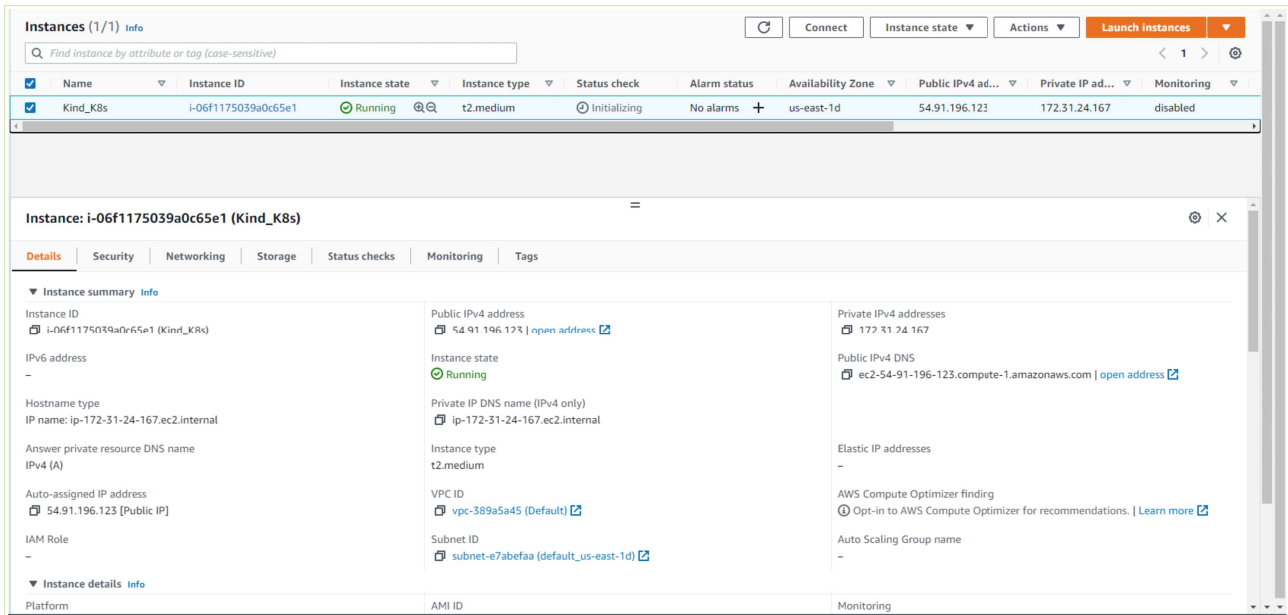


Assignment 1 :: Write a step by step process to create kubernetes kind cluster. Also write process to containerize any sample application. Steps to deploy that application on kind cluster.

Table of Contents

Assignment 1 :: Write a step by step process to create kubernetes kind cluster. Also write process to containerize any sample application. Steps to deploy that application on kind cluster.....	1
Step 1 :: Create an instance with t2.medium and storage 60 GB.....	2
Step 2 :: Install Docker and Docker-Compose.....	2
Step 3 :: Install Kubectl.....	2
Step 4 :: Install Kind.....	3
Step 5 :: Create a cluster using Kind.....	4
Step 6 :: Creating a sample application using dockerfile.....	5
Step 6.1 :: Create a docker file.....	5
Step 6.2 :: Placing the necessary scripts.....	5
Step 6.3 :: Creation and pushing an image to docker hub.....	6
Step 7 :: Deploy the application in Kind Cluster.....	7
Step 8 :: Result.....	8

Step 1 :: Create an instance with t2.medium and storage 60 GB



The screenshot shows the AWS Management Console interface for an EC2 instance. At the top, there's a search bar and a table of instances. The instance 'Kind_K8s' is selected, showing its details in a sidebar. The instance is in the 'Running' state, has a public IP address of 54.91.196.123, and is of type 't2.medium'. The console also shows the instance's VPC ID, subnet ID, and other configuration details.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 ad...	Private IP ad...	Monitoring
Kind_K8s	i-06f1175039a0c65e1	Running	t2.medium	Initializing	No alarms	us-east-1d	54.91.196.123	172.31.24.167	disabled

