Hello Minikube:

Use the `kubectl create` command to create a Deployment that manages a Pod. The Pod runs a Container based on the provided Docker image.

```
C:\Users\Shiroy>kubectl create deployment hello-node --image=k8s.gcr.io/echoserver:1.4
deployment.apps/hello-node created
```

View the Deployment:

```
C:\Users\Shiroy>kubectl get deployments
NAME         READY   UP-TO-DATE   AVAILABLE   AGE
hello-node   1/1     1            1           2m40s
```

View the Pod:

```
C:\Users\Shiroy>kubectl get pods
NAME                          READY   STATUS    RESTARTS   AGE
hello-node-7567d9fdc9-jbk8c   1/1     Running   0          3m6s
```

View cluster events:

```
C:\Users\Shiroy>kubectl get events
LAST SEEN   TYPE      REASON                    OBJECT                         MESSAGE
3m22s       Normal    Scheduled                 pod/hello-node-7567d9fdc9-jbk8c   Successfully assigned default/hello-node-7567d9fdc9-jbk8c to minikub
e
3m21s       Normal    Pulling                   pod/hello-node-7567d9fdc9-jbk8c   Pulling image "k8s.gcr.io/echoserver:1.4"
3m14s       Normal    Pulled                    pod/hello-node-7567d9fdc9-jbk8c   Successfully pulled image "k8s.gcr.io/echoserver:1.4" in 7.1967351s
3m11s       Normal    Created                   pod/hello-node-7567d9fdc9-jbk8c   Created container echoserver
3m11s       Normal    Started                   pod/hello-node-7567d9fdc9-jbk8c   Started container echoserver
3m22s       Normal    SuccessfulCreate          replicaset/hello-node-7567d9fdc9  Created pod: hello-node-7567d9fdc9-jbk8c
3m22s       Normal    ScalingReplicaSet         deployment/hello-node             Scaled up replica set hello-node-7567d9fdc9 to 1
22m         Normal    NodeHasSufficientMemory   node/minikube                     Node minikube status is now: NodeHasSufficientMemory
22m         Normal    NodeHasNoDiskPressure     node/minikube                     Node minikube status is now: NodeHasNoDiskPressure
22m         Normal    NodeHasSufficientPID      node/minikube                     Node minikube status is now: NodeHasSufficientPID
22m         Normal    Starting                  node/minikube                     Starting kubelet.
22m         Normal    NodeHasSufficientMemory   node/minikube                     Node minikube status is now: NodeHasSufficientMemory
22m         Normal    NodeHasNoDiskPressure     node/minikube                     Node minikube status is now: NodeHasNoDiskPressure
22m         Normal    NodeHasSufficientPID      node/minikube                     Node minikube status is now: NodeHasSufficientPID
22m         Normal    NodeNotReady              node/minikube                     Node minikube status is now: NodeNotReady
22m         Normal    NodeAllocatableEnforced   node/minikube                     Updated Node Allocatable limit across pods
22m         Normal    RegisteredNode            node/minikube                     Node minikube event: Registered Node minikube in Controller
22m         Warning   readOnlySysFS             node/minikube                     CRI error: /sys is read-only: cannot modify conntrack limits, proble
ms may arise later (If running Docker, see docker issue #24000)
22m         Normal    Starting                  node/minikube                     Starting kube-proxy.
22m         Normal    NodeReady                 node/minikube                     Node minikube status is now: NodeReady
```

View the `kubectl` configuration:

```
C:\Users\Shiroy>kubectl config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority: C:\Users\Shiroy\.minikube\ca.crt
    server: https://127.0.0.1:32776
  name: minikube
contexts:
- context:
    cluster: minikube
    user: minikube
  name: minikube
current-context: minikube
kind: Config
preferences: {}
users:
- name: minikube
  user:
    client-certificate: C:\Users\Shiroy\.minikube\profiles\minikube\client.crt
    client-key: C:\Users\Shiroy\.minikube\profiles\minikube\client.key
```

