

**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 51**  
**Microservices Practical 15**

**Aim : To understand the basics of Kubernetes**

**Scenario :** Your company and its application are scaling, and you hence need to make use of tools and services that would help you manage multiple containers and its management easily. On that note, understand the basics of Kubernetes and perform the following.

**Tasks and Screenshots containing commands and output :**

1. Learn what a Kubernetes cluster is.

Kubernetes coordinates a highly available cluster of computers that are connected to work as a single unit. The abstractions in Kubernetes allow you to deploy containerized applications to a cluster without tying them specifically to individual machines. To make use of this new model of deployment, applications need to be packaged in a way that decouples them from individual hosts: they need to be containerized. Containerized applications are more flexible and available than in past deployment models, where applications were installed directly onto specific machines as packages deeply integrated into the host. Kubernetes automates the distribution and scheduling of application containers across a cluster in a more efficient way. Kubernetes is an open-source platform and is production-ready. A Kubernetes cluster consists of two types of resources:

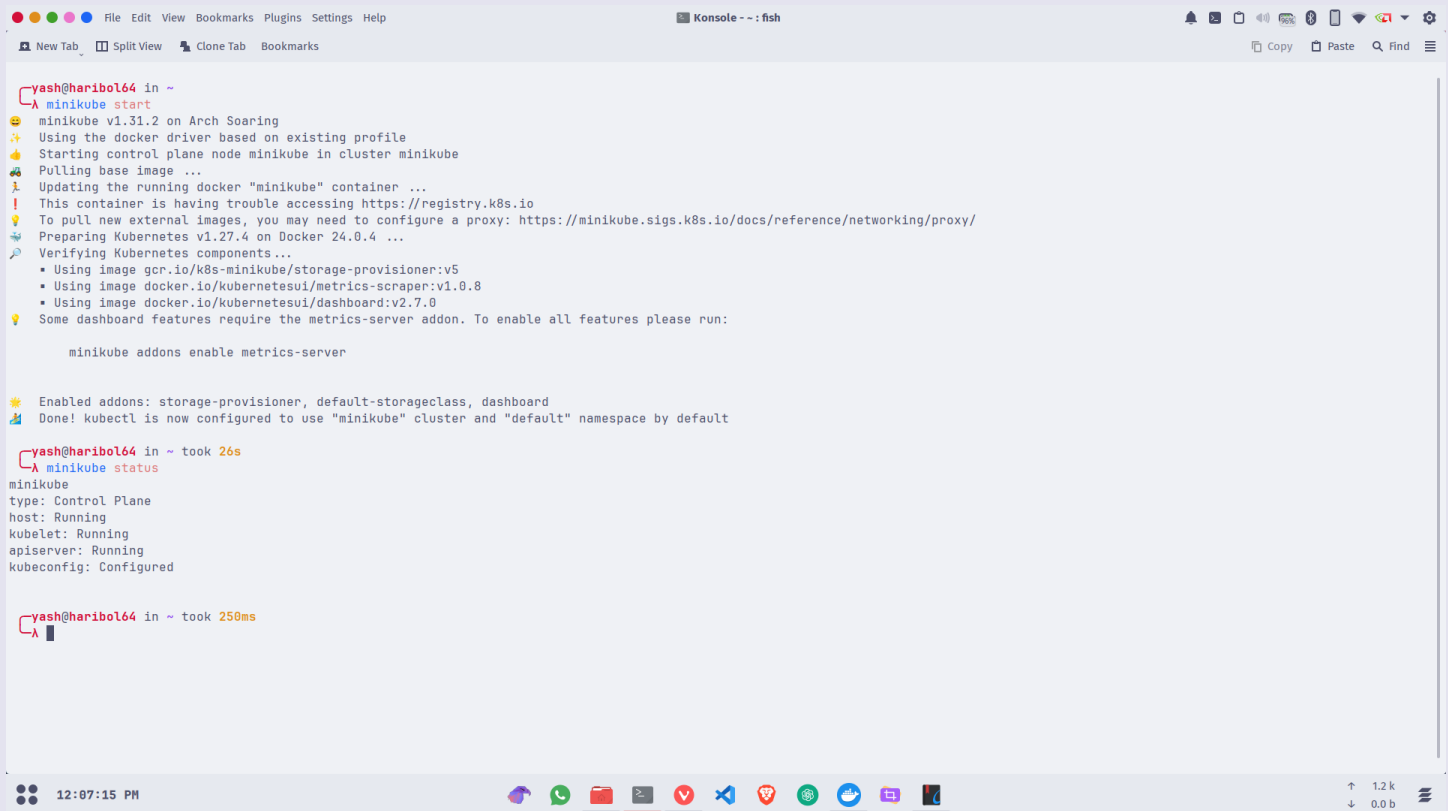
- The Control Plane coordinates the cluster.
- Nodes are the workers that run applications.

