



Department of Computer Science and Engineering
Cloud Architecture

Lab5A: Kubernetes using Minikube

UCID:2022301008

Student Name: Adnan Khan

Branch: Comps A

Objective:

Learn basic Kubernetes commands for resource inspection.

Understand the process of making deployed applications accessible both internally and externally through service exposure.

Learn to deploy servers like NGINX on Kubernetes pods using YAML for effective resource management.

Outcomes: After successful completion of the lab, students should be able to:

- [1] Gain proficiency in Kubernetes concepts such as pods, services, and deployments.
- [2] Acquire practical experience with Minikube, kubectl, and YAML file handling.
- [3] Develop skills in creating, managing, and exposing deployments and services within a Kubernetes cluster.
- [4] Access applications deployed in a Kubernetes cluster using various methods like port-forwarding and Minikube's service command, and learn to create YAML files to define Kubernetes resources with ease.
- [5] Confidently troubleshoot and solve issues within Kubernetes environments, thereby boosting technical expertise and preparedness for future obstacles.

System Requirements:

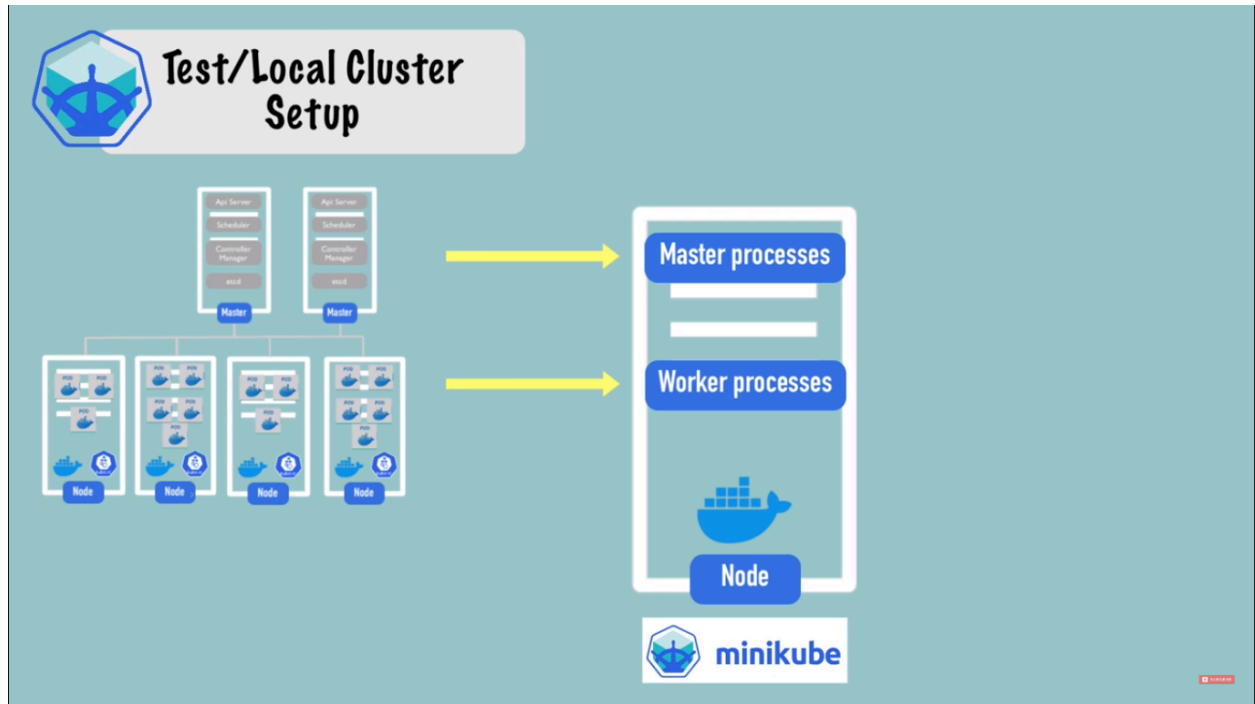
- 1. A computer running a Unix-based operating system (e.g., Ubuntu Linux, macOS).
- 2. Minikube for running applications with kubernetes(k8s)
- 3. Superuser (root) privileges or sudo access.
- 4. Internet connectivity for downloading VirtualBox VM (Ubuntu 22.04)

