

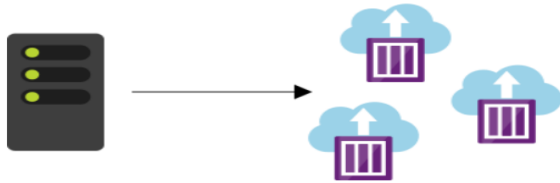
What is Kubernetes

Kubernetes is a portable, extensible open-source platform for managing and orchestrating containerized workloads. Kubernetes abstracts away complex container management tasks, and provides you with declarative configuration to orchestrate containers in different computing environments. This orchestration platform gives you the same ease of use and flexibility you may already know from platform as a service (PaaS) or infrastructure as a service (IaaS) offerings.

What is container management?

Container management is the process of organizing, adding, removing, or updating a significant number of containers.

The drone tracking app consists of multiple microservices, responsible for tasks like caching, queuing, or data processing. Each of these services is hosted in a container and can be deployed, updated, and scaled independently from one another.



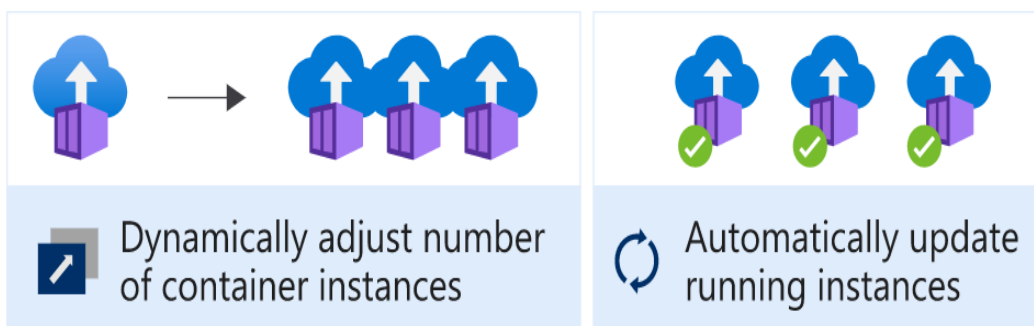
For example, with the drone tracking app's website, you find that at specific times during the day, you need more instances of the site's caching service to keep up performance, so you add more caching service container instances.

Now, assume that you've increased the number of caching instances, and need to roll out a new version of the microservice. You'll have to make sure to update *all* the active containers.

Container management helps you with these otherwise manual tasks.

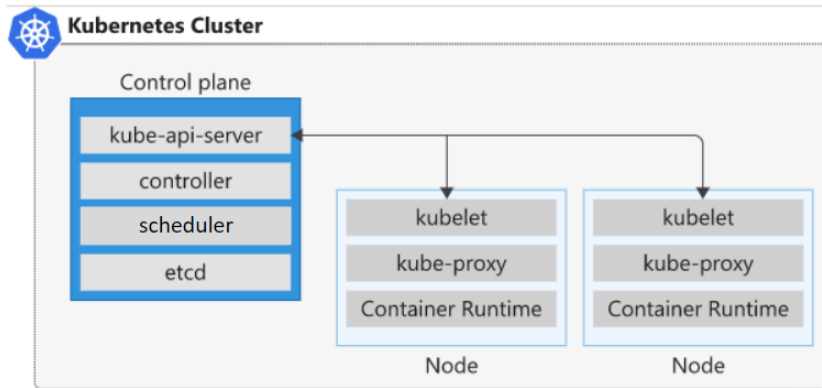
What is container orchestration?

A container orchestrator is a system that automatically deploys and manages containerized apps. For example, the orchestrator can dynamically respond to changes in the environment to increase or decrease the deployed instances of the managed app. Or, it can ensure all deployed container instances get updated if a new version of a service is released.