Create a Service

Expose the Pod to the public internet using the `kubectl expose` command:

```
C:\Users\Shiroy>kubectl expose deployment hello-node --type=LoadBalancer --port=8080
service/hello-node exposed
```

View the Service you just created:

```
C:\Users\Shiroy>kubectl get services
NAME         TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
hello-node   LoadBalancer   10.103.91.57    <pending>     8080:30071/TCP   14s
kubernetes   ClusterIP      10.96.0.1       <none>        443/TCP          23m
```

Run the following command:

```
C:\Users\Shiroy>minikube service hello-node
|-----------|------------|-------------|------------------------|
| NAMESPACE |    NAME    | TARGET PORT |          URL           |
|-----------|------------|-------------|------------------------|
| default   | hello-node |        8080 | http://192.168.49.2:30071 |
|-----------|------------|-------------|------------------------|
* Starting tunnel for service hello-node.
|-----------|------------|-------------|------------------------|
| NAMESPACE |    NAME    | TARGET PORT |          URL           |
|-----------|------------|-------------|------------------------|
| default   | hello-node |             | http://127.0.0.1:10728 |
|-----------|------------|-------------|------------------------|
* 正通过默认浏览器打开服务 default/hello-node...
! Because you are using a Docker driver on windows, the terminal needs to be open to run it.
```

Enable addons

List the currently supported addons:

```
C:\Users\Shiroy>minikube addons list
|-----------------------------|----------|--------------|
|         ADDON NAME          | PROFILE  |    STATUS    |
|-----------------------------|----------|--------------|
| ambassador                  | minikube | disabled     |
| csi-hostpath-driver         | minikube | disabled     |
| dashboard                   | minikube | disabled     |
| default-storageclass        | minikube | enabled ☐    |
| efk                         | minikube | disabled     |
| freshpod                    | minikube | disabled     |
| gcp-auth                    | minikube | disabled     |
| gvisor                      | minikube | disabled     |
| helm-tiller                 | minikube | disabled     |
| ingress                     | minikube | disabled     |
| ingress-dns                 | minikube | disabled     |
| istio                       | minikube | disabled     |
| istio-provisioner           | minikube | disabled     |
| kubevirt                    | minikube | disabled     |
| logviewer                   | minikube | disabled     |
| metallb                     | minikube | disabled     |
| metrics-server              | minikube | disabled     |
| nvidia-driver-installer     | minikube | disabled     |
| nvidia-gpu-device-plugin    | minikube | disabled     |
| olm                         | minikube | disabled     |
| pod-security-policy         | minikube | disabled     |
| registry                    | minikube | disabled     |
| registry-aliases            | minikube | disabled     |
| registry-creds              | minikube | disabled     |
| storage-provisioner         | minikube | enabled ☐    |
| storage-provisioner-gluster | minikube | disabled     |
| volumesnapshots             | minikube | disabled     |
|-----------------------------|----------|--------------|
```

Enable an addon, for example, `metrics-server`:

```
C:\Users\Shiroy>minikube addons enable metrics-server
* 启动 'metrics-server' 插件
```

View the Pod and Service you just created:

```
C:\Users\Shiroy>kubectl get pod,svc -n kube-system
NAME                                   READY   STATUS    RESTARTS   AGE
pod/coredns-f9fd979d6-tkbgk            1/1     Running   0          26m
pod/etcd-minikube                      1/1     Running   0          26m
pod/kube-apiserver-minikube            1/1     Running   0          26m
pod/kube-controller-manager-minikube   1/1     Running   0          26m
pod/kube-proxy-k886n                   1/1     Running   0          26m
pod/kube-scheduler-minikube            1/1     Running   0          26m
pod/metrics-server-d9b576748-24p5t     1/1     Running   0          17s
pod/storage-provisioner                1/1     Running   0          26m

NAME                     TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)                  AGE
service/kube-dns         ClusterIP   10.96.0.10     <none>        53/UDP,53/TCP,9153/TCP   26m
service/metrics-server   ClusterIP   10.99.71.164   <none>        443/TCP                  17s
```