Introduction to Minikube

Watch the introductory videos on Minikube, Kubernetes available on youtube to understand the basics of Minikube and its capabilities. [1][2][3][4]

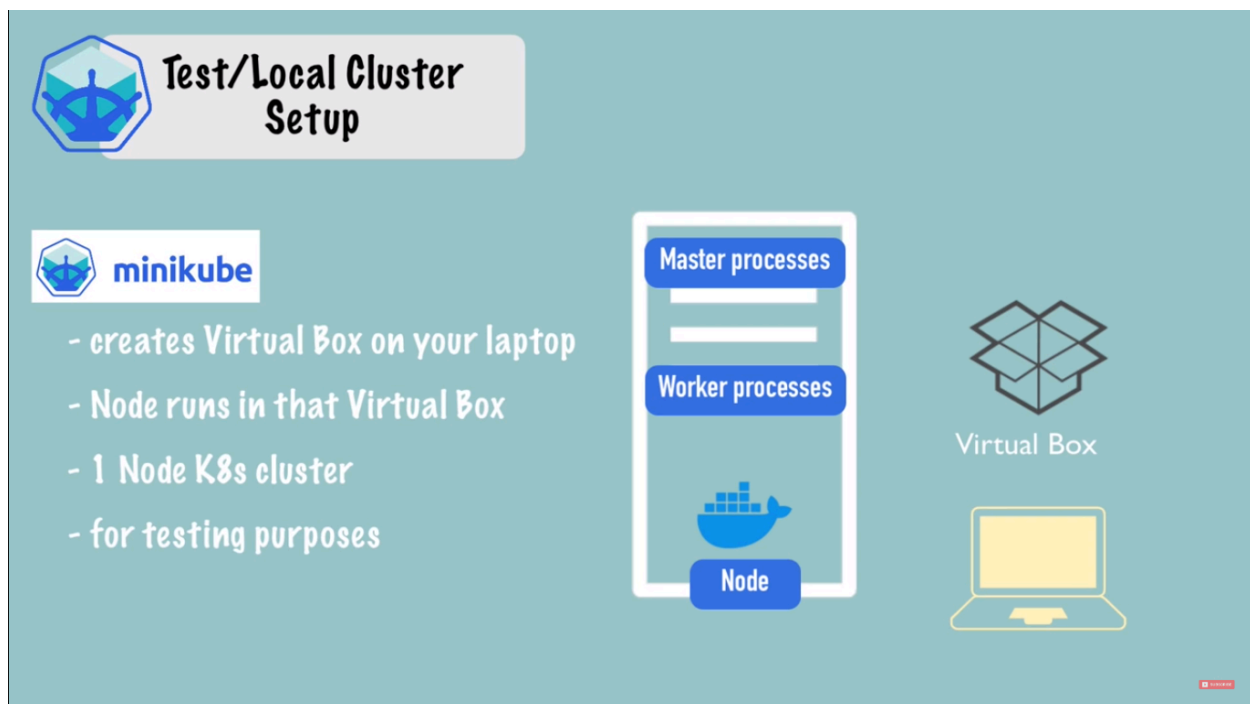
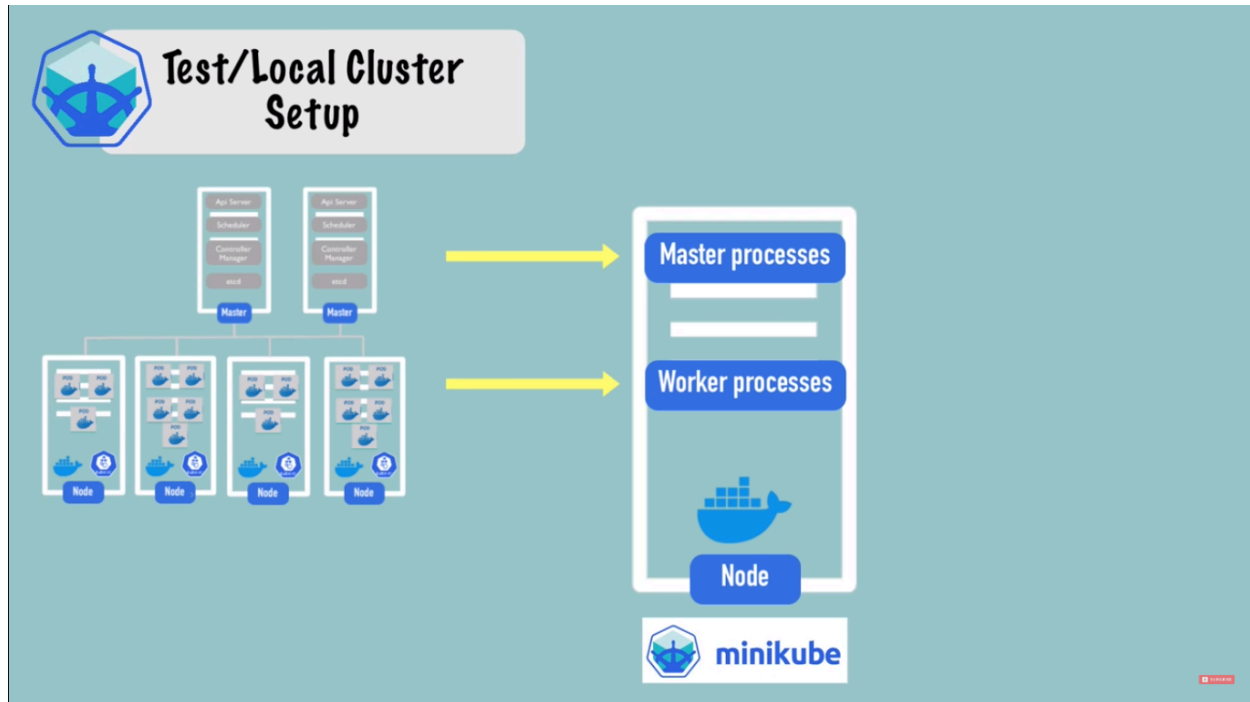


