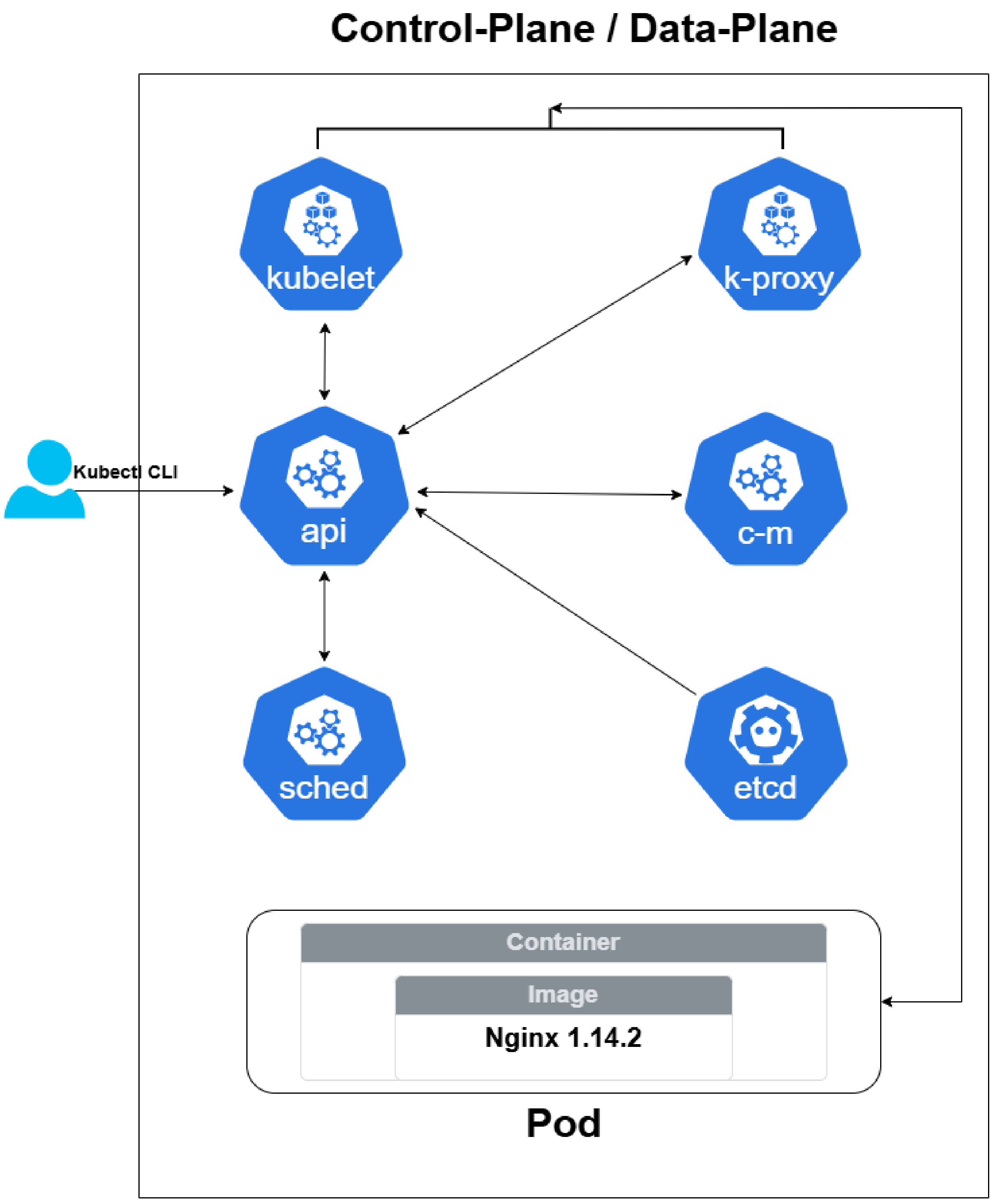
KUBERNETES DEPLOYING A POD USING SINGLE NODE HANDS-ON COMMANDS

SINGLE NODE ARCHITECTURE