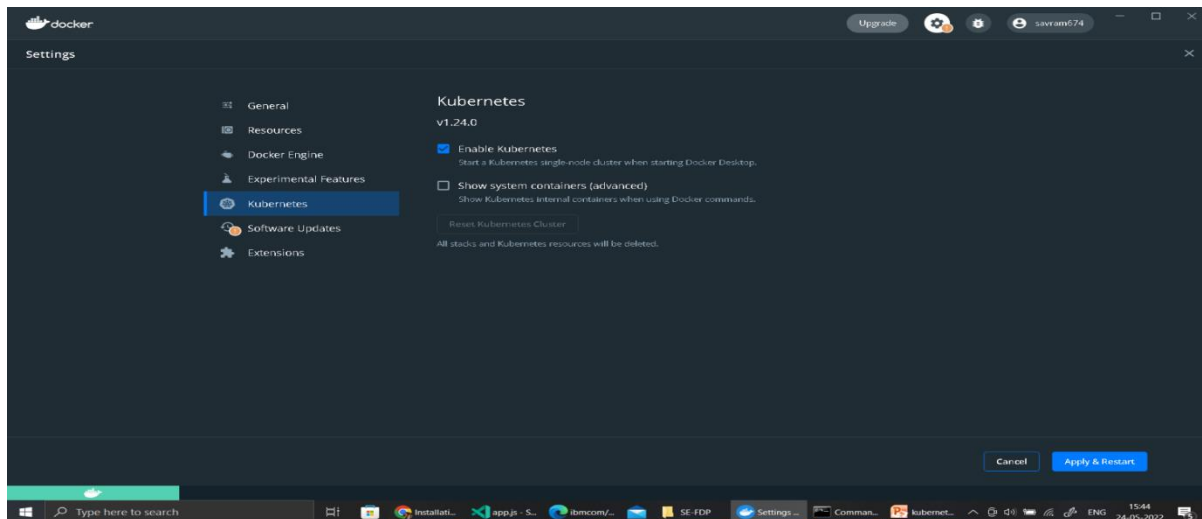
Kubernetes architecture

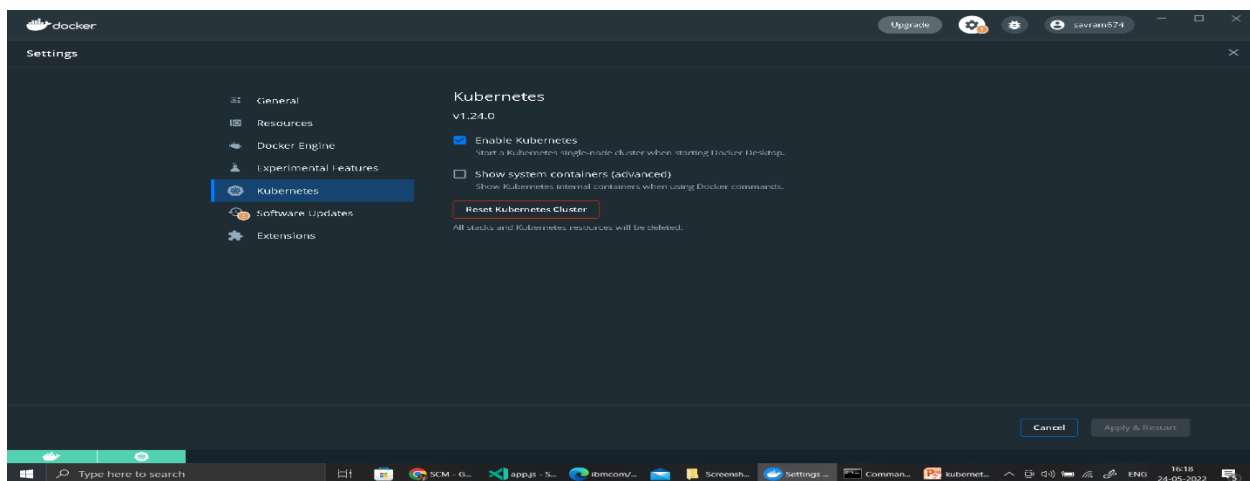
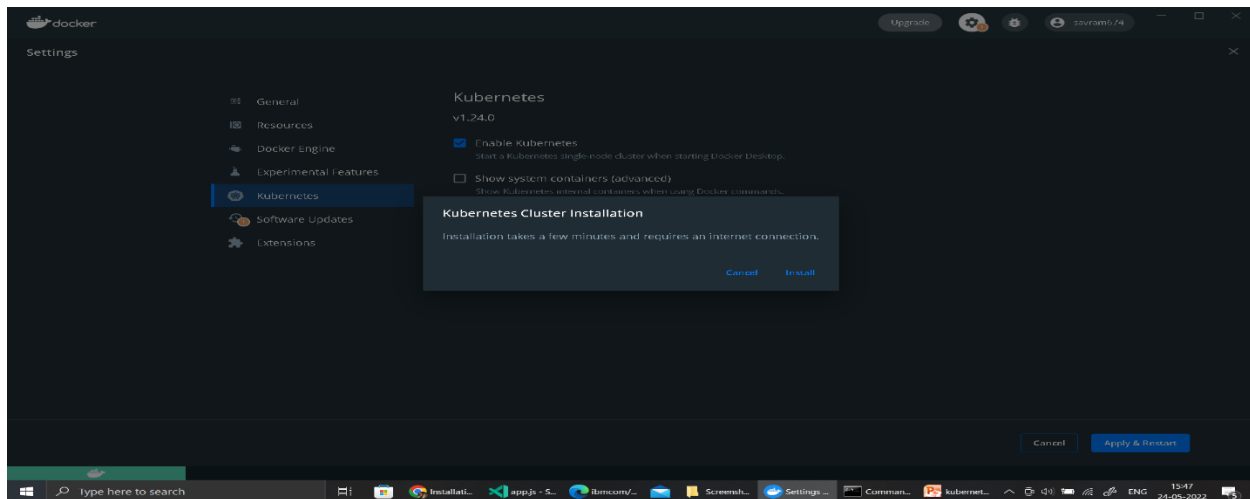
Recall from earlier that an orchestrator is a system that deploys and manages apps. You also learned a cluster is a set of computers that work together and are viewed as a single system. You use Kubernetes as the orchestration and cluster software to deploy your apps and respond to changes in compute resource needs.



Install using Docker Desktop

Settings → Kubernetes → Check Enable Kubernetes → Apply





[Minikube](#) is a tool that allows you to run a single-node Kubernetes cluster inside a virtual machine, on your own computer.

```
C:\Users\91984>docker run --rm -v $(pwd)/tmp:/tmp ibmcom/minikube-ppc64le:0.30.0 sh -c "cp /minikube /tmp"
Unable to find image 'ibmcom/minikube-ppc64le:0.30.0' locally
0.30.0: Pulling from ibmcom/minikube-ppc64le
4af4b9131647: Pull complete
714eea448cf8: Pull complete
Digest: sha256:eee5c4cd37e9b0087cc74fbf618766c2987d823d306cd63ddd70b350c777aaab
Status: Downloaded newer image for ibmcom/minikube-ppc64le:0.30.0
docker: Error response from daemon: create $(pwd)/tmp: "$(pwd)/tmp" includes invalid characters for a local volume name, only "[a-zA-Z0-9_@.:-]" is supported.
See 'docker run --help'.

C:\Users\91984>
C:\Users\91984>
C:\Users\91984>docker image ls
REPOSITORY          TAG
nodeimg              2.0
savnam674/nodeimg   latest
<none>               <none>
nodeimg              1.0
v1.24.0
```

[Minikube](#) is a tool that allows you to run a single-node Kubernetes cluster inside a virtual machine, on your own computer.

Download the required minikube installer for the windows platform.

1 Installation

Click on the buttons that describe your target platform. For other architectures, see [the release page](#) for a complete list of minikube binaries.

