Manual

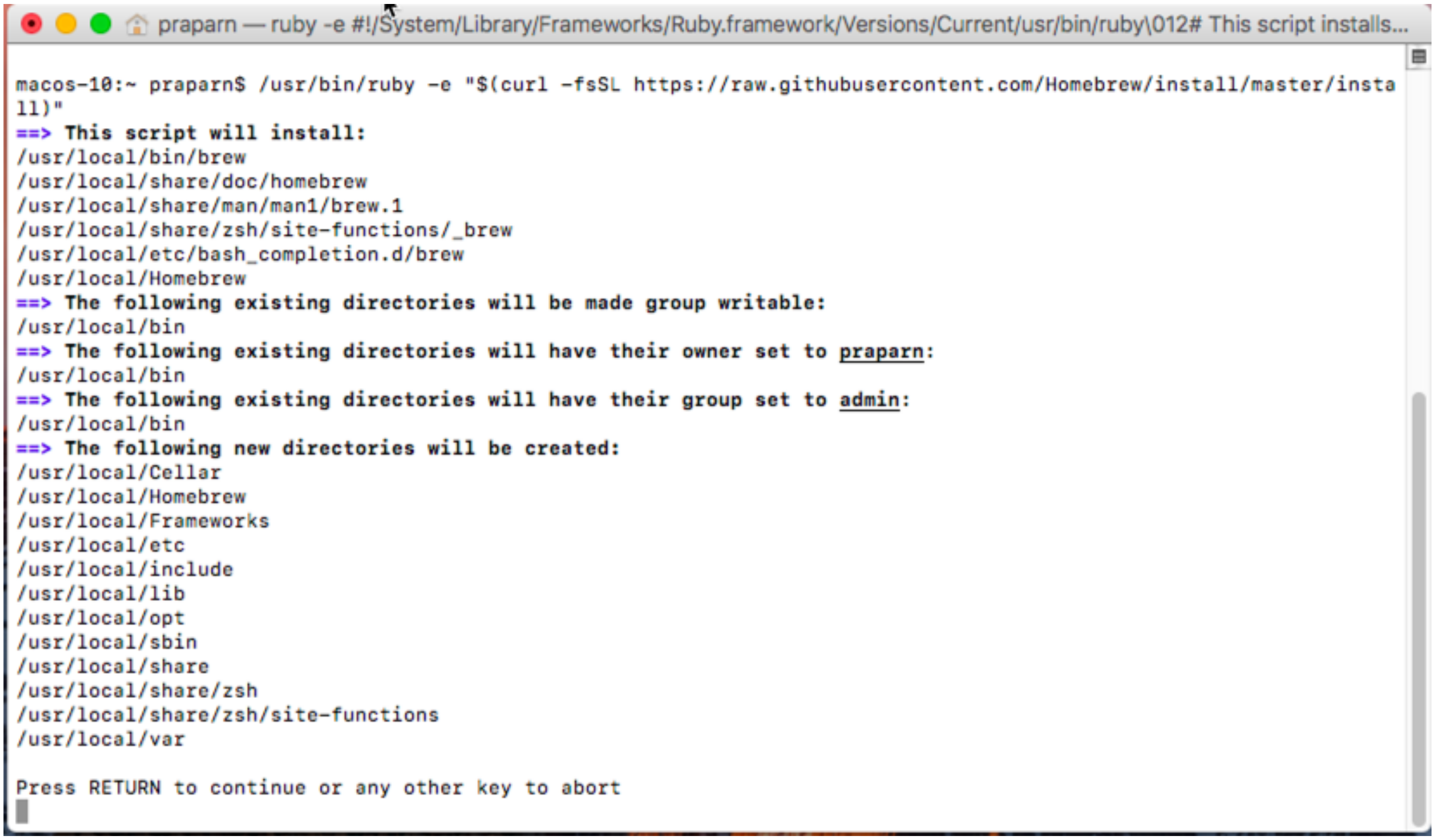
Install Minikube Software Set for OSX Platform

Prerequisite

Install brew for MACOS

1. Install brew module by command:l.

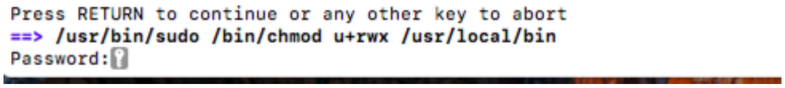
   /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)" and enter key stroke for continute



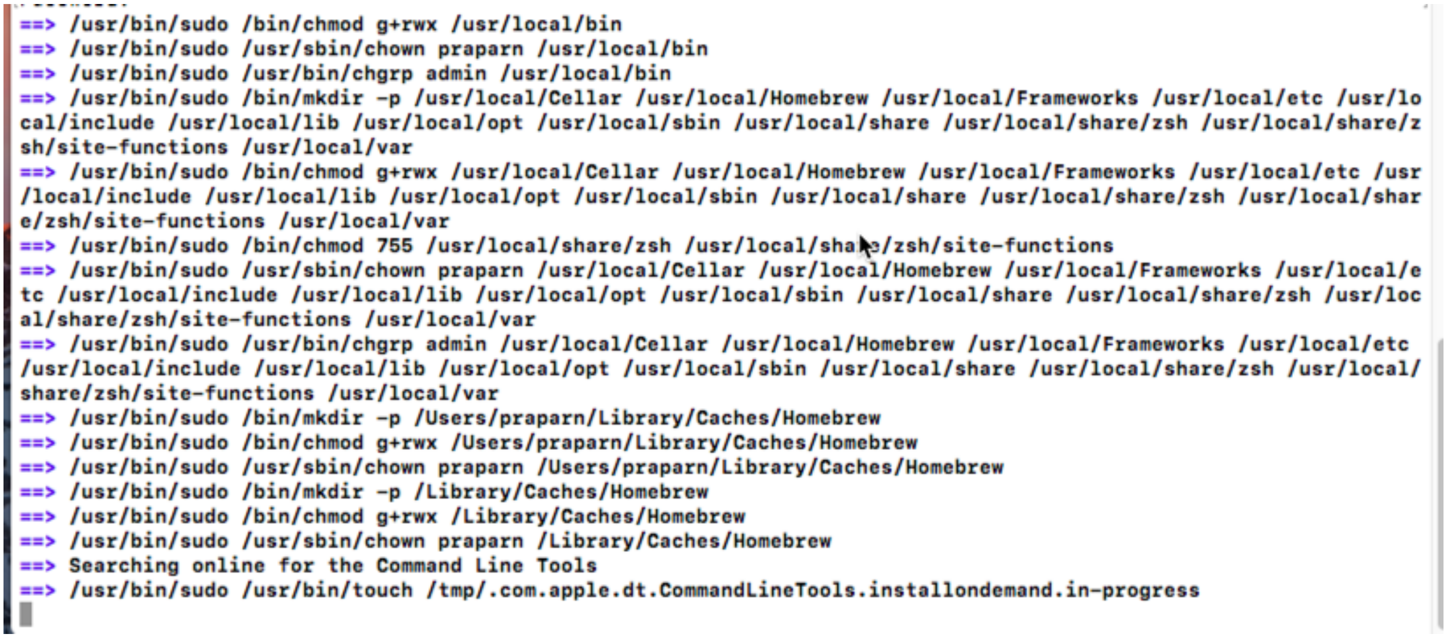
praparn — ruby -e #!/System/Library/Frameworks/Ruby.framework/Versions/Current/usr/bin/ruby\012# This script installs...

```
macos-10:~ praparn$ /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/insta
ll)"
==> This script will install:
/usr/local/bin/brew
/usr/local/share/doc/homebrew
/usr/local/share/man/man1/brew.1
/usr/local/share/zsh/site-functions/_brew
/usr/local/etc/bash_completion.d/brew
/usr/local/Homebrew
==> The following existing directories will be made group writable:
/usr/local/bin
==> The following existing directories will have their owner set to praparn:
/usr/local/bin
==> The following existing directories will have their group set to admin:
/usr/local/bin
==> The following new directories will be created:
/usr/local/Cellar
/usr/local/Homebrew
/usr/local/Frameworks
/usr/local/etc
/usr/local/include
/usr/local/lib
/usr/local/opt
/usr/local/sbin
/usr/local/share
/usr/local/share/zsh
/usr/local/share/zsh/site-functions
/usr/local/var

Press RETURN to continue or any other key to abort
```

