



Java Techie

Follow

Dec 11, 2021 · 3 min read · Listen



Save



Kubernetes Tutorial | Install & Run Minikube in mac OS | K8s Cluster

Kubernetes Tutorial | Install & Run Minikube in mac OS | K8s Cluster | Java Te

Open in app



you can try out Kubernetes for your daily development work.

H

ow to install Minikube ?

Let's start with installation steps

Step 1:

To check if virtualization is supported on macOS, run the following command on your terminal.

```
sysctl -a | grep -E --color 'machdep.cpu.features|VMX'
```

If you see `vmx` in the output (should be colored), the VT-x feature is enabled in your machine.

```
javatechie@Basantas-iMac ~ % sysctl -a | grep -E --color 'machdep.cpu.features|VMX'
machdep.cpu.features: FPU VME DE PSE TSC MSR PAE MCE CX8 APIC SEP MTRR PGE MCA CMOV PAT PSE36 CLFSH DS ACPI MMX FXSR
SSE SSE2 SS HTT TM PBE SSE3 PCLMULQDQ DTES64 MON DSCPL VMX SMX EST TM2 SSSE3 FMA CX16 TPR PDCM SSE4.1 SSE4.2 x2APIC
MOVBE POPCNT AES PCID XSAVE OSXSAVE SEGLIM64 TSCTMR AVX1.0 RDRAND F16C
```

Step 2:

Make sure you have kubectl installed. You can install kubectl using below command

```
brew install kubectl
```

I hope everyone aware about what Homebrew does ? if not just recap once

Homebrew is a free and open-source software package manager in mac. that simplifies the installation of software on macOS, as well as Linux

```
javatechie@Basantas-iMac ~ % brew install kubectl
=> Downloading https://ghcr.io/v2/homebrew/core/kubernetes-cli/manifests/1.23.0
Already downloaded: /Users/javatechie/Library/Caches/Homebrew/downloads/edd60eba52e1a8fe2dc2236a0a89674be9ee292b11f2a41f97
7ed148e74de9d--kubernetes-cli-1.23.0.bottle_manifest.json
=> Downloading https://ghcr.io/v2/homebrew/core/kubernetes-cli/blobs/sha256:c53
Already downloaded: /Users/javatechie/Library/Caches/Homebrew/downloads/fce8c4569ae6e27851605a82d383501c376b208e3c47ff40b1
c607857500939--kubernetes-cli--1.23.0.monterey.bottle.tar.gz
=> Pouring kubernetes-cli--1.23.0.monterey.bottle.tar.gz
=> Caveats
zsh completions have been installed to:
  /usr/local/share/zsh/site-functions
=> Summary
🍺 /usr/local/Cellar/kubernetes-cli/1.23.0: 227 files, 56.8MB
=> Running `brew cleanup kubernetes-cli`...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
```

Verify kubectl version

```
kubectl version
```

```
javatechie@Basantas-iMac ~ % kubectl version
Client Version: version.Info{Major:"1", Minor:"23", GitVersion:"v1.23.0", GitCommit:"ab69524f795c42094a6630298ff53f3c3ebab7f4", GitTreeState:"clean", BuildDate:"2021-12-07T18:08:39Z", GoVersion:"go1.17.3", Compiler:"gc", Platform:"darwin/amd64"}
The connection to the server localhost:8080 was refused - did you specify the right host or port?
javatechie@Basantas-iMac ~ %
```

Step 4 :

Install a Hypervisor

If you do not already have a hypervisor installed, install one of these now:

- HyperKit
- VirtualBox
- VMware Fusion

We will install HyperKit to run our Minikube

```
brew install hyperkit
```

```
javatechie@Basantas-iMac ~ % brew install hyperkit
==> Downloading https://ghcr.io/v2/homebrew/core/hyperkit/manifests/0.20200908
Already downloaded: /Users/javatechie/Library/Caches/Homebrew/downloads/db4baa34ee2a779e1db4861dd34db8b2fc91786b2b7d32775b5e5de7f47af42a--hyperkit-0.20200908.bottle_manifest.json
==> Downloading https://ghcr.io/v2/homebrew/core/hyperkit/blobs/sha256:26a203b17733ff5166d8c31069e3ecd5af15c74448a51d8b682689cb07e911e8
Already downloaded: /Users/javatechie/Library/Caches/Homebrew/downloads/0ba7a770088a60dd6275860b0407ad95191aa6ddde0207b446e17a144b0e7a33--hyperkit--0.20200908.catalina.bottle.tar.gz
==> Pouring hyperkit--0.20200908.catalina.bottle.tar.gz
🍺 /usr/local/Cellar/hyperkit/0.20200908: 5 files, 4.3MB
==> Running 'brew cleanup hyperkit'...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see 'man brew').
javatechie@Basantas-iMac ~ %
```