Operating system:

Architecture:

Release type:

Installer type:

To install the latest minikube **stable** release on **x86-64 Windows** using **.exe download**:

1. Download and run the installer for the [latest release](#).

Run the installer to complete the installation process

Set the path for **minikube** in the Environment variables

Start your cluster by running the following command in powershell

> minikube start [Docker service should be running]

```
Windows PowerShell
PS C:\Users\91984> minikube start
* minikube v1.25.2 on Microsoft Windows 10 Home Single Language 10.0.19043 Build 19043
* Automatically selected the docker driver
* Starting control plane node minikube in cluster minikube
* Pulling base image ...
* Downloading Kubernetes v1.23.3 preload ...
  > gcr.io/k8s-minikube/kicbase: 12.51 MiB / 379.06 MiB   3.30% 181.80 KiB p/s
```

To check which ip address minikube is running type

> minikube ip

Then type `kubectl get po -A` to see the nodes

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> minikube ip
! Executing "docker container inspect minikube --format={{.State.Status}}" took an unusually long time: 15.5115666s
* Restarting the docker service may improve performance.
192.168.49.2
PS C:\WINDOWS\system32> kubectl get po -A
NAMESPACE      NAME                                READY   STATUS    RESTARTS   AGE
kube-system    coredns-64897985d-wx4sz            1/1     Running   0           5m13s
kube-system    etcd-minikube                      1/1     Running   0           5m51s
kube-system    kube-apiserver-minikube            1/1     Running   0           5m58s
kube-system    kube-controller-manager-minikube   1/1     Running   2 (6m7s ago) 6m6s
kube-system    kube-proxy-hhsvb                  1/1     Running   0           5m13s
kube-system    kube-scheduler-minikube            1/1     Running   0           5m51s
kube-system    storage-provisioner                1/1     Running   0           4m39s
PS C:\WINDOWS\system32>
```

Create a sample deployment and expose it on port 8080:.

`kubectl create deployment hello-minikube --image=k8s.gcr.io/echoserver:1.4`

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> minikube ip
! Executing "docker container inspect minikube --format={{.State.Status}}" took an unusually long time: 15.5115666s
* Restarting the docker service may improve performance.
192.168.49.2
PS C:\WINDOWS\system32> kubectl get po -A
NAMESPACE      NAME                                READY   STATUS    RESTARTS   AGE
kube-system    coredns-64897985d-wx4sz            1/1     Running   0           5m13s
kube-system    etcd-minikube                      1/1     Running   0           5m51s
kube-system    kube-apiserver-minikube            1/1     Running   0           5m58s
kube-system    kube-controller-manager-minikube   1/1     Running   2 (6m7s ago) 6m6s
kube-system    kube-proxy-hhsvb                  1/1     Running   0           5m13s
kube-system    kube-scheduler-minikube            1/1     Running   0           5m51s
kube-system    storage-provisioner                1/1     Running   0           4m39s
PS C:\WINDOWS\system32> kubectl create deployment hello-minikube --image=k8s.gcr.io/echoserver:1.4
deployment.apps/hello-minikube created
PS C:\WINDOWS\system32>
```

Create a sample deployment and expose it on port 8080:.

`kubectl expose deployment hello-minikube --type=NodePort --port=8080`

```
Administrator Windows PowerShell
PS C:\WINDOWS\system32> minikube ip
! Executing "docker container inspect minikube --format={{.State.Status}}" took an unusually long time: 15.5115666s
* Restarting the docker service may improve performance.
192.168.49.2
PS C:\WINDOWS\system32> kubectl get po -A
NAMESPACE      NAME                                     READY   STATUS    RESTARTS   AGE
kube-system     coredns-64897985d-wx4sz                1/1     Running   0           5m13s
kube-system     etcd-minikube                           1/1     Running   0           5m51s
kube-system     kube-apiserver-minikube                 1/1     Running   0           5m58s
kube-system     kube-controller-manager-minikube        1/1     Running   2 (6m7s ago) 6m6s
kube-system     kube-proxy-hhsvb                        1/1     Running   0           5m13s
kube-system     kube-scheduler-minikube                 1/1     Running   0           5m51s
kube-system     storage-provisioner                     1/1     Running   0           4m39s
PS C:\WINDOWS\system32> kubectl create deployment hello-minikube --image=k8s.gcr.io/echoserver:1.4
deployment.apps/hello-minikube created
PS C:\WINDOWS\system32> kubectl expose deployment hello-minikube --type=NodePort --port=8080
service/hello-minikube exposed
PS C:\WINDOWS\system32>
```