Disable `metrics-server`:

```
C:\Users\Shiroy>minikube addons disable metrics-server
* "The 'metrics-server' addon is disabled
```

Clean up

Now you can clean up the resources you created in your cluster:

```
C:\Users\Shiroy>kubectl delete service hello-node
service "hello-node" deleted

C:\Users\Shiroy>kubectl delete deployment hello-node
deployment.apps "hello-node" deleted
```

Optionally, stop the Minikube virtual machine (VM):

```
C:\Users\Shiroy>minikube stop
* Stopping node "minikube"  ...
* 正在通过 SSH 关闭“minikube”…
* 1 nodes stopped.
```

Optionally, delete the Minikube VM:

```
C:\Users\Shiroy>minikube delete
* 正在删除 docker 中的“minikube”…
* Deleting container "minikube" ...
* 正在移除 C:\Users\Shiroy\.minikube\machines\minikube…
* Removed all traces of the "minikube" cluster.
```

Interact with cluster

From a terminal with administrator access (but not logged in as root), run:

```
C:\Users\Shiroy>minikube start
* Microsoft Windows 10 Home 10.0.19041 Build 19041 上的 minikube v1.14.1
* Automatically selected the docker driver
* Starting control plane node minikube in cluster minikube
* Pulling base image ...
* Downloading Kubernetes v1.19.2 preload ...
    > preloaded-images-k8s-v6-v1.19.2-docker-overlay2-amd64.tar.lz4: 486.33 MiB
* Creating docker container (CPUs=2, Memory=6100MB) ...
* 正在 Docker 19.03.8 中准备 Kubernetes v1.19.2…
* Verifying Kubernetes components...
* Enabled addons: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" by default
```

If you already have kubectl installed, you can now use it to access your shiny new cluster:

```
C:\Users\Shiroy>kubectl get po -A
NAMESPACE     NAME                               READY   STATUS    RESTARTS   AGE
kube-system   coredns-f9fd979d6-xmxc9            1/1     Running   0          72s
kube-system   etcd-minikube                      0/1     Running   0          77s
kube-system   kube-apiserver-minikube            1/1     Running   0          77s
kube-system   kube-controller-manager-minikube   0/1     Running   0          77s
kube-system   kube-proxy-2gtkj                   1/1     Running   0          72s
kube-system   kube-scheduler-minikube            1/1     Running   0          77s
kube-system   storage-provisioner                1/1     Running   0          77s
```

Alternatively, minikube can download the appropriate version of kubectl, if you don't mind the double-dashes in the command-line:

```
C:\Users\Shiroy>minikube kubectl -- get po -A
    > kubectl.exe.sha256: 65 B / 65 B [----------------------] 100.00% ? p/s 0s
    > kubectl.exe: 42.26 MiB / 42.26 MiB [----------] 100.00% 33.78 MiB p/s 2s
NAMESPACE     NAME                               READY   STATUS    RESTARTS   AGE
kube-system   coredns-f9fd979d6-xmxc9            1/1     Running   0          3m2s
kube-system   etcd-minikube                      1/1     Running   0          3m7s
kube-system   kube-apiserver-minikube            1/1     Running   0          3m7s
kube-system   kube-controller-manager-minikube   1/1     Running   0          3m7s
kube-system   kube-proxy-2gtkj                   1/1     Running   0          3m2s
kube-system   kube-scheduler-minikube            1/1     Running   0          3m7s
kube-system   storage-provisioner                1/1     Running   0          3m7s
```

Initially, some services such as the storage-provisioner, may not yet be in a Running state. This is a normal condition during cluster bring-up, and will resolve itself momentarily. For additional insight into your cluster state, minikube bundles the Kubernetes Dashboard, allowing you to get easily acclimated to your new environment:

```
C:\Users\Shiroy>minikube dashboard
* 正在开启 dashboard ...
* 正在验证 dashboard 运行情况 ...
* Launching proxy ...
* 正在验证 proxy 运行状况 ...
* Opening http://127.0.0.1:6011/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ in you
r default browser...
```

