



# Application Containerization

## EXPERIMENT-9

### Working With Minikube

NIPUN SINGAL

R171218069

500069052

B.Tech-CSE-DevOps-III

# A) launch single node cluster

ApplicationsPlacesFirefox ESR

Mon Apr 12 9:37 AM

1

80 %

Launch Single Node Kubernetes Cluster | Kubernetes | Katacoda - Mozilla Firefox

Launch Single Node Kube X +

https://www.katacoda.com/courses/kubernetes/launch-single-node-cluster

Most VisitedGetting StartedKali TrainingKali ToolsKali DocsKali ForumsNetHunterOffensive SecurityGHDBUWatchFreeWorldfree4u.trade

O'REILLY  
Katacoda

KATACODA OVERVIEW & SOLUTIONS

TRY O'REILLY

LOG IN >

## Launch Single Node Kubernetes Cluster

Step 1 of 4 ▶

Minikube has been installed and configured in the environment. Check that it is properly installed, by running the *minikube version* command:

```
minikube version ✓
```

Start the cluster, by running the *minikube start* command:

```
minikube start --wait=false ✓
```

Great! You now have a running Kubernetes cluster in your online terminal. Minikube started a virtual machine for you, and a Kubernetes cluster is now running in that VM.

CONTINUE

TerminalDashboard +

```
$ minikube start --wait=false
* minikube v1.8.1 on Ubuntu 18.04
* Using the none driver based on existing profile
* Reconfiguring existing host ...
* Using the running none "minikube" bare metal machine ...
* OS release is Ubuntu 18.04.4 LTS
* Preparing Kubernetes v1.17.3 on Docker 19.03.6 ...
  - kubelet.resolv-conf=/run/systemd/resolve/resolv.conf
* Launching Kubernetes ...
* Enabling addons: default-storageclass, storage-provisioner
* Configuring local host environment ...
* Done! kubectl is now configured to use "minikube"
$
$
$ minikube version
minikube version: v1.8.1
commit: cbda04cf6bbe65e987ae52bb393c10099ab62014
$
```



KATACODA OVERVIEW & SOLUTIONS

TRY O'REILLY

LOG IN >

# Launch Single Node Kubernetes Cluster

Step 2 of 4

## Step 2 - Cluster Info

The cluster can be interacted with using the `kubectl` CLI. This is the main approach used for managing Kubernetes and the applications running on top of the cluster.

Details of the cluster and its health status can be discovered via `kubectl cluster-info` ✓

To view the nodes in the cluster using `kubectl get nodes` ✓

If the node is marked as **NotReady** then it is still starting the components.

This command shows all nodes that can be used to

TerminalDashboard +

```
$ kubectl cluster-info
Kubernetes master is running at https://172.17.0.26:8443
KubeDNS is running at https://172.17.0.26:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
$
$
$
$
$
$
$ kubectl get nodes
NAME          STATUS    ROLES    AGE   VERSION
minikube      Ready     master   19m   v1.17.3
$
```



KATACODA OVERVIEW & SOLUTIONS

TRY O'REILLY

LOG IN >

# Launch Single Node Kubernetes Cluster

Step 3 of 4

that provides a dynamic port to a container.

```
kubectl expose deployment first-deployment
--port=80 --type=NodePort ✓
```

The command below finds the allocated port and executes a HTTP request.

```
export PORT=$(kubectl get svc first-deployment
-o go-template='{{range.spec.ports}}{{if
.nodePort}}{{.nodePort}}{{"\n"}}{{end}}
{{end}}')
echo "Accessing host01:$PORT"
curl host01:$PORT ✓
```

The result is the container that processed the request.

TerminalDashboard +

```
$ kubectl create deployment first-deployment --image=katacoda/docker-http-server
deployment.apps/first-deployment created
$
$
$
$
$
$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
first-deployment-666c48b44-wn9r2    0/1     ContainerCreating   0          6s
$
$
$
$ kubectl expose deployment first-deployment --port=80 --type=NodePort
service/first-deployment exposed
$
$
$ export PORT=$(kubectl get svc first-deployment -o go-template='{{range.spec.ports}}{{if .nodePort}}{{.nodePort}}{{"\n"}}{{end}}{{end}}')
$ echo "Accessing host01:$PORT"
Accessing host01:32619
$ curl host01:$PORT
<h1>This request was processed by host: first-deployment-666c48b44-wn9r2</h1>
$
```

CONTINUE

ApplicationsPlacesFirefox ESR

Mon Apr 12 9:49 AM

176%

Launch Single Node Kubernetes Cluster | Kubernetes | Katacoda - Mozilla Firefox

Launch Single Node Kube x +

https://www.katacoda.com/courses/kubernetes/launch-single-node-cluster

Most VisitedGetting StartedKali TrainingKali ToolsKali DocsKali ForumsNetHunterOffensive SecurityGHDBUWatchFreeWorldfree4u.trade

O'REILLYKatacodaKATACODA OVERVIEW & SOLUTIONS

TRY O'REILLYLOG IN >

Launch Single Node Kubernetes Cluster

Step 4 of 4

but may take a while to start.