Instance: i-06f1175039a0c65e1 (Kind_K8s)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Instance summary [Info](#)

Instance ID: i-06f1175039a0c65e1 (Kind_K8s)
Public IPv4 address: 54.91.196.123 | [open address](#)
Private IPv4 addresses: 172.31.24.167
Instance state: **Running**
Private IP DNS name (IPv4 only): ip-172-31-24-167.ec2.internal
Instance type: t2.medium
VPC ID: vpc-389a5a45 (Default) | [open address](#)
Subnet ID: subnet-e7abefaa (default_us-east-1d) | [open address](#)
Elastic IP addresses: -
AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | [Learn more](#)
Auto Scaling Group name: -

Instance details [Info](#)

Platform: AMI ID: Monitoring:

Step 2 :: Install Docker and Docker-Compose

```
# yum update
# yum install -y docker
# yum install -y python3-pip
# pip3 install docker-compose
# systemctl enable docker.service
# systemctl start docker.service
# systemctl status docker.service
# docker -v
# docker-compose -v
```

```
[root@ip-172-31-24-167 ~]# docker -v
Docker version 20.10.17, build 100c701
[root@ip-172-31-24-167 ~]# docker-compose -v
docker-compose version 1.29.2, build unknown
[root@ip-172-31-24-167 ~]#
```

Step 3 :: Install Kubectl

```
# curl -o kubectl https://s3.us-west-2.amazonaws.com/amazon-eks/1.23.7/2022-06-29/bin/linux/amd64/kubectl
# curl -o kubectl.sha256 https://s3.us-west-2.amazonaws.com/amazon-eks/1.23.7/2022-06-29/bin/linux/amd64/kubectl.sha256
```

```
# openssl sha1 -sha256 kubectl
# chmod +x ./kubectl
# mkdir -p $HOME/bin && cp ./kubectl $HOME/bin/kubectl && export
PATH=$PATH:$HOME/bin
# echo 'export PATH=$PATH:$HOME/bin' >> ~/.bashrc
# kubectl version --short --client
```

```
[root@ip-172-31-24-167 ~]# kubectl version --short --client
Client Version: v1.23.7-eks-4721010
[root@ip-172-31-24-167 ~]#
```

Step 4 :: Install Kind

```
# curl -Lo ./kind https://kind.sigs.k8s.io/dl/v0.15.0/kind-linux-amd64
# chmod +x ./kind
# mv ./kind /usr/local/bin/kind
# kind version
```

```
[root@ip-172-31-24-167 ~]# kind version
kind v0.15.0 go1.19 linux/amd64
[root@ip-172-31-24-167 ~]#
```

Step 5 :: Create a cluster using Kind

```
# vi kind-config.yaml
```

```
kind: Cluster
apiVersion: kind.x-k8s.io/v1alpha4
nodes:
- role: control-plane
- role: worker
- role: worker
- role: worker
```

```
# kind create cluster --config kind-config.yaml --name=assignment-1a
```

```
[root@ip-172-31-24-167 ~]# kind create cluster --config kind-config.yaml --name=assignment-1a
Creating cluster "assignment-1a" ...
 ✓ Ensuring node image (kindest/node:v1.25.0)
 ✓ Preparing nodes
 ✓ Writing configuration
 ✓ Starting control-plane
 ✓ Installing CNI
 ✓ Installing StorageClass
 ✓ Joining worker nodes
Set kubectl context to "kind-assignment-1a"
You can now use your cluster with:

kubectl cluster-info --context kind-assignment-1a

Have a question, bug, or feature request? Let us know! https://kind.sigs.k8s.io/#community ☺
[root@ip-172-31-24-167 ~]#
```

```
# kubectl get all
```

```
# kubectl get node -o wide
```

```
[root@ip-172-31-24-167 ~]# kubectl get all
NAME                                TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
service/kubernetes                  ClusterIP     10.96.0.1     <none>         443/TCP    62s
[root@ip-172-31-24-167 ~]# kubectl get node -o wide
NAME                                STATUS    ROLES    AGE   VERSION   INTERNAL-IP    EXTERNAL-IP    OS-IMAGE             KERNEL-VERSION      CONTAINER-RUNTIME
assignment-1a-control-plane         Ready    control-plane   68s   v1.25.0   172.18.0.2     <none>         Ubuntu 22.04.1 LTS   5.10.135-122.509.amzn2.x86_64   containerd://1.6.7
assignment-1a-worker                Ready    <none>         34s   v1.25.0   172.18.0.3     <none>         Ubuntu 22.04.1 LTS   5.10.135-122.509.amzn2.x86_64   containerd://1.6.7
assignment-1a-worker2               Ready    <none>         34s   v1.25.0   172.18.0.5     <none>         Ubuntu 22.04.1 LTS   5.10.135-122.509.amzn2.x86_64   containerd://1.6.7
assignment-1a-worker3               Ready    <none>         34s   v1.25.0   172.18.0.4     <none>         Ubuntu 22.04.1 LTS   5.10.135-122.509.amzn2.x86_64   containerd://1.6.7
[root@ip-172-31-24-167 ~]#
```

```
# docker ps -a
```

