Manual

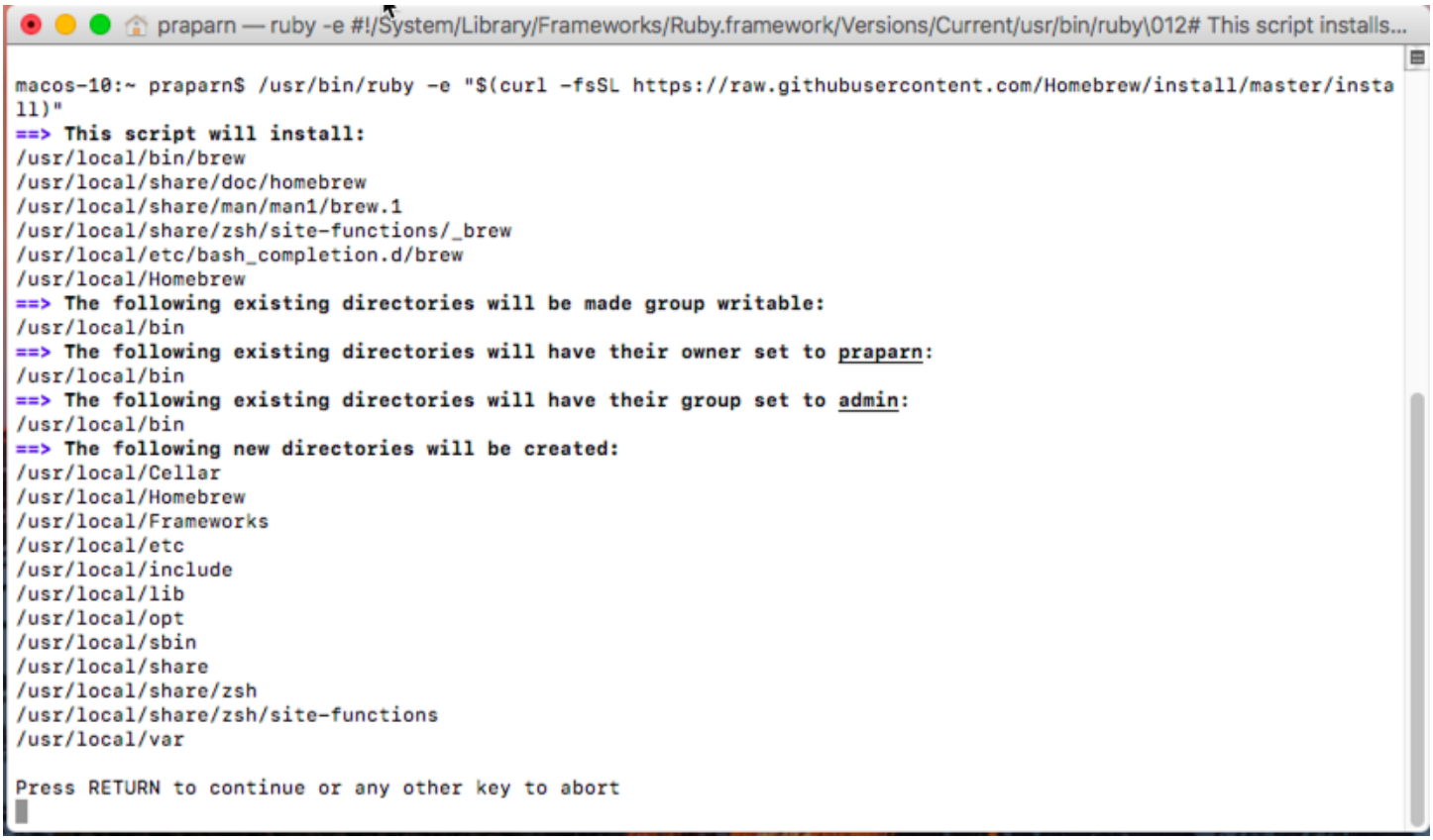
Install Minikube Software Set for OSX Platform

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



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:
```

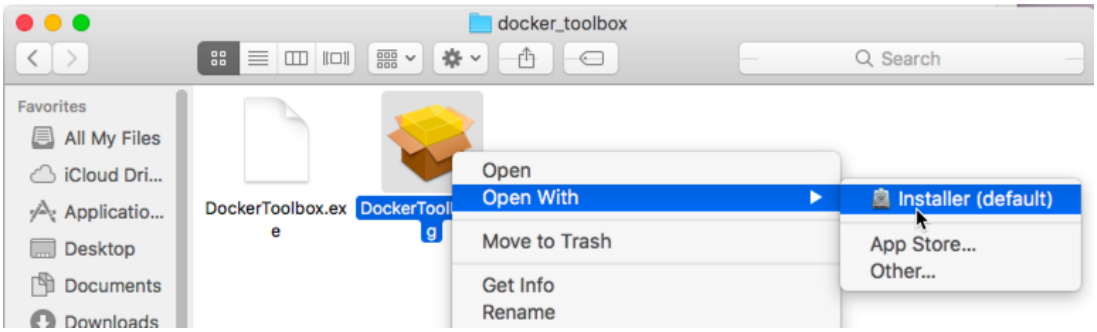
```
==> /usr/bin/sudo /bin/chmod g+rwx /usr/local/bin
==> /usr/bin/sudo /usr/sbin/chown praparn /usr/local/bin
==> /usr/bin/sudo /usr/bin/chgrp admin /usr/local/bin
==> /usr/bin/sudo /bin/mkdir -p /usr/local/Cellar /usr/local/Homebrew /usr/local/Frameworks /usr/local/etc /usr/lo
cal/include /usr/local/lib /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

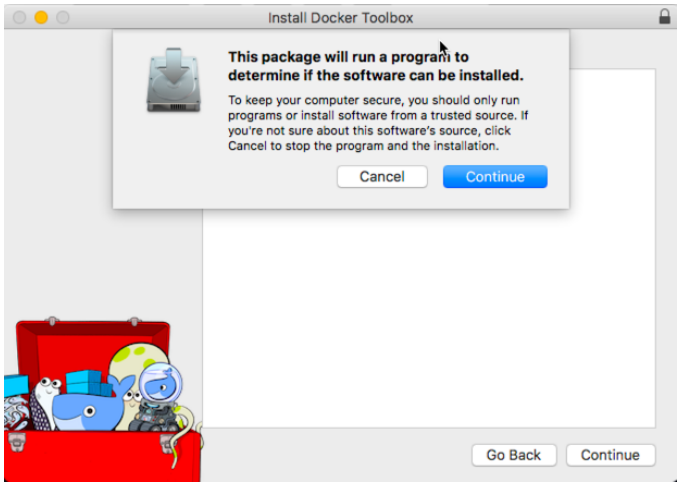
```
==> Tapping homebrew/core
Cloning into '/usr/local/Homebrew/Library/Taps/homebrew/homebrew-core'...
remote: Counting objects: 4449, done.
remote: Compressing objects: 100% (4250/4250), done.
remote: Total 4449 (delta 34), reused 462 (delta 13), pack-reused 0
Receiving objects: 100% (4449/4449), 3.53 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$
```

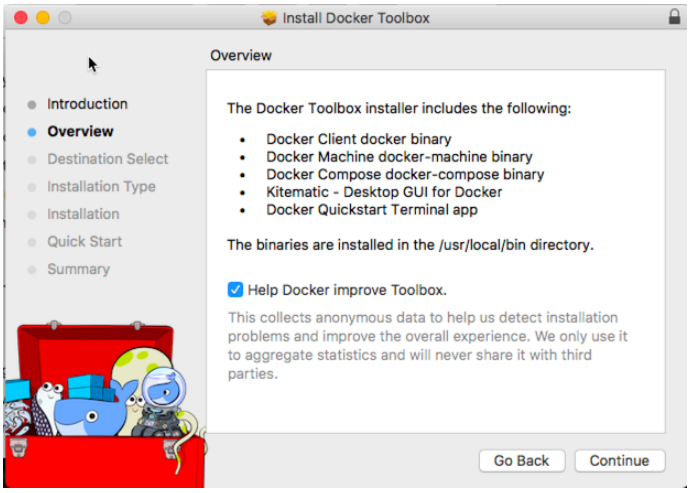
Install Docker Toolbox

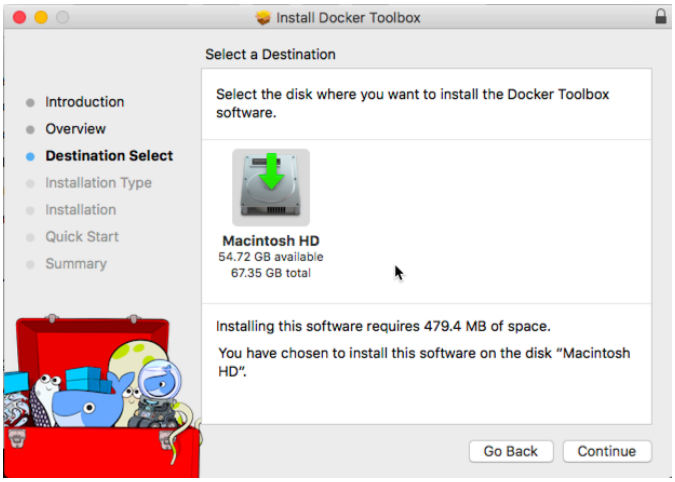
1. Right Click on "DockerToolbox.dmg" and select installer

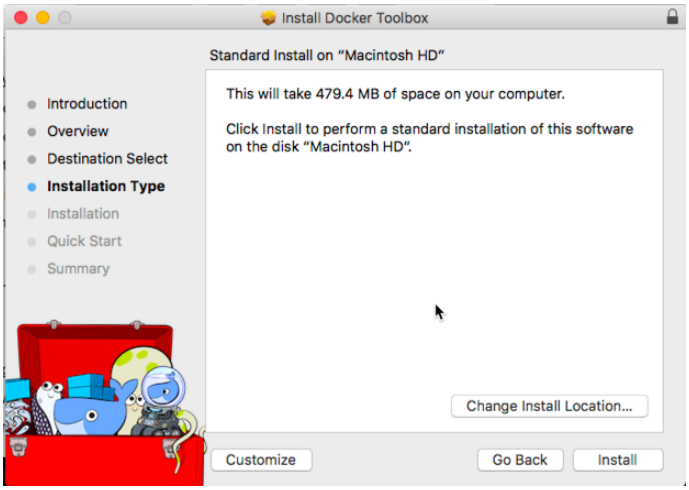


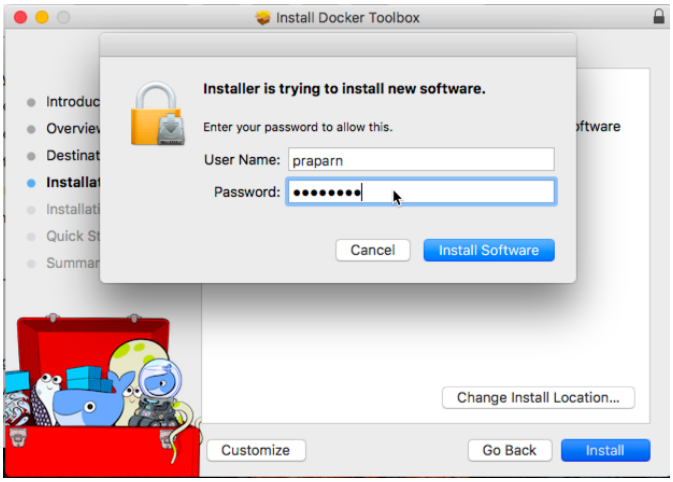
2. Following screen for setup

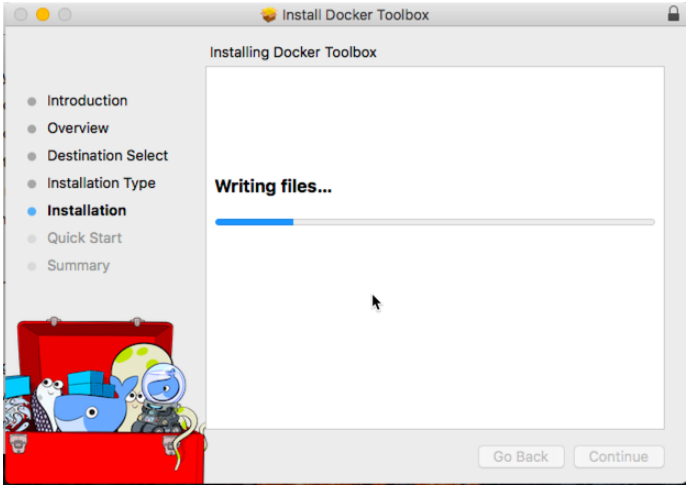