To see the progress of the Dashboard starting, watch the Pods within the *kube-system* namespace using

kubectl get pods -n kube-system -w ✓

Once running, the URL to the dashboard is

https://2886795283-30000-ollie07.environments.katacoda.com/

CONTINUE

TerminalDashboard +

\$ minikube addons enable dashboard  
\* The 'dashboard' addon is enabled  
\$  
\$  
\$  
\$  
\$  
\$ kubectl apply -f /opt/kubernetes-dashboard.yaml  
namespace/kubernetes-dashboard configured  
service/kubernetes-dashboard-katacoda unchanged  
\$  
\$  
\$  
\$ kubectl get pods -n kube-system -w  
NAME READY STATUS RESTARTS AGE  
dashboard-metrics-scraper-7b64584c5c-g69jp 1/1 Running 0 18s  
kubernetes-dashboard-79d9cd965-k756r 1/1 Running 0 18s

ApplicationsPlacesFirefox ESR

Mon Apr 12 9:50 AM

176%

Kubernetes Dashboard - Mozilla Firefox

Launch Single Node Kube x Kubernetes Dashboard x +

https://2886795283-30000-ollie07.environments.katacoda.com/#/overview?namesp...

Most VisitedGetting StartedKali TrainingKali ToolsKali DocsKali ForumsNetHunterOffensive SecurityGHDBUWatchFreeWorldfree4u.trade

kubernetes

Search

+

Overview

Cluster

Cluster RolesNamespacesNodesPersistent VolumesStorage Classes

Namespace

default

Overview

Workloads

Cron JobsDaemon SetsDeployments

Workload Status

Running: 1100

DeploymentsPodsReplica Sets

Deployments

Name	Namespace	Labels	Pods	Age ↑	Images
✓ first-deployment	default	app: first-deployment	1 / 1	3 minutes	katacoda/docker-http-server

1 - 1 of 1

## B) Using kubectl run containers

ApplicationsPlacesFirefox ESR

Mon Apr 12 12:03 PM

1

46 %

Start containers using Kubectl | Kubernetes | Katacoda - Mozilla Firefox

Start containers using Ku x +

https://www.katacoda.com/courses/kubernetes/kubectl-run-containers

Most VisitedGetting StartedKali TrainingKali ToolsKali DocsKali ForumsNetHunterOffensive SecurityGHDBUWatchFreeWorldfree4u.trade

O'REILLY  
katacoda

KATACODA OVERVIEW & SOLUTIONS

CLAIM YOUR PROFILELOG OUT >

Start containers using Kubectl

Step 1 of 5 ▶

### Step 1 - Launch Cluster

To start we need to launch a Kubernetes cluster.

Execute the command below to start the cluster components and download the Kubectl CLI.

```
minikube start --wait=false ✓
```

Wait for the Node to become Ready by checking

```
kubectl get nodes ✓
```

CONTINUE

Terminal +

Your Interactive Bash Terminal. A safe place to learn and execute commands.

```
$
$ minikube start --wait=false
* minikube v1.6.2 on Ubuntu 18.04
* Selecting 'none' driver from user configuration (alternates: [])
* Running on localhost (CPUs=2, Memory=2461MB, Disk=47990MB) ...
* OS release is Ubuntu 18.04.3 LTS
* Preparing Kubernetes v1.17.0 on Docker '18.09.7' ...
  - kubelet.resolv-conf=/run/systemd/resolve/resolv.conf
* Pulling images ...
* Launching Kubernetes ...
* Configuring local host environment ...
* Done! kubectl is now configured to use "minikube"
$
$
$ kubectl get nodes
NAME      STATUS    ROLES    AGE   VERSION
minikube  Ready     master   90s   v1.17.0
$
```



KATACODA OVERVIEW & SOLUTIONS

CLAIM YOUR PROFILE

LOG OUT >

## Start containers using Kubectl

Step 2 of 5

To find out what Kubernetes created you can describe the deployment process.

```
kubectl describe deployment http ✓
```

The description includes how many replicas are available, labels specified and the events associated with the deployment. These events will highlight any problems and errors that might have occurred.

In the next step we'll expose the running service.

CONTINUE

Terminal +

```
$ kubectl run http --image=katacoda/docker-http-server:latest --replicas=1
kubectl run --generator=deployment/apps.v1 is DEPRECATED and will be removed in a future version.
Use kubectl run --generator=run-pod/v1 or kubectl create instead.
deployment.apps/http created
$
$
$
$ kubectl get deployments
NAME    READY   UP-TO-DATE   AVAILABLE   AGE
http    1/1     1            1           5s
$
$
$ kubectl describe deployment http
Name:                http
Namespace:           default
CreationTimestamp:    Mon, 12 Apr 2021 06:33:32 +0000
Labels:               run=http
Annotations:          deployment.kubernetes.io/revision: 1
Selector:             run=http
Replicas:             1 desired | 1 updated | 1 total | 1 available | 0 unavailable
StrategyType:         RollingUpdate
MinReadySeconds:      0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels:  run=http
  Containers:
    http:
      Image:  katacoda/docker-http-server:latest
```



KATACODA OVERVIEW & SOLUTIONS

CLAIM YOUR PROFILE

LOG OUT >

## Start containers using Kubectl

Step 3 of 5

expose the deployment.