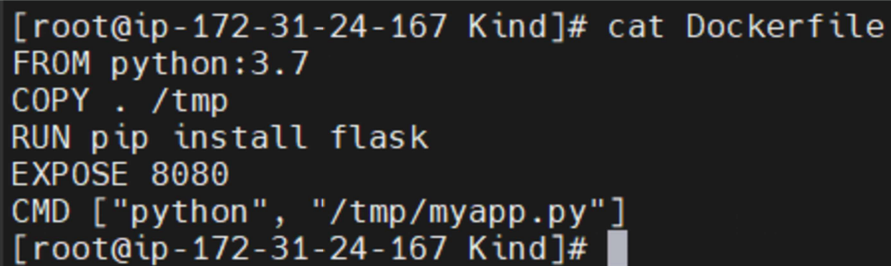
```
[root@ip-172-31-24-167 ~]# docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                               NAMES
9b7b65cf1889   kindest/node:v1.25.0               "/usr/local/bin/entr..." 2 minutes ago  Up 2 minutes                assignment-1a-worker
e8588eae5cd4   kindest/node:v1.25.0               "/usr/local/bin/entr..." 2 minutes ago  Up 2 minutes                assignment-1a-worker2
0167e5e43811   kindest/node:v1.25.0               "/usr/local/bin/entr..." 2 minutes ago  Up 2 minutes                assignment-1a-worker3
7177e46c486d   kindest/node:v1.25.0               "/usr/local/bin/entr..." 2 minutes ago  Up 2 minutes   127.0.0.1:37743->6443/tcp         assignment-1a-control-plane
[root@ip-172-31-24-167 ~]#
```

Step 6 :: Creating a sample application using dockerfile

Step 6.1 :: Create a docker file

vi Dockerfile

```
FROM python:3.7
COPY . /tmp
RUN pip install flask
EXPOSE 8080
CMD ["python", "/tmp/myapp.py"]
```

A terminal window with a dark background and light-colored text. The prompt is [root@ip-172-31-24-167 Kind]#. The user has entered the command cat Dockerfile. The output shows the contents of the Dockerfile: FROM python:3.7, COPY . /tmp, RUN pip install flask, EXPOSE 8080, and CMD ["python", "/tmp/myapp.py"]. The prompt returns to [root@ip-172-31-24-167 Kind]#.

```
[root@ip-172-31-24-167 Kind]# cat Dockerfile
FROM python:3.7
COPY . /tmp
RUN pip install flask
EXPOSE 8080
CMD ["python", "/tmp/myapp.py"]
[root@ip-172-31-24-167 Kind]#
```

Step 6.2 :: Placing the necessary scripts

Create myapp.py where the Dockerfile is available

vi myapp.py

```
from flask import Flask
import os
app = Flask(__name__)
@app.route('/')
def hello():
    return('Hello from container.\n')
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8080, debug=True)
```