Department of Computer Science and Engineering
Cloud Architecture





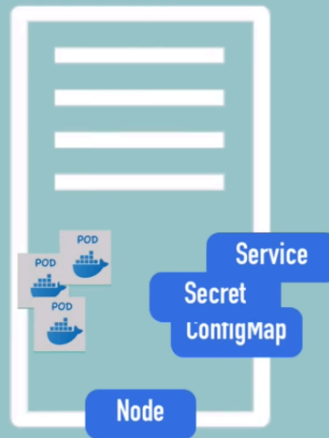
Department of Computer Science and Engineering
Cloud Architecture



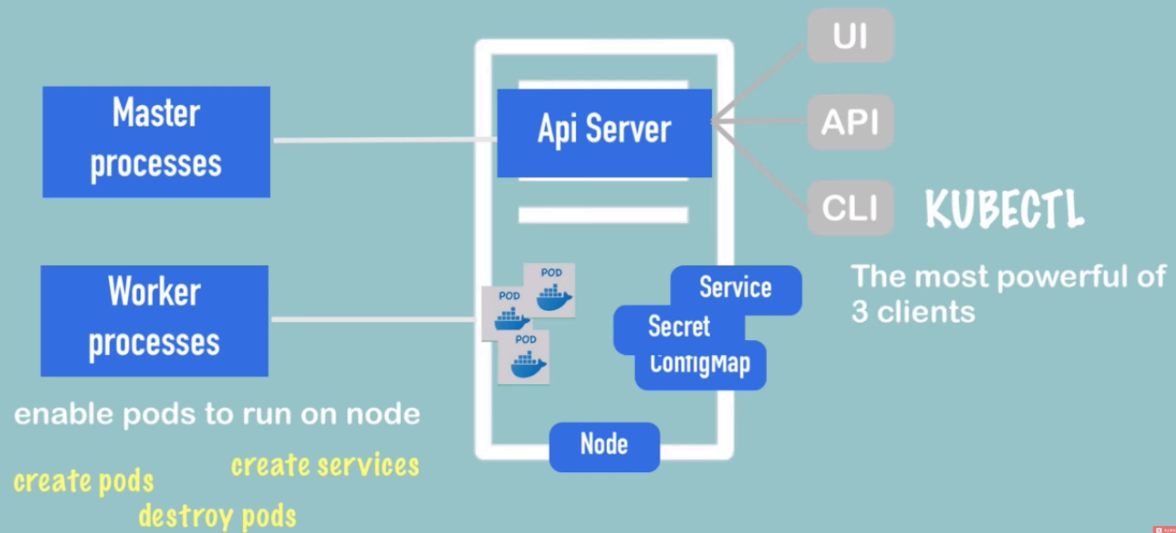


Department of Computer Science and Engineering
Cloud Architecture

What is kubectl?