Verify that you installed kubectl & HyperKit successfully in your mac using

```
brew list
```

```
javatechie@Basantas-iMac ~ % brew list
==> Formulae
gettext      git          hyperkit     kubernetes-cli  libev        pcre2
javatechie@Basantas-iMac ~ %
```

Step 5 :

Install Minikube

The easiest way to install Minikube on macOS is using Homebrew

```
brew install minikube
```

```
javatechie@Basantas-iMac ~ % brew install minikube

==> Downloading https://ghcr.io/v2/homebrew/core/minikube/manifests/1.24.0
Already downloaded: /Users/javatechie/Library/Caches/Homebrew/downloads/78d32b154b6ae966179798034dc5869da8f21722d375ee7aa5f8be7d202118ad--minikube-1.24.0.bottle_manifest.json
==> Downloading https://ghcr.io/v2/homebrew/core/minikube/blobs/sha256:0385fcb25a2009995119471d968d4c04925ffb29413ad07e87d9
==> Downloading from https://pkg-containers.githubusercontent.com/ghcr1/blobs/sha256:0385fcb25a2009995119471d968d4c04925ffb
##### 100.0%
==> Pouring minikube--1.24.0.monterey.bottle.tar.gz
==> Caveats
zsh completions have been installed to:
  /usr/local/share/zsh/site-functions
==> Summary
📦 /usr/local/Cellar/minikube/1.24.0: 9 files, 69.3MB
==> Running `brew cleanup minikube`...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
javatechie@Basantas-iMac ~ %
```

```
minikube version
```

```
javatechie@Basantas-iMac ~ % minikube version
minikube version: v1.24.0
commit: 76b94fb3c4e8ac5062daf70d60cf03ddcc0a741b
javatechie@Basantas-iMac ~ %
```

We successfully setup minikube in our mac now we are good to start minikube

Step 6 :

```
minikube start
```

```
javatechie@Basantas-iMac ~ % minikube start
minikube v1.24.0 on Darwin 12.0.1
Automatically selected the hyperkit driver
Starting control plane node minikube in cluster minikube
Creating hyperkit VM (CPUs=2, Memory=4000MB, Disk=20000MB) ...
Preparing Kubernetes v1.22.3 on Docker 20.10.8 ...
  ▪ Generating certificates and keys ...
  ▪ Booting up control plane ...
  ▪ Configuring RBAC rules ...
Verifying Kubernetes components...
  ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
Enabled addons: storage-provisioner, default-storageclass
Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
javatechie@Basantas-iMac ~ %
```

if you observe the statement above , minikube choose default driver as hyperkit , that's what we installed as Hypervisor

Once minikube started successfully , we can verify its status

```
minikube status
```

If your cluster is running, the output from `minikube status` should be similar to:

```
javatechie@Basantas-iMac ~ % minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
javatechie@Basantas-iMac ~ %
```

Step 7:

After you have confirmed whether Minikube is working with your chosen hypervisor, you can continue to use Minikube or you can stop your cluster. To stop your cluster, run:

```
minikube stop
```

```
javatechie@Basantas-iMac ~ % minikube stop
👋 Stopping node "minikube" ...
🔴 1 node stopped.
javatechie@Basantas-iMac ~ %
```

Step 8:

Delete minikube

```
minikube delete
```

Deletes a local Kubernetes cluster. This command deletes the VM, and removes all associated files.

```
javatechie@Basantas-iMac ~ % minikube delete
🔥 Deleting "minikube" in hyperkit ...
💀 Removed all traces of the "minikube" cluster.
javatechie@Basantas-iMac ~ %
```

Hope you will enjoy this blog , If you like this article then please do share with your colleagues ..

Thank you

Kubernetes

Minikube

Javatechie