Deploy applications

Create a sample deployment and expose it on port 8080:

```
C:\Users\Shiroy>kubectl create deployment hello-minikube --image=k8s.gcr.io/echoserver:1.4
deployment.apps/hello-minikube created
```

It may take a moment, but your deployment will soon show up when you run:

```
C:\Users\Shiroy>kubectl expose deployment hello-minikube --type=NodePort --port=8080
service/hello-minikube exposed
```

The easiest way to access this service is to let minikube launch a web browser for you:

```
C:\Users\Shiroy>kubectl get services hello-minikube
NAME             TYPE       CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
hello-minikube   NodePort   10.109.45.12    <none>        8080:31487/TCP   61s
```

Alternatively, use kubectl to forward the port:

```
C:\Users\Shiroy>kubectl get services hello-minikube
NAME             TYPE       CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
hello-minikube   NodePort   10.109.45.12    <none>        8080:31487/TCP   61s

C:\Users\Shiroy>minikube service hello-minikube
|-----------|----------------|-------------|-----------------------|
| NAMESPACE |      NAME      | TARGET PORT |          URL          |
|-----------|----------------|-------------|-----------------------|
| default   | hello-minikube |        8080 | http://192.168.49.2:31487 |
|-----------|----------------|-------------|-----------------------|
* Starting tunnel for service hello-minikube.
|-----------|----------------|-------------|-----------------------|
| NAMESPACE |      NAME      | TARGET PORT |          URL          |
|-----------|----------------|-------------|-----------------------|
| default   | hello-minikube |             | http://127.0.0.1:6821 |
|-----------|----------------|-------------|-----------------------|
* 正通过默认浏览器打开服务 default/hello-minikube...
! Because you are using a Docker driver on windows, the terminal needs to be open to run it.
```


```
CLIENT VALUES:
client_address=192.168.49.2
command=GET
real path=/
query=nil
request_version=1.1
request_uri=http://127.0.0.1:8080/

SERVER VALUES:
server_version=nginx: 1.10.0 - lua: 10001

HEADERS RECEIVED:
accept=text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
accept-encoding=gzip, deflate, br
accept-language=zh-CN,zh;q=0.9,en;q=0.8,en-US;q=0.7,pl;q=0.6
connection=keep-alive
host=127.0.0.1:6821
sec-ch-ua="Chromium";v="86", "\"Not\\A;Brand";v="99", "Google Chrome";v="86"
sec-ch-ua-mobile=?0
sec-fetch-dest=document
sec-fetch-mode=navigate
sec-fetch-site=none
sec-fetch-user=?1
upgrade-insecure-requests=1
user-agent=Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.111 Safari/537.36
BODY:
-no body in request-
```

```
C:\Users\Shiroy>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
Handling connection for 7080
```


```
CLIENT VALUES:
client_address=127.0.0.1
command=GET
real path=/
query=nil
request_version=1.1
request_uri=http://localhost:8080/

SERVER VALUES:
server_version=nginx: 1.10.0 - lua: 10001

HEADERS RECEIVED:
accept=text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
accept-encoding=gzip, deflate, br
accept-language=zh-CN,zh;q=0.9,en;q=0.8,en-US;q=0.7,pl;q=0.6
connection=keep-alive
host=localhost:7080
sec-ch-ua="Chromium";v="86", "\"Not\\A;Brand";v="99", "Google Chrome";v="86"
sec-ch-ua-mobile=?0
sec-fetch-dest=document
sec-fetch-mode=navigate
sec-fetch-site=cross-site
sec-fetch-user=?1
upgrade-insecure-requests=1
user-agent=Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.111 Safari/537.36
BODY:
-no body in request-
```

LoadBalancer deployments

To access a LoadBalancer deployment, use the “minikube tunnel” command. Here is an example deployment:

```
C:\Users\Shiroy>kubectl create deployment blanced —image=k8.gcr.io/echoserver:1.4
deployment.apps/blanced created
```

```
C:\Users\Shiroy>kubectl expose deployment blanced —type=LoadBalancer —port=8080
service/blanced exposed
```