2. Learn what Minikube is.

Minikube is a lightweight tool that enables you to set up and run a single-node Kubernetes cluster locally, making it easier for developers to develop and test Kubernetes applications on their own machines.

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA Batch - 51  
Microservices Practical 15

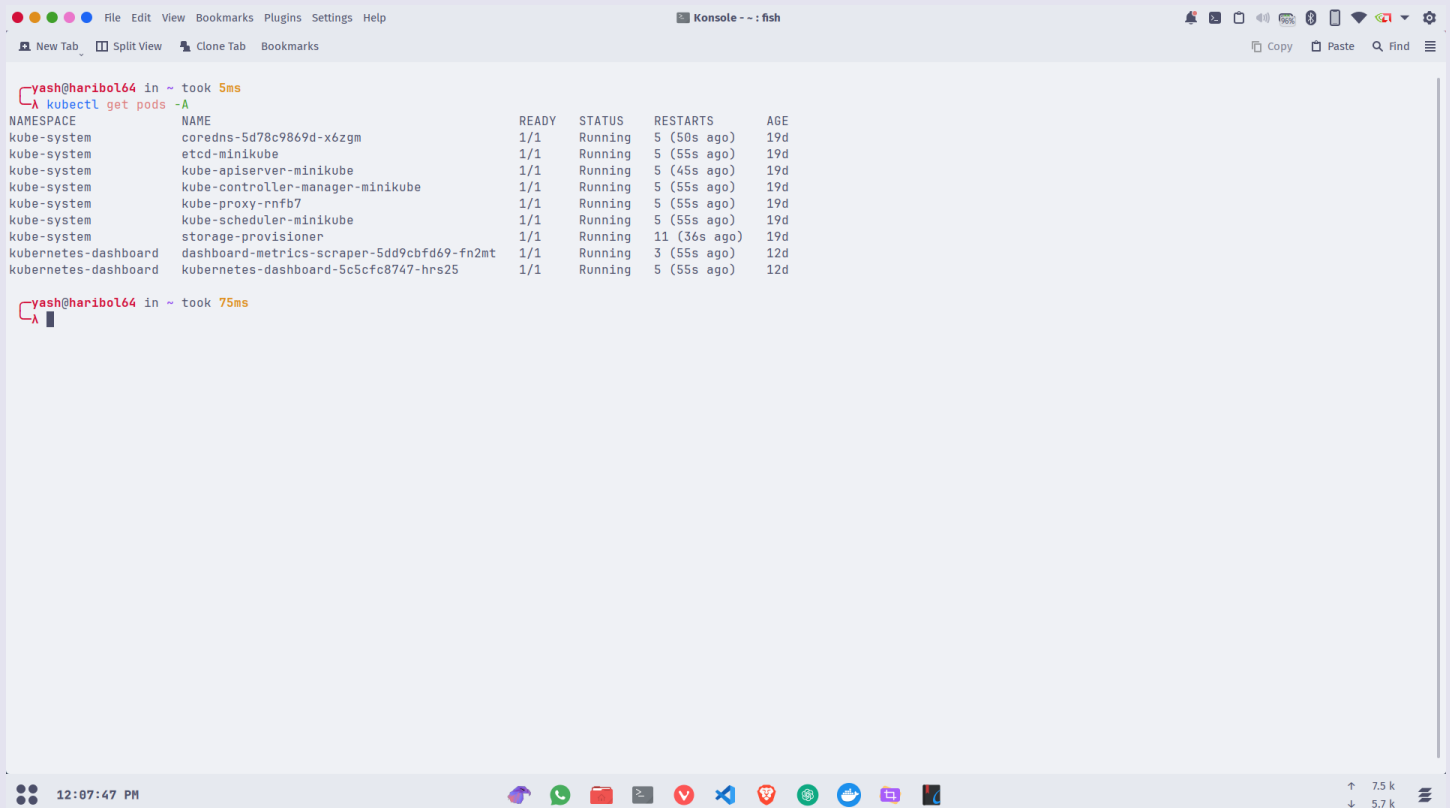
### 3. Start a Kubernetes cluster using a terminal.



```
yash@haribol64 in ~  
$ minikube start  
minikube v1.31.2 on Arch Soaring  
Using the docker driver based on existing profile  
Starting control plane node minikube in cluster minikube  
Pulling base image ...  
Updating the running docker "minikube" container ...  
! 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 ...  
Verifying Kubernetes components...  
  ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5  
  ▪ Using image docker.io/kubernetes/metrics-scraper:v1.0.8  
  ▪ Using image docker.io/kubernetes/dashboard:v2.7.0  
Some dashboard features require the metrics-server addon. To enable all features please run:  
  
    minikube addons enable metrics-server  
  
Enabled addons: storage-provisioner, default-storageclass, dashboard  
Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default  
  
yash@haribol64 in ~ took 26s  
$ minikube status  
minikube  
type: Control Plane  
host: Running  
kubelet: Running  
apiserver: Running  
kubeconfig: Configured  
  
yash@haribol64 in ~ took 250ms  
$
```

**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 51**  
**Microservices Practical 15**

The command `kubectl get pods -A` is used to list all the pods in all namespaces within a Kubernetes cluster. It provides a comprehensive view of all running pods, regardless of the namespace they belong to, making it a valuable tool for monitoring and managing the entire cluster's pod resources.



