**Install minikube. Deploy, run, and test a public (image image=k8s.gcr.io/echoserver:1.4) in minikube**

Prerequisites: Minikube, Kubernetes, docker daemon. Minikube is shipped with a kubernetes client/server. It is also shipped with a docker daemon. The version of minikube installed with the link below is **Kubernetes v1.17.3.** The kubernetes installed at the link

curl -LO [https://storage.googleapis.com/kubernetes-release/release/](https://storage.googleapis.com/kubernetes-release/release/v1.19.0/bin/windows/amd64/kubectl.exe)**[v1.19.0](https://storage.googleapis.com/kubernetes-release/release/v1.19.0/bin/windows/amd64/kubectl.exe)**[/bin/windows/amd64/](https://storage.googleapis.com/kubernetes-release/release/v1.19.0/bin/windows/amd64/kubectl.exe)**[kubectl.exe](https://storage.googleapis.com/kubernetes-release/release/v1.19.0/bin/windows/amd64/kubectl.exe).**

Because of this version discrepancy the warning message appears at the end of minikube start:

! **C:\bin\kubectl.exe is v1.19.0**, which **may be incompatible with Kubernetes v1.17.3**.

To resolve this incompatibility issue I downloaded kubectl from a different link, placed the executable in a directory bin that I have added to my path.

<https://storage.googleapis.com/kubernetes-release/release/v1.17.3/bin/windows/amd64/kubectl.exe>

Now, that the version of kubectl matches the version of kubectl in minikube, the warning message does not appear. This is from the powershell console:

PS C:\WINDOWS\system32> **minikube status**

host: Running

kubelet: Running

apiserver: Running

kubeconfig: Configured

PS C:\WINDOWS\system32> **minikube stop**

\* Stopping "minikube" in hyperv ...

\* Powering off "minikube" via SSH ...

\* Node "m01" stopped.

PS C:\WINDOWS\system32> **minikube start**

\* minikube v1.8.2 on Microsoft Windows 10 Pro 10.0.18362 Build 18362

\* Using the hyperv driver based on existing profile

\* Reconfiguring existing host ...

\* Starting existing hyperv VM for "minikube" ...

\* Preparing Kubernetes v1.17.3 on Docker 19.03.6 ...

\* Launching Kubernetes ...

\* Enabling addons: dashboard, default-storageclass, storage-provisioner

\* **Done! kubectl is now configured to use "minikube"**

**Steps:**

1. I have stopped and deleted the existing minikube

2. Downloaded

<https://github.com/kubernetes/minikube/releases/download/v1.8.2/minikube-windows-amd64.exe>

and renamed it to minikube.exe

3. Placed the above minikube.exe in the same folder in which my kubectl.exe is existing. (C:\kube). This path is added to the environment variable "Path" of my user. You don’t need to install kubectl separately. You can use the kubectl utility that comes with minikube installation.

4. Created a hyperv switch, Open Hyper-V Manager, Click on Virtual Switch Manager, Create New Virtual Network Switch, Select external type, and OK. (instructions at: <https://docs.docker.com/machine/drivers/hyper-v/>)

5. Start minikube with admin rights "

**minikube start --vm-driver="hyperv" --hyperv-virtual-switch="myswitch"**

7. minikube successfully started.

PS C:\WINDOWS\system32> minikube start

\* **minikube v1.8.2** on Microsoft Windows 10 Pro 10.0.18362 Build 18362

\* Using the hyperv driver based on existing profile

\* Reconfiguring existing host ...

\* Starting existing hyperv VM for "minikube" ...

\* Preparing **Kubernetes v1.17.3** on **Docker 19.03.6** ...

\* Launching Kubernetes ...

\* Enabling addons: dashboard, default-storageclass, storage-provisioner

\* Done! kubectl is now configured to use "minikube"

! **C:\bin\kubectl.exe is v1.19.0**, which **may be incompatible with Kubernetes v1.17.3**.

\* You can also use 'minikube kubectl -- get pods' to invoke a matching version

PS C:\WINDOWS\system32> **minikube kubectl version**

**Client Version**: version.Info{**Major:"1", Minor:"17",** GitVersion:"v1.17.3", GitCommit:"06ad960bfd03b39c8310aaf92d1e7c12ce618213", GitTreeState:"clean", BuildDate:"2020-02-11T18:14:22Z", GoVersion:"go1.13.6", Compiler:"gc", Platform:"windows/amd64"}

**Server Version**: version.Info{**Major:"1", Minor:"17",** GitVersion:"v1.17.3", GitCommit:"06ad960bfd03b39c8310aaf92d1e7c12ce618213", GitTreeState:"clean", BuildDate:"2020-02-11T18:07:13Z", GoVersion:"go1.13.6", Compiler:"gc", Platform:"linux/amd64"}