Step 6.3 :: Creation and pushing an image to docker hub

docker build -t webapp .

```
[root@ip-172-31-24-167 ~]# docker build -t webapp .
Sending build context to Docker daemon 102.5MB
Step 1/5 : FROM python:3.7
3.7: Pulling from library/python
23858da423a6: Pull complete
326f452ade5c: Pull complete
a42821cd14fb: Pull complete
8471b75885ef: Pull complete
8ffa7aef404: Pull complete
15132af73342: Pull complete
a81a13cb42bc: Pull complete
39bba166dda6: Pull complete
28d172f530ba: Pull complete
Digest: sha256:2341ac5eadd71f1e6481afe854af572f5ec1b78fa3aea2293dba65942108e663
Status: Downloaded newer image for python:3.7
--> 7d2ecbd72983
Step 2/5 : COPY . /tmp
--> 08fe6245fe29
Step 3/5 : RUN pip install flask
--> Running in 162d6d2cbc65
Collecting flask
  Downloading Flask-2.2.2-py3-none-any.whl (101 kB)
    101.5/101.5 KB 11.0 MB/s eta 0:00:00
Collecting Jinja2>=3.0
  Downloading Jinja2-3.1.2-py3-none-any.whl (133 kB)
    133.1/133.1 KB 24.9 MB/s eta 0:00:00
Collecting click>=8.0
  Downloading click-8.1.3-py3-none-any.whl (96 kB)
    96.6/96.6 KB 20.7 MB/s eta 0:00:00
Collecting Werkzeug>=2.2.2
  Downloading Werkzeug-2.2.2-py3-none-any.whl (232 kB)
    232.7/232.7 KB 36.1 MB/s eta 0:00:00
Collecting itsdangerous>=2.0
  Downloading itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Collecting importlib-metadata>=3.6.0
  Downloading importlib-metadata-5.0.0-py3-none-any.whl (21 kB)
Collecting typing-extensions>=3.6.4
  Downloading typing-extensions-4.3.0-py3-none-any.whl (25 kB)
Collecting zipp>=0.5
  Downloading zipp-3.8.1-py3-none-any.whl (5.6 kB)
Collecting MarkupSafe>=2.0
  Downloading MarkupSafe-2.1.1-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (25 kB)
Installing collected packages: zipp, typing-extensions, MarkupSafe, itsdangerous, Werkzeug, Jinja2, importlib-metadata, click, flask
Successfully installed Jinja2-3.1.2 MarkupSafe-2.1.1 Werkzeug-2.2.2 click-8.1.3 flask-2.2.2 importlib-metadata-5.0.0 itsdangerous-2.1.2 typing-extensions-4.3.0 zipp-3.8.1
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended
to use a virtual environment instead: https://pip.pypa.io/warnings/venv
WARNING: You are using pip version 22.0.4; however, version 22.2 is available.
You should consider upgrading via the '/usr/local/bin/python -m pip install --upgrade pip' command.
Removing intermediate container 162d6d2cbc65
--> 49ee7b2c06dd
Step 4/5 : EXPOSE 8080
--> Running in bffa573844f0
Removing intermediate container bffa573844f0
--> 7e3cc1674ffc
Step 5/5 : CMD ["python", "/tmp/myapp.py"]
--> Running in edf2a017dbca
Removing intermediate container edf2a017dbca
--> 2a494d267023
Successfully built 2a494d267023
Successfully tagged webapp:latest
[root@ip-172-31-24-167 ~]#
```

docker images

```
[root@ip-172-31-24-167 Kind]# docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
webapp              latest          c7a91035a38e   6 seconds ago  918MB
python              3.7            7d2ecbd72983   2 weeks ago    907MB
kindest/node        <none>         d3da246e125a   4 weeks ago    870MB
[root@ip-172-31-24-167 Kind]#
```