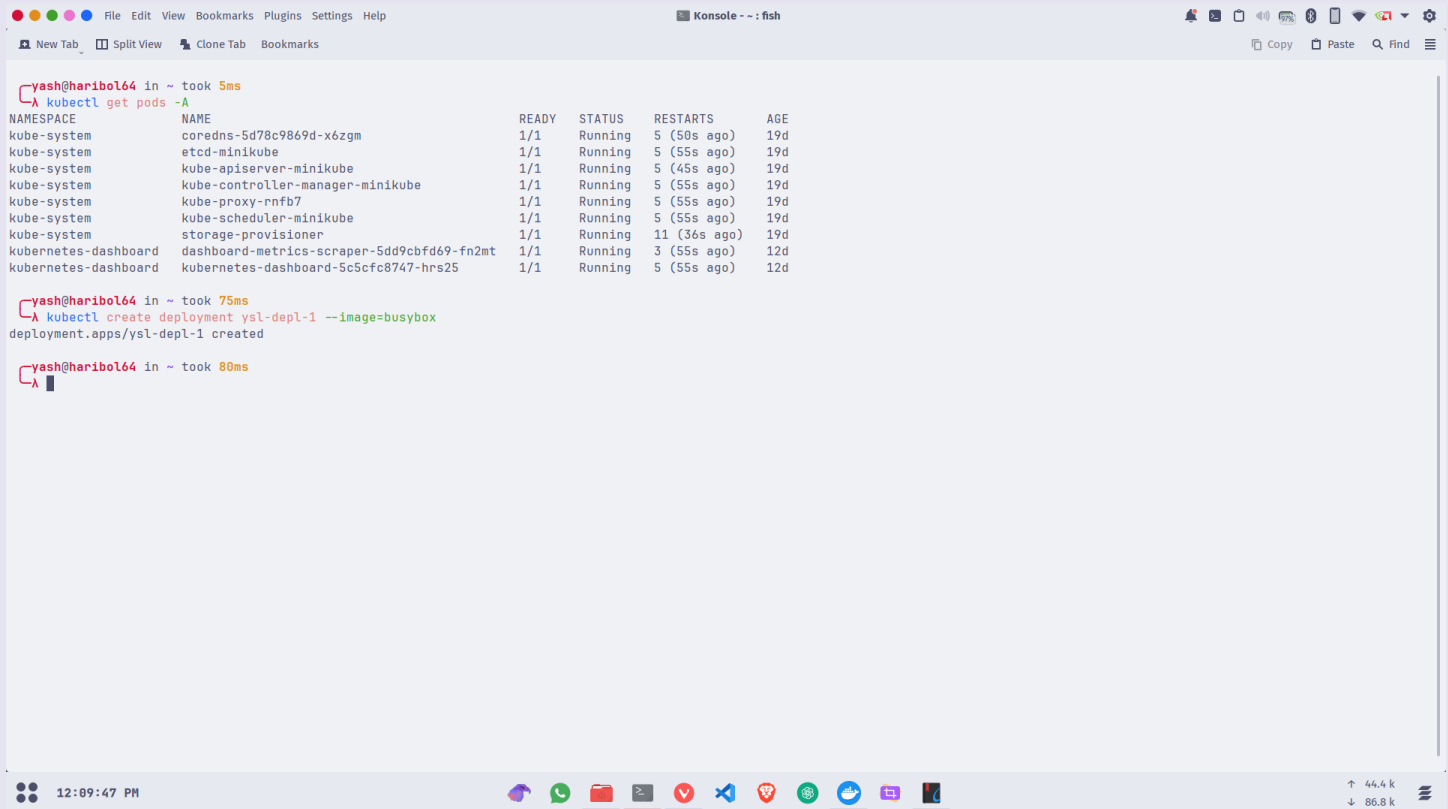
```
yash@haribol64 in ~ took 5ms
kubectl get pods -A
NAMESPACE      NAME                                     READY   STATUS    RESTARTS   AGE
kube-system     coredns-5d78c9869d-x6zgm              1/1     Running   5 (50s ago) 19d
kube-system     etcd-minikube                         1/1     Running   5 (55s ago) 19d
kube-system     kube-apiserver-minikube               1/1     Running   5 (45s ago) 19d
kube-system     kube-controller-manager-minikube      1/1     Running   5 (55s ago) 19d
kube-system     kube-proxy-rnfb7                     1/1     Running   5 (55s ago) 19d
kube-system     kube-scheduler-minikube               1/1     Running   5 (55s ago) 19d
kube-system     storage-provisioner                   1/1     Running   11 (36s ago) 19d
kubernetes-dashboard dashboard-metrics-scraper-5dd9cbfd69-fn2mt 1/1     Running   3 (55s ago) 12d
kubernetes-dashboard kubernetes-dashboard-5c5cfc8747-hrs25 1/1     Running   5 (55s ago) 12d

yash@haribol64 in ~ took 75ms
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA Batch - 51  
Microservices Practical 15

#### 4. Learn about application Deployments.

##### a) Create deployment with image specified in command



The screenshot shows a terminal window with the following commands and output:

```
yash@haribol64 in ~ took 5ms
kubectll get pods -A
```

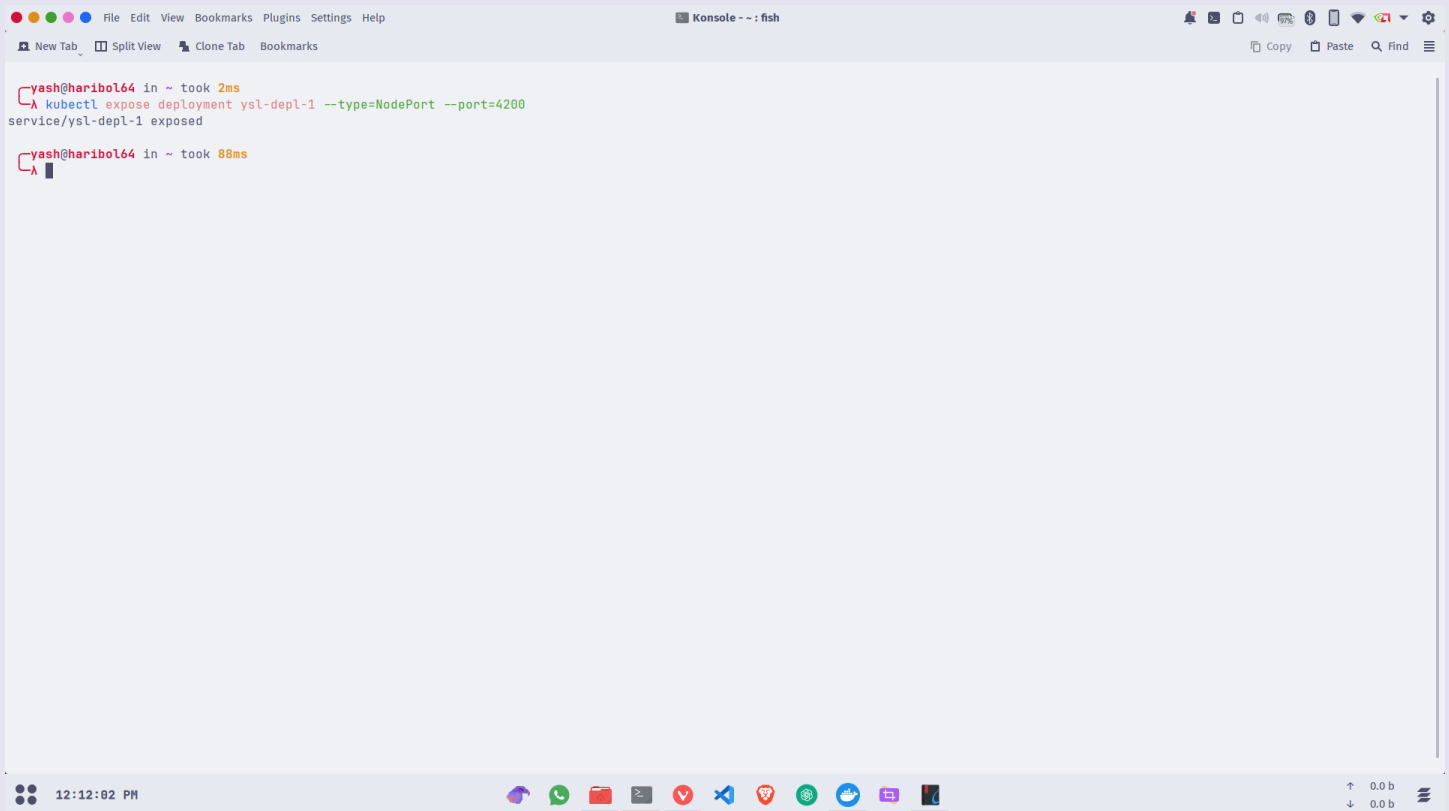
NAMESPACE	NAME	READY	STATUS	RESTARTS	AGE
kube-system	coredns-5d78c9869d-x6zgm	1/1	Running	5 (50s ago)	19d
kube-system	etcd-minikube	1/1	Running	5 (55s ago)	19d
kube-system	kube-apiserver-minikube	1/1	Running	5 (45s ago)	19d
kube-system	kube-controller-manager-minikube	1/1	Running	5 (55s ago)	19d
kube-system	kube-proxy-rnfb7	1/1	Running	5 (55s ago)	19d
kube-system	kube-scheduler-minikube	1/1	Running	5 (55s ago)	19d
kube-system	storage-provisioner	1/1	Running	11 (36s ago)	19d
kubernetes-dashboard	dashboard-metrics-scraper-5dd9cbfd69-fn2mt	1/1	Running	3 (55s ago)	12d
kubernetes-dashboard	kubernetes-dashboard-5c5cfc8747-hrs25	1/1	Running	5 (55s ago)	12d