2. Input password for grant privilege for install and wait until all install process was done.

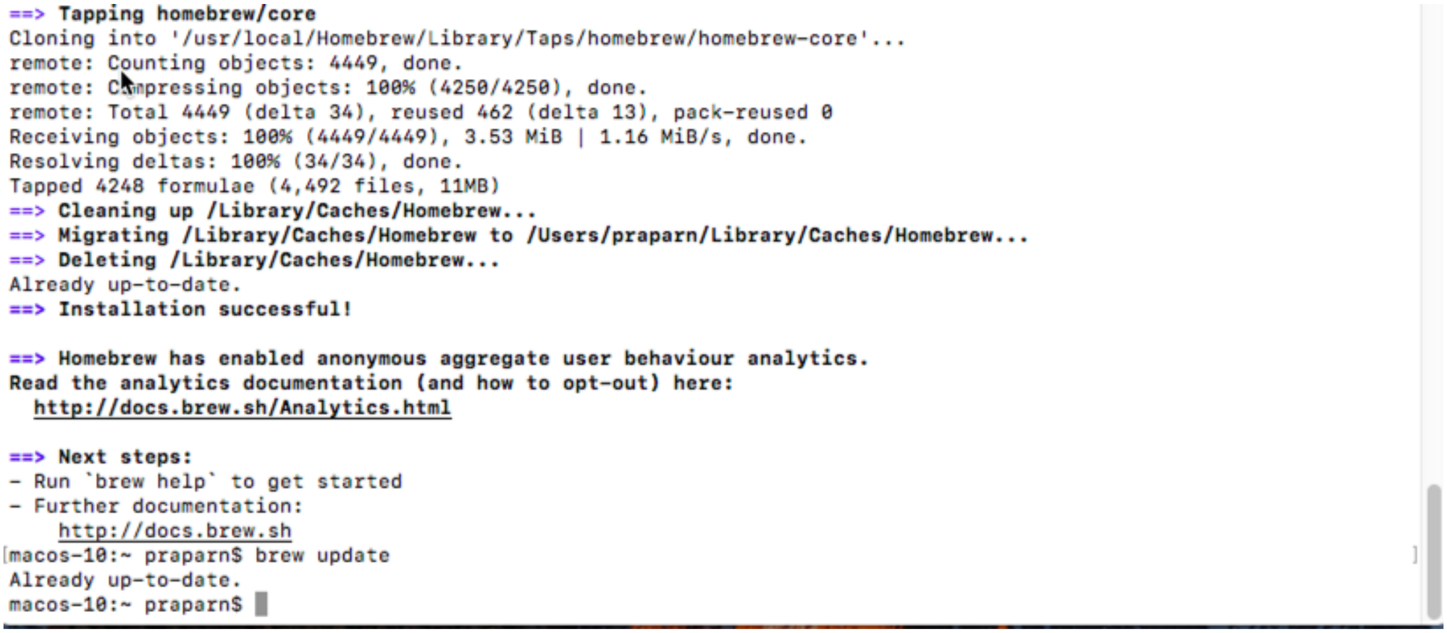


```
Press RETURN to continue or any other key to abort
==> /usr/bin/sudo /bin/chmod u+rwx /usr/local/bin
Password:
```



```
==> /usr/bin/sudo /bin/chmod g+rwx /usr/local/bin
==> /usr/bin/sudo /usr/sbin/chown praparn /usr/local/bin
==> /usr/bin/sudo /usr/bin/chgrp admin /usr/local/bin
==> /usr/bin/sudo /bin/mkdir -p /usr/local/Cellar /usr/local/Homebrew /usr/local/Frameworks /usr/local/etc /usr/lo
cal/include /usr/local/lib /usr/local/opt /usr/local/sbin /usr/local/share /usr/local/share/zsh /usr/local/share/z
sh/site-functions /usr/local/var
==> /usr/bin/sudo /bin/chmod g+rwx /usr/local/Cellar /usr/local/Homebrew /usr/local/Frameworks /usr/local/etc /usr
/local/include /usr/local/lib /usr/local/opt /usr/local/sbin /usr/local/share /usr/local/share/zsh /usr/local/shar
e/zsh/site-functions /usr/local/var
==> /usr/bin/sudo /bin/chmod 755 /usr/local/share/zsh /usr/local/share/zsh/site-functions
==> /usr/bin/sudo /usr/sbin/chown praparn /usr/local/Cellar /usr/local/Homebrew /usr/local/Frameworks /usr/local/e
tc /usr/local/include /usr/local/lib /usr/local/opt /usr/local/sbin /usr/local/share /usr/local/share/zsh /usr/loc
al/share/zsh/site-functions /usr/local/var
==> /usr/bin/sudo /usr/bin/chgrp admin /usr/local/Cellar /usr/local/Homebrew /usr/local/Frameworks /usr/local/etc
/usr/local/include /usr/local/lib /usr/local/opt /usr/local/sbin /usr/local/share /usr/local/share/zsh /usr/local/
share/zsh/site-functions /usr/local/var
==> /usr/bin/sudo /bin/mkdir -p /Users/praparn/Library/Caches/Homebrew
==> /usr/bin/sudo /bin/chmod g+rwx /Users/praparn/Library/Caches/Homebrew
==> /usr/bin/sudo /usr/sbin/chown praparn /Users/praparn/Library/Caches/Homebrew
==> /usr/bin/sudo /bin/mkdir -p /Library/Caches/Homebrew
==> /usr/bin/sudo /bin/chmod g+rwx /Library/Caches/Homebrew
==> /usr/bin/sudo /usr/sbin/chown praparn /Library/Caches/Homebrew
==> Searching online for the Command Line Tools
==> /usr/bin/sudo /usr/bin/touch /tmp/.com.apple.dt.CommandLineTools.installondemand.in-progress
```

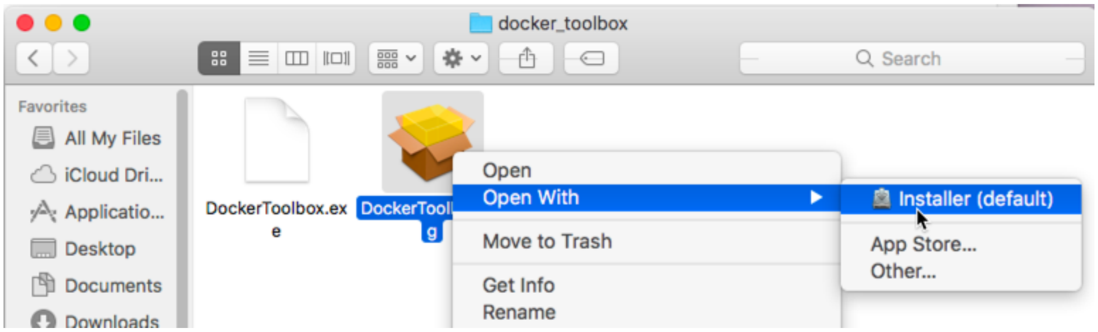
3. After install brew have finished install. Check brew by command: brew update



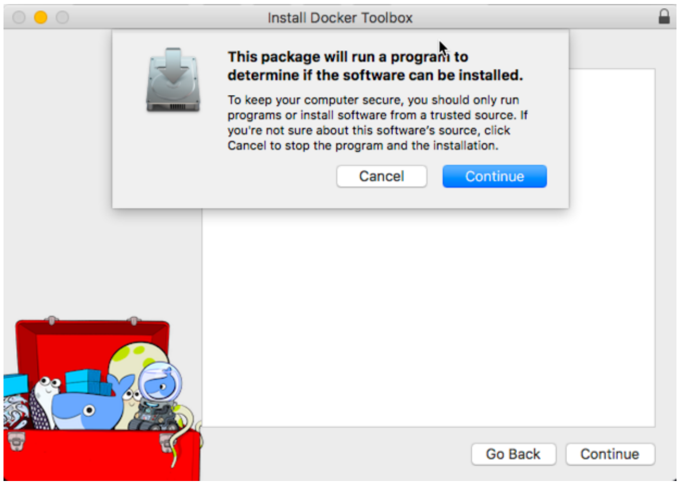
```
==> Tapping homebrew/core
Cloning into '/usr/local/Homebrew/Library/Taps/homebrew/homebrew-core'...
remote: Counting objects: 4449, done.
remote: Compressing objects: 100% (4250/4250), done.
remote: Total 4449 (delta 34), reused 462 (delta 13), pack-reused 0
Receiving objects: 100% (4449/4449), 3.53 MiB | 1.16 MiB/s, done.
Resolving deltas: 100% (34/34), done.
Tapped 4248 formulae (4,492 files, 11MB)
==> Cleaning up /Library/Caches/Homebrew...
==> Migrating /Library/Caches/Homebrew to /Users/praparn/Library/Caches/Homebrew...
==> Deleting /Library/Caches/Homebrew...
Already up-to-date.
==> Installation successful!

==> Homebrew has enabled anonymous aggregate user behaviour analytics.
Read the analytics documentation (and how to opt-out) here:
  http://docs.brew.sh/Analytics.html

==> Next steps:
- Run `brew help` to get started
- Further documentation:
    http://docs.brew.sh
macos-10:~ praparn$ brew update
Already up-to-date.
macos-10:~ praparn$
```

