

Kubernetes Lab

To use Kubernetes minimum requirement is 2 cpu , but in our aws free tier account we get only t2 micro ec2 instance which has only 1 cpu. For 2 cpu machine we have to pay to aws.

That's why for this lab we will setting up kubernetes on our local Machines. (Laptop)

We will use minikube to setup single node kubernetes cluster.

Steps:-

1. Download oracle vm to your windows or mac machines.

<https://www.oracle.com/in/virtualization/technologies/vm/downloads/virtualbox-downloads.html>

The latest release is version 6.1.34.

- Oracle VM VirtualBox Base Packages - 6.1.34
- Oracle VM VirtualBox Extension Pack
- Source Code for Oracle VM VirtualBox Base Packages
- Oracle VM VirtualBox Pre-built Appliances
- Oracle Vagrant Boxes for Oracle VM VirtualBox - GitHub
- Programming Guide and Reference (PDF)
- VBox GuestAdditions

Oracle VM VirtualBox Base Packages - 6.1.34

Freely available for Windows, Mac OS X, Linux and Solaris x86 platforms under GPLv2:

Platform	64-bit
<u>Windows</u>	↓ Windows Installer
<u>Mac OS X</u>	↓ dmg Image
Solaris 10 5/08 and later or Solaris 11	↓ Solaris Package

Linux Platforms

[Chat with sales](#) [Contact or call](#) [×](#)

2. Install oracle vm and verify like this.

Recycle Bin desktop1 Microsoft Teams (wo...) WinSCP UltraViewer

jenkins Nox Acrobat Reader DC 164873690... streak123

Internet Download... VSDC Free Video Editor FortiClient VPN KishanRayD...

Visual Studio Code AnyDesk test mail

OBS Studio Remote Mouse TrafficMonit... kubernetes... Barrier

VLC media player Avast Free Antivirus TrafficMonit... requirement streak devops

VMware Workstati... Avast Secure Browser Oracle VM VirtualBox

Mouse Witho... MongoDB... Nox Asst mongo

testqvwwe dockerfile aws

Search: orac

All Apps Documents Web More

Best match

Oracle VM VirtualBox App

Search the web

- orac - See web results
- Oracle - Computer Technology Corporation
- oracle support
- oracle live sql
- oracle login
- oracle java
- oracle database
- oracle virtualbox
- oracle download

Folders (1+)

Oracle VM VirtualBox App

- Open
- Run as administrator
- Open file location
- Pin to Start
- Pin to taskbar
- Uninstall

ENG IN 19:50 27-07-2022

3. Visit minikube website there you will see the minimum requirement for minikube

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

minikube start

minikube is local Kubernetes, focusing on making it easy to learn and develop for Kubernetes.

All you need is Docker (or similarly compatible) container or a Virtual Machine environment, and Kubernetes is a single command away: `minikube start`

What you'll need

- 2 CPUs or more
- 2GB of free memory
- 20GB of free disk space
- Internet connection
- Container or virtual machine manager, such as: [Docker](#), [Hyperkit](#), [Hyper-V](#), [KVM](#), [Parallels](#), [Podman](#), [VirtualBox](#), or [VMware Fusion/Workstation](#)

For windows installation

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	Linux	macOS	Windows
Architecture	x86-64		
Release type	Stable Beta		
Installer type	.exe download Windows Package Manager Chocolatey		

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

1. Download and run the installer for the latest release.

Or if using `PowerShell`, use this command:

```
New-Item -Path 'c:\' -Name 'minikube' -ItemType Directory -Force
Invoke-WebRequest -OutFile 'c:\minikube\minikube.exe' -Uri 'https://github.com/kubernetes/minikube/releases/latest/download/minikube.exe'
```

2. Add the `minikube.exe` binary to your `PATH`.

Make sure to run PowerShell as Administrator.

Download File Info


URL

Category


Save As

☐ Remember this path for "Programs" category

Description

 31.00 MB

Installer Language

 Please select a language.



Welcome to the Kubernetes - Minikube Setup Wizard

This wizard will guide you through the installation of
Kubernetes - Minikube.

It is recommended that you close all other applications
before starting Setup. This will make it possible to update
relevant system files without having to reboot your
computer.

Click Next to continue.

Next >

Cancel

Choose Install Location

Choose the folder in which to install Kubernetes - Minikube.



Setup will install Kubernetes - Minikube in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder

C:\Program Files\Kubernetes\Minikube

Browse...

Space required: 71.5MB

Space available: 24.1GB

< Back

Install

Cancel



Installation Complete

Setup was completed successfully.