In another window, start the tunnel to create a routable IP for the ‘balanced’ deployment:

```
C:\Users\Shiroy>minikube tunnel
* Starting tunnel for service blanced.
```

To find the routable IP, run this command and examine the `EXTERNAL-IP` column:

```
C:\Users\Shiroy>kubectl get services blanced
NAME      TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
blanced   LoadBalancer   10.107.140.0   127.0.0.1     8080:31775/TCP   85s
```

Manage cluster

Pause Kubernetes without impacting deployed applications:

```
C:\Users\Shiroy>minikube pause
* Pausing node minikube ...
* Paused 18 containers in: kube-system, kubernetes-dashboard, storage-gluster, istio-operator
```

Halt the cluster:

```
C:\Users\Shiroy>minikube stop
* Stopping node "minikube"  ...
* 正在通过 SSH 关闭“minikube”…
* 1 nodes stopped.
```

Increase the default memory limit (requires a restart):

```
C:\Users\Shiroy>minikube config set memory 16384
! These changes will take effect upon a minikube delete and then a minikube start
```

Browse the catalog of easily installed Kubernetes services:

```
C:\Users\Shiroy>minikube addons list
|-----------------------------|----------|--------------|
|         ADDON NAME          | PROFILE  |    STATUS    |
|-----------------------------|----------|--------------|
| ambassador                  | minikube | disabled     |
| csi-hostpath-driver         | minikube | disabled     |
| dashboard                   | minikube | disabled     |
| default-storageclass        | minikube | enabled ☐    |
| efk                         | minikube | disabled     |
| freshpod                    | minikube | disabled     |
| gcp-auth                    | minikube | disabled     |
| gvisor                      | minikube | disabled     |
| helm-tiller                 | minikube | disabled     |
| ingress                     | minikube | disabled     |
| ingress-dns                 | minikube | disabled     |
| istio                       | minikube | disabled     |
| istio-provisioner           | minikube | disabled     |
| kubevirt                    | minikube | disabled     |
| logviewer                   | minikube | disabled     |
| metallb                     | minikube | disabled     |
| metrics-server              | minikube | disabled     |
| nvidia-driver-installer     | minikube | disabled     |
| nvidia-gpu-device-plugin    | minikube | disabled     |
| olm                         | minikube | disabled     |
| pod-security-policy         | minikube | disabled     |
| registry                    | minikube | disabled     |
| registry-aliases            | minikube | disabled     |
| registry-creds              | minikube | disabled     |
| storage-provisioner         | minikube | enabled ☐    |
| storage-provisioner-gluster | minikube | disabled     |
| volumesnapshots             | minikube | disabled     |
|-----------------------------|----------|--------------|
```

Create a second cluster running an older Kubernetes release:

```
C:\Users\Shiroy>minikube start -p aged —kubernetes-version=v1.16.1
* Microsoft Windows 10 Home 10.0.19042 Build 19042 上的 [aged] minikube v1.14.1
* Automatically selected the docker driver
* Starting control plane node aged in cluster aged
* Creating docker container (CPUs=2, Memory=16384MB) ...
* 正在 Docker 19.03.8 中准备 Kubernetes v1.16.1…
* Verifying Kubernetes components...
* Enabled addons: storage-provisioner, default-storageclass

! C:\Program Files\Docker\Docker\resources\bin\kubectl.exe is version 1.18.8, which may have incompatibilites with Kuber
netes 1.16.1.
* Want kubectl v1.16.1? Try 'minikube kubectl — get pods -A'
* Done! kubectl is now configured to use "aged" by default
```

Delete all of the minikube clusters:

```
C:\Users\Shiroy>minikube delete —all
* 正在删除 docker 中的“aged” …
* 正在移除 C:\Users\Shiroy\.minikube\machines\aged…
* Removed all traces of the "aged" cluster.
* 正在删除 docker 中的“minikube” …
* 正在移除 C:\Users\Shiroy\.minikube\machines\minikube…
* Removed all traces of the "minikube" cluster.
* 成功删除所有配置文件
```