

## Lab Assignment 10

**AIM:** To understand the Kubernetes Cluster Architecture, install and Spin Up a Kubernetes Cluster on Linux Machines/Cloud Platforms

**LO2.** To deploy single and multiple container applications and manage application deployments with rollouts in Kubernetes.

### **THEORY:**

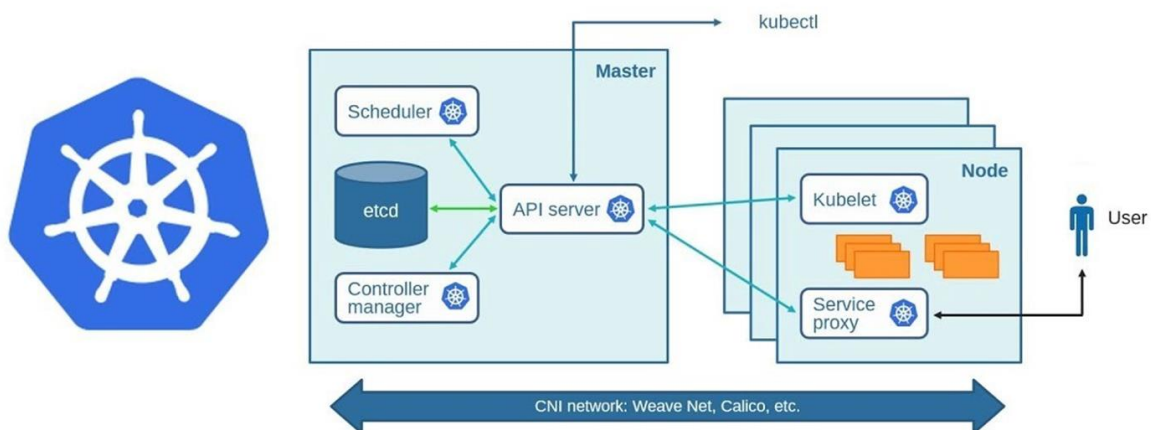
Kubernetes is an open-source container management tool that automates container deployment, scaling & load balancing.

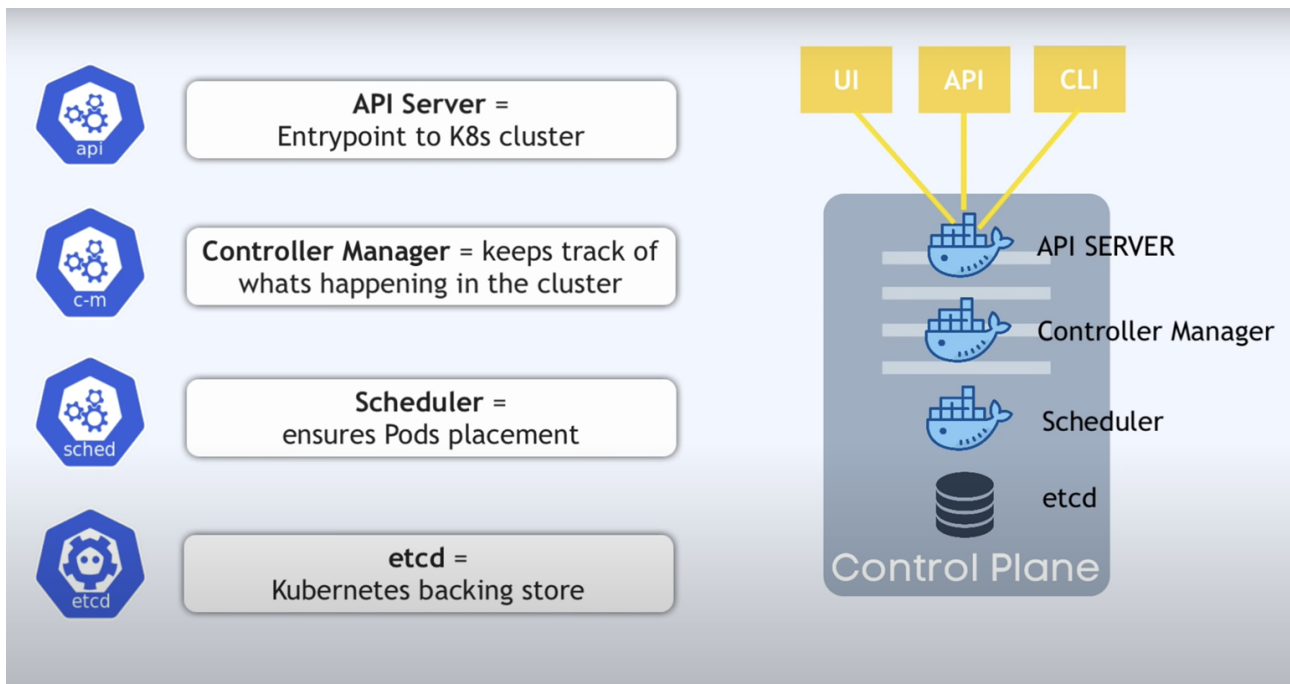
It schedules, runs, and manages isolated containers that are running on virtual/physical/cloud machines.