Install Docker Toolbox

1. Right Click on "DockerToolbox.dmg" and select installer

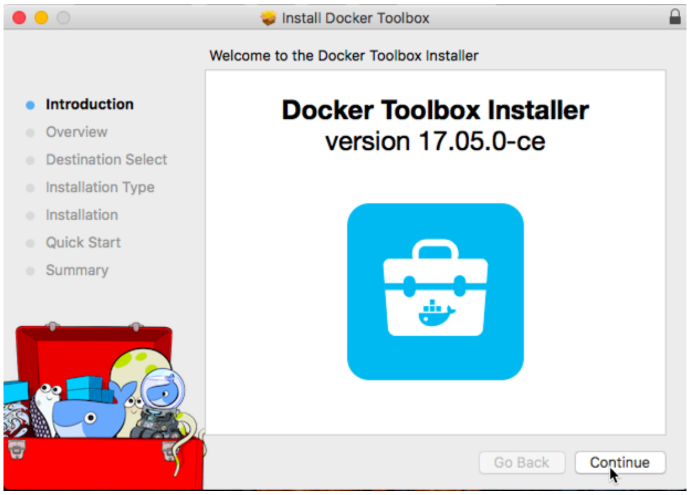


2. Following screen for setup



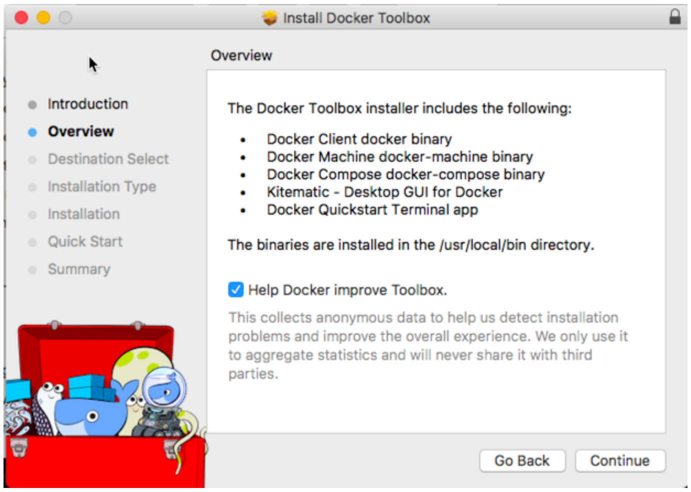
This package will run a program to determine if the software can be installed.

To keep your computer secure, you should only run programs or install software from a trusted source. If you're not sure about this software's source, click Cancel to stop the program and the installation.



Welcome to the Docker Toolbox Installer

Docker Toolbox Installer
version 17.05.0-ce



Overview

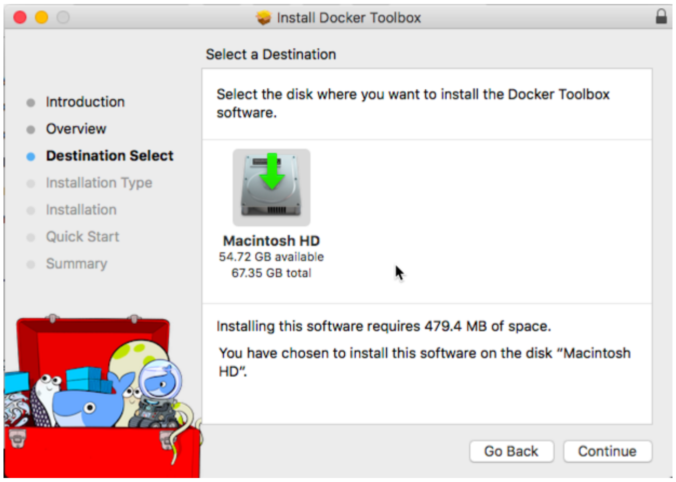
The Docker Toolbox installer includes the following:

- Docker Client docker binary
- Docker Machine docker-machine binary
- Docker Compose docker-compose binary
- Kitematic - Desktop GUI for Docker
- Docker Quickstart Terminal app

The binaries are installed in the /usr/local/bin directory.

☑ Help Docker improve Toolbox.

This collects anonymous data to help us detect installation problems and improve the overall experience. We only use it to aggregate statistics and will never share it with third parties.

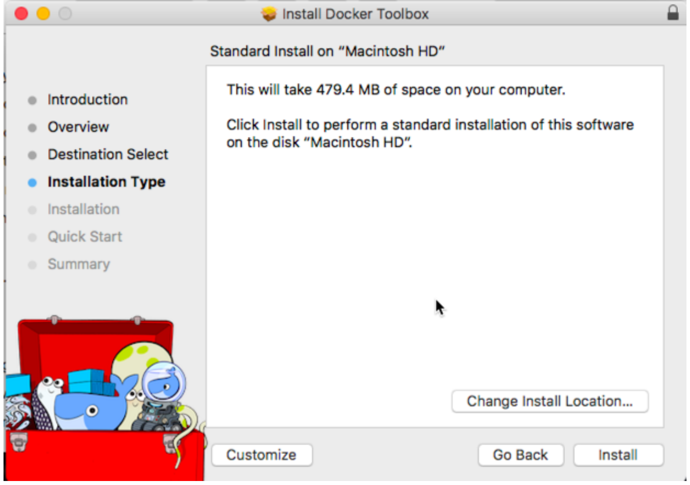


Select a Destination

Select the disk where you want to install the Docker Toolbox software.

Installing this software requires 479.4 MB of space.

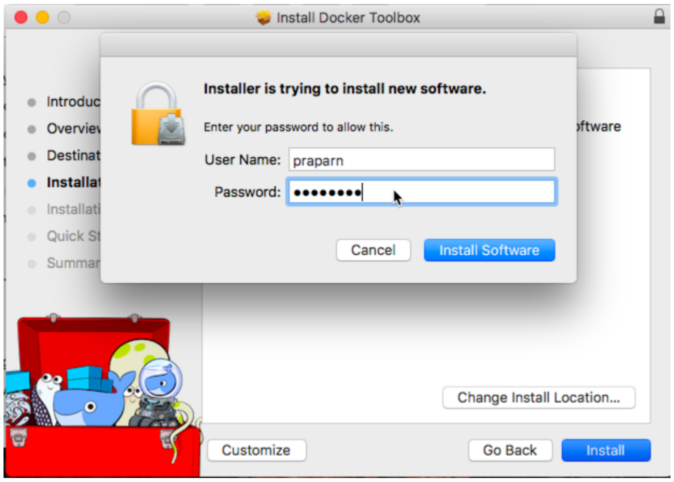
You have chosen to install this software on the disk "Macintosh HD".



Standard Install on "Macintosh HD"

This will take 479.4 MB of space on your computer.

Click Install to perform a standard installation of this software on the disk "Macintosh HD".

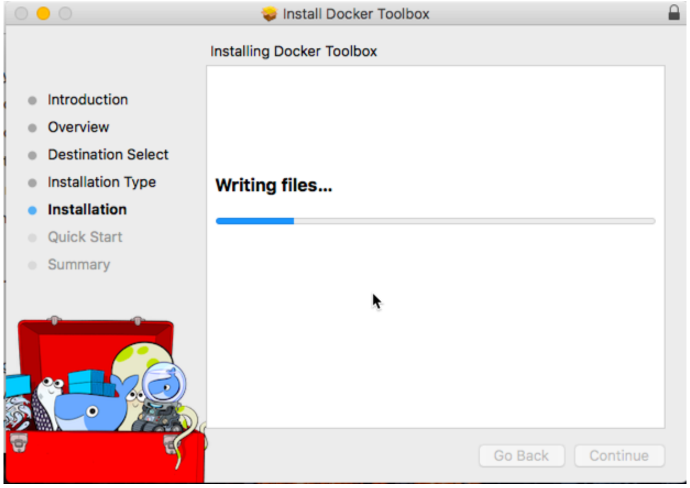


Installer is trying to install new software.

Enter your password to allow this.