Completed

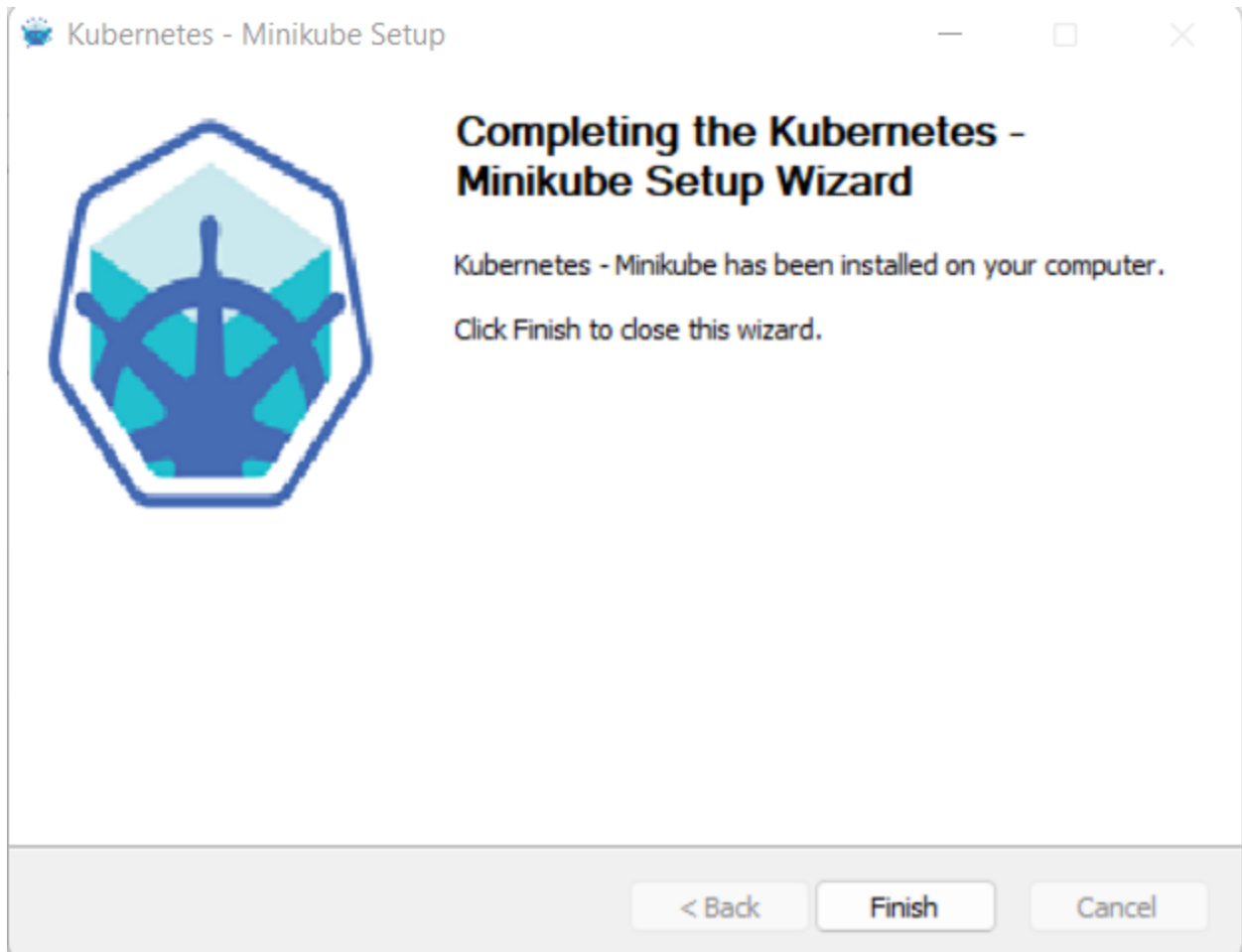


Show details

< Back

Next >

Cancel



For MacOS Installation

Run following command in terminal

```
curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-darwin-amd64  
sudo install minikube-darwin-amd64 /usr/local/bin/minikube
```

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

Linux

macOS

Windows

Architecture

x86-64

ARM64

Release type

Stable

Beta

Installer type

Binary download

Homebrew

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

```
curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-darwin-amd64
sudo install minikube-darwin-amd64 /usr/local/bin/minikube
```

[Edit](#)

[Create](#)

[Create](#)

[Issue](#)

What you

LoadBe

Take the

Now once installation is done , run this command in cmd {windows user} and terminal {macos},

to verify whether minikube installed or not.

```
minikube version
```

```
C:\Users\royki>minikube version
minikube version: v1.26.0
commit: f4b412861bb746be73053c9f6d2895f12cf78565
```

To start minikube

```
minikube start --driver=virtualbox --no-vtx-check
#--driver you have to use only one time
```

```
C:\Users\royki>minikube start --driver=virtualbox --no-vtx-check
* minikube v1.26.0 on Microsoft Windows 11 Home Single Language 10.0.22000 Build 22000
* Using the virtualbox driver based on existing profile
* Starting control plane node minikube in cluster minikube
* Downloading Kubernetes v1.24.1 preload ...
  > preloaded-images-k8s-v18-v1...: 8.16 MiB / 405.83 MiB  2.01% 1.51 MiB p/s
```

```
C:\Users\royki>minikube start --driver=virtualbox --no-vtx-check
* minikube v1.26.0 on Microsoft Windows 11 Home Single Language 10.0.22000 Build 22000
* Using the virtualbox driver based on existing profile
* Starting control plane node minikube in cluster minikube
* Downloading Kubernetes v1.24.1 preload ...
  > preloaded-images-k8s-v18-v1...: 405.83 MiB / 405.83 MiB 100.00% 1.52 MiB
* Creating virtualbox VM (CPUs=2, Memory=4500MB, Disk=20000MB) ...
! This VM is having trouble accessing https://k8s.gcr.io
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
```



```

C:\Users\royki>minikube start --driver=virtualbox --no-vtx-check
* minikube v1.26.0 on Microsoft Windows 11 Home Single Language 10.0.22000 Build 22000
* Using the virtualbox driver based on existing profile
* Starting control plane node minikube in cluster minikube
* Downloading Kubernetes v1.24.1 preload ...
  > preloaded-images-k8s-v18-v1...: 405.83 MiB / 405.83 MiB 100.00% 1.52 MiB
* Creating virtualbox VM (CPUs=2, Memory=4500MB, Disk=20000MB) ...
! This VM is having trouble accessing https://k8s.gcr.io
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
* Preparing Kubernetes v1.24.1 on Docker 20.10.16 ...
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Verifying Kubernetes components...
* Enabled addons: storage-provisioner, default-storageclass
* kubectl not found. If you need it, try: 'minikube kubectl -- get pods -A'
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
C:\Users\royki>