All top cloud providers support Kubernetes.

One popular name for Kubernetes is K8s.

### **ARCHITECTURE**





## Working with Kubernetes

- We create a Manifest (.yaml) file
- Apply those to cluster (to master) to bring it into the desired state.
- POD runs on a node, which is controlled by the master.

### ● Role of Master Node

- Kubernetes cluster contains containers running or Bare Metal / VM instances/cloud instances/ all mix.
- Kubernetes designates one or more of these as masters and all others as workers.
- The master is now going to run a set of K8s processes. These processes will ensure the smooth functioning of the cluster. These processes are called the 'Control Plane'.
- Can be Multi-Master for high availability.
- Master runs control plane to run cluster smoothly.

### ● Components of Control Plane

#### ■ Kube-api-server → (For all communications)

- This api-server interacts directly with the user (i.e we apply .yaml or .json manifest to kube-api-server)
- This kube-api-server is meant to scale automatically as per load.
- Kube-api-server is the front end of the control plane.

#### ■ etcd

- Stores metadata and status of the cluster.
- etcd is a consistent and high-available store (key-value-store)

- Source of truth for cluster state (info about the state of the cluster)

→ **etcd has the following features**

1. Fully Replicated → The entire state is available on every node in the cluster.
2. Secure → Implements automatic TLS with optional client-certificate authentication.
3. Fast → Benchmarked at 10,000 writes per second.

### ■ **Kube-scheduler (action)**

- When users request the creation & management of Pods, Kube-scheduler is going to take action on these requests.
- Handles POD creation and Management.
- Kube-scheduler match/assign any node to create and run pods.
- A scheduler watches for newly created pods that have no node assigned. For every pod that the scheduler discovers, the scheduler becomes responsible for finding the best node for that pod to run.
- The scheduler gets the information for hardware configuration from configuration files and schedules the Pods on nodes accordingly.

### ■ **Controller-Manager**

- Make sure the actual state of the cluster matches the desired state.

→ Two possible choices for controller manager—

1. If K8s is on the cloud, then it will be a cloud controller manager.
2. If K8s is on non-cloud, then it will be kube-controller-manager.

### **Components on the master that runs the controller**

**Node Controller** → For checking the cloud provider to determine if a node has been detected in the cloud after it stops responding.

**Route-Controller** → Responsible for setting up a network, and routes on your cloud.

**Service-Controller** → Responsible for load Balancers on your cloud against services of type Load Balancer.

**Volume-Controller** → For creating, attaching, and mounting volumes and interacting with the cloud provider to orchestrate volume.

### ■ **Nodes (Kubelet and Container Engine)**

- Node is going to run 3 important pieces of software/process.

#### **Kubelet**

- The agent running on the node.
- Listens to Kubernetes master (eg- Pod creation request).
- Use port 10255.
- Send success/Fail reports to master.