8. issue command to retrieve minikubes ip address

**minikube ip**

8. Issue command **minikube status** to check the status of minikube (both commands are run with admin privileges) :

monica@DESKTOP-0MV2FVN /cygdrive/c/windows/System32/Drivers/etc

$ **minikube ip**

192.168.1.24

monica@DESKTOP-0MV2FVN /cygdrive/c/windows/System32/Drivers/etc

$ **minikube status**

host: Running

kubelet: Running

apiserver: Running

kubeconfig: Configured

monica@DESKTOP-0MV2FVN /cygdrive/c/windows/System32/Drivers/etc

By default, kubectl gets configured to access the kubernetes cluster control plane inside minikube when the minikube start command is executed. However if kubectl is not installed locally, kubectl can be used inside the minikube as well.

minikube kubectl -- <kubectl commands>

You can also alias kubectl="minikube kubectl --" for easier usage.

Get pods

**minikube kubectl -- get pods**

Creating a deployment inside kubernetes cluster

9. **minikube kubectl -- create deployment hello-minikube --image=k8s.gcr.io/echoserver:1.4**

Exposing the deployment with a NodePort service. Do not change --port=8080

10. **minikube kubectl -- expose deployment hello-minikube --type=NodePort --port=8080**

11. **minikube kubectl -- get services**

12. $ minikube kubectl get services

monica@DESKTOP-0MV2FVN /cygdrive/c/bin

$ **minikube kubectl get services**

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

hello-minikube NodePort 10.100.106.123 <none> 8080:30938/TCP 17m

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 56m

13. test hello-minikube service with curl:

**Use the ip address of minikube (command: minikube ip. You must run this command with admin privileges) and the port published under PORT(S)**

when you ran the command kubectl get services:

$ curl http://192.168.1.24:30938

$ curl http://192.168.1.24:30938

CLIENT VALUES:

client\_address=172.17.0.1

command=GET

real path=/

query=nil

request\_version=1.1

request\_uri=http://192.168.1.24:8080/

SERVER VALUES:

server\_version=nginx: 1.10.0 - lua: 10001

HEADERS RECEIVED:

accept=\*/\*

host=192.168.1.24:30938

user-agent=curl/7.55.1

BODY:

-no body in request-

14. test hello-minikube service with browser at url:

**http://192.168.1.24:30938**



**CLEANUP:**

15. **minikube -- kubectl delete service hello-minikube**

16. **minikube -- kubectl delete deployment hello-minikube**

Optionally, stop the Minikube virtual machine (VM):

17. **minikube stop**

Optionally, delete the Minikube VM:

18. **minikube delete**

**View/Manage hello-minikube service in kubernetes dashboard**

1. **Install kubectl**

monica@DESKTOP-0MV2FVN /cygdrive/c/bin

$ **curl -LO https://storage.googleapis.com/kubernetes-release/release/v1.19.0/bin/windows/amd64/kubectl.exe**