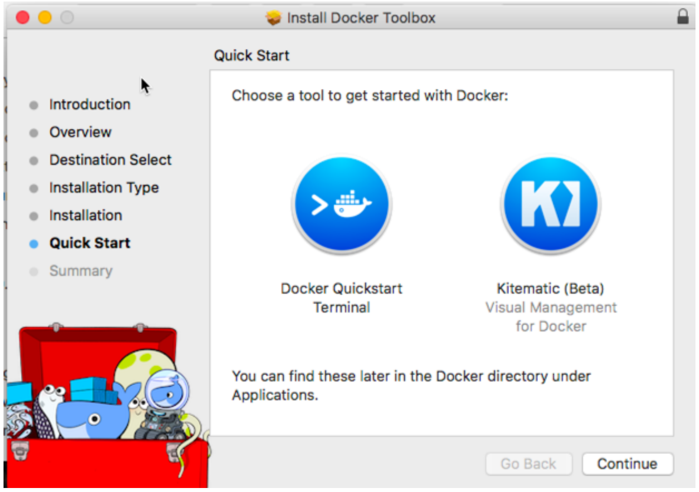
User Name: praparn

Password: ••••••••



Installing Docker Toolbox

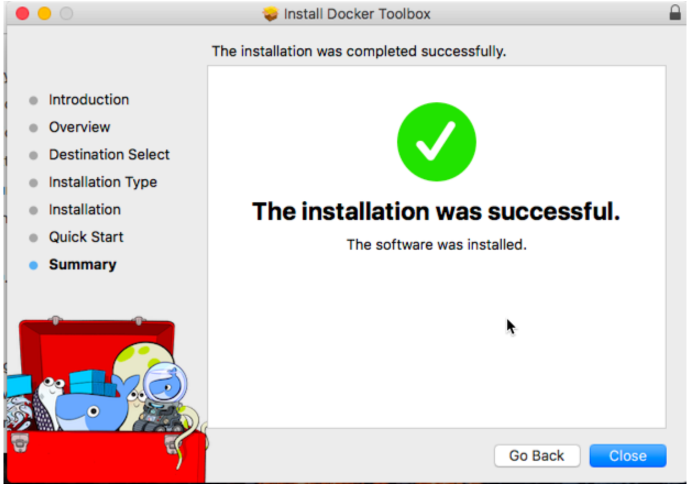
Writing files...



Quick Start

Choose a tool to get started with Docker:

You can find these later in the Docker directory under Applications.

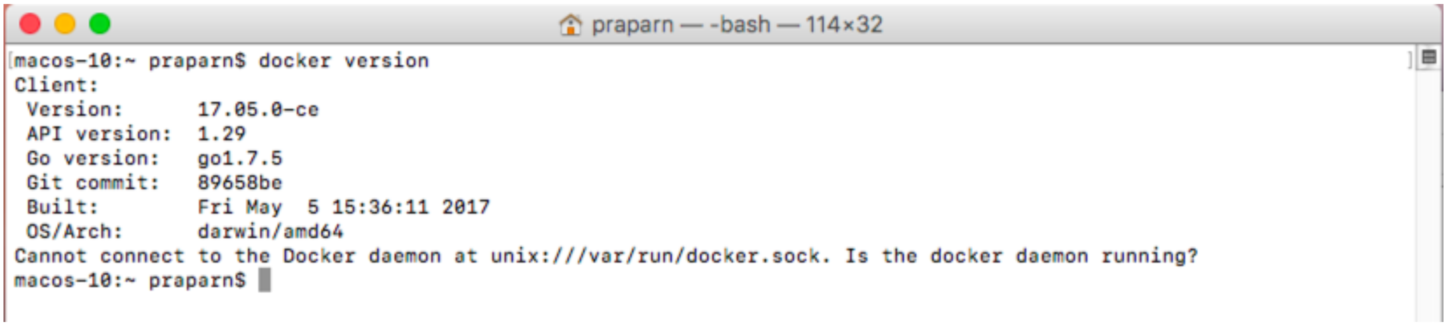


The installation was completed successfully.

The installation was successful.

The software was installed.

3. Check version of Docker Tools by command: docker version



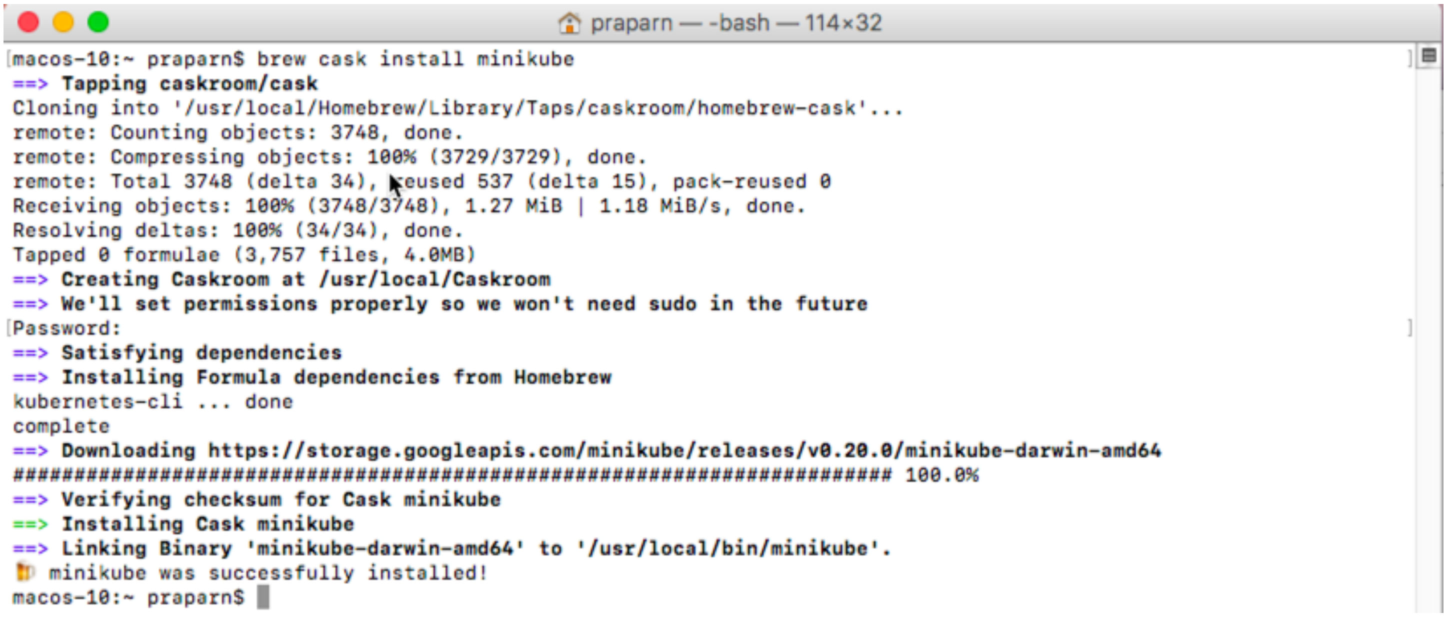
```
macos-10:~ praparn$ docker version
Client:
 Version:      17.05.0-ce
 API version:  1.29
 Go version:   go1.7.5
 Git commit:   89658be
 Built:        Fri May  5 15:36:11 2017
 OS/Arch:      darwin/amd64
Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?
macos-10:~ praparn$ 
```

Install minikube / Initial minikube machine

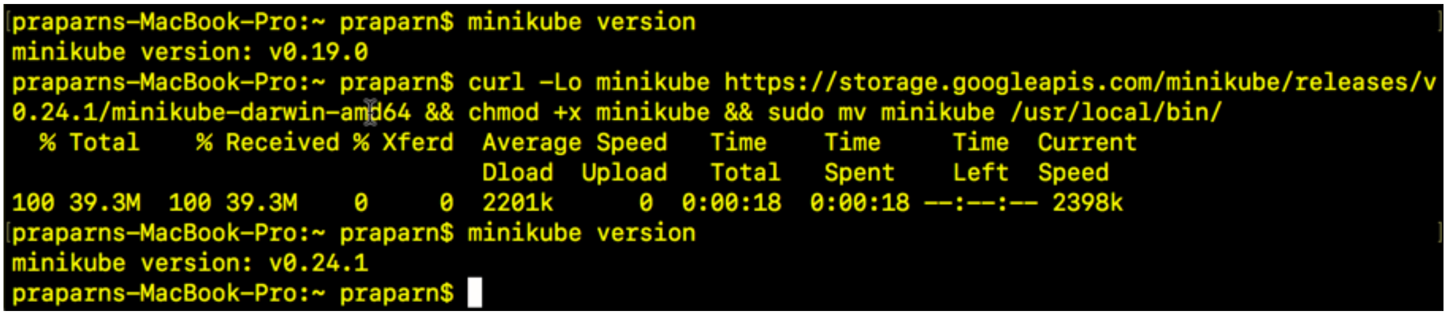
1. Install minikube by command: brew cask install minikube

   Or

   “curl -Lo minikube https://storage.googleapis.com/minikube/releases/v0.25.0/minikube-darwin-amd64 && chmod +x minikube && sudo mv minikube /usr/local/bin/”