```

To install kubectl inside minikube

```
minikube kubectl get pods
```

```

C:\Users\royki>kubectl get pods
'kubectl' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\royki>minikube kubectl get pods
  > kubectl.exe.sha256: 64 B / 64 B [-----] 100.00% ? p/s 0s
  > kubectl.exe: 36.95 MiB / 44.14 MiB [----->_] 83.72% 1.67 MiB p/s ETA 4s_

```

some basic minikube commands

```

minikube status
minikube stop
minikube start

```

```

C:\Users\royki>minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

```

```
C:\Users\royki>minikube stop
* Stopping node "minikube" ...
* 1 node stopped.
```

```
C:\Users\royki>
```

```
C:\Users\royki>minikube start
* minikube v1.26.0 on Microsoft Windows 11 Home Single Language 10.0.22000 Build 22000
* Using the virtualbox driver based on existing profile
* Starting control plane node minikube in cluster minikube
* Restarting existing virtualbox VM for "minikube" ...
! This VM is having trouble accessing https://k8s.gcr.io
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
* Preparing Kubernetes v1.24.1 on Docker 20.10.16 ...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Verifying Kubernetes components...
* Enabled addons: default-storageclass, storage-provisioner
* kubectl not found. If you need it, try: 'minikube kubectl -- get pods -A'
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

```
C:\Users\royki>
```

To use kubectl everytime you have to first start minikube

```
C:\Users\royki>minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
```

```
C:\Users\royki>
```

To install kubectl (windows user only)

<https://kubernetes.io/docs/tasks/tools/install-kubectl-windows/>

latest compatible version of kubect helps avoid unforeseen issues.

Install kubectl on Windows

The following methods exist for installing kubectl on Windows:

- [Install kubectl binary with curl on Windows](#)
- [Install on Windows using Chocolatey or Scoop](#)



Install kubectl binary with curl on Windows

1. Download the [latest release v1.24.0.](#)

Or if you have `curl` installed, use this command:

```
curl -LO "https://dl.k8s.io/release/v1.24.0/bin/windows/amd64/kubectl.exe"
```

copy the downloaded file to .minikube folder in cdrive

> This PC > Downloads > Programs				
 Name	Date modified	Type	Size	
 kubectl	27-07-2022 20:42	Application	45,195 KB	

File Explorer window showing the contents of the path: This PC > Windows (C:) > Users > royki > .minikube

Name	Date modified	Type	Size
addons	27-07-2022 19:32	File folder	
cache	27-07-2022 20:37	File folder	
certs	27-07-2022 20:20	File folder	
config	27-07-2022 19:32	File folder	
files	27-07-2022 19:32	File folder	
logs	27-07-2022 20:17	File folder	
machines	27-07-2022 20:31	File folder	
profiles	27-07-2022 19:32	File folder	
ca	27-07-2022 20:35	Security Certificate	2 KB
ca.key	27-07-2022 20:35	KEY File	2 KB
ca	27-07-2022 21:02	PEM File	2 KB
cert	27-07-2022 21:02	PEM File	2 KB
key	27-07-2022 21:02	PEM File	2 KB
<u>kubectl</u>	27-07-2022 20:42	Application	45,195 KB
machine_client.lock	27-07-2022 20:20	LOCK File	0 KB
proxy-client-ca	27-07-2022 20:35	Security Certificate	2 KB
proxy-client-ca.key	27-07-2022 20:35	KEY File	2 KB

Now right click on kubectl and copy as path

Now search environment variable

System Properties



Computer Name Hardware **Advanced** System Protection Remote

You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage, and virtual memory

Settings...

User Profiles

Desktop settings related to your sign-in

Settings...

Startup and Recovery

System startup, system failure, and debugging information

Settings...

Environment Variables...

OK

Cancel

Apply

User variables for royki

Variable	Value
ChocolateyLastPathUpdate	132977898405548243
JD2_HOME	E:\downloader
node	E:\nodejs
OneDrive	C:\Users\royki\OneDrive
OneDriveConsumer	C:\Users\royki\OneDrive
Path	C:\Program Files\MySQL\MySQL Shell 8.0\bin\;C:\Users\royki\...
PyCharm Community Editi...	E:\pycharm\PyCharm Community Edition 2020.2\bin;
TEMP	C:\Users\royki\AppData\Local\Temp

New...

Edit...

Delete

System variables

Variable	Value
OnlineServices	Online Services
OS	Windows_NT
Path	C:\Program Files\Common Files\Oracle\Java\javapath;E:\pyth...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC;.PY;.PYW
platformcode	KV
PROCESSOR_ARCHITECTU...	AMD64
PROCESSOR_IDENTIFIER	AMD64 Family 23 Model 24 Stepping 1, AuthenticAMD
PROCESSOR_LEVEL	23

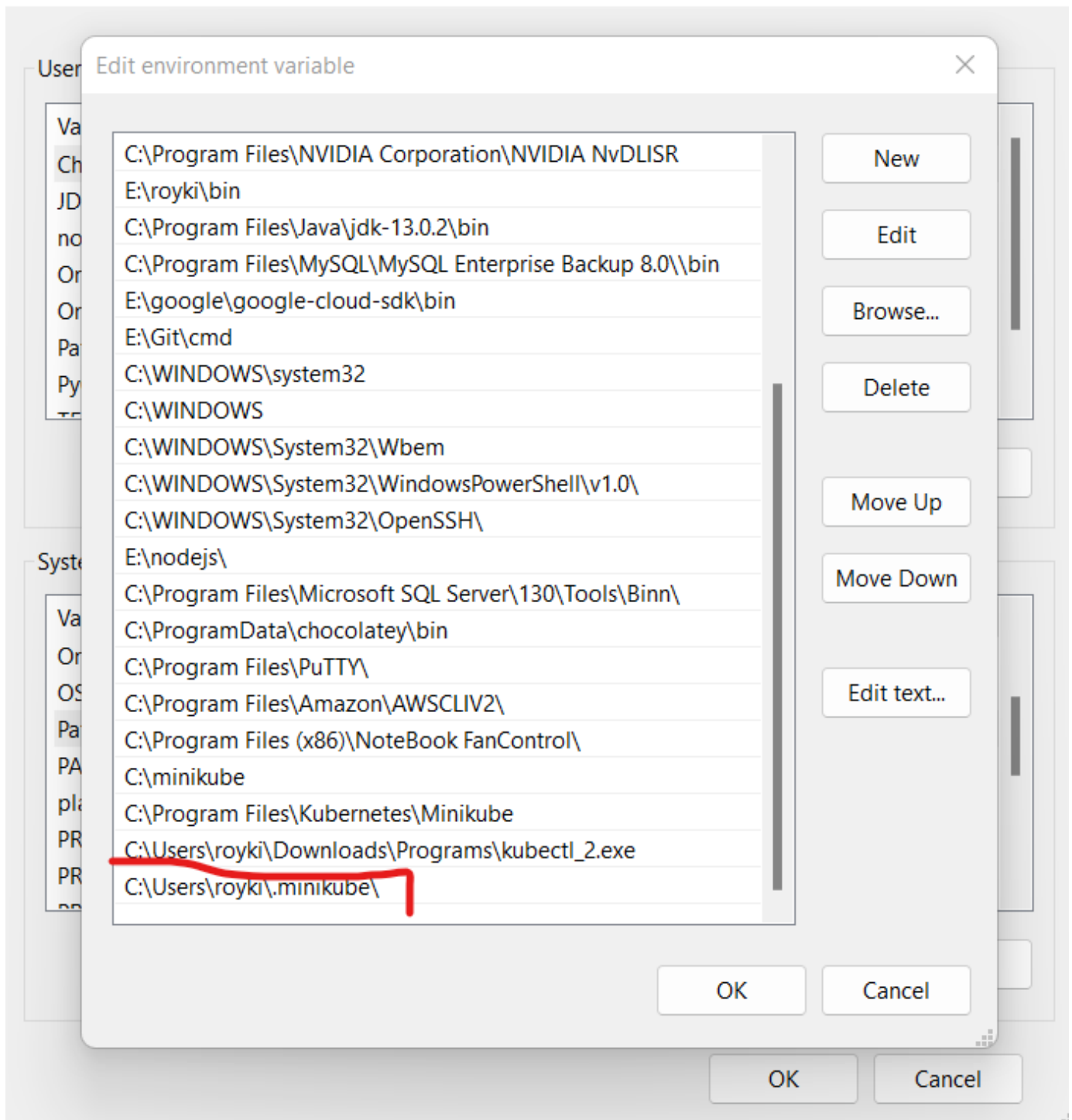
New...

Edit...

Delete

OK

Cancel



To check running pods