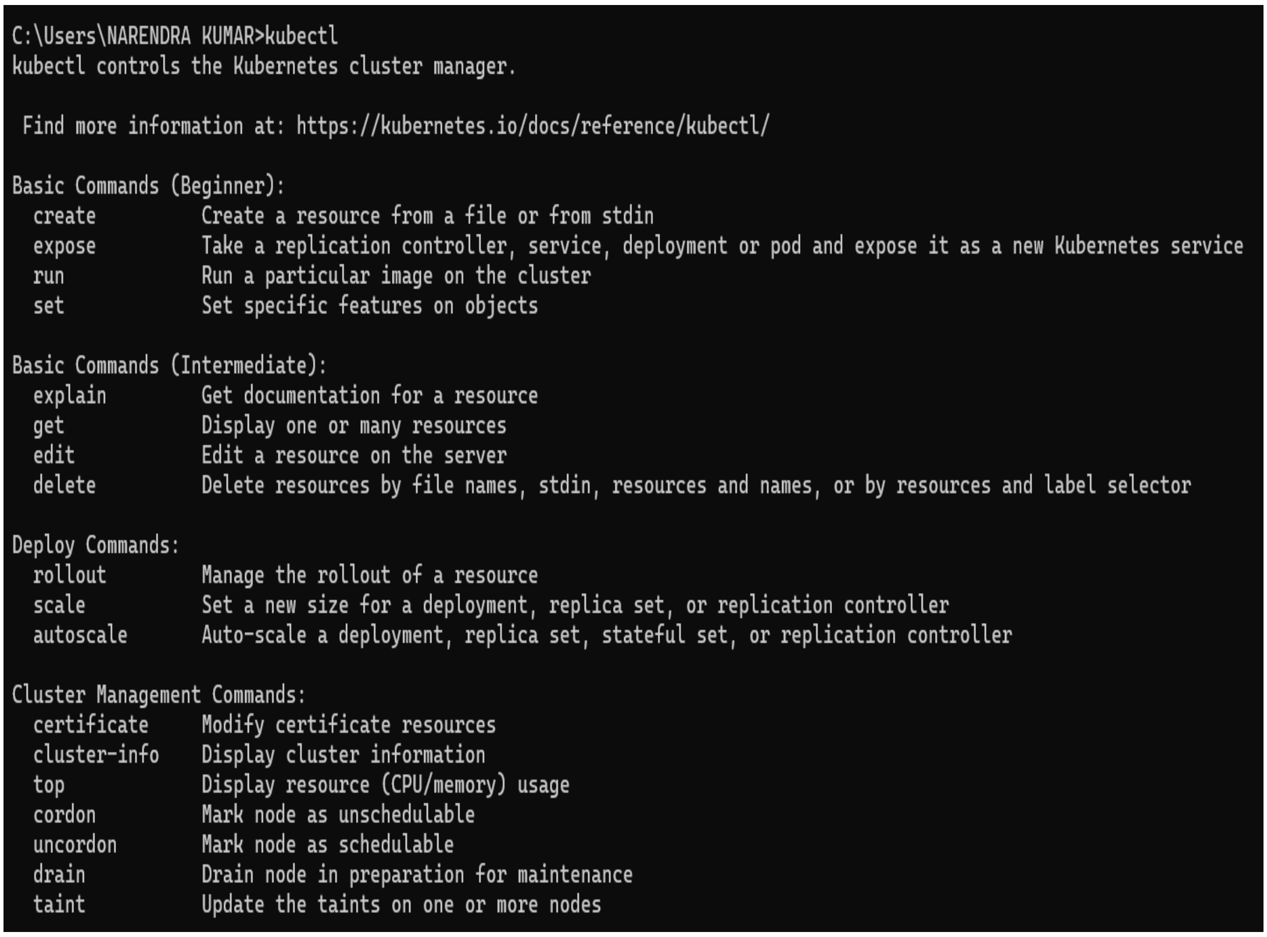


Prerequisites:

1. Install Kubectl
2. Install Minikube

Kubectl Installation Check:

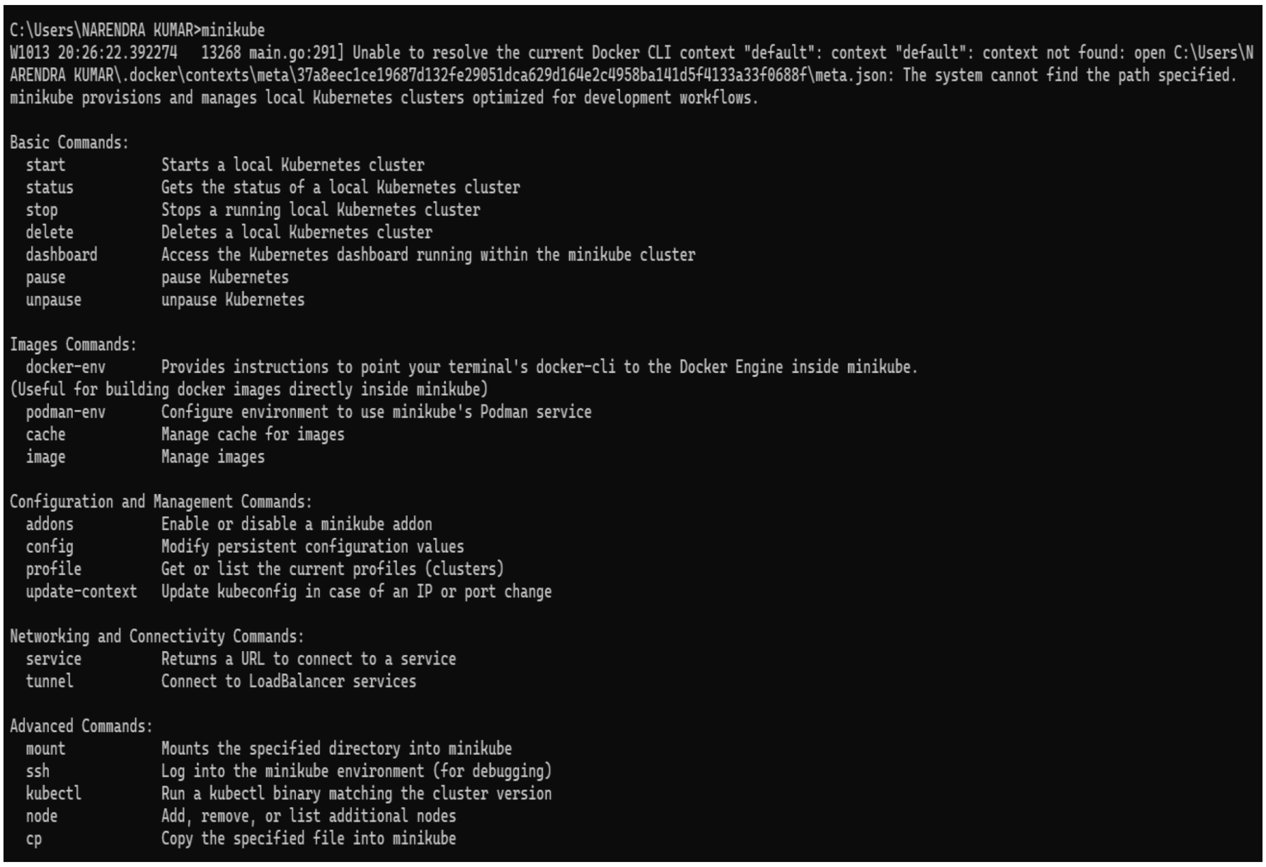
After installing kubectl, open Command prompt and type **“kubectl”**. If you are getting the same output as below, then you can confirm kubectl is successfully installed.



```
C:\Users\NARENDRA KUMAR>kubectl
kubectl controls the Kubernetes cluster manager.

 Find more information at: https://kubernetes.io/docs/reference/kubectl/

Basic Commands (Beginner):
  create          Create a resource from a file or from stdin
  expose          Take a replication controller, service, deployment or pod and expose it as a new Kubernetes service
  run             Run a particular image on the cluster
  set             Set specific features on objects

Basic Commands (Intermediate):
  explain         Get documentation for a resource
  get             Display one or many resources
  edit            Edit a resource on the server
  delete          Delete resources by file names, stdin, resources and names, or by resources and label selector

Deploy Commands:
  rollout         Manage the rollout of a resource
  scale           Set a new size for a deployment, replica set, or replication controller
  autoscale       Auto-scale a deployment, replica set, stateful set, or replication controller

Cluster Management Commands:
  certificate     Modify certificate resources
  cluster-info    Display cluster information
  top             Display resource (CPU/memory) usage
  cordon          Mark node as unschedulable
  uncordon        Mark node as schedulable
  drain           Drain node in preparation for maintenance
  taint           Update the taints on one or more nodes
```

Minikube Installation Check:

After installing Minikube, open Command prompt and type **“minikube”**. If you are getting the same output as below, then you can confirm minikube is successfully installed.



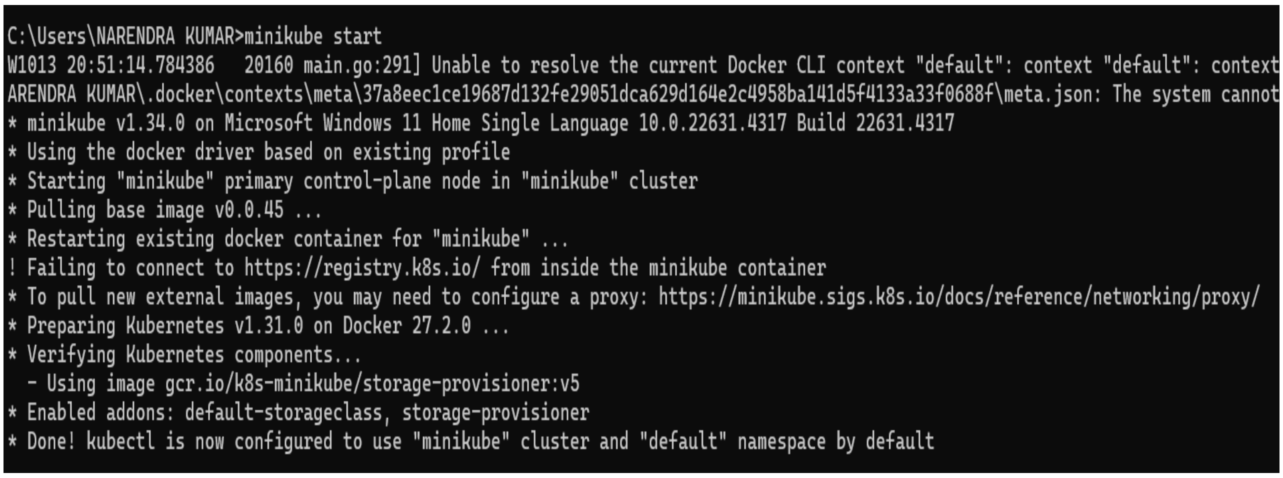
```
C:\Users\NARENDRA KUMAR>minikube
W1013 20:26:22.392274   13268 main.go:291] Unable to resolve the current Docker CLI context "default": context "default": context not found: open C:\Users\N
ARENDRA KUMAR\.docker\contexts\meta\37a8eec1ce19687d132fe29051dca629d164e2c4958ba141d5f4133a33f0688f\meta.json: The system cannot find the path specified.
minikube provisions and manages local Kubernetes clusters optimized for development workflows.

Basic Commands:
  start            Starts a local Kubernetes cluster
  status           Gets the status of a local Kubernetes cluster
  stop             Stops a running local Kubernetes cluster
  delete           Deletes a local Kubernetes cluster
  dashboard        Access the Kubernetes dashboard running within the minikube cluster
  pause            pause Kubernetes
  unpause          unpause Kubernetes

Images Commands:
  docker-env       Provides instructions to point your terminal's docker-cli to the Docker Engine inside minikube.
(Useful for building docker images directly inside minikube)
  podman-env       Configure environment to use minikube's Podman service
  cache            Manage cache for images
  image            Manage images

Configuration and Management Commands:
  addons           Enable or disable a minikube addon
  config           Modify persistent configuration values
  profile          Get or list the current profiles (clusters)
  update-context   Update kubeconfig in case of an IP or port change

Networking and Connectivity Commands:
  service          Returns a URL to connect to a service
  tunnel           Connect to LoadBalancer services

Advanced Commands:
  mount            Mounts the specified directory into minikube
  ssh              Log into the minikube environment (for debugging)
  kubectl          Run a kubectl binary matching the cluster version
  node             Add, remove, or list additional nodes
  cp               Copy the specified file into minikube
```

Note: For installation of Kubectl and Minikube follow the Kubernetes documentation. Links are given at the end of the document.

DEPLOYING AN NGINX WEB SERVER USING PODS

STEP 1: Create and start the Kubernetes cluster. You can do that using the below single command. After executing you can see the below output.

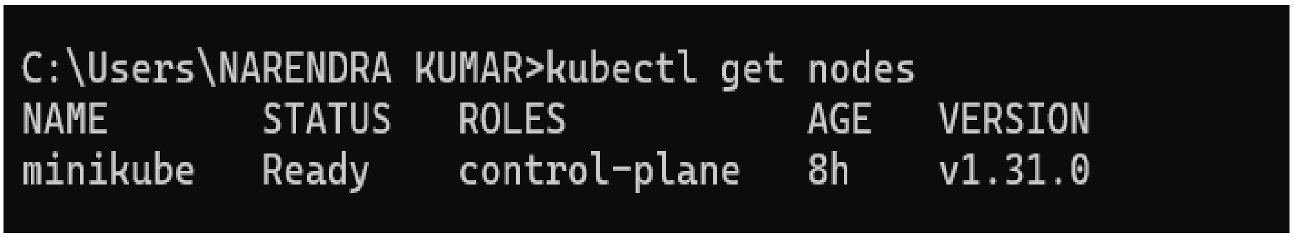
$ minikube start



```
C:\Users\NARENDRA KUMAR>minikube start
W1013 20:51:14.784386   20160 main.go:291] Unable to resolve the current Docker CLI context "default": context "default": context
ARENDRA KUMAR\.docker\contexts\meta\37a8eec1ce19687d132fe29051dca629d164e2c4958ba141d5f4133a33f0688f\meta.json: The system cannot
* minikube v1.34.0 on Microsoft Windows 11 Home Single Language 10.0.22631.4317 Build 22631.4317
* Using the docker driver based on existing profile
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.45 ...
* Restarting existing docker container for "minikube" ...
! Failing to connect to https://registry.k8s.io/ from inside the minikube container
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
* Preparing Kubernetes v1.31.0 on Docker 27.2.0 ...
* 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
```

STEP 2: After having the cluster, you have to connect with that cluster and get the nodes. After executing the below command, you will get the nodes you have.

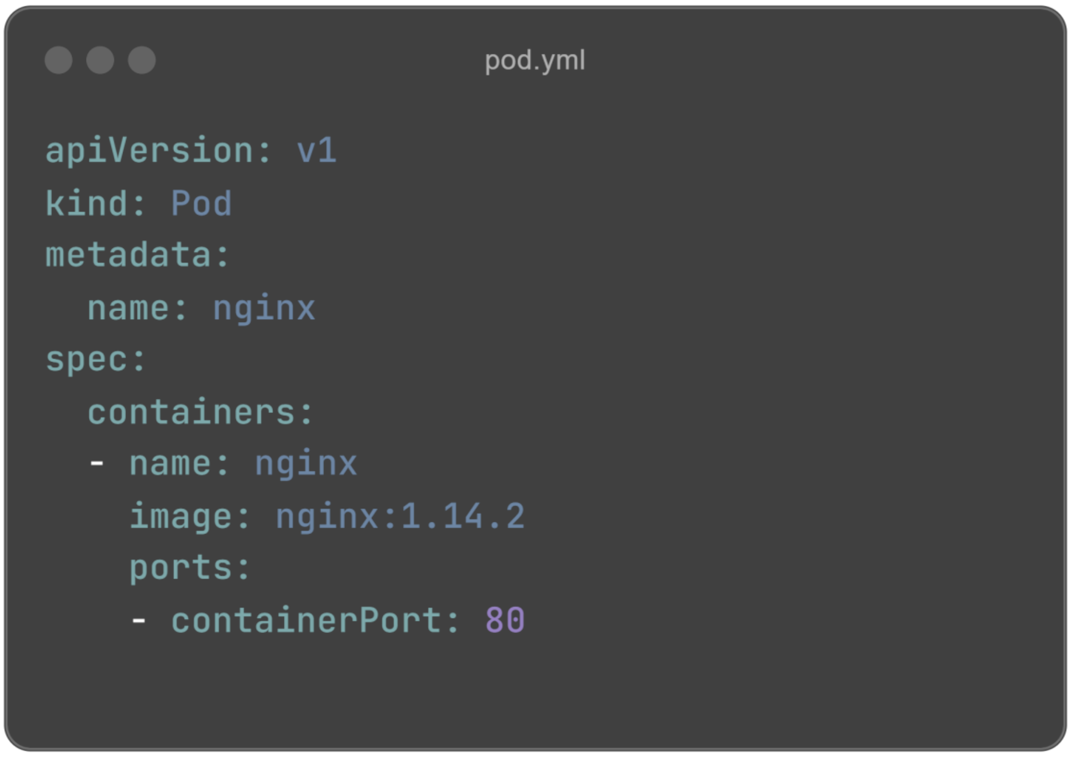
$ kubectl get nodes



```
C:\Users\NARENDRA KUMAR>kubectl get nodes
NAME       STATUS   ROLES           AGE   VERSION
minikube   Ready    control-plane   8h    v1.31.0
```

By default, only one node will get created that acts as both control-pane and data-plane. Here the node is in ready status and name is minikube. This is Single Node Architecture.

STEP 3: Next we have to create pods. For creating pods, we should have an yml code that specifies how the pod should get created. Below is the **“pod.yml”** code that I have followed for simplicity.



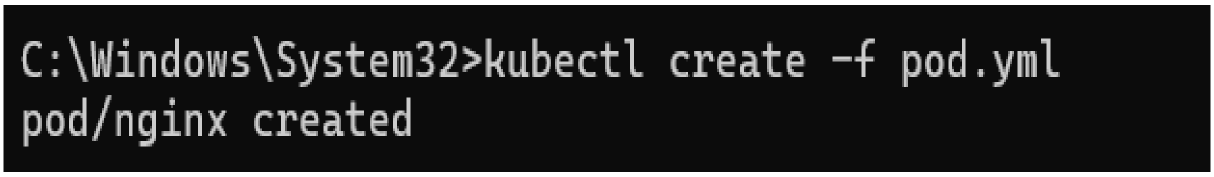
```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - name: nginx
    image: nginx:1.14.2
    ports:
    - containerPort: 80
```

- Since we are creating pod, the **kind** key should have the value, **Pod**. And we are creating a container inside the pod using the image **nginx:1.14.2**. And the **containerPort** is **80**.

- Without the yml code we are unable to create the pods in Kubernetes.

STEP 4: Now create the pod using the “pod.yml” file. By Executing the below command, the pod will get created and automatically runs. Here -f represents file.

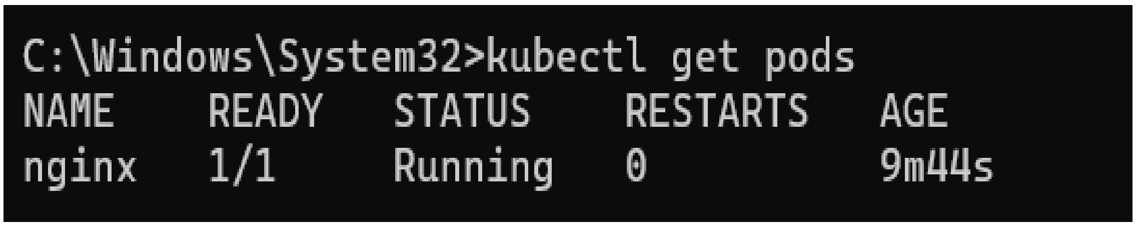
$ kubectl create -f pod.yml



```
C:\Windows\System32>kubectl create -f pod.yml
pod/nginx created
```

- If you want to see the running pods, use the below command.

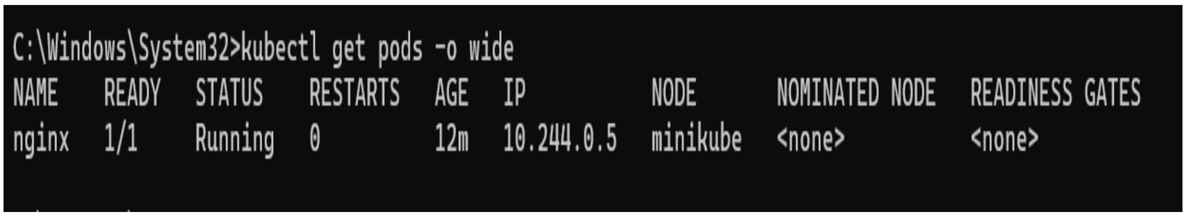
  $ kubect get pods



```
C:\Windows\System32>kubectl get pods
NAME    READY   STATUS    RESTARTS   AGE
nginx   1/1     Running   0          9m44s
```

- If you want to see the full details about the pods, use the below command.

  $ kubectl get pods -o wide



```
C:\Windows\System32>kubectl get pods -o wide
NAME    READY   STATUS    RESTARTS   AGE   IP           NODE       NOMINATED NODE   READINESS GATES
nginx   1/1     Running   0          12m   10.244.0.5   minikube   <none>           <none>
```

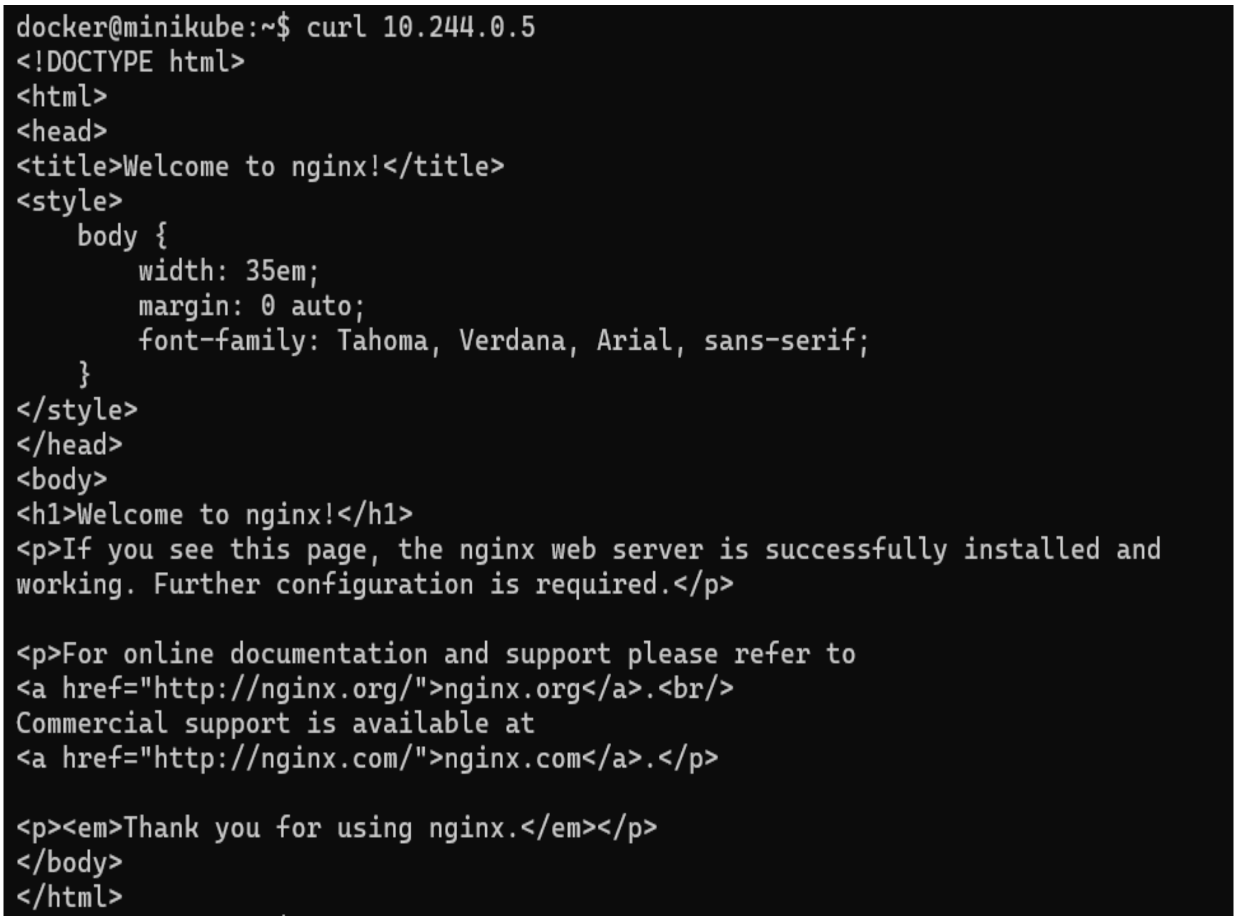
Here **-o** represents the output.

STEP 5: Access the Nginx server. In order to access the server, first you need to login to the Kubernetes cluster. For logging in you need to do SSH. Use the below command to do that.

$ minikube ssh

STEP 6: After doing SSH, request that IP address using Curl command. After doing that you will see the basic home page html code of Nginx Webserver.

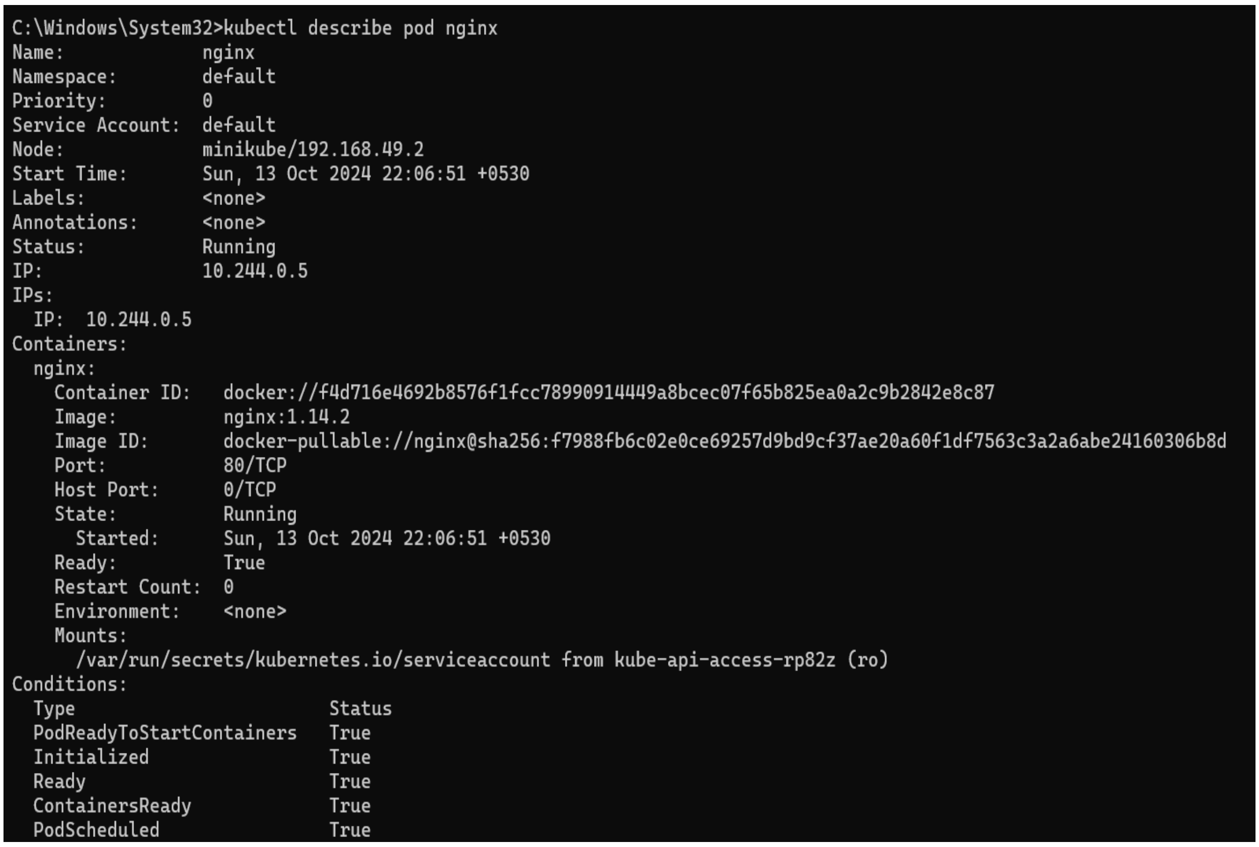
$ curl 10.244.0.5



```
docker@minikube:~$ curl 10.244.0.5
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

STEP 7: We can also see the complete details about the particular pod using the following command.

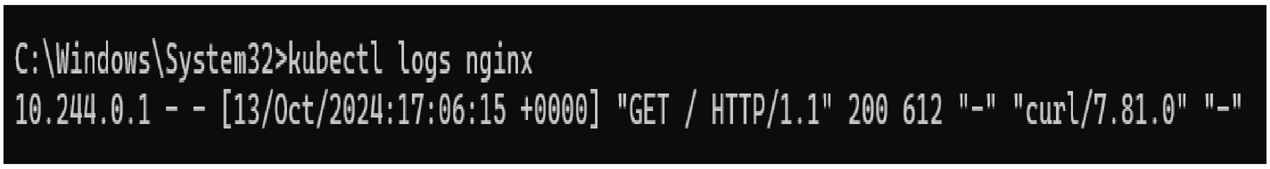
$ kubectl describe pod nginx



```
C:\Windows\System32>kubectl describe pod nginx
Name:             nginx
Namespace:        default
Priority:         0
Service Account:  default
Node:             minikube/192.168.49.2
Start Time:       Sun, 13 Oct 2024 22:06:51 +0530
Labels:           <none>
Annotations:      <none>
Status:           Running
IP:               10.244.0.5
IPs:
  IP:  10.244.0.5
Containers:
  nginx:
    Container ID:   docker://f4d716e4692b8576f1fcc78990914449a8bcec07f65b825ea0a2c9b2842e8c87
    Image:          nginx:1.14.2
    Image ID:       docker-pullable://nginx@sha256:f7988fb6c02e0ce69257d9bd9cf37ae20a60f1df7563c3a2a6abe24160306b8d
    Port:           80/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Sun, 13 Oct 2024 22:06:51 +0530
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-rp82z (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       True
  ContainersReady             True
  PodScheduled                True
```

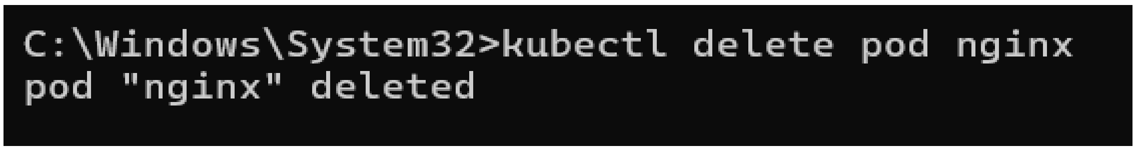
STEP 8: We can also see the logs of the pod using the below command.

$ kubectl logs ngnix



```
C:\Windows\System32>kubectl logs nginx
10.244.0.1 - - [13/Oct/2024:17:06:15 +0000] "GET / HTTP/1.1" 200 612 "-" "curl/7.81.0" "-"
```

- You can also delete the pod using the following command.
  $kubectl delete pod nginx



```
C:\Windows\System32>kubectl delete pod nginx
pod "nginx" deleted
```

THIS IS HOW WE CAN DEPLOY THE APPLICATION USING PODS

<u>REFERENCES</u>

[1] Install and Set Up kubectl on Windows | Kubernetes

[2] minikube start | minikube (k8s.io)

[3] Pods | Kubernetes

[4] Kubernetes Pods | Deploy Your First App | YouTube