### Task

Use the following command to expose the container port 80 on the host 8000 binding to the external-ip of the host.

```
kubectl expose deployment http --external-ip="172.17.0.49" --port=8000 --target-port=80 ✓
```

You will then be able to ping the host and see the result from the HTTP service.

```
curl http://172.17.0.49:8000 ✓
```

CONTINUE

Terminal +

```
$ kubectl expose deployment http --external-ip="172.17.0.49" --port=8000 --target-port=80
service/http exposed
$
$
$
$
$
$
$ curl http://172.17.0.49:8000
<h1>This request was processed by host: http-774bb756bb-p5nr6</h1>
$ □
```

ApplicationsPlacesFirefox ESR

Mon Apr 12 12:04 PM

145 %

Start containers using Kubectl | Kubernetes | Katacoda - Mozilla Firefox

Start containers using Ku x +

https://www.katacoda.com/courses/kubernetes/kubectl-run-containers

Most VisitedGetting StartedKali TrainingKali ToolsKali DocsKali ForumsNetHunterOffensive SecurityGHDBUWatchFreeWorldfree4u.trade

O'REILLYKatacoda

KATACODA OVERVIEW & SOLUTIONS

CLAIM YOUR PROFILE

LOG OUT >

Start containers using Kubectl

Step 4 of 5

## Step 4 - Kubectl Run and Expose

With *kubectl run* it's possible to create the deployment and expose it as a single command.

### Task

Use the command command to create a second http service exposed on port 8001.

```
kubectl run httpexposed --image=katacoda/docker-http-server:latest --replicas=1 --port=80 --hostport=8001 ✓
```

You should be able to access it using

```
curl http://172.17.0.49:8001 ✓
```

Terminal +

\$ kubectl run httpexposed --image=katacoda/docker-http-server:latest --replicas=1 --port=80 --hostport=8001  
kubectl run --generator=deployment/apps.v1 is DEPRECATED and will be removed in a future version. Use kubectl run --generator=run-pod/v1 or kubectl create instead.  
deployment.apps/httpexposed created  
\$  
\$  
\$ curl http://172.17.0.49:8001  
curl: (7) Failed to connect to 172.17.0.49 port 8001: Connection refused  
\$  
\$  
\$ kubectl get svc  
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE  
http ClusterIP 10.96.184.198 172.17.0.49 8000/TCP 29s  
kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 2m32s  
\$  
\$  
\$ docker ps | grep httpexposed  
042baelad51c katacoda/docker-http-server "/app" 8 seconds ago Up  
8 seconds k8s\_httpexposed\_httpexposed-68cb8c8d4-c9vfj\_default\_178b24  
ce-6673-4860-b344-29277f02232b\_0  
da08c5e9712d k8s.gcr.io/pause:3.1 "/pause" 10 seconds ago Up  
9 seconds 0.0.0.0:8001->80/tcp k8s\_POD\_httpexposed-68cb8c8d4-c9vfj\_default\_178b24ce-6673-4860-b344-29277f02232b\_0  
\$ □

ApplicationsPlacesFirefox ESR

Mon Apr 12 12:04 PM

145 %

Start containers using Kubectl | Kubernetes | Katacoda - Mozilla Firefox

Start containers using Ku x +

https://www.katacoda.com/courses/kubernetes/kubectl-run-containers

Most VisitedGetting StartedKali TrainingKali ToolsKali DocsKali ForumsNetHunterOffensive SecurityGHDBUWatchFreeWorldfree4u.trade

O'REILLYKatacoda

KATACODA OVERVIEW & SOLUTIONS

CLAIM YOUR PROFILE

LOG OUT >

Start containers using Kubectl

Step 5 of 5

Once each Pod starts it will be added to the load balancer service. By describing the service you can view the endpoint and the associated Pods which are included.

```
kubectl describe svc http ✓
```

Making requests to the service will request in different nodes processing the request.

```
curl http://172.17.0.49:8000 ↵
```

CONTINUE

Terminal +

\$ kubectl scale --replicas=3 deployment http  
deployment.apps/http scaled  
\$  
\$  
\$  
\$  
\$ kubectl get pods  
NAME READY STATUS RESTARTS AGE  
http-774bb756bb-cdzx9 1/1 Running 0 5s  
http-774bb756bb-p5nr6 1/1 Running 0 74s  
http-774bb756bb-wdkzx 1/1 Running 0 5s  
httpexposed-68cb8c8d4-c9vfj 1/1 Running 0 28s  
\$  
\$  
\$  
\$ kubectl describe svc http  
Name: http  
Namespace: default  
Labels: run=http  
Annotations: <none>  
Selector: run=http  
Type: ClusterIP  
IP: 10.96.184.198  
External IPs: 172.17.0.49  
Port: <unset> 8000/TCP  
TargetPort: 80/TCP  
Endpoints: 172.18.0.6:80,172.18.0.8:80,172.18.0.9:80  
Session Affinity: None  
Events: <none>