```
kubectl get pods
```

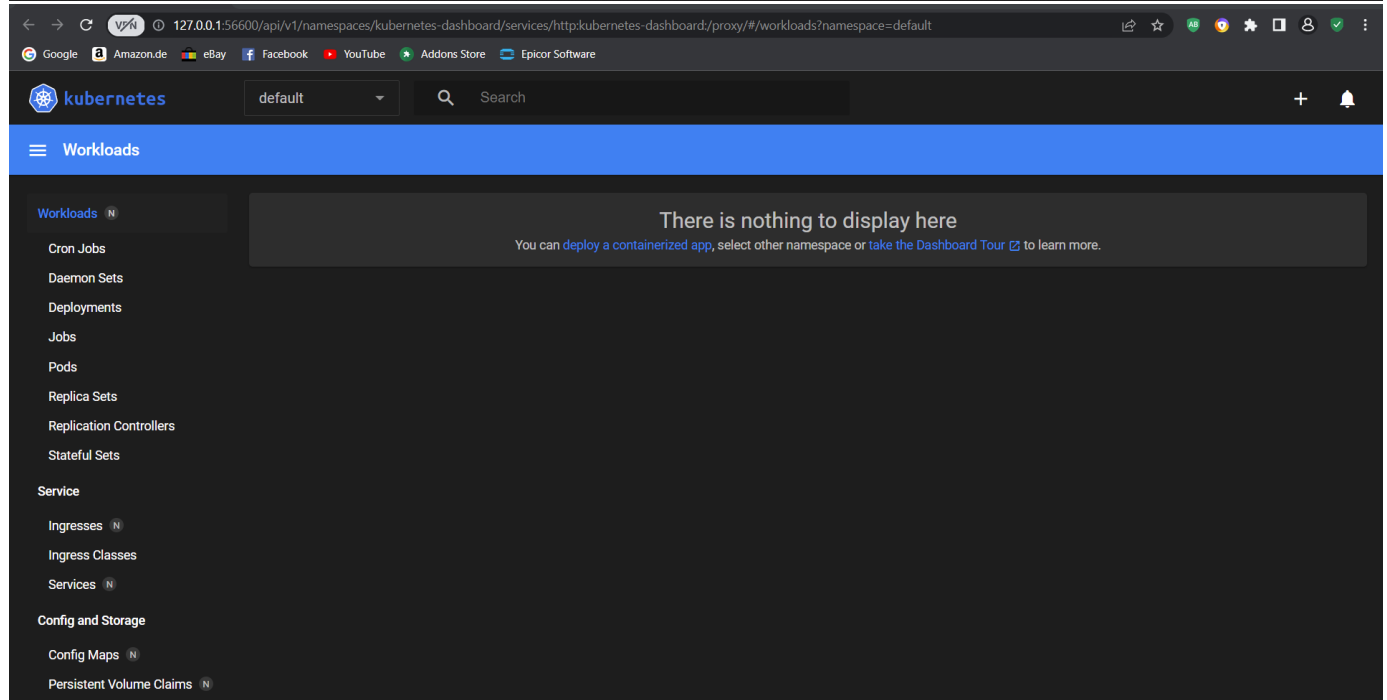
```
C:\Users\royki>kubectl get pods
No resources found in default namespace.