- ## Kube-Proxy

- ## INSTALLATION:

## 1. Install Docker

[illegible]

## T11 PRASAD AROTE 06

The image shows a terminal window titled "Activities Terminal" with a dark background. The terminal output displays the Docker installation process and usage instructions. It starts with a prompt "prasad@prasad-VirtualBox: ~" and shows the command "update" being executed, which updates the configuration of one or more containers. This is followed by a "wait" command, which blocks until one or more containers stop, then prints their exit codes. The output then shows the "Global Options:" section, which lists various command-line options for the Docker client, such as "--config", "--context", "--debug", "--host", "--log-level", "--tls", "--tlscacert", "--tlscert", "--tlskey", "--tlsverify", and "--version". The terminal then shows the command "Run 'docker COMMAND --help' for more information on a command." followed by a link to the Docker documentation. The next section shows the command "prasad@prasad-VirtualBox:~\$ sudo docker run hello-world" being executed, which results in the message "Unable to find image 'hello-world:latest' locally" and "latest: pulling from library/hello-world". The output then shows the command "719385e32844: Pull complete" and "Digest: sha256:88ecbaca3ec199d3b7eaf73580f4518c25f9d34f58ce9a0df68429c5af48e8d". The terminal then shows the command "Status: Downloaded newer image for hello-world:latest" and the output "Hello from Docker!". The final section shows the command "This message shows that your installation appears to be working correctly." followed by a list of steps to generate this message, such as "1. The Docker client contacted the Docker daemon." and "2. The Docker daemon pulled the 'hello-world' image from the Docker Hub." The terminal then shows the command "To try something more ambitious, you can run an Ubuntu container with:" followed by the command "\$ docker run -it ubuntu bash". The final section shows the command "Share images, automate workflows, and more with a free Docker ID:" followed by the link "https://hub.docker.com/". The terminal then shows the command "For more examples and ideas, visit:" followed by the link "https://docs.docker.com/get-started/". The terminal ends with the prompt "prasad@prasad-VirtualBox:~\$".

```
Activities Terminal Oct 14 22:11 • prasad@prasad-VirtualBox: ~

update      Update configuration of one or more containers
wait        Block until one or more containers stop, then print their exit codes

Global Options:
--config string      Location of client config files (default "/home/prasad/.docker")
--context string     Name of the context to use to connect to the daemon (overrides DOCKER_HOST env var and default context set with "docker context use")
-D, --debug          Enable debug mode
-H, --host list      Daemon socket to connect to
-l, --log-level string Set the logging level ("debug", "info", "warn", "error", "fatal") (default "info")
--tls               Use TLS; implied by --tlsverify
--tlscacert string   Trust certs signed only by this CA (default "/home/prasad/.docker/ca.pem")
--tlscert string     Path to TLS certificate file (default "/home/prasad/.docker/cert.pem")
--tlskey string      Path to TLS key file (default "/home/prasad/.docker/key.pem")
--tlsverify          Use TLS and verify the remote
-v, --version        Print version information and quit

Run 'docker COMMAND --help' for more information on a command.

For more help on how to use Docker, head to https://docs.docker.com/go/guides/
prasad@prasad-VirtualBox:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: pulling from library/hello-world
719385e32844: Pull complete
Digest: sha256:88ecbaca3ec199d3b7eaf73580f4518c25f9d34f58ce9a0df68429c5af48e8d
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

prasad@prasad-VirtualBox:~$
```

```

Activities Terminal Oct 14 22:11 • prasad@prasad-VirtualBox: ~
Unpacking curl (7.88.0-ubuntu20.20) ...
Setting up curl (7.88.0-ubuntu20.20) ...
Processing triggers for man-db (2.9.1-1) ...
prasad@prasad-VirtualBox:~$ sudo install -m 0755 -d /etc/apt/keyrings
prasad@prasad-VirtualBox:~$ sudo -E sh -c 'cat </dev/null > /dev/null'
prasad@prasad-VirtualBox:~$ sudo apt-get update
Get:1 https://download.docker.com/linux/ubuntu focal InRelease [57.7 kB]
Get:2 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages [33.3 kB]
Hit:1 http://archive.ubuntu.com/ubuntu focal InRelease
Hit:4 http://archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:5 http://archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:6 http://archive.ubuntu.com/ubuntu focal-security InRelease
Fetched 91.0 kB in 1s (9.0 kB/s)
Reading package lists... Done
prasad@prasad-VirtualBox:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