What is kubectl?





Department of Computer Science and Engineering Cloud Architecture

Part-I: Minikube Installation and Setup

1. Visit the Minikube website and follow the instructions to download and install Minikube on your system.

[1]

<https://minikube.sigs.k8s.io/docs/start/>

To install the latest minikube stable release on x86-64 Linux using binary download:

`$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64`

`$ sudo install minikube-linux-amd64 /usr/local/bin/minikube && rm minikube-linux-amd64`

```
kali@kali: ~/Desktop

(kali@kali)-[~/Desktop]
└─$ docker --version
Docker version 20.10.25+dfsg1, build b82b9f3

(kali@kali)-[~/Desktop]
└─$ sudo install minikube-linux-amd64 /usr/local/bin/minikube && rm minikube-linux-amd64
[sudo] password for kali:
install: cannot stat 'minikube-linux-amd64': No such file or directory

(kali@kali)-[~/Desktop]
└─$ sudo install minikube-linux-amd64 /usr/local/bin/minikube && rm minikube-linux-amd64
install: cannot stat 'minikube-linux-amd64': No such file or directory

(kali@kali)-[~/Desktop]
└─$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
sudo install minikube-linux-amd64 /usr/local/bin/minikube && rm minikube-linux-amd64

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100  99.0M  100  99.0M    0     0  8973k      0  0:00:11  0:00:11 --:--:-- 10.9M

(kali@kali)-[~/Desktop]
└─$ minikube start

🐳 minikube v1.34.0 on Debian kali-rolling
💡 Unable to pick a default driver. Here is what was considered, in preference order:
   • docker: Not healthy: "docker version --format {{.Server.Os}}-{{.Server.Version}}:{{.Server.Platform.Name}}}" exit status 1: permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/version": dial unix /var/run/docker.sock: connect: permission denied
   • docker: Suggestion: Add your user to the 'docker' group: 'sudo usermod -aG docker $USER && newgrp docker' <https://docs.docker.com/engine/install/linux-postinstall/>
   • virtualbox: Not healthy: warning from virtualbox WARNING: The character device /dev/vboxdrv does not exist.

Please install the virtualbox-dkms package and the appropriate headers, most likely linux-headers-.

You will not be able to start VMs until this problem is fixed.
7.0.14_Debianr161095

   • virtualbox: Suggestion: Read the docs for resolution <https://minikube.sigs.k8s.io/docs/reference/drivers/virtualbox/>
💡 Alternatively you could install one of these drivers:
   • kvm2: Not installed: exec: "virsh": executable file not found in $PATH
   • podman: Not installed: exec: "podman": executable file not found in $PATH
   • qemu2: Not installed: stat /usr/share/OVMF/OVMF_CODE.fd: no such file or directory

❌ Exiting due to DRV_NOT_HEALTHY: Found driver(s) but none were healthy. See above for suggestions how to fix installed drivers.

(kali@kali)-[~/Desktop]
```



Department of Computer Science and Engineering Cloud Architecture

Start your cluster

From a terminal with administrator access (but not logged in as root), run:

\$minikube start

```
techz@techz:~$ sudo usermod -aG docker $USER
techz@techz:~$ newgrp docker
techz@techz:~$ minikube start
minikube v1.34.0 on Ubuntu 20.04
🌟 Automatically selected the docker driver. Other choices: ssh, none
🔧 Using Docker driver with root privileges
👉 Starting "minikube" primary control-plane node in "minikube" cluster
📦 Pulling base image v0.0.45 ...
📦 Downloading Kubernetes v1.31.0 preload ...
> gcr.io/k8s-minikube/kicbase...: 487.90 MiB / 487.90 MiB 100.00% 7.57 Mi
> preloaded-images-k8s-v18-v1...: 326.69 MiB / 326.69 MiB 100.00% 4.31 Mi
🔧 Creating docker container (CPUs=2, Memory=3900MB) ...
🔧 Preparing Kubernetes v1.31.0 on Docker 27.2.0 ...
   ▪ 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: default-storageclass, storage-provisioner
🎉 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
techz@techz:~$ sudo docker attach a7ad30070bd9
```