C:\Users\royki>
```

To see UI

```
minikube dashboard
```

```
C:\Users\royki>minikube dashboard
* Verifying dashboard health ...
* Launching proxy ...
* Verifying proxy health ...
* Opening http://127.0.0.1:56600/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ in your default browser...
```



To start a pod : We are using the same image which we created using our dockerfile

```
kubectl run myweb --image=roykishan8/kishanwebserver
```



```
C:\Users\royki>kubectl get pods
No resources found in default namespace.

C:\Users\royki>kubectl run myweb --image=roykishan8/kishanwebserver
pod/myweb created

C:\Users\royki>
```

```
C:\Users\royki>kubectl run myweb --image=roykishan8/kishanwebserver
pod/myweb created
```

```
C:\Users\royki>kubectl get pods
NAME      READY   STATUS             RESTARTS   AGE
myweb     0/1     ImagePullBackOff   0          5s
```

```
C:\Users\royki>kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
myweb     1/1     Running   0          71s
```

```
C:\Users\royki>
```

To delete a pod

```
kubectl delete pod myweb
kubectl get pods
```

```
C:\Users\royki>kubectl delete pods myweb
pod "myweb" deleted
```

```
C:\Users\royki>kubectl get pods
No resources found in default namespace.
```

```
C:\Users\royki>
```

To create a deployment

```
kubectl create deployment myd --image=roykishan8/kishanwebserver
kubectl get deployments
kubectl get pods
```

```
C:\Users\royki>kubect1 create deployment myd --image=roykishan8/kishanwebserver
deployment.apps/myd created

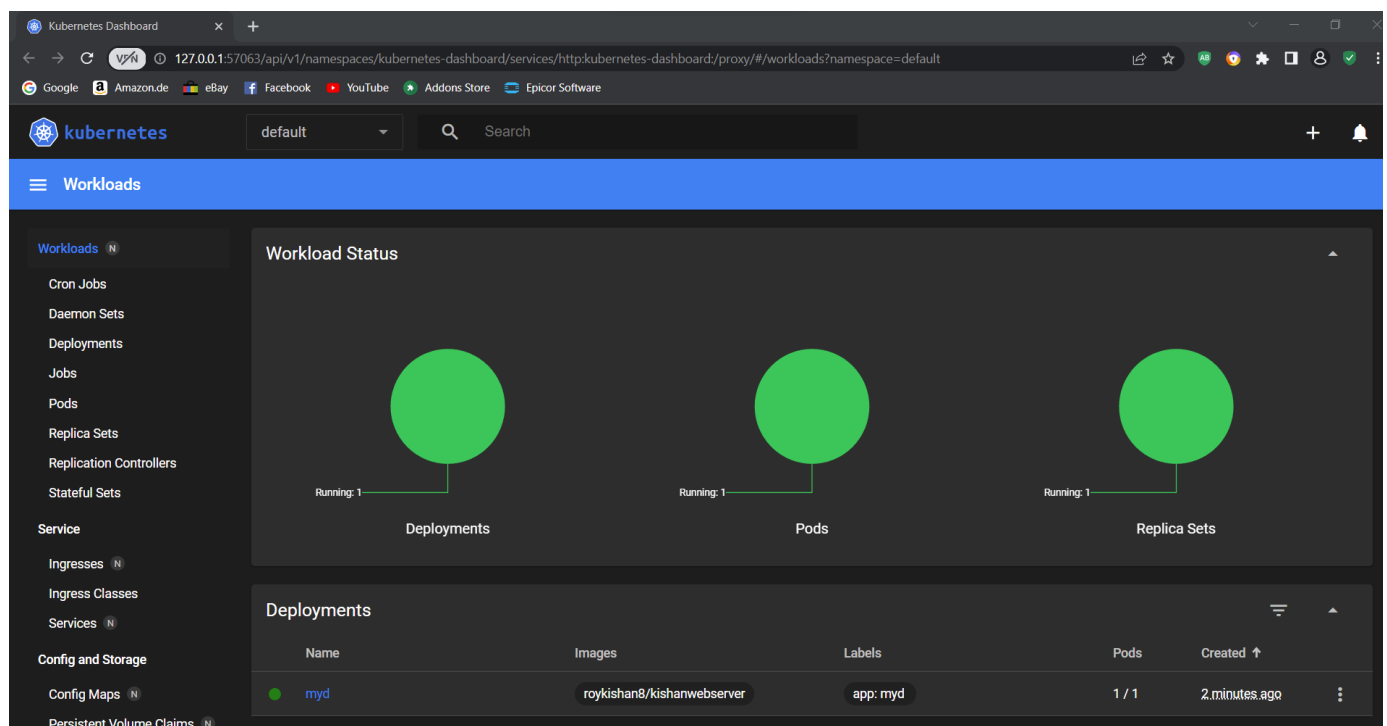
C:\Users\royki>kubect1 get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
myd       1/1     1            1           80s

C:\Users\royki>kubect1 get pods
NAME                               READY   STATUS    RESTARTS   AGE
myd-7787c768c5-rpwqm             1/1     Running   0          86s

C:\Users\royki>
```

To see this in webui

minikube dashboard



Now to create a loadbalancer to see the content of webserver, we have to create a service.

```
kubect1 expose deployment myd --port=80 --type=NodePort
kubect1 get svc
```

```
C:\Users\royki>kubectl expose deployment myd --port=80 --type=NodePort
service/myd exposed
```

```
C:\Users\royki>kubectl get svc
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	10m
myd	NodePort	10.100.44.247	<none>	80:30453/TCP	8s

```
C:\Users\royki>
```

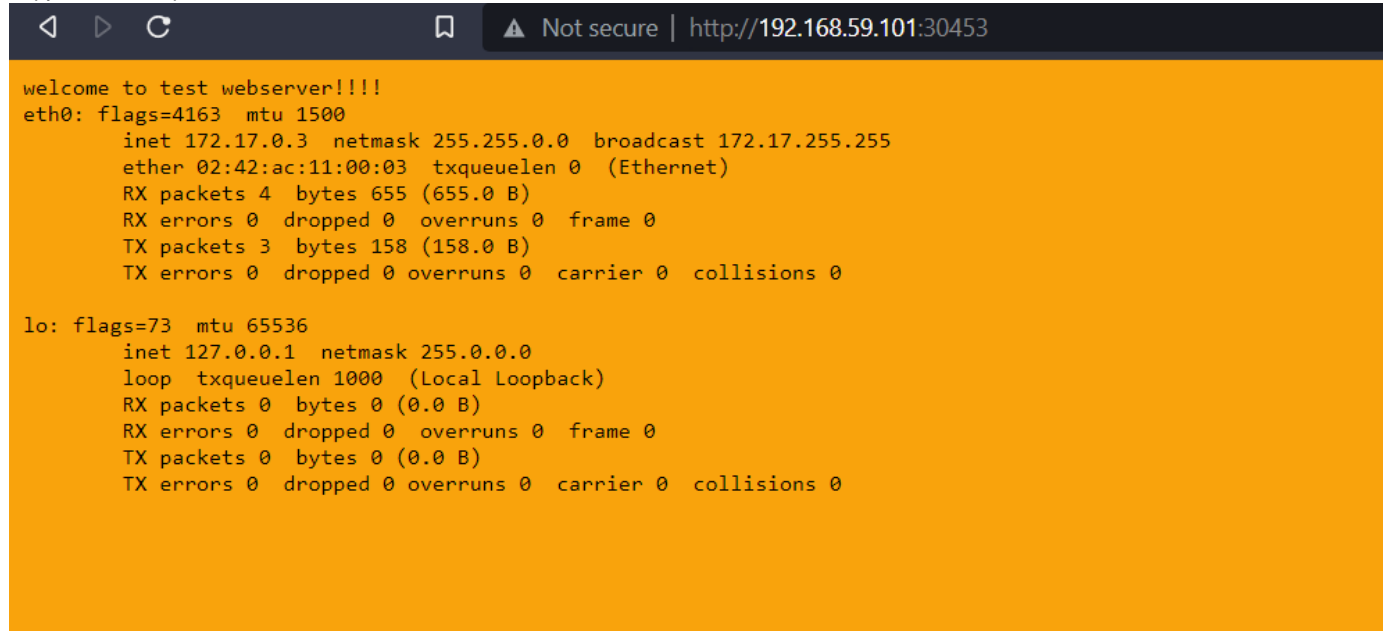
To get the weburl of webserver:

```
minikube service myd --url
```

```
C:\Users\royki>minikube service myd --url
http://192.168.59.101:30453
```

```
C:\Users\royki>
```

Copy this url and paste it in browser



The screenshot shows a web browser window with the address bar displaying "http://192.168.59.101:30453". The page content is a terminal output with an orange background. It starts with "welcome to test webserver!!!!" and then shows network statistics for the eth0 and lo interfaces.

```
welcome to test webserver!!!!
eth0: flags=4163  mtu 1500
    inet 172.17.0.3  netmask 255.255.0.0  broadcast 172.17.255.255
    ether 02:42:ac:11:00:03  txqueuelen 0  (Ethernet)
    RX packets 4  bytes 655 (655.0 B)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 3  bytes 158 (158.0 B)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    loop  txqueuelen 1000  (Local Loopback)
    RX packets 0  bytes 0 (0.0 B)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 0  bytes 0 (0.0 B)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Now our webserver is up.

To verify number of pods is running:

```
kubectl get pods
```

```
C:\Users\royki>kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
myd-7787c768c5-rpwqm               1/1     Running   0           7m39s

C:\Users\royki>
```

To scale the deployment so our webserver should always up, like creating three instance of webserver.

```
kubectl scale deployment myd --replicas=3
```

```
C:\Users\royki>kubectl scale deployment myd --replicas=3
deployment.apps/myd scaled

C:\Users\royki>
```

```
C:\Users\royki>kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
myd-7787c768c5-dsdf8               1/1     Running   0           23s
myd-7787c768c5-fsntm               1/1     Running   0           23s
myd-7787c768c5-rpwqm               1/1     Running   0           23m

C:\Users\royki>
```

The screenshot shows the Kubernetes dashboard interface. On the left is a navigation menu with categories like Workloads, Service, and Config and Storage. The main area displays the 'Workload Status' card with three green circles indicating the status of Deployments (Running: 1), Pods (Running: 3), and Replica Sets (Running: 1). Below this, the 'Deployments' table lists the 'myd' deployment with 3 pods running, created 24 minutes ago.

Name	Images	Labels	Pods	Created
myd	roykishan8/kishanwebserver	app: myd	3 / 3	24 minutes ago

Go to browser again and refresh the page



Not secure | http://192.168.59.101:30453

welcome to test webserver!!!!

eth0: flags=4163 mtu 1500

inet 172.17.0.7 netmask 255.255.0.0 broadcast 172.17.255.255

ether 02:42:ac:11:00:07 txqueuelen 0 (Ethernet)

RX packets 10 bytes 2412 (2.3 KiB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 9 bytes 3634 (3.5 KiB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73 mtu 65536

inet 127.0.0.1 netmask 255.0.0.0

loop txqueuelen 1000 (Local Loopback)

RX packets 0 bytes 0 (0.0 B)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 0 bytes 0 (0.0 B)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

welcome to test webserver!!!!

eth0: flags=4163 mtu 1500

inet 172.17.0.6 netmask 255.255.0.0 broadcast 172.17.255.255

ether 02:42:ac:11:00:06 txqueuelen 0 (Ethernet)

RX packets 5 bytes 723 (723.0 B)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 3 bytes 158 (158.0 B)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73 mtu 65536

inet 127.0.0.1 netmask 255.0.0.0

loop txqueuelen 1000 (Local Loopback)

RX packets 0 bytes 0 (0.0 B)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 0 bytes 0 (0.0 B)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```
← → ↻ V/N ⚠ Not secure | 192.168.59.101:30453
Google Amazon.de eBay Facebook YouTube Addons Store Epicor Software

welcome to test webserver!!!!
eth0: flags=4163 mtu 1500
    inet 172.17.0.3 netmask 255.255.0.0 broadcast 172.17.255.255
    ether 02:42:ac:11:00:03 txqueuelen 0 (Ethernet)
    RX packets 15 bytes 2070 (2.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 10 bytes 2017 (1.9 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73 mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

We are able to see that our webserver is running on three different container.

To delete all the resources we created inside kubernetes cluster

```
kubectl delete all --all
```

```
C:\Users\royki>kubectl delete all --all
pod "myd-7787c768c5-dsdf8" deleted
pod "myd-7787c768c5-fsntm" deleted
pod "myd-7787c768c5-rpwqm" deleted
service "kubernetes" deleted
service "myd" deleted
deployment.apps "myd" deleted
```

We can do same things using YAML file also.

To create a pod using YAML file:-

save this code with a name "pod.yaml"

```
apiVersion : v1
kind : Pod
metadata :
  name : "mypod1"
  labels :
    app : web
spec :
  containers :
    - name: "myc1"
      image : "roykishan8/kishanwebserver"
```

> This PC > DATA (D:) > DEVOPSTRaining > kubernetesWS

Name	Date modified	Type	Size
pod	28-07-2022 16:51	YAML File	1 KB

pod - Notepad

File Edit View

```
apiVersion : v1
kind : Pod
metadata :
  name : "mypod"
  labels :
    app : web
spec :
  containers :
    - name: "myc1"
      image : "roykishan8/kishanwebserver"
```

Open this location in cmd

```
D:\DEVOPSTRaining\kubernetesWS>dir
Volume in drive D is DATA
Volume Serial Number is 28A5-1DAC

Directory of D:\DEVOPSTRaining\kubernetesWS

28-07-2022  16:51    <DIR>          .
28-07-2022  16:48    <DIR>          ..
28-07-2022  16:51                163 pod.yaml
                1 File(s)                163 bytes
                2 Dir(s)  228,704,968,704 bytes free

D:\DEVOPSTRaining\kubernetesWS>
```

To run this "YAML" file

```
kubectl get pods
kubectl apply -f pod.yaml
kubectl get pods
```

To delete this pod

```
kubectl delete -f pod.yaml
kubectl get pods
```

```
D:\DEVOPSTRaining\kubernetesWS>kubectl delete -f pod.yaml
pod "mypod" deleted

D:\DEVOPSTRaining\kubernetesWS>kubectl get pods
No resources found in default namespace.

D:\DEVOPSTRaining\kubernetesWS>
```

To create a deployment using YAML file


```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mydeployment
spec:
  selector:
    matchLabels:
      app: mypod
  replicas: 2
  template:
    metadata:
      labels:
        app: mypod
    spec:
      containers:
      - name: myapp
        image: roykishan8/kishanwebserver
```

save this code with a name "deployment.yaml"

```
D:\DEVOPSTRaining\kubernetesWS>dir
Volume in drive D is DATA
Volume Serial Number is 28A5-1DAC

Directory of D:\DEVOPSTRaining\kubernetesWS

28-07-2022  17:08    <DIR>          .
28-07-2022  16:48    <DIR>          ..
28-07-2022  17:08                298 deployment.yaml
28-07-2022  16:51                163 pod.yaml
                2 File(s)                461 bytes
                2 Dir(s)  228,704,968,704 bytes free

D:\DEVOPSTRaining\kubernetesWS>
```

To run this deployment file

```
kubectl get deployent
kubectl apply -f deployment.yaml
kubectl get deployment
kubectl get pods
```

```

D:\DEVOPSTRAINING\kubernetesWS>kubectl get deployment
No resources found in default namespace.

D:\DEVOPSTRAINING\kubernetesWS>kubectl apply -f deployment.yaml
deployment.apps/mydeployment created

D:\DEVOPSTRAINING\kubernetesWS>kubectl get deployment
NAME                READY    UP-TO-DATE    AVAILABLE    AGE
mydeployment        0/3      3              0            6s

D:\DEVOPSTRAINING\kubernetesWS>kubectl get deployment
NAME                READY    UP-TO-DATE    AVAILABLE    AGE
mydeployment        3/3      3              3            33s

D:\DEVOPSTRAINING\kubernetesWS>kubectl get pods
NAME                                READY    STATUS    RESTARTS    AGE
mydeployment-86b79b5df6-7jp2n      1/1      Running   0            40s
mydeployment-86b79b5df6-h6mjh      1/1      Running   0            40s
mydeployment-86b79b5df6-mmd64      1/1      Running   0            40s

D:\DEVOPSTRAINING\kubernetesWS>

```

To create a service using YAML file

```

apiVersion: v1
kind: Service
metadata:
  name: myservice
spec:
  type: NodePort
  selector:
    app: mypod
  ports:
  - port: 8080
    targetPort: 80
    nodePort: 30000

```

save this code in a file with name "service.yaml"

To run this "service.yaml" file

```

kubectl get svc
kubectl apply -f service.yaml
kubectl get svc

```

```
D:\DEVOPSTRaining\kubernetesWS>kubect1 get svc
NAME          TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
kubernetes    ClusterIP     10.96.0.1     <none>         443/TCP    18m

D:\DEVOPSTRaining\kubernetesWS>kubect1 apply -f service.yaml
service/myservice created

D:\DEVOPSTRaining\kubernetesWS>kubect1 get svc
NAME          TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
kubernetes    ClusterIP     10.96.0.1     <none>         443/TCP    18m
myservice     NodePort      10.105.154.254 <none>         8080:30000/TCP 6s

D:\DEVOPSTRaining\kubernetesWS>
```


To get the loadbalancer/service url

```
minikube service myservice --url
```

```
D:\DEVOPSTRaining\kubernetesWS>minikube service myservice --url
http://192.168.59.101:30000

D:\DEVOPSTRaining\kubernetesWS>
```

Paste this url in browser

< > ↻  Not secure | http://192.168.59.101:30000

```
welcome to test webserver!!!!
eth0: flags=4163  mtu 1500
    inet 172.17.0.2  netmask 255.255.0.0  broadcast 172.17.255.255
    ether 02:42:ac:11:00:02  txqueuelen 0  (Ethernet)
    RX packets 4  bytes 655 (655.0 B)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 3  bytes 158 (158.0 B)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    loop txqueuelen 1000  (Local Loopback)
    RX packets 0  bytes 0 (0.0 B)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 0  bytes 0 (0.0 B)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Refresh the page

```
⏪ ⏩ ↺ ⏹ Not secure | http://192.168.59.101:30000

welcome to test webserver!!!!
eth0: flags=4163 mtu 1500
    inet 172.17.0.4 netmask 255.255.0.0 broadcast 172.17.255.255
    ether 02:42:ac:11:00:04 txqueuelen 0 (Ethernet)
    RX packets 6 bytes 785 (785.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 3 bytes 158 (158.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73 mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Refresh the page

```
⏪ ⏩ ↺ ⏹ Not secure | http://192.168.59.101:30000

welcome to test webserver!!!!
eth0: flags=4163 mtu 1500
    inet 172.17.0.3 netmask 255.255.0.0 broadcast 172.17.255.255
    ether 02:42:ac:11:00:03 txqueuelen 0 (Ethernet)
    RX packets 14 bytes 2620 (2.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 9 bytes 3636 (3.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73 mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

To delete all setup

```
kubectl delete all --all
```

To delete pod,deployment,service

```
kubectl delete pod <podname>
kubectl delete deployment <deploymentname>
kubectl delete svc <servicename>
```

To get more details of resources

```
kubectl describe pod <podname>
kubectl describe deployment <deploymentname>
kubectl describe service <servicename>
```

What is replicaset?

It is like autoscaling group of aws. Autoscaling groups keep eye on the number of ec2-instance running and if any ec2-instance terminate or fails it launches new ec2-instance and register that ec2-instance to autoscaling group.

Similar to this replicaset keep eyes on the number of replicas of pod and if any pod fails it try to launch new pod and then again attach that pod to replicaset. Whenever we create a deployment, replicaset automatically get created in backend which keeps eye on the the number of replicas of pod.

```
kubectl apply -f deployment.yaml
kubectl get deployments
kubectl get rs
kubectl get pods
```

```
D:\DEVOPSTRAINING\kubernetesWS>kubect1 apply -f deployment.yaml
deployment.apps/mydeployment created
```

```
D:\DEVOPSTRAINING\kubernetesWS>kubect1 get deployments
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
mydeployment	0/3	3	0	7s

```
D:\DEVOPSTRAINING\kubernetesWS>kubect1 get rs
```

NAME	DESIRED	CURRENT	READY	AGE
mydeployment-86b79b5df6	3	3	1	19s

```
D:\DEVOPSTRAINING\kubernetesWS>kubect1 get rs
```

NAME	DESIRED	CURRENT	READY	AGE
mydeployment-86b79b5df6	3	3	2	44s

```
D:\DEVOPSTRAINING\kubernetesWS>kubect1 get rs
```

NAME	DESIRED	CURRENT	READY	AGE
mydeployment-86b79b5df6	3	3	3	72s

```
D:\DEVOPSTRAINING\kubernetesWS>kubect1 get pods
```

NAME	READY	STATUS	RESTARTS	AGE
mydeployment-86b79b5df6-5v4nm	1/1	Running	0	78s
mydeployment-86b79b5df6-jbw6t	1/1	Running	0	78s
mydeployment-86b79b5df6-q55xr	1/1	Running	0	78s

```
D:\DEVOPSTRAINING\kubernetesWS>
```

over.....