adun-theme ubuntu-command-not-found-data cpp-7 dfdiffstat enchant example-content-fuupdate gcc-7-base gcc-8-base gettext gir1.2-goa-1.0 gir1.2-gtksource-3.0 gir1.2-mutter-2 gnome-software-common
gnome-user-paths gutil-2.0 libs fopdown intool debian iptools arping libappstream-glib libapt-pkg-perl libarchive-zip-perl libargon2-0 libart-2.0 libasyncns-mempoint-perl
libbsd-hacks-andofuse-op-checker libbz2-dev libbz2-perl libboost-date-time1.65.1 libboost-filesystem1.65.1 libboost-log-streams1.65.1 libboost-system1.65.1 libboost-thread1.65.1
libbrlapi-0 libcamel-1.2-61 libcapture-tiny-perl libcdio17 libcurl-fast-perl libdbi-pm-perl libclass-accessor-perl libclass-method-modifiers-perl libclass-xscaccessor-perl libclone-perl
libcpnapi-jon-xen perl libcroco3 libdvel-clchecker perl libdvel-size perl libdvgst-bubbletable-perl libdns-extractor1100 libdns1100 libdouble-conversion1 libdynalynator-functions-perl
libgmp-compat1.2 libical-1.2-19 libidnata-server-1.2-23 libichantica2 libevent-2.1.6 libexempi3 libexttextile4 libexpat-perl libfclm libfclm libfclm libfclm libfclm libfclm libfclm libfclm
libfontconfig1 perl libfpall-0.0 libgsasl3 libgsasl-perl libgsasl-perl libgsasl-perl libgsasl-perl libgsasl-perl libgsasl-perl libgsasl-perl libgsasl-perl libgsasl-perl libgsasl-perl libgsasl-perl
libgutenprints libgweather-3.5 libhwseed4 libhugetlbfs libhunspell-1.6-0 libicu60 libimbnase2 libinotify-intro-perl libindictor3 libio-async-loop-epoll-perl libio-async-perl libio-pty-perl
libio-string-perl libip4tc libiproute2 libipc-run-perl libiptc librtsp160 libisc-exp169 libisc-exp169 libisc-exp169 libisbi19 libjsbin3 libjson-maybes-perl liblinux-epoll-perl
liblist-compare-perl liblist-noreutsils-perl liblvlnv liblvlnv liblvlnvsudrh liblvlnvsudrh liblvlnvsudrh liblvlnvsudrh liblvlnvsudrh liblvlnvsudrh liblvlnvsudrh liblvlnvsudrh liblvlnvsudrh liblvlnvsudrh
libmodule-implements libmodule-runtime-perl libmoose-alases-perl libmozjs-52.0 libmutter-2.0 libnamespace-clean-perl libncurses5 libncursesw5 libndis-dns-perl libnet-dnssec-perl
libnet-ip-perl libnetlites libssh-myhostname libtnfrs-3q88 libpython-compare-perl libpython libpython libpython libpython libpython libpython libpython libpython libpython libpython libpython
libparse-debianchangelog-perl libpath-tiny-perl libperl4-corelibs-perl libperl5.26 libperlito-gzip-perl libplymouth4 libpoppler73 libprotobuf17 libpython2.7-minimal libpython2.7-stdlib
libpython3.6 libpython3.6-minimal libpython3.6-stdlib libtbbgit21 libtwine libreadline libreadline libref-util-perl libref-util-perl libref-util-perl libreffice-awmenda-backend-gstreamer
libreoffice-style-play libreoffice-style-play libreoffice-style-play libreoffice-style-play libreoffice-style-play libreoffice-style-play libreoffice-style-play libreoffice-style-play libreoffice-style-play
libusb-identify-perl libusb-quote-perl libtest-fatal-perl libtest-recount-perl libtest-levenshtein-perl libtinfnos libtype-tiny-perl libtype-tiny-perl libtype-tiny-perl libtype-tiny-perl libtype-tiny-perl
libunibicode-utf8-perl libusbusux4 libvariable-magic-perl libvpx libvresharaki1 libwetrapp libwscodec2 libwsuttl19 libxmlplan30 libxml-simple-perl libxml-writer-perl libxmlbi libxmlbi libxmlbi libxmlbi
linux-generic linux-generic-hwe-18.04 linux-headers-5.4.0-147 linux-headers-5.4.0-147 linux-headers-5.4.0-147 linux-headers-5.4.0-147 linux-headers-5.4.0-147 linux-headers-5.4.0-147
linux-headers-5.4.0-147 linux-headers-5.4.0-147 linux-headers-5.4.0-147 linux-headers-5.4.0-147 linux-headers-5.4.0-147 linux-headers-5.4.0-147 linux-headers-5.4.0-147 linux-headers-5.4.0-147
linux-image-generic-hwe-18.04 linux-modules-5.4.0-147-generic linux-modules-5.4.0-147-generic linux-modules-5.4.0-147-generic linux-modules-extra-5.4.0-147-generic linux-modules-extra-5.4.0-147-generic
linux-modules-extra-5.4.0-147-generic linux-modules-extra-5.4.0-147-generic linux-modules-extra-5.4.0-147-generic linux-modules-extra-5.4.0-147-generic linux-modules-extra-5.4.0-147-generic
patchutils pppoeconf pppoeconf printer-driver-gutenprint python-tallico python3-asciitropy python3-oauth python3-zope.interface python3.6 python3.6 python3.6 python3.6 python3.6 python3.6
ubuntu-system-service server-xorg-core-hwe-18.04 server-xorg-video-all-hwe-18.04 server-xorg-input-all-hwe-18.04 server-xorg-video-all-hwe-18.04 server-xorg-video-all-hwe-18.04
server-xorg-video-intel-hwe-18.04 server-xorg-video-nouveau-hwe-18.04 server-xorg-video-nouveau-hwe-18.04 server-xorg-video-nouveau-hwe-18.04 server-xorg-video-nouveau-hwe-18.04
server-xorg-video-nouveau-hwe-18.04 server-xorg-video-nouveau-hwe-18.04 server-xorg-video-nouveau-hwe-18.04 server-xorg-video-nouveau-hwe-18.04 server-xorg-video-nouveau-hwe-18.04