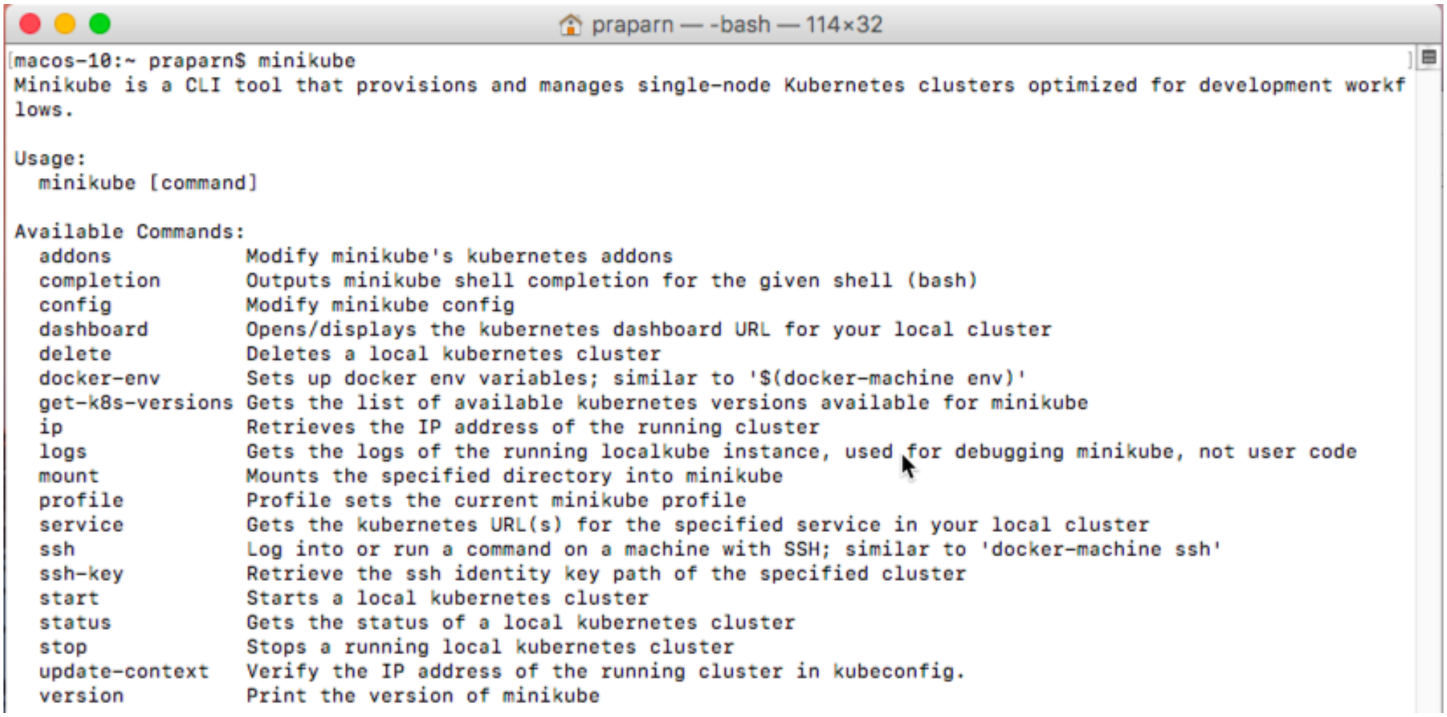


```
[macos-10:~ praparn$ brew cask install minikube
==> Tapping caskroom/cask
Cloning into '/usr/local/Homebrew/Library/Taps/caskroom/homebrew-cask'...
remote: Counting objects: 3748, done.
remote: Compressing objects: 100% (3729/3729), done.
remote: Total 3748 (delta 34), reused 537 (delta 15), pack-reused 0
Receiving objects: 100% (3748/3748), 1.27 MiB | 1.18 MiB/s, done.
Resolving deltas: 100% (34/34), done.
Tapped 0 formulae (3,757 files, 4.0MB)
==> Creating Caskroom at /usr/local/Caskroom
==> We'll set permissions properly so we won't need sudo in the future
[Password:
==> Satisfying dependencies
==> Installing Formula dependencies from Homebrew
kubernetes-cli ... done
complete
==> Downloading https://storage.googleapis.com/minikube/releases/v0.20.0/minikube-darwin-amd64
########################################################################## 100.0%
==> Verifying checksum for Cask minikube
==> Installing Cask minikube
==> Linking Binary 'minikube-darwin-amd64' to '/usr/local/bin/minikube'.
🍺 minikube was successfully installed!
macos-10:~ praparn$
```



```
[praparns-MacBook-Pro:~ praparn$ minikube version
minikube version: v0.19.0
praparns-MacBook-Pro:~ praparn$ curl -Lo minikube https://storage.googleapis.com/minikube/releases/v
0.24.1/minikube-darwin-amd64 && chmod +x minikube && sudo mv minikube /usr/local/bin/
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 39.3M  100 39.3M    0     0  2201k      0  0:00:18  0:00:18 --:--:-- 2398k
[praparns-MacBook-Pro:~ praparn$ minikube version
minikube version: v0.24.1
praparns-MacBook-Pro:~ praparn$
```

2. Check minikube interactive command

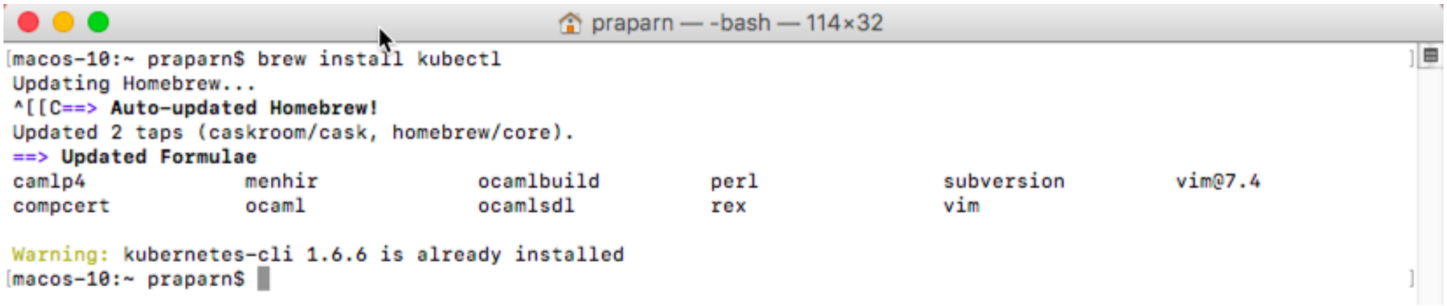


```
macos-10:~ praparn$ minikube
Minikube is a CLI tool that provisions and manages single-node Kubernetes clusters optimized for development workf
lows.

Usage:
  minikube [command]

Available Commands:
  addons          Modify minikube's kubernetes addons
  completion      Outputs minikube shell completion for the given shell (bash)
  config          Modify minikube config
  dashboard       Opens/displays the kubernetes dashboard URL for your local cluster
  delete          Deletes a local kubernetes cluster
  docker-env      Sets up docker env variables; similar to '$(docker-machine env)'
  get-k8s-versions Gets the list of available kubernetes versions available for minikube
  ip              Retrieves the IP address of the running cluster
  logs            Gets the logs of the running localkube instance, used for debugging minikube, not user code
  mount           Mounts the specified directory into minikube
  profile         Profile sets the current minikube profile
  service         Gets the kubernetes URL(s) for the specified service in your local cluster
  ssh             Log into or run a command on a machine with SSH; similar to 'docker-machine ssh'
  ssh-key         Retrieve the ssh identity key path of the specified cluster
  start           Starts a local kubernetes cluster
  status          Gets the status of a local kubernetes cluster
  stop            Stops a running local kubernetes cluster
  update-context  Verify the IP address of the running cluster in kubeconfig.
  version         Print the version of minikube
```

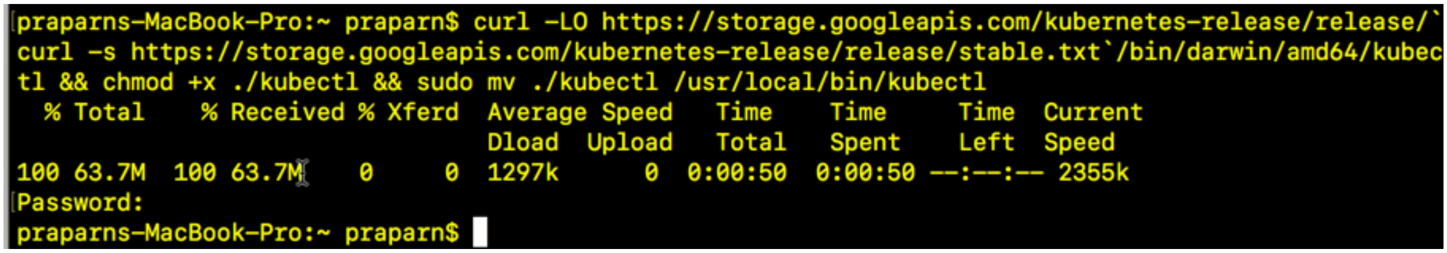
3. Install kubectl by command: brew install kubectl

   Or

"curl -LO https://storage.googleapis.com/kubernetes-release/release/`curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt`/bin/darwin/amd64/kubectl && chmod +x ./kubectl && sudo mv ./kubectl /usr/local/bin/kubectl"

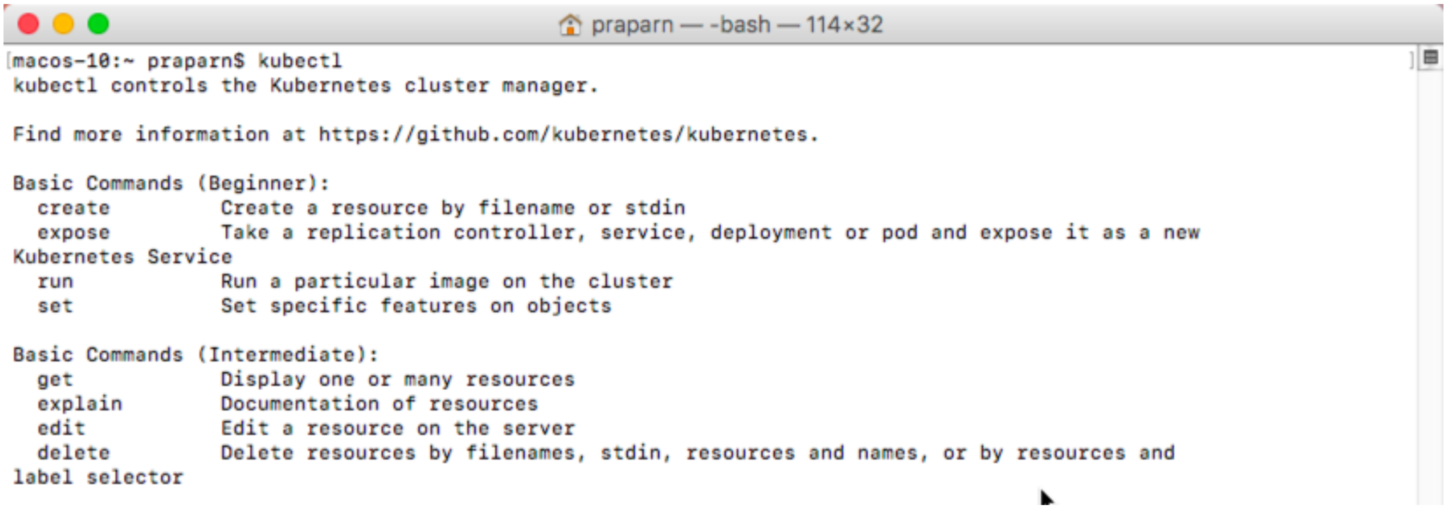


```
[macos-10:~ praparn$ brew install kubectl
 Updating Homebrew...
 ^[[C==> Auto-updated Homebrew!
 Updated 2 taps (caskroom/cask, homebrew/core).
 ==> Updated Formulae
 camlp4            menhir            ocamlbuild        perl              subversion        vim@7.4
 compcert          ocaml             ocamlsdl          rex               vim

 Warning: kubernetes-cli 1.6.6 is already installed
[macos-10:~ praparn$ 
```



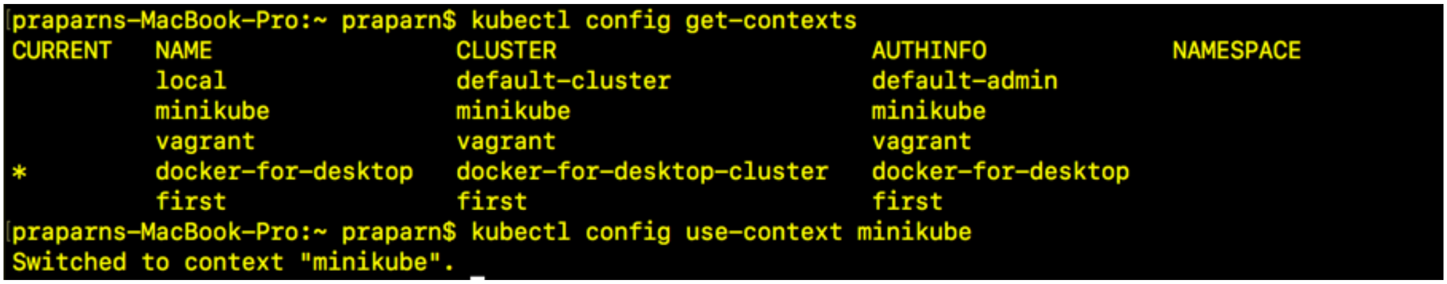
```
[praparns-MacBook-Pro:~ praparn$ curl -LO https://storage.googleapis.com/kubernetes-release/release/`
 curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt`/bin/darwin/amd64/kubec
 tl && chmod +x ./kubectl && sudo mv ./kubectl /usr/local/bin/kubectl
   % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                  Dload  Upload   Total   Spent    Left  Speed
 100 63.7M  100 63.7M    0     0  1297k      0  0:00:50  0:00:50 --:--:-- 2355k
[Password:
 praparns-MacBook-Pro:~ praparn$ 
```

4. Check kubectl interactive command



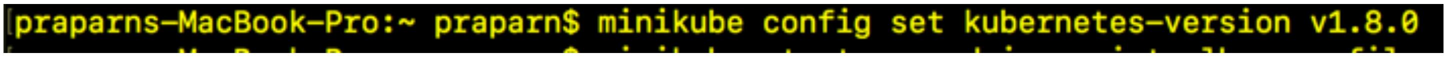
```
[macos-10:~ praparn$ kubectl
 kubectl controls the Kubernetes cluster manager.

 Find more information at https://github.com/kubernetes/kubernetes.

 Basic Commands (Beginner):
   create         Create a resource by filename or stdin
   expose         Take a replication controller, service, deployment or pod and expose it as a new
 Kubernetes Service
   run            Run a particular image on the cluster
   set            Set specific features on objects

 Basic Commands (Intermediate):
   get            Display one or many resources
   explain        Documentation of resources
   edit           Edit a resource on the server
   delete         Delete resources by filenames, stdin, resources and names, or by resources and
 label selector
```

5. **Special** for pilot "docker for mac with K8S" you need to switch context to minikube before start next



```
[praparns-MacBook-Pro:~ praparn$ kubectl config get-contexts
CURRENT   NAME                 CLUSTER                      AUTHINFO             NAMESPACE
          local                default-cluster              default-admin
          minikube             minikube                     minikube
          vagrant              vagrant                      vagrant
*         docker-for-desktop   docker-for-desktop-cluster   docker-for-desktop
          first                first                        first
[praparns-MacBook-Pro:~ praparn$ kubectl config use-context minikube
Switched to context "minikube".
```

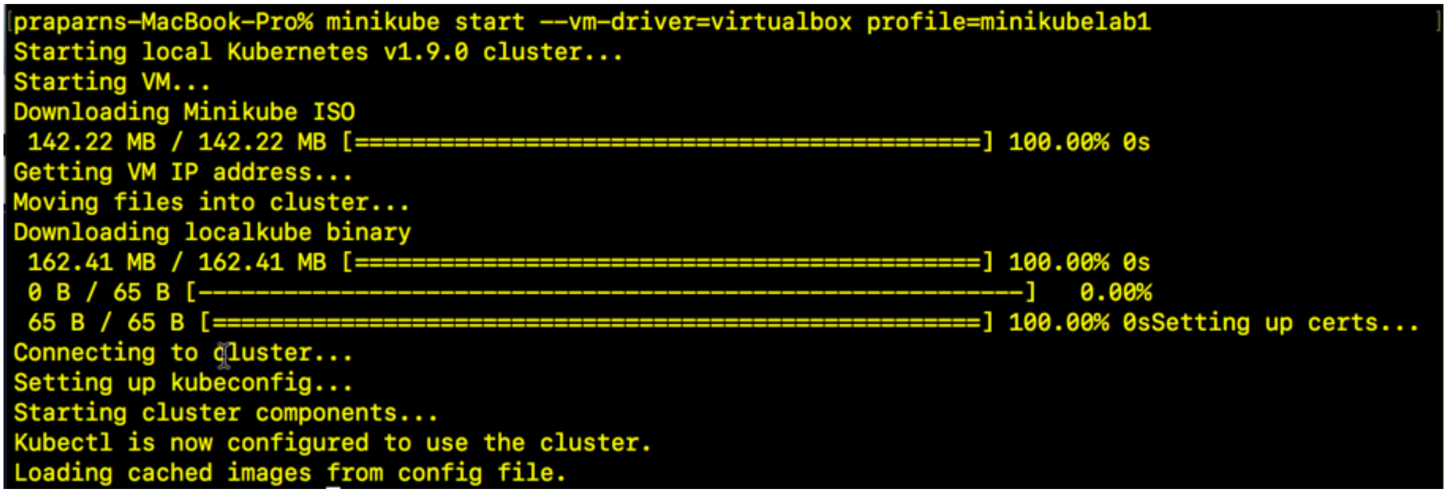
6. Configure minikube for use kubernetes version 1.7.0 by command: minikube config set kubernetes-version v1.9.0



```
[praparns-MacBook-Pro:~ praparn$ minikube config set kubernetes-version v1.8.0
```

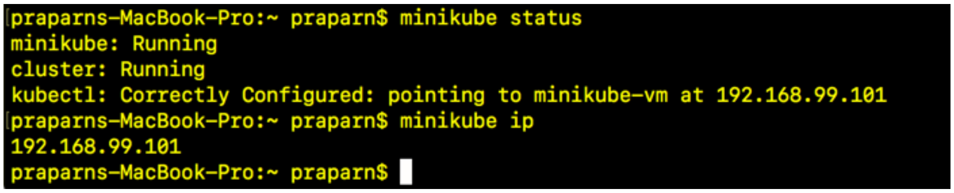
7. Create minikube machine by command:

“minikube start --vm-driver=virtualbox profile=minikubelab1”



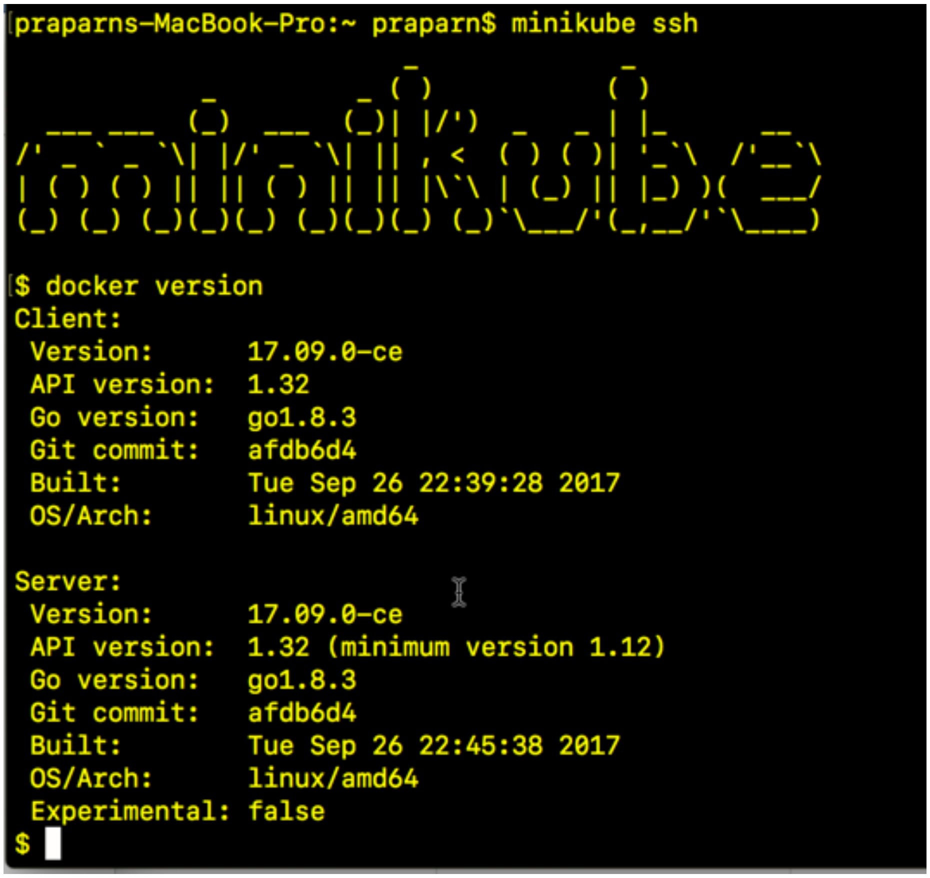
```
[praparns-MacBook-Pro% minikube start --vm-driver=virtualbox profile=minikubelab1
Starting local Kubernetes v1.9.0 cluster...
Starting VM...
Downloading Minikube ISO
 142.22 MB / 142.22 MB [============================================] 100.00% 0s
Getting VM IP address...
Moving files into cluster...
Downloading localkube binary
 162.41 MB / 162.41 MB [============================================] 100.00% 0s
 0 B / 65 B [----------------------------------------------------------]   0.00%
 65 B / 65 B [======================================================] 100.00% 0sSetting up certs...
Connecting to cluster...
Setting up kubeconfig...
Starting cluster components...
Kubectl is now configured to use the cluster.
Loading cached images from config file.
```

8. Check status of minikube’s machine by command: “minikube status”,”minikube ip”



```
[praparns-MacBook-Pro:~ praparn$ minikube status
minikube: Running
cluster: Running
kubectl: Correctly Configured: pointing to minikube-vm at 192.168.99.101
[praparns-MacBook-Pro:~ praparn$ minikube ip
192.168.99.101
praparns-MacBook-Pro:~ praparn$ 
```

9. Test ssh to minikube's machine by command (user: docker, password: tcuser): minikube ssh

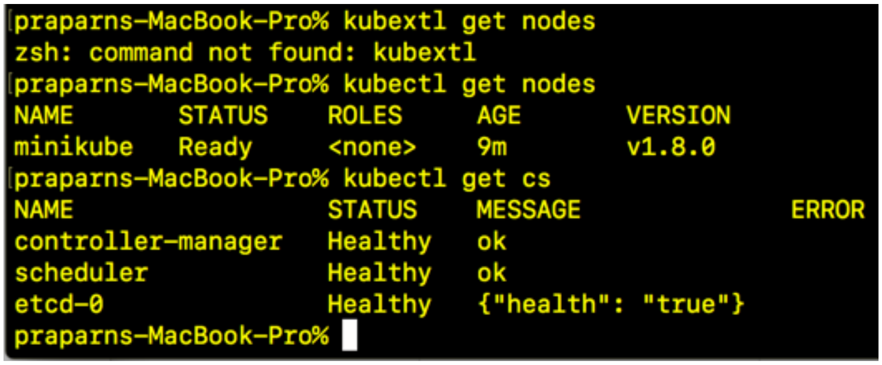


```
[praparns-MacBook-Pro:~ praparn$ minikube ssh
                         _             _
            _         _ ( )           ( )
  ___ ___  (_)  ___  (_)| |/')  _   _ | |_      __
/' _ ` _ `\| |/' _ `\| || , <  ( ) ( )| '_`\  /'__`\
| ( ) ( ) || || ( ) || || |\`\ | (_) || |_) )(  ___/
(_) (_) (_)(_)(_) (_)(_)(_) (_)`\___/'(_,__/'`\____)

[$ docker version
Client:
 Version:       17.09.0-ce
 API version:   1.32
 Go version:    go1.8.3
 Git commit:    afdb6d4
 Built:         Tue Sep 26 22:39:28 2017
 OS/Arch:       linux/amd64

Server:
 Version:       17.09.0-ce
 API version:   1.32 (minimum version 1.12)
 Go version:    go1.8.3
 Git commit:    afdb6d4
 Built:         Tue Sep 26 22:45:38 2017
 OS/Arch:       linux/amd64
 Experimental:  false
$ 
```

10. Check health of kubenetes cluster by command

   a. kubectl get nodes ➔ check node status

   b. kubectl get cs ➔ check cluster status



```
[praparns-MacBook-Pro% kubextl get nodes
zsh: command not found: kubextl
[praparns-MacBook-Pro% kubectl get nodes
NAME       STATUS   ROLES    AGE      VERSION
minikube   Ready    <none>   9m       v1.8.0
[praparns-MacBook-Pro% kubectl get cs
NAME                 STATUS    MESSAGE              ERROR
controller-manager   Healthy   ok
scheduler            Healthy   ok
etcd-0               Healthy   {"health": "true"}
praparns-MacBook-Pro% 
```

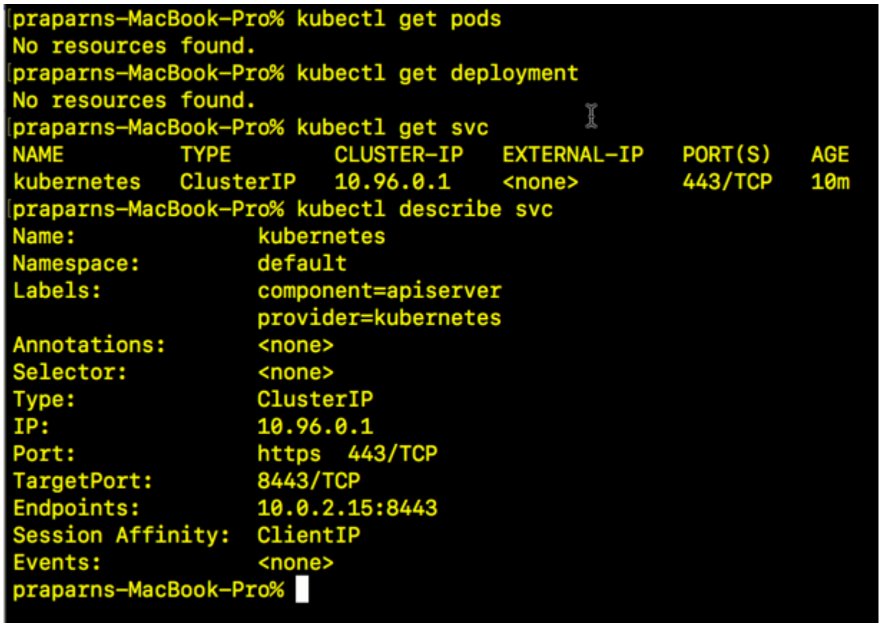
11. Check status of kubenetest's elements by command

   a. kubectl get pods ➔ check pods element

   b. kubectl get deployment ➔ check deployment element

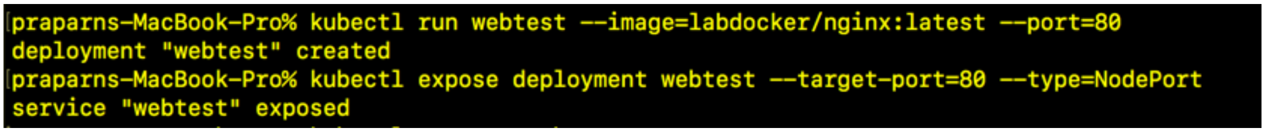
   c. kubectl get svc ➔ check service deploy on kubenetes

   d. kubectl describe svc ➔ check service description on kubenetes



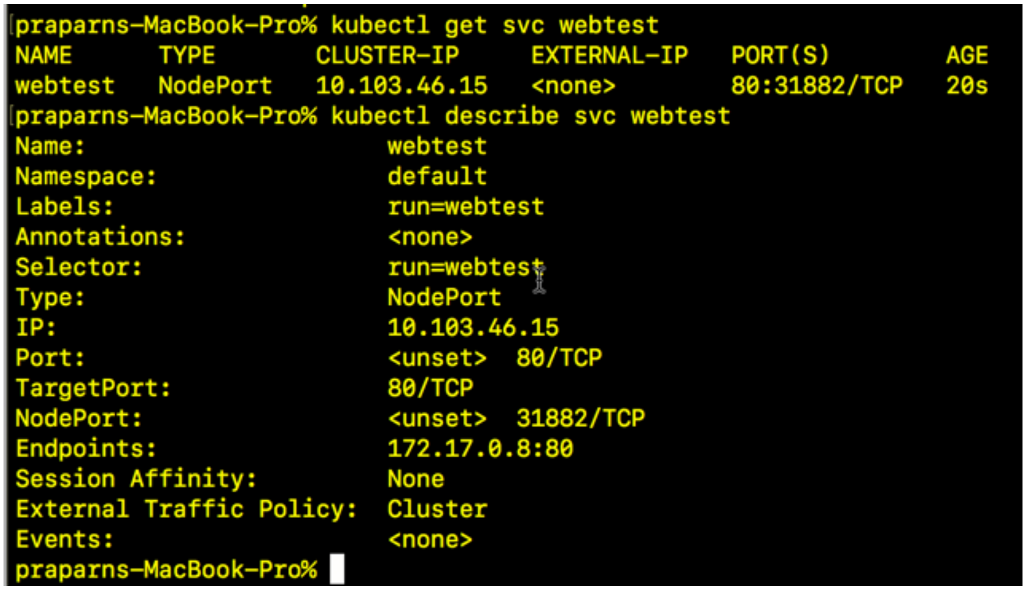
```
[praparns-MacBook-Pro% kubectl get pods
No resources found.
[praparns-MacBook-Pro% kubectl get deployment
No resources found.
[praparns-MacBook-Pro% kubectl get svc
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   10m
[praparns-MacBook-Pro% kubectl describe svc
Name:              kubernetes
Namespace:         default
Labels:            component=apiserver
                   provider=kubernetes
Annotations:       <none>
Selector:          <none>
Type:              ClusterIP
IP:                10.96.0.1
Port:              https  443/TCP
TargetPort:        8443/TCP
Endpoints:         10.0.2.15:8443
Session Affinity:  ClientIP
Events:            <none>
praparns-MacBook-Pro%
```

12. Test deployment “nginx” web server by command:
    a. kubectl run webtest --image=labdocker/nginx:latest --port=80 → deployment nginx (image: labdocker/nginx:latest) with port 80 service
    b. kubectl expose deployment webtest --target-port=80 --type=NodePort → expose pods with service 80 (http)



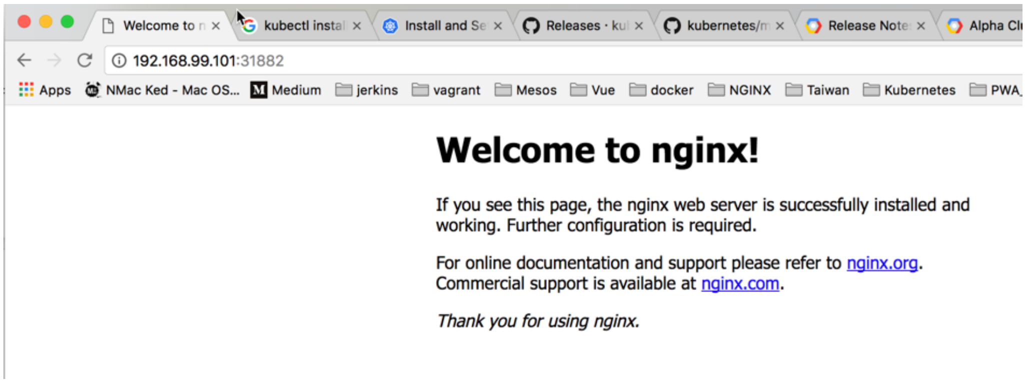
```
[praparns-MacBook-Pro% kubectl run webtest --image=labdocker/nginx:latest --port=80
deployment "webtest" created
[praparns-MacBook-Pro% kubectl expose deployment webtest --target-port=80 --type=NodePort
service "webtest" exposed
```

13. Check port mapping for service with host by command:
    a. kubectl get svc webtest → check mapping service
    b. kubectl describe svc webtest → check description of service



```
[praparns-MacBook-Pro% kubectl get svc webtest
NAME      TYPE       CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
webtest   NodePort   10.103.46.15   <none>        80:31882/TCP   20s
[praparns-MacBook-Pro% kubectl describe svc webtest
Name:                     webtest
Namespace:                default
Labels:                   run=webtest
Annotations:              <none>
Selector:                 run=webtest
Type:                     NodePort
IP:                       10.103.46.15
Port:                     <unset>  80/TCP
TargetPort:               80/TCP
NodePort:                 <unset>  31882/TCP
Endpoints:                172.17.0.8:80
Session Affinity:         None
External Traffic Policy:  Cluster
Events:                   <none>
praparns-MacBook-Pro%
```

14. Test open webpage with port describe on command above (This example: [http://192.168.99.101:31882)](http://192.168.99.101:31882)



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

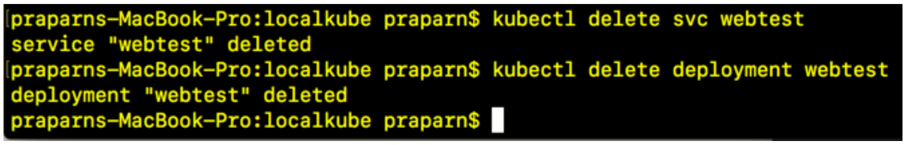
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

15. Stop deployment by command and recheck again

    a. kubectl delete svc webtest

    b. kubectl delete deployment webtest



```
praparns-MacBook-Pro:localkube praparn$ kubectl delete svc webtest
service "webtest" deleted
praparns-MacBook-Pro:localkube praparn$ kubectl delete deployment webtest
deployment "webtest" deleted
praparns-MacBook-Pro:localkube praparn$ 
```