```
yash@haribol64 in ~ took 75ms
kubectll create deployment ysl-depl-1 --image=busybox
deployment.apps/ysl-depl-1 created

yash@haribol64 in ~ took 80ms
```

**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 51**  
**Microservices Practical 15**

b) Expose the deployment by creating a service of type and port specified.



```
yash@haribol64 in ~ took 2ms
$ kubectl expose deployment ysl-depl-1 --type=NodePort --port=4200
service/ysl-depl-1 exposed

yash@haribol64 in ~ took 88ms
```

The screenshot shows a terminal window with a light gray background. The terminal title is "Konsole --: fish". The menu bar includes "File", "Edit", "View", "Bookmarks", "Plugins", "Settings", and "Help". The toolbar has icons for "New Tab", "Split View", "Clone Tab", and "Bookmarks". The right side of the toolbar includes "Copy", "Paste", "Find", and a menu icon. The terminal content shows two commands being executed. The first command is `kubectl expose deployment ysl-depl-1 --type=NodePort --port=4200`, which results in the output `service/ysl-depl-1 exposed`. The second command is an empty prompt, and the terminal shows `took 88ms`. The bottom status bar shows the time `12:12:02 PM` and a row of application icons. On the right side of the status bar, there are network speed indicators: `↑ 0.0 b` and `↓ 0.0 b`.

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA Batch - 51  
Microservices Practical 15

Vivaldi - Kubernetes Dashboard

Workloads

Workload Status

Running: 1 Deployments

Running: 1 Pods

Running: 1 Replica Sets

Deployments

Name	Images	Labels	Pods	Created ↑
ysl-depl-1	gcr.io/google-samples/hello-app:1.0	app: ysl-depl-1	1 / 1	43 seconds ago

Pods

Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created ↑
ysl-depl-1-6574bdf4f-5k49r	gcr.io/google-samples/hello-app:1.0	app: ysl-depl-1 pod-template-hash: 6574bdf4f	minikube	Running	0	-	-	43 seconds ago

12:16:34 PM

Vivaldi - Kubernetes Dashboard

Workloads > Pods

Pods

Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created ↑
ysl-depl-1-6574bdf4f-5k49r	gcr.io/google-samples/hello-app:1.0	app: ysl-depl-1 pod-template-hash: 6574bdf4f	minikube	Running	0	-	-	a minute ago

12:17:27 PM

**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 51**  
**Microservices Practical 15**

5. Deploy your first app on Kubernetes with kubectl.

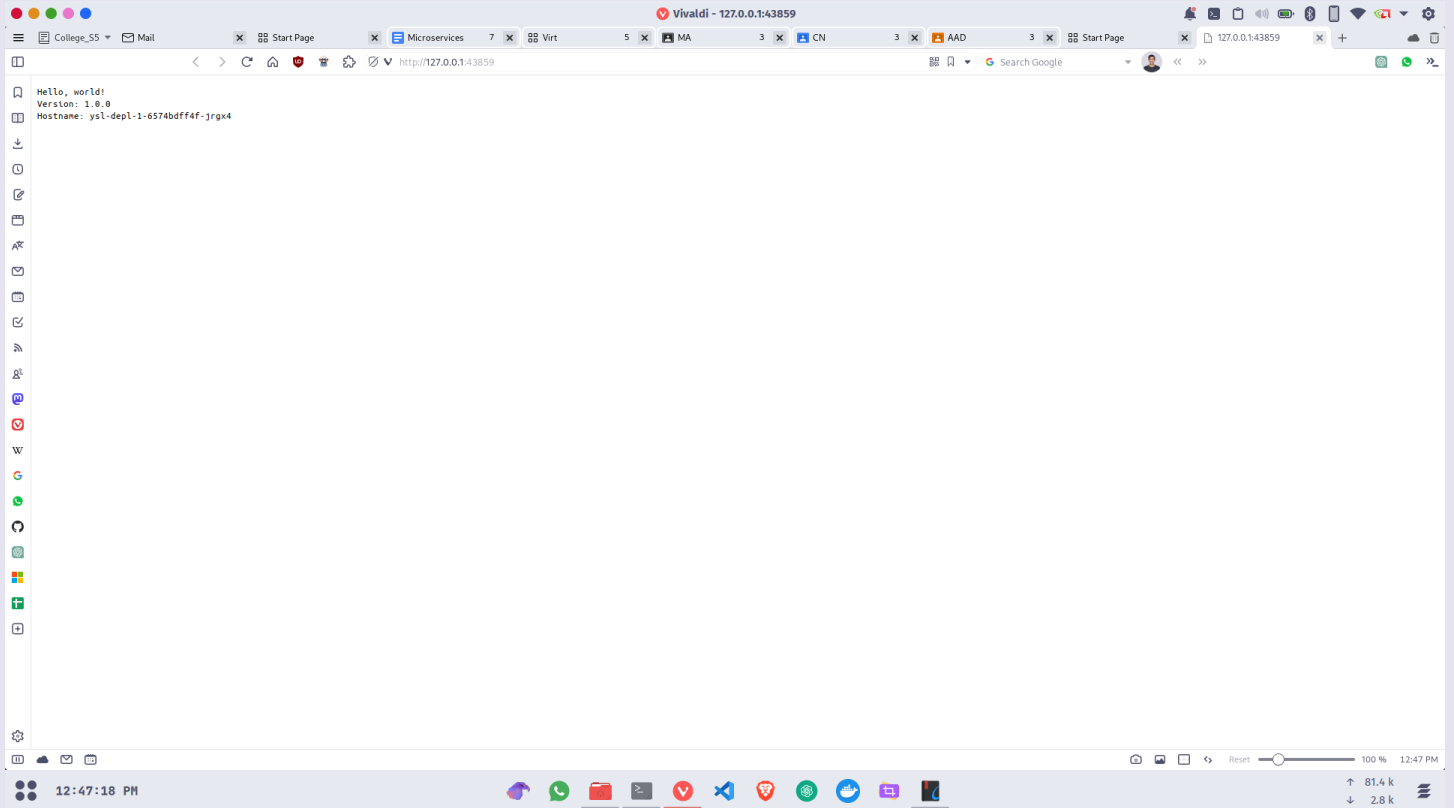
Get service info using kubectl and access it using minikube

The screenshot shows a terminal window titled "Konsole -- minikube". The terminal output is as follows:

```
yash@haribol64 in ~  
$ kubectl get service ysl-depl-1  
NAME          TYPE        CLUSTER-IP    EXTERNAL-IP  PORT(S)          AGE  
ysl-depl-1    NodePort    10.111.144.136 <none>       8080:31406/TCP  16m  
  
yash@haribol64 in ~ took 74ms  
$ minikube service ysl-depl-1 --url  
http://127.0.0.1:43859  
! Because you are using a Docker driver on linux, the terminal needs to be open to run it.
```

The terminal window has a standard Linux desktop environment at the bottom with a taskbar showing various application icons and a system clock displaying 12:46:59 PM.

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 51  
Microservices Practical 15

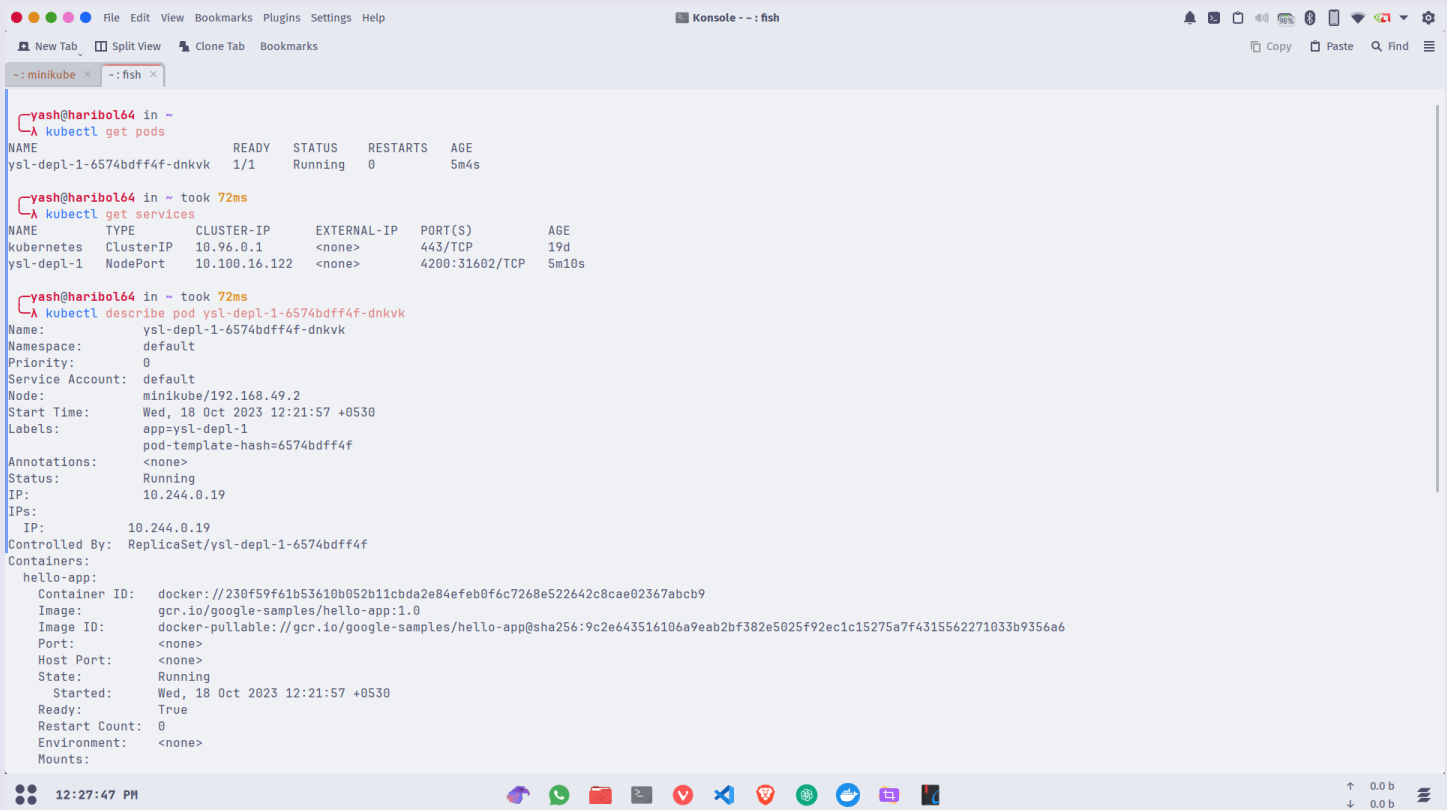




Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA Batch - 51  
Microservices Practical 15

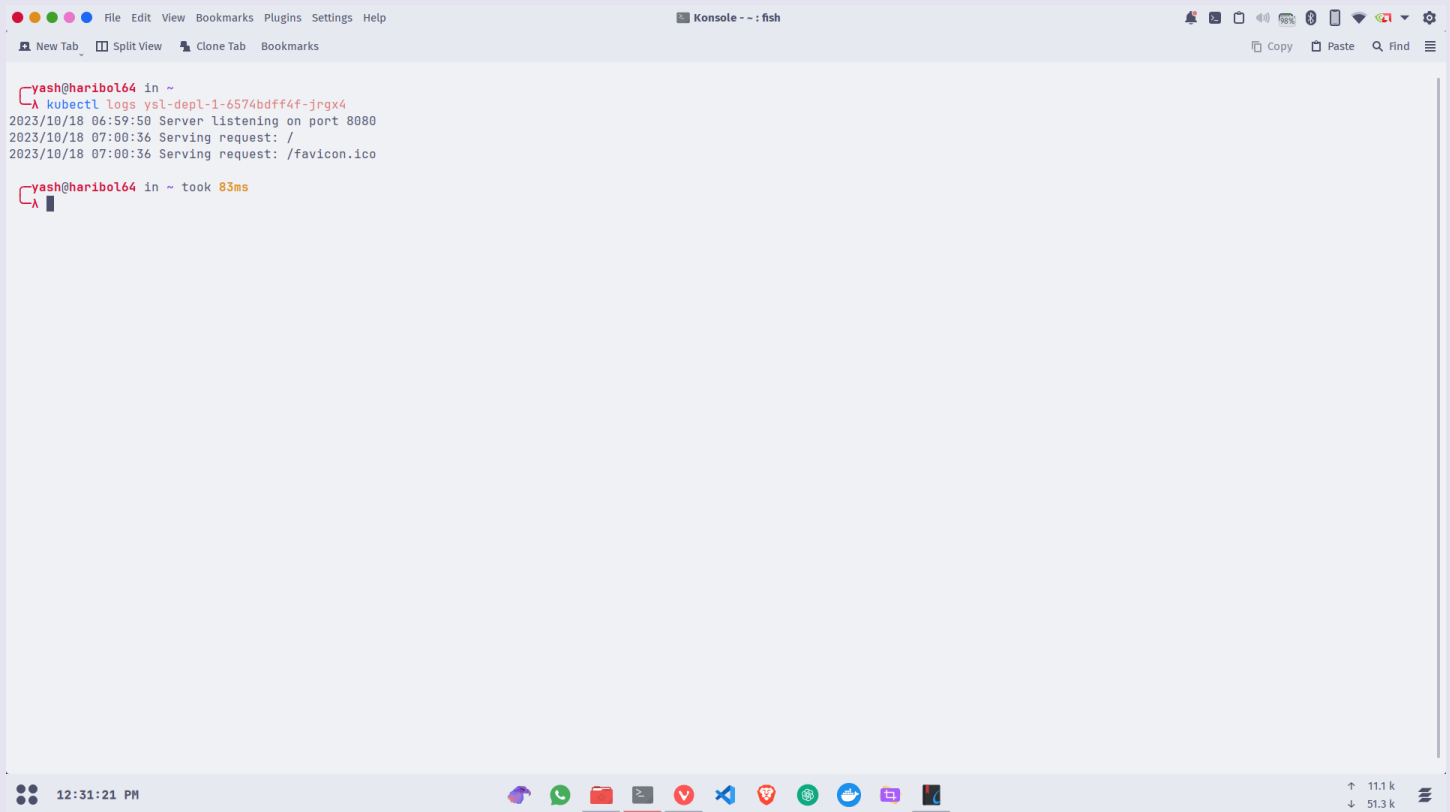
6. Learn how to troubleshoot Kubernetes applications using the `kubectl` `get`, `describe`, `logs` and `exec` commands

Use **`kubectl get`** to list resources, then **`kubectl describe`** to get detailed information of a specific Kubernetes resource, also **`kubectl logs`** can be used to get logs of pods. Optionally, **`kubectl exec`** to execute command inside a container (useful for interactive shell apps or containers).