```

## 2. Install minikube using following commands



```

prasad@prasad-VirtualBox:~$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
% Total % Received % Xferd Average Speed Time Time Time Current
100 82.4M 100 82.4M 0 0 5315k 0 0:00:15 0:00:15 --:--:-- 5910k
prasad@prasad-VirtualBox:~$ sudo install minikube-linux-amd64 /usr/local/bin/minikube
[sudo] password for prasad:
prasad@prasad-VirtualBox:~$ minikube start --driver=docker
minikube v1.31.2 on Ubuntu 20.04 (vbox/amd64)
Using the docker driver based on user configuration
* Extling due to PROVIDER_DOCKER_NEWCRP: "docker version --format '{{.Server.Version}}' exit status 1: permission denied while trying to connect to the Docker daemon socket at unix:///var/r
un/docker.sock: Get "http://192.168.2.2Frun2Fdocker.sock/v1.24/version": dial unix /var/run/docker.sock: connect: permission denied
Suggestion: Add your user to the 'docker' group: 'sudo usermod -aG docker $USER && newgrp docker'
Documentation: https://docs.docker.com/engine/install/linux-postinstall/
prasad@prasad-VirtualBox:~$ sudo usermod -aG docker $USER && newgrp docker
prasad@prasad-VirtualBox:~$ minikube start --driver=docker
minikube v1.31.2 on Ubuntu 20.04 (vbox/amd64)
Using the docker driver based on user configuration
* The requested memory allocation of 1971MiB does not leave room for system overhead (total system memory: 1971MiB). You may face stability issues.
Suggestion: Start minikube with less memory allocated: 'minikube start --memory=1971mb'
* Using Docker driver with root privileges
Starting control plane node minikube in cluster minikube
Pulling base image ...
Downloading Kubernetes v1.27.4 preload ...
> preloaded-images-k8s-v18-v1...: 393.21 MiB / 393.21 MiB 100.00% 2.85 Mi
> gcr.io/k8s-minikube/kicbase...: 447.62 MiB / 447.62 MiB 100.00% 2.99 Mi
Creating docker container (CPUs=2, Memory=1971MiB) ...
* Docker is nearly out of disk space, which may cause deployments to fail! (94% of capacity). You can pass '--force' to skip this check.
Suggestion:
Try one or more of the following to free up space on the device:
1. Run "docker system prune" to remove unused Docker data (optionally with "-a")
2. Increase the storage allocated to Docker for Desktop by clicking on:
Docker icon > Preferences > Resources > Disk Image Size
3. Run "minikube ssh -- docker system prune" if using the Docker container runtime
Related issue: https://github.com/kubernetes/minikube/issues/9024
* This container is having trouble accessing https://registry.k8s.io
To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
Preparing Kubernetes v1.27.4 on Docker 24.0.4 ...
* Generating certificates and keys ...
* Booting up control plane ...
* Configuring RBAC rules ...
* Configuring bridge CNI (Container Networking Interface) ...
* 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
prasad@prasad-VirtualBox:~$ ^C
prasad@prasad-VirtualBox:~$