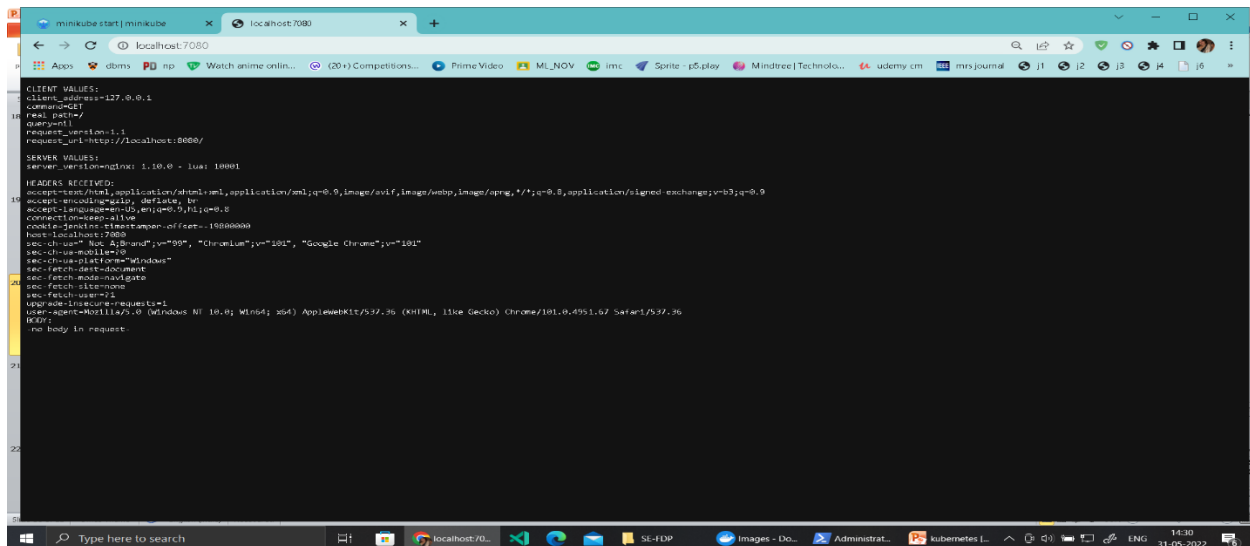
Use kubectl to forward the port to any other port

kubectl port-forward service/hello-minikube 7080:8080

```
Administrator Windows PowerShell
PS C:\WINDOWS\system32> minikube ip
! Executing "docker container inspect minikube --format={{.State.Status}}" took an unusually long time: 15.5115666s
* Restarting the docker service may improve performance.
192.168.49.2
PS C:\WINDOWS\system32> kubectl get po -A
NAMESPACE      NAME                                     READY   STATUS    RESTARTS   AGE
kube-system     coredns-64897985d-wx4sz                1/1     Running   0           5m13s
kube-system     etcd-minikube                           1/1     Running   0           5m51s
kube-system     kube-apiserver-minikube                 1/1     Running   0           5m58s
kube-system     kube-controller-manager-minikube        1/1     Running   2 (6m7s ago) 6m6s
kube-system     kube-proxy-hhsvb                        1/1     Running   0           5m13s
kube-system     kube-scheduler-minikube                 1/1     Running   0           5m51s
kube-system     storage-provisioner                     1/1     Running   0           4m39s
PS C:\WINDOWS\system32> kubectl create deployment hello-minikube --image=k8s.gcr.io/echoserver:1.4
deployment.apps/hello-minikube created
PS C:\WINDOWS\system32> kubectl expose deployment hello-minikube --type=NodePort --port=8080
service/hello-minikube exposed
PS C:\WINDOWS\system32> kubectl port-forward service/hello-minikube 7080:8080
Forwarding from 127.0.0.1:7080 -> 8080
Forwarding from [::1]:7080 -> 8080
```