% Total % Received % Xferd Average Speed Time Time Time Current

Dload Upload Total Spent Left Speed

100 42.2M 100 42.2M 0 0 7213k 0 0:00:06 0:00:06 --:--:-- 6837k

monica@DESKTOP-0MV2FVN /cygdrive/c/bin

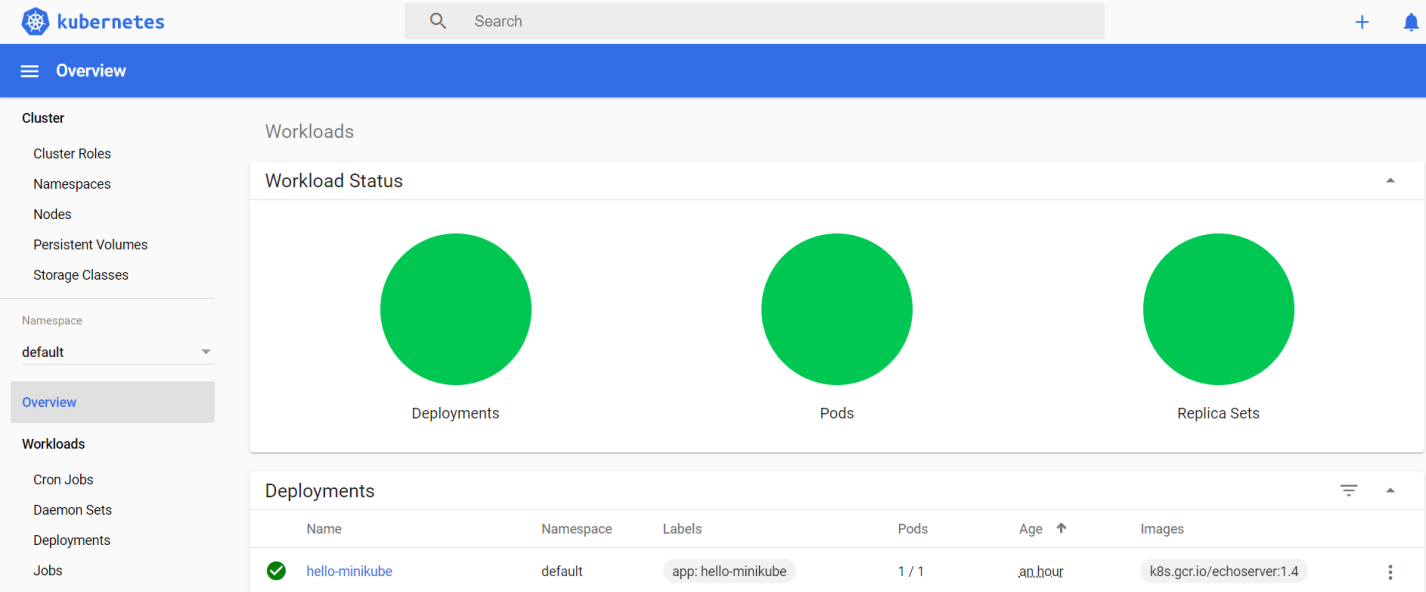
$ ls

kompose.exe\* kubectl.exe\* minikube.exe\*

monica@DESKTOP-0MV2FVN /cygdrive/c/bin

$

1. **Add kubectl.exe to you PATH.**
2. **Start the kubernetes dashboard with command. Run this command with admin priviliges:**
3. $ **minikube dashboard**
4. \* Enabling dashboard ...
5. \* Verifying dashboard health ...
6. \* Launching proxy ...
7. \* Verifying proxy health ...
8. \* Opening http://127.0.0.1:51519/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ in your defau
9. lt browser...



monica@DESKTOP-0MV2FVN /cygdrive/c/bin

$ **kubectl get nodes**

NAME STATUS ROLES AGE VERSION

m01 Ready master 103m v1.17.3

monica@DESKTOP-0MV2FVN /cygdrive/c/bin

$ **kubectl get services**

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

hello-minikube NodePort 10.100.106.123 <none> 8080:30938/TCP 64m

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 103m

monica@DESKTOP-0MV2FVN /cygdrive/c/bin

$ **kubectl get deployments**

NAME READY UP-TO-DATE AVAILABLE AGE

hello-minikube 1/1 1 1 92m

Helpful Documentation:

<https://medium.com/@mudrii/kubernetes-local-development-with-minikube-on-hyper-v-windows-10-75f52ad1ed42>

<https://medium.com/containers-101/local-kubernetes-for-windows-minikube-vs-docker-desktop-25a1c6d3b766>

<https://minikube.sigs.k8s.io/docs/handbook/kubectl/#:~:text=By%20default%2C%20kubectl%20gets%20configured,inside%20the%20minikube%20as%20well.&text=You%20can%20also%20alias%20kubectl,kubectl%20%2D%2D%22%20for%20easier%20usage.>

<https://kubernetes.io/docs/tasks/tools/install-kubectl/>

<https://kubernetes.io/docs/tutorials/hello-minikube/>

<https://docs.docker.com/machine/drivers/hyper-v/>

$ **minikube docker-env**

export DOCKER\_TLS\_VERIFY="1"

export DOCKER\_HOST="tcp://192.168.1.24:2376"

export DOCKER\_CERT\_PATH="C:\Users\monica\.minikube\certs"

export MINIKUBE\_ACTIVE\_DOCKERD="minikube"

PS C:\WINDOWS\system32> **choco version**

Chocolatey v0.10.15

PS C:\applications> **choco install docker-cli**

Chocolatey v0.10.15

Installing the following packages:

docker-cli

Progress: 100% - Completed download of C:\ProgramData\chocolatey\lib\docker-cli\tools\docker.exe (60.52 MB).

Download of docker.exe (60.52 MB) completed.

Hashes match.

**C:\ProgramData\chocolatey\lib\docker-cli\tools\docker.exe**

PS C:\applications> **docker**

Usage: docker [OPTIONS] COMMAND

PS C:\applications> **docker version**

Client:

Version: 19.03.12

API version: 1.40

Go version: go1.13.12

Git commit: 0ed913b8-

Built: 07/28/2020 16:36:03

OS/Arch: windows/amd64

Experimental: false

set windows environment variables:

PS C:\applications> **setx DOCKER\_CERT\_PATH 'C:\Users\monica\.minikube\certs'**

SUCCESS: Specified value was saved.

PS C:\applications> **setx DOCKER\_HOST "tcp://192.168.1.24:2376"**

SUCCESS: Specified value was saved.

PS C:\applications> minikube ip

192.168.1.24

echo 'monica ALL=(ALL) NOPASSWD:ALL' >> /etc/sudoers