```

### 3. Install kubectl

```

Activities Terminal Oct 14 22:36
prasad@prasad-VirtualBox: ~
prasad@prasad-VirtualBox: ~
prasad@prasad-VirtualBox: ~

Using Docker driver with root privileges
Starting control plane node minikube in cluster minikube
Pulling base image ...
Downloading Kubernetes v1.27.4 preload ...
> preload-images-k8s-v18-v1...: 393.21 MiB / 393.21 MiB 100.00% 2.85 Mi
> gcr.io/k8s-minikube/kicbase...: 447.62 MiB / 447.62 MiB 100.00% 2.99 Mi
Creating docker container (CPUs=2, Memory=1971MB) ...

Docker is nearly out of disk space, which may cause deployments to fail! (94% of capacity). You can pass '--force' to skip this check.
Suggestion:

Try one or more of the following to free up space on the device:
1. Run "docker system prune" to remove unused Docker data (optionally with "--a")
2. Increase the storage allocated to Docker for Desktop by clicking on:
   Docker icon > Preferences > Resources > Disk Image Size
3. Run "minikube ssh -- docker system prune" if using the Docker container runtime
Related issue: https://github.com/kubernetes/minikube/issues/9024

This container is having trouble accessing https://registry.k8s.io
To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
Preparing Kubernetes v1.27.4 on Docker 24.0.4 ...
  Generating certificates and keys ...
  Booting up control plane ...
  Configuring RBAC rules ...
  Configuring bridge CNI (Container Networking Interface) ...
  Using image gcr.io/k8s-minikube/storage-provisioner:v5
  Verifying Kubernetes components...
  Enabled addons: default-storageclass, storage-provisioner
  kubectrl not found. If you need it, try: 'minikube kubectrl -- get pods -A'
Done! kubectrl is now configured to use "minikube" cluster and "default" namespace by default
prasad@prasad-VirtualBox: ~$ sudo snap install kubectrl --classic
[sudo] password for prasad:
kubectrl 1.28.2 from Canonical LLC installed
prasad@prasad-VirtualBox: ~$ kubectrl get po -A
prasad@prasad-VirtualBox: ~$ kubectl get po -A

```

NAME	READY	STATUS	RESTARTS	AGE
kube-system coredns-5d78c9869d-6wdgp	1/1	Running	0	16m
kube-system etcd-minikube	1/1	Running	0	16m
kube-system kube-apiserver-minikube	1/1	Running	0	16m
kube-system kube-controller-manager-minikube	1/1	Running	0	17m
kube-system kube-proxy-snjnt	1/1	Running	0	16m
kube-system kube-scheduler-minikube	1/1	Running	0	16m
kube-system storage-provisioner	1/1	Running	1 (16m ago)	16m