```
yash@haribol64 in ~  
$ kubectl get pods  
NAME                                READY   STATUS    RESTARTS   AGE  
ysl-depl-1-6574bdf4f-dnkvk         1/1     Running   0           5m4s  
  
yash@haribol64 in ~ took 72ms  
$ kubectl get services  
NAME      TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE  
kubernetes ClusterIP   10.96.0.1     <none>        443/TCP       19d  
ysl-depl-1 NodePort    10.100.16.122 <none>        4200:31602/TCP 5m10s  
  
yash@haribol64 in ~ took 72ms  
$ kubectl describe pod ysl-depl-1-6574bdf4f-dnkvk  
Name:             ysl-depl-1-6574bdf4f-dnkvk  
Namespace:        default  
Priority:          0  
Service Account:  default  
Node:             minikube/192.168.49.2  
Start Time:       Wed, 18 Oct 2023 12:21:57 +0530  
Labels:           app=ysl-depl-1  
                  pod-template-hash=6574bdf4f  
Annotations:      <none>  
Status:           Running  
IP:               10.244.0.19  
IPs:              10.244.0.19  
Controlled By:    ReplicaSet/ysl-depl-1-6574bdf4f  
Containers:  
  hello-app:  
    Container ID:  docker://230f59f61b53610b052b11cbda2e84efeb0f6c7268e522642c8cae02367abcb9  
    Image:         gcr.io/google-samples/hello-app:1.0  
    Image ID:      docker-pullable://gcr.io/google-samples/hello-app@sha256:9c2e643516106a9eab2bf382e5025f92ec1c15275a7f4315562271033b9356a6  
    Port:          <none>  
    Host Port:     <none>  
    State:         Running  
      Started:     Wed, 18 Oct 2023 12:21:57 +0530  
    Ready:         True  
    Restart Count: 0  
    Environment:  <none>  
    Mounts:        <none>
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 51  
Microservices Practical 15



The screenshot shows a terminal window titled "Konsole - -: fish". The terminal content includes the command `kubectl logs ysl-depl-1-6574bdf4f-jrgx4` and its output, which consists of three log lines: `2023/10/18 06:59:50 Server listening on port 8080`, `2023/10/18 07:00:36 Serving request: /`, and `2023/10/18 07:00:36 Serving request: /favicon.ico`. Below the logs, the prompt `yash@haribol64 in ~` is shown, followed by the text `took 83ms` and a cursor. The terminal window has a menu bar with "File", "Edit", "View", "Bookmarks", "Plugins", "Settings", and "Help". The top right corner shows system status icons, and the bottom right corner shows memory usage: `↑ 11.1 k` and `↓ 51.3 k`.

```
yash@haribol64 in ~  
kubectl logs ysl-depl-1-6574bdf4f-jrgx4  
2023/10/18 06:59:50 Server listening on port 8080  
2023/10/18 07:00:36 Serving request: /  
2023/10/18 07:00:36 Serving request: /favicon.ico  
  
yash@haribol64 in ~ took 83ms  
█
```