Push the image to docker hub

docker login

Enter username and password of docker hub

docker tag <image-id> hariharan410/webapp:latest

docker push hariharan410/webapp:latest

Step 7 :: Deploy the application in Kind Cluster

```
# kubectl run mywebapp --image=hariharan410/webapp --restart=Never --dry-run=client -o yaml > web-app.yaml
```

```
# cat web-app.yaml
```

```
[root@ip-172-31-24-167 Kind]# cat web-app.yaml
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: mywebapp
  name: mywebapp
spec:
  containers:
  - image: hariharan410/webapp
    name: mywebapp
    resources: {}
  dnsPolicy: ClusterFirst
  restartPolicy: Never
status: {}
```

```
# kubectl apply -f web-app.yaml
```

```
[root@ip-172-31-24-167 Kind]# kubectl run mywebapp --image=hariharan410/webapp --restart=Never --dry-run=client -o yaml > web-app.yaml
[root@ip-172-31-24-167 Kind]# cat web-app.yaml
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: mywebapp
  name: mywebapp
spec:
  containers:
  - image: hariharan410/webapp
    name: mywebapp
    resources: {}
  dnsPolicy: ClusterFirst
  restartPolicy: Never
status: {}
[root@ip-172-31-24-167 Kind]# kubectl apply -f web-app.yaml
pod/mywebapp created
[root@ip-172-31-24-167 Kind]# kubectl get pod -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP          NODE          NOMINATED NODE   READINESS GATES
mywebapp      0/1     ContainerCreating   0           10s   <none>      assignment-1a-worker   <none>           <none>
```

```
# kubectl get pod -o wide
```

```
[root@ip-172-31-24-167 Kind]# kubectl get pod -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP          NODE          NOMINATED NODE   READINESS GATES
mywebapp      1/1     Running   0           117s   10.244.3.3   assignment-1a-worker   <none>           <none>
```

```
# kubectl expose pod mywebapp --type=NodePort --port=8080 --name webservice
```

```
[root@ip-172-31-24-167 Kind]# kubectl expose pod mywebapp --type=NodePort --port=8080 --name webservice
service/webservice exposed
```


kubectl get all -o wide

```
[root@ip-172-31-24-167 Kind]# kubectl get all -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP            NODE          NOMINATED NODE   READINESS GATES
pod/mywebapp   1/1     Running   0           4m48s  10.244.3.3    assignment-1a-worker  <none>           <none>

NAME          TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE   SELECTOR
service/kubernetes  ClusterIP    10.96.0.1    <none>        443/TCP          73m   <none>
service/websocket   NodePort     10.96.61.93  <none>        8080:30672/TCP   96s   run=mywebapp
[root@ip-172-31-24-167 Kind]#
```

Step 8 :: Result

kubectl get all -o wide

kubectl get node -o wide

curl 172.18.0.3:30672

```
[root@ip-172-31-24-167 Kind]# kubectl get all -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP            NODE          NOMINATED NODE   READINESS GATES
pod/mywebapp   1/1     Running   0           6m38s  10.244.3.3    assignment-1a-worker  <none>           <none>

NAME          TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE   SELECTOR
service/kubernetes  ClusterIP    10.96.0.1    <none>        443/TCP          75m   <none>
service/websocket   NodePort     10.96.61.93  <none>        8080:30672/TCP   3m26s  run=mywebapp
[root@ip-172-31-24-167 Kind]#
[root@ip-172-31-24-167 Kind]# kubectl get node -o wide
NAME          STATUS    ROLES          AGE   VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE          KERNEL-VERSION   CONTAINER-RUNTIME
assignment-1a-control-plane  Ready   control-plane  75m   v1.25.0   172.18.0.2    <none>         Ubuntu 22.04.1 LTS  5.10.135-122.509.amzn2.x86_64  containerd://1.6.7
assignment-1a-worker         Ready   <none>         75m   v1.25.0   172.18.0.3    <none>         Ubuntu 22.04.1 LTS  5.10.135-122.509.amzn2.x86_64  containerd://1.6.7
assignment-1a-worker2        Ready   <none>         75m   v1.25.0   172.18.0.5    <none>         Ubuntu 22.04.1 LTS  5.10.135-122.509.amzn2.x86_64  containerd://1.6.7
assignment-1a-worker3        Ready   <none>         75m   v1.25.0   172.18.0.4    <none>         Ubuntu 22.04.1 LTS  5.10.135-122.509.amzn2.x86_64  containerd://1.6.7
[root@ip-172-31-24-167 Kind]#
[root@ip-172-31-24-167 Kind]# curl 172.18.0.3:30672
Hello from container..
[root@ip-172-31-24-167 Kind]#
```