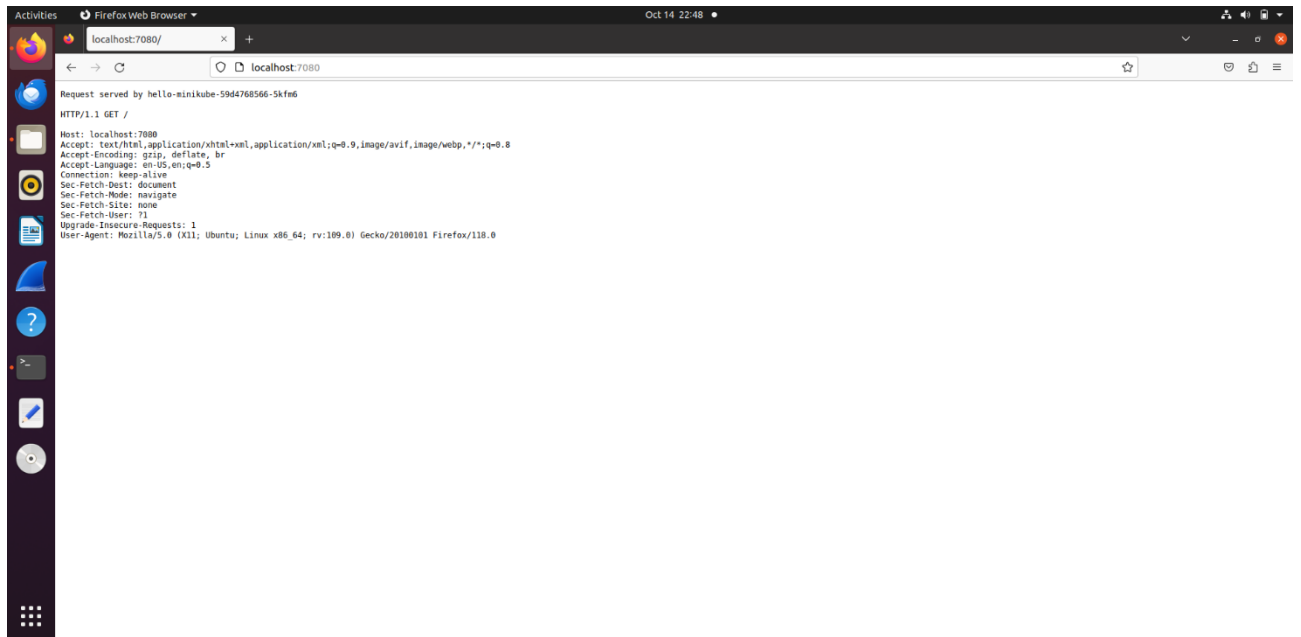
#### 4. Create a sample deployment.

```

Activities Terminal Oct 14 22:48
prasad@prasad-VirtualBox: ~
prasad@prasad-VirtualBox: ~
prasad@prasad-VirtualBox: ~

prasad@prasad-VirtualBox: ~$ kubectl create deployment hello-minikube --image=kicbase/echo-server:1.0
deployment.apps/hello-minikube created
prasad@prasad-VirtualBox: ~$ kubectl expose deployment hello-minikube --type=NodePort --port=8080
service/hello-minikube exposed
prasad@prasad-VirtualBox: ~$ kubectl get services hello-minikube
NAME          TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
hello-minikube NodePort    10.108.233.37 <none>         8080:31242/TCP   86s
prasad@prasad-VirtualBox: ~$ kubectl port-forward service/hello-minikube 7080:8080
Forwarding from 127.0.0.1:7080 -> 8080
Forwarding from [::1]:7080 -> 8080
Handling connection for 7080

```



## CONCLUSION:

Here we studied Kubernetes cluster architecture in detail. Also we installed Kubernetes in ubuntu machine and created a sample deployment.