Interact with your cluster

If you already have kubectl installed (see documentation), you can now use it to access your shiny new cluster:

\$kubectl get po -A

Alternatively, minikube can download the appropriate version of kubectl and you should be able to use it like this:

\$minikube kubectl -- get po -A

```
techz@techz:~$ kubectl get po -A
NAMESPACE   NAME                                     READY   STATUS    RESTARTS   AGE
kube-system  coredns-6f6b679f8f-fdkcx              1/1     Running   0           92s
kube-system  etcd-minikube                          1/1     Running   0           97s
kube-system  kube-apiserver-minikube                1/1     Running   0           97s
kube-system  kube-controller-manager-minikube       1/1     Running   0           97s
kube-system  kube-proxy-ht8gj                       1/1     Running   0           92s
kube-system  kube-scheduler-minikube                1/1     Running   0           97s
kube-system  storage-provisioner                    1/1     Running   1 (62s ago)  96s
techz@techz:~$
```

You can also make your life easier by adding the following to your shell config: (for more details see: kubectl)



Department of Computer Science and Engineering Cloud Architecture

```
techz@techz:~$ minikube kubectl -- get po -A
> kubectl.sha256: 64 B / 64 B [-----] 100.00% ? p/s 0s
> kubectl: 53.77 MiB / 53.77 MiB [-----] 100.00% 11.50 MiB p/s 4.9s
error: no server found for cluster "minikube"
techz@techz:~$
```

\$alias kubectl="minikube kubectl --"

Deploy applications

Create a sample deployment and expose it on port 8080:

\$kubectl create deployment hello-minikube --image=kicbase/echo-server:1.0

\$kubectl expose deployment hello-minikube --type=NodePort --port=8080

```
techz@techz:~$ kubectl create deployment hello-minikube --image=kicbase/echo-server:1.0
deployment.apps/hello-minikube created
techz@techz:~$ kubectl expose deployment hello-minikube --type=NodePort --port=8080
service/hello-minikube exposed
techz@techz:~$
```

It may take a moment, but your deployment will soon show up when you run:

\$kubectl get services hello-minikube

```
techz@techz:~$ kubectl get services hello-minikube
NAME          TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
hello-minikube NodePort    10.108.34.81 <none>        8080:32489/TCP   62s
techz@techz:~$
```

The easiest way to access this service is to let minikube launch a web browser for you:

\$minikube service hello-minikube

```
techz@techz:~$ minikube service hello-minikube
+-----+-----+-----+-----+
| NAMESPACE | NAME          | TARGET PORT | URL                               |
+-----+-----+-----+-----+
| default   | hello-minikube | 8080        | http://192.168.49.2:32489       |
+-----+-----+-----+-----+
🌐 Opening service default/hello-minikube in default browser...
```




Department of Computer Science and Engineering Cloud Architecture

```
Lab4A: Kubernetes using x | Adnan Khan - Lab-5A_Ku x | ChatGPT x | 192.168.49.2:32489 x +
< → ↻ ⚠ Not secure 192.168.49.2:32489
Request served by hello-minikube-7d48979fd6-trnrk
HTTP/1.1 GET /
Host: 192.168.49.2:32489
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=0.7
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/129.0.0.0 Safari/537.36
```

Alternatively, use kubectl to forward the port:

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

```
techz@techz:~$ Opening in existing browser session.
[52591:52591:0100/000000.675374:ERROR:zygote_linux.cc(678)] write: Broken pipe (32)
kubectl port-forward service/hello-minikube 7080:8080
Forwarding from 127.0.0.1:7080 -> 8080
Forwarding from [::1]:7080 -> 8080
```

Manage your cluster

Pause Kubernetes without impacting deployed applications:

\$minikube pause

```
techz@techz:~$ minikube pause
|| Pausing node minikube ...
>|| Paused 14 containers in: kube-system, kubernetes-dashboard, storage-gluster, istio-operator
```

Unpause a paused instance:

\$minikube unpause

```
techz@techz:~$ minikube unpause
|| Pausing node minikube ...
>|| Paused 14 containers in: kube-system, kubernetes-dashboard, storage-gluster, istio-operator
```

Halt the cluster:

\$minikube stop

```
techz@techz:~$ minikube stop
👋 Stopping node "minikube" ...
🔴 Powering off "minikube" via SSH ...
🔴 1 node stopped.
```




Department of Computer Science and Engineering Cloud Architecture

Change the default memory limit (requires a restart):

\$minikube config set memory 9001

```
techz@techz:~$ minikube config set memory 9001
! These changes will take effect upon a minikube delete and then a minikube start
```

Browse the catalog of easily installed Kubernetes services:

\$minikube addons list

```
techz@techz:~$ minikube addons list
```

ADDON NAME	PROFILE	STATUS	MAINTAINER
ambassador	minikube	disabled	3rd party (Ambassador)
auto-pause	minikube	disabled	minikube
cloud-spanner	minikube	disabled	Google
csi-hostpath-driver	minikube	disabled	Kubernetes
dashboard	minikube	disabled	Kubernetes
default-storageclass	minikube	enabled ✓	Kubernetes
efk	minikube	disabled	3rd party (Elastic)
freshpod	minikube	disabled	Google
gcp-auth	minikube	disabled	Google
gvisor	minikube	disabled	minikube
headlamp	minikube	disabled	3rd party (kinvolk.io)
helm-tiller	minikube	disabled	3rd party (Helm)
inaccel	minikube	disabled	3rd party (InAccel [info@inacel.com])
ingress	minikube	disabled	Kubernetes
ingress-dns	minikube	disabled	minikube
inspektor-gadget	minikube	disabled	3rd party (inspektor-gadget.io)
istio	minikube	disabled	3rd party (Istio)
istio-provisioner	minikube	disabled	3rd party (Istio)
kong	minikube	disabled	3rd party (Kong HQ)
kubeflow	minikube	disabled	3rd party
kubevirt	minikube	disabled	3rd party (KubeVirt)
logviewer	minikube	disabled	3rd party (unknown)
metallb	minikube	disabled	3rd party (MetalLB)
metrics-server	minikube	disabled	Kubernetes
nvidia-device-plugin	minikube	disabled	3rd party (NVIDIA)
nvidia-driver-installer	minikube	disabled	3rd party (NVIDIA)
nvidia-gpu-device-plugin	minikube	disabled	3rd party (NVIDIA)
olm	minikube	disabled	3rd party (Operator Framework)
pod-security-policy	minikube	disabled	3rd party (unknown)
portainer	minikube	disabled	3rd party (Portainer.io)
registry	minikube	disabled	minikube
registry-aliases	minikube	disabled	3rd party (unknown)



Department of Computer Science and Engineering Cloud Architecture

Create a second cluster running an older Kubernetes release:

\$minikube start -p aged --kubernetes-version=v1.16.1

```
techz@techz:~$ minikube start -p aged --kubernetes-version=v1.16.1
[aged] minikube v1.34.0 on Ubuntu 20.04
Specified Kubernetes version 1.16.1 is less than the oldest supported version: v1.20.0. Use 'minikube config defaults kubernetes-version' for details.
You can force an unsupported Kubernetes version via the --force flag
Exiting due to K8S_OLD_UNSUPPORTED: Kubernetes 1.16.1 is not supported by this release of minikube
techz@techz:~$
```

Delete all of the minikube clusters:

\$minikube delete --all

```
techz@techz:~$ minikube delete --all
Deleting "minikube" in docker ...
Removing /home/techz/.minikube/machines/minikube ...
Removed all traces of the "minikube" cluster.
Removed all traces of the "aged" cluster.
Successfully deleted all profiles
techz@techz:~$
```



Department of Computer Science and Engineering Cloud Architecture

Part-II:Run Nginx on Kubernetes Using Minikube

Refer to:

<https://medium.com/cloud-native-daily/how-to-run-nginx-on-kubernetes-using-minikube-df3319b80511>

```
$mkdir my_directory
$cd my_directory
Create a service yaml file
$nano service.yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
  labels:
    env: sandbox
spec:
  type: LoadBalancer
  ports:
    - port: 80
  selector:
    env: sandbox
```

Save it. (Ctrl+X)

Create deployment.yaml

```
$nano deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    env: sandbox
spec:
  replicas: 3
  selector:
    matchLabels:
      env: sandbox
  template:
```



Department of Computer Science and Engineering Cloud Architecture

```
metadata:  
  labels:  
    env: sandbox  
spec:  
  containers:  
  - name: nginx  
    image: nginx  
    ports:  
    - containerPort: 80
```

Save it. (Ctrl+X)

Start Minikube.

\$minikube start

Create the Kubernetes service.

\$kubectl create -f service.yaml

Create the Kubernetes deployment.

\$kubectl create -f deployment.yaml

Wait until the 3 deployment pods have been created. Once they state 'Running', they have successfully created.

```
techz@techz:~$ mkdir my_directory  
techz@techz:~$ cd my_directory  
techz@techz:~/my_directory$ nano service.yaml  
techz@techz:~/my_directory$ nano deployment.yaml  
techz@techz:~/my_directory$ minikube start  
🐳 minikube v1.34.0 on Ubuntu 20.04  
🔧 Automatically selected the docker driver. Other choices: none, ssh  
👉 Using Docker driver with root privileges  
👉 Starting "minikube" primary control-plane node in "minikube" cluster  
📦 Pulling base image v0.0.45 ...  
  
🔥 Creating docker container (CPUs=2, Memory=9001MB) ...  
🐳 Preparing Kubernetes v1.31.0 on Docker 27.2.0 ...  
   ■ 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  
techz@techz:~/my_directory$  
techz@techz:~/my_directory$ kubectl create -f service.yaml  
service/nginx-service created  
techz@techz:~/my_directory$ kubectl create -f service.yaml  
Error from server (AlreadyExists): error when creating "service.yaml": services "nginx-service" already exists  
techz@techz:~/my_directory$ kubectl get pods  
No resources found in default namespace.  
techz@techz:~/my_directory$ kubectl create -f deployment.yaml  
deployment.apps/nginx-deployment created
```



Department of Computer Science and Engineering

Cloud Architecture

\$kubectl get pods

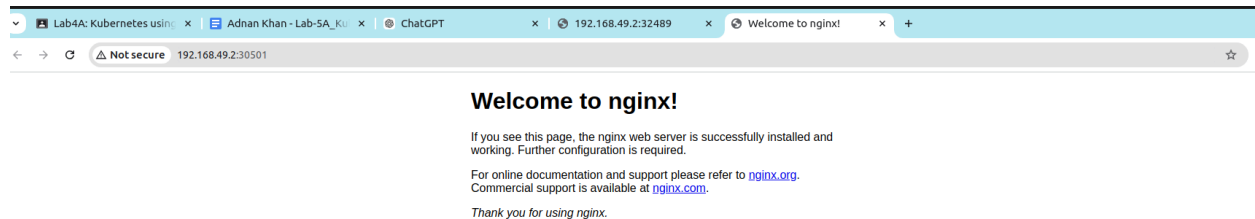
```
techz@techz:~/my_directory$ kubectl get pods
NAME                                READY   STATUS              RESTARTS   AGE
nginx-deployment-7d6f8cc7-6mqmz     0/1     ContainerCreating   0           5s
nginx-deployment-7d6f8cc7-ghjf4     0/1     ContainerCreating   0           5s
nginx-deployment-7d6f8cc7-hx6qv     0/1     ContainerCreating   0           5s
techz@techz:~/my_directory$ kubectl describe deployment nginx-deployment
Name:                               nginx-deployment
Namespace:                           default
CreationTimestamp:                   Wed, 18 Sep 2024 19:22:37 +0530
Labels:                               env=sandbox
Annotations:                           deployment.kubernetes.io/revision: 1
Selector:                             env=sandbox
Replicas:                             3 desired | 3 updated | 3 total | 0 available | 3 unavailable
StrategyType:                         RollingUpdate
MinReadySeconds:                       0
RollingUpdateStrategy:                 25% max unavailable, 25% max surge
Pod Template:
  Labels:                               env=sandbox
  Containers:
    nginx:
      Image:                             nginx
      Port:                               80/TCP
      Host Port:                           0/TCP
      Environment:                         <none>
      Mounts:                              <none>
      Volumes:                             <none>
      Node-Selectors:                       <none>
      Tolerations:                         <none>
Conditions:
  Type                               Status  Reason
  ---                               -
  Available                           False   MinimumReplicasUnavailable
  Progressing                         True    ReplicaSetUpdated
OldReplicaSets:                       <none>
NewReplicaSet:                         nginx-deployment-7d6f8cc7 (3/3 replicas created)
Events:
  Type    Reason              Age    From                      Message
  ---    -
  Normal  ScalingReplicaSet   12s    deployment-controller     Scaled up replica set nginx-deployment-7d6f8cc7 to 3
```



Department of Computer Science and Engineering Cloud Architecture

Assign the service URL with Minikube .

\$minikube service nginx-service



Your default web browser should open, showing you the ‘Welcome to nginx’ default homepage.

Clean up by deleting all resources.

\$minikube delete --all

```
techz@techz:~/my_directory$ minikube service nginx-service
|-----|
| NAMESPACE | NAME          | TARGET PORT | URL                      |
|-----|
| default   | nginx-service | 80          | http://192.168.49.2:30501 |
|-----|
🌐 Opening service default/nginx-service in default browser...
techz@techz:~/my_directory$ Opening in existing browser session.
[60765:60765:0100/000000.394951:ERROR:zygote_linux.cc(678)] write: Broken pipe (32)
techz@techz:~/my_directory$ minikube delete --all
🔥 Deleting "minikube" in docker ...
🔥 Removing /home/techz/.minikube/machines/minikube ...
💀 Removed all traces of the "minikube" cluster.
🔥 Successfully deleted all profiles
techz@techz:~/my_directory$
```



Department of Computer Science and Engineering Cloud Architecture

Grafana

```
techz@techz:~/my_directory$ kubectl create -f grafana-service.yaml
service/grafana created
techz@techz:~/my_directory$ kubectl get pods
NAME                                READY   STATUS             RESTARTS   AGE
grafana-d79654855-l655n            0/1     ContainerCreating   0           9s
techz@techz:~/my_directory$ kubectl get services
NAME          TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
grafana       NodePort    10.105.7.25   <none>         80:31917/TCP     9s
kubernetes    ClusterIP   10.96.0.1     <none>         443/TCP          107s
techz@techz:~/my_directory$
```




Department of Computer Science and Engineering

Cloud Architecture

Conclusion:

In this lab, I got to learn how to use Minikube and Kubernetes by setting up a simple NGINX website. It was a great experience that helped me understand how to manage applications using configuration files called YAML. Seeing how Kubernetes organizes and runs different applications made everything click for me.

I also practiced different ways to make my application available to users, which was both fun and a bit challenging. Overall, this experience really built my confidence in working with cloud technologies, and I feel much more ready to take on similar projects in the future.

References:

- [1] <https://minikube.sigs.k8s.io/docs/start/>
- [2] <https://medium.com/cloud-native-daily/how-to-run-nginx-on-kubernetes-using-minikube-df3319b80511>
- [3] https://youtu.be/s_o8dwzRlu4
- [4] <https://youtu.be/E2pP1MOfo3g>

List of commands on my setup: history command

```
sudo apt update
21 curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
22 sudo install minikube-linux-amd64 /usr/local/bin/minikube && rm minikube-linux-amd64
28 minikube start
29 sudo chmod 777 /var/run/docker.sock
30 minikube start
31 kubectl get po -A
32 docker ps
33 docker ps -a
34 docker ps -aq
35 docker ps -a
```



Department of Computer Science and Engineering
Cloud Architecture

```
36 minikube kubectl -- get po -A
37 docker ps -a
38 alias kubectl="minikube kubectl --"
39 minikube dashboard
40 sudo minikube dashboard
41 sudo minikube start
42 minikube dashboard
43 kubectl create deployment hello-minikube --image=kicbase/echo-server:1.0
44 kubectl expose deployment hello-minikube --type=NodePort --port=8080
45 kubectl get services hello-minikube
46 minikube service hello-minikube
47 kubectl port-forward service/hello-minikube 7080:8080
48 ifconfig
49 kubectl port-forward service/hello-minikube 7080:8080
50 minikube kubectl -- get pods
51 mkdir my_directory
52 cd my_directory/
53 nano service.yaml
54 nano deployment.yaml
55 kubectl create -f service.yaml
56 kubectl create -f deployment.yaml
57 Kubectl get pods
58 kubectl get pods
59 minikube service nginx-service
60 kubectl get pods
61 docker ps -a
```