Check your application on the browser at the port 7080

in browser open <http://localhost:7080/>



```
CLIENT VALUES:
client_address=127.0.0.1
command=GET
real_path=/
query=
request_version=1.1
request_uri=http://localhost:7080/

SERVER VALUES:
server_version=nginx: 1.10.0 - lua: 10001

HEADERS RECEIVED:
accept=text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
accept-encoding=gzip, deflate, br
accept-language=en-US,en;q=0.5,hi;q=0.8
connection=keep-alive
cookie=3rdpartycookie=10000000
host=localhost:7080
sec-chua="No_A_Sound";v="99", "Chromium";v="101", "Google Chrome";v="101"
sec-chua-mobile=?
sec-chua-platform="Windows"
sec-fetch-dest=document
sec-fetch-mode=navigate
sec-fetch-site=none
sec-fetch-user=?
upgrade-insecure-requests=1
user-agent=Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.67 Safari/537.36
body:
no body in request
```

Create a new app using Node.js and Mongodb for creating and updating user profile.

Steps : Create the required configuration files

mongo-config.yaml

mongo-secret.yaml

mongo.yaml

webapp.yaml

Mongo-config.yaml

apiVersion: v1

kind: ConfigMap

metadata:

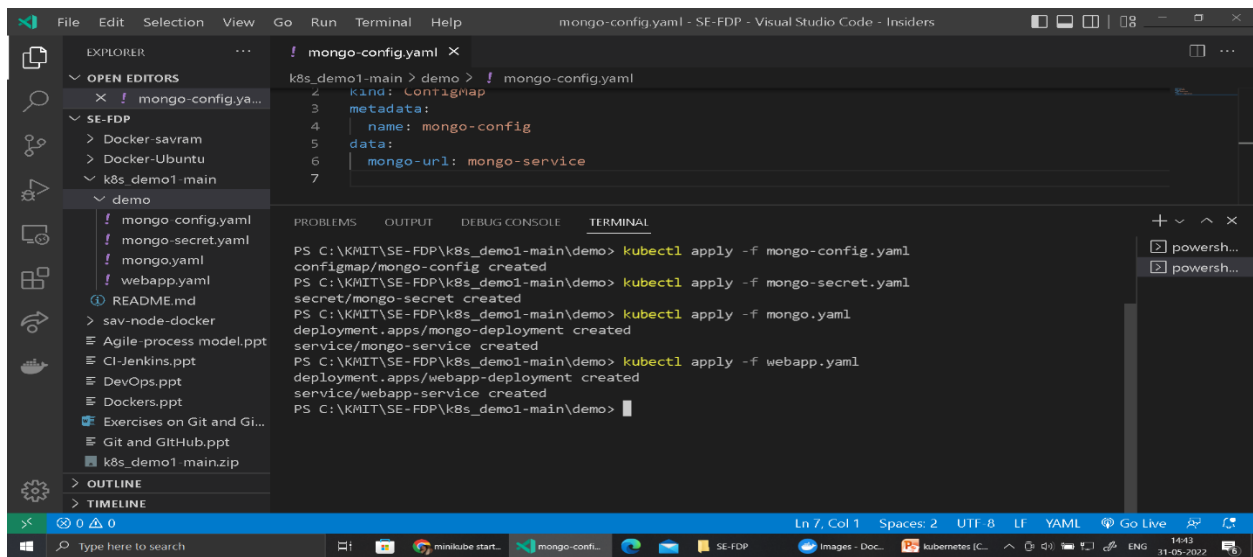
name: mongo-config

data:

mongo-url: mongo-service

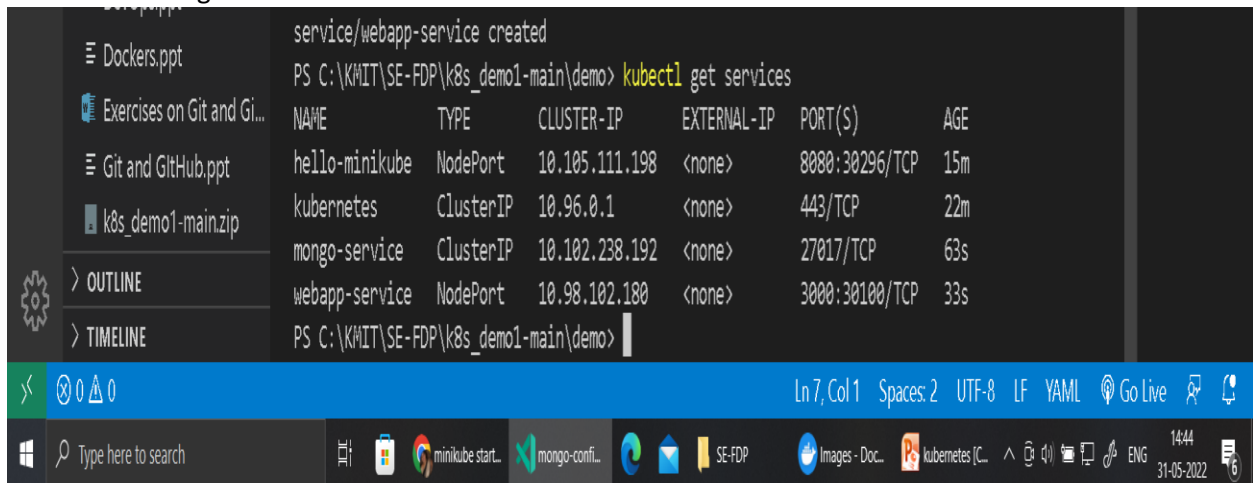
Run all the configuration files

> kubectl apply -f mongo-config.yaml



Check the services running using the command:

`kubectl get services`



Use kubectl to forward the port:

`kubectl port-forward service/webapp-service 6080:8080`

Visual Studio Code - Insiders

File Edit Selection View Go Run Terminal Help mongo.yaml - SE-FDP

EXPLORER

OPEN EDITORS

- ! mongo-config.ya...
- ! mongo.yaml k8s_d...

SE-FDP

- demo
- ! mongo-config.yaml
- ! mongo-secret.yaml
- ! mongo.yaml
- ! webapp.yaml
- README.md
- > sav node docker
- Agile process model.ppt
- CI-Jenkins.ppt
- DevOps.ppt
- Dockers.ppt
- Exercises on Git and Gi...
- Git and GitHub.ppt
- k8s_demo1_main.zip
- kubernetes.ppt

OUTLINE

TIMELINE

! mongo.yaml

```
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: mongo-deployment
5   labels:
6     app: mongo
7 spec:
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Handling connection for 9080
Handling connection for 9080
Handling connection for 9080
PS C:\KMIT\SE-FDP\k8s_demo1-main\demo> kubectl port-forward service/webapp-service 6080:3000
Forwarding from 127.0.0.1:6080 -> 3000
Forwarding from [::1]:6080 -> 3000
Handling connection for 6080
Handling connection for 6080
Handling connection for 6080
Handling connection for 6080
Handling connection for 6080
Handling connection for 6080
Handling connection for 6080
Handling connection for 6080

1 In 14, Col 14 Spaces: 2 UTF-8 LF YAMI Go Live

type here to search

localhost:6080

Apps dbms np Watch anime onlin... (20+) Competitions... Prime Video ML_NOV imc Sprite - p5.play Mindtree | Technolo... udemy cm

User profile



Name: **Narendra Modi**

Email: **ndmodi@india.gov.in**

Interests: **patriotism**

Edit Profile