

- create minikube cluster

```
(kali@kali)~$ minikube start
minikube v1.30.1 on Debian kali-rolling
+ Automatically selected the docker driver. Other choices: ssh, none

! The requested memory allocation of 2200MiB does not leave room for system overhead (total system memory: 3156MiB). You may face stability issues.
! Suggestion: Start minikube with less memory allocated: 'minikube start --memory=2200mb'

✖ Using Docker driver with root privileges
🔥 Starting control plane node minikube in cluster minikube
📡 Pulling base image ...
📦 Downloading Kubernetes v1.26.3 preload ...
> preloaded-images-k8s-v18-v1 ...: 397.02 MiB / 397.02 MiB 100.00% 860.33 ^[[OS
> gcr.io/k8s-minikube/kicbase ...: 373.53 MiB / 373.53 MiB 100.00% 783.17
🔥 Creating docker container (CPUs=2, Memory=2200MB) ...
📦 Preparing Kubernetes v1.26.3 on Docker 23.0.2 ...
  • 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: storage-provisioner, default-storageclass

! /usr/bin/kubectl is version 1.20.2, which may have incompatibilities with Kubernetes 1.26.3.
  • Want kubectl v1.26.3? Try 'minikube kubectl -- get pods -A'
🏠 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default

(kali@kali)~$ kubectl cluster-info
Kubernetes control plane is running at https://192.168.49.2:8443
CoreDNS is running at https://192.168.49.2:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
```

create nginx deployment

```
(kali㉿kali)-[~/nginx]
$ touch nginx-deployment.yaml

(kali㉿kali)-[~/nginx]
$ nano nginx-deployment.yaml

(kali㉿kali)-[~/nginx]
$ cat nginx-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:latest
        ports:
        - containerPort: 80

(kali㉿kali)-[~/nginx]
$ kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx-deployment created

(kali㉿kali)-[~/nginx]
$ kubectl get deployments
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment    0/3     3            0           31s

(kali㉿kali)-[~/nginx]
$ kubectl get deployments

NAME                READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment    3/3     3            3           94s
```

create nginx service to point to this deployment, type cluster IP

```
(kali㉿kali)-[~/nginx]
$ nano nginx-service.yaml

(kali㉿kali)-[~/nginx]
$ cat nginx-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
  type: NodePort

(kali㉿kali)-[~/nginx]
$ kubectl apply -f nginx-service.yaml

service/nginx-service created

(kali㉿kali)-[~/nginx]
$ kubectl get services
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	48m
nginx-service	NodePort	10.111.196.252	<none>	80:32528/TCP	10s

create debug pod to test the service

```
(kali㉿kali)-[~/nginx]
$ nano debug-pod.yaml

(kali㉿kali)-[~/nginx]
$ cat debug-pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: debug-pod
spec:
  containers:
  - name: debug-container
    image: busybox
    command: ['sleep', '3600']
    restartPolicy: Always

(kali㉿kali)-[~/nginx]
$ kubectl apply -f debug-pod.yaml

pod/debug-pod created
```

```
(kali㉿kali)-[~/nginx]
$ kubectl get pods

NAME                                READY   STATUS    RESTARTS   AGE
debug-pod                           1/1     Running   0           2m36s
nginx-deployment-6b7f675859-8g4p8   1/1     Running   0           14m
nginx-deployment-6b7f675859-q7mdz   1/1     Running   0           14m
nginx-deployment-6b7f675859-zs49w   1/1     Running   0           14m

(kali㉿kali)-[~/nginx]
$ kubectl exec -it debug-pod -- sh

/ #
/ # wget -qO- nginx-service
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
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>
/ # █
```