservices-Task/Microservices# docker push akshayap2901/user-service:latest
The push refers to repository [docker.io/akshayap2901/user-service]
fcfa8667d3db: Pushed
fed8e3d1082e: Mounted from akshayap2901/order-service
9a9dec96257a: Mounted from akshayap2901/order-service
0d1e55ea2ef4: Mounted from akshayap2901/order-service
365cccd918307: Mounted from akshayap2901/order-service
1bba629c69e9: Mounted from akshayap2901/order-service
139c1270acf1: Mounted from akshayap2901/order-service
4693057ce236: Mounted from akshayap2901/order-service
latest: digest: sha256:700e53cb14443b448202ee933c555986156e66b0144d66a02a75ec7b5c103203 size: 1989
root@ip-10-0-0-136:/home/ubuntu/Microservices-Task/Microservices#
```

Create and run apply for manifest yaml files to create pods/deployment and services.

```
kubectl apply -f product-svc.yml  
kubectl apply -f gateway-svc.yml  
kubectl apply -f user-svc.yml  
kubectl apply -f order-svc.yml
```

```
root@ip-10-0-0-136:/home/ubuntu/Microservices-Task/Microservices/user-service# kubectl get pods  
NAME           READY   STATUS    RESTARTS   AGE  
gateway-service-6fd8d79d4d-pvmct  1/1     Running   0          46s  
order-service-cbcfc69bb-gfhhs   1/1     Running   0          100s  
product-service-85fb57f987-h4gpl 1/1     Running   0          59s  
user-service-77d88dccfc-97ndt   1/1     Running   0          26s  
root@ip-10-0-0-136:/home/ubuntu/Microservices-Task/Microservices/user-service# kubectl get deployment  
NAME        READY   UP-TO-DATE   AVAILABLE   AGE  
gateway-service  1/1      1           1           48s  
order-service   1/1      1           1           102s  
product-service 1/1      1           1           61s  
user-service    1/1      1           1           28s  
root@ip-10-0-0-136:/home/ubuntu/Microservices-Task/Microservices/user-service# kubectl get svc  
NAME        TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE  
gateway-service ClusterIP  10.104.93.200 <none>       3003/TCP  52s  
kubernetes   ClusterIP  10.96.0.1    <none>       443/TCP   78m  
order-service ClusterIP  10.109.129.57 <none>       3002/TCP  106s  
product-service ClusterIP  10.100.35.196 <none>       3001/TCP  65s  
user-service   ClusterIP  10.107.163.168 <none>       3000/TCP  32s  
root@ip-10-0-0-136:/home/ubuntu/Microservices-Task/Microservices/user-service#
```

Create ingress file and apply

```
kubectl apply -f ingress.yaml
```

```
root@ip-10-0-0-136:/home/ubuntu/Microservices-Task/Microservices#  
root@ip-10-0-0-136:/home/ubuntu/Microservices-Task/Microservices# kubectl apply -f ingress.yaml  
ingress.networking.k8s.io/microservices-ingress created  
root@ip-10-0-0-136:/home/ubuntu/Microservices-Task/Microservices#
```

```
kubectl get services -n ingress-nginx
```

```
kubectl get service ingress-nginx-controller -n ingress-nginx
```

```
root@ip-10-0-0-136:/home/ubuntu/Microservices-Task/Microservices#
root@ip-10-0-0-136:/home/ubuntu/Microservices-Task/Microservices# kubectl get services -n ingress-nginx
NAME           TYPE      CLUSTER-IP   EXTERNAL-IP     PORT(S)          AGE
ingress-nginx-controller   LoadBalancer  10.99.111.41 <pending>    80:32680/TCP,443:31672/TCP   32m
ingress-nginx-controller-admission ClusterIP  10.101.187.196 <none>        443/TCP         32m
root@ip-10-0-0-136:/home/ubuntu/Microservices-Task/Microservices#
root@ip-10-0-0-136:/home/ubuntu/Microservices-Task/Microservices# kubectl get ingress
NAME           CLASS    HOSTS          ADDRESS       PORTS   AGE
microservices-ingress  nginx   microservices.local   80          48m
```

```
root@ip-10-0-0-136:/home/ubuntu/Microservices-Task/Microservices#
root@ip-10-0-0-136:/home/ubuntu/Microservices-Task/Microservices# sudo echo "54.166.167.139 microservices.local" | sudo tee -a /etc/hosts
54.166.167.139 microservices.local
root@ip-10-0-0-136:/home/ubuntu/Microservices-Task/Microservices#
root@ip-10-0-0-136:/home/ubuntu/Microservices-Task/Microservices# minikube ip
192.168.49.2
root@ip-10-0-0-136:/home/ubuntu/Microservices-Task/Microservices#
```

ALB:

The screenshot shows the AWS CloudFormation console with the 'ingress-nginx-controller' stack selected. The left sidebar lists various resources and stacks. The main pane displays the 'Details' section for the 'ingress-nginx-controller' stack, which includes information about the VPC (vpc-09f02049d6176fe30), Hosted zone (Z55SXDOTRQ7X7K), and Availability Zones (us-east-1a, us-east-1b). Below this, the 'Listeners and rules' tab is active, showing one listener rule for port 80 that forwards traffic to the 'nginx-ctrl' target group. The rule has a 100% weight and is set to 'Target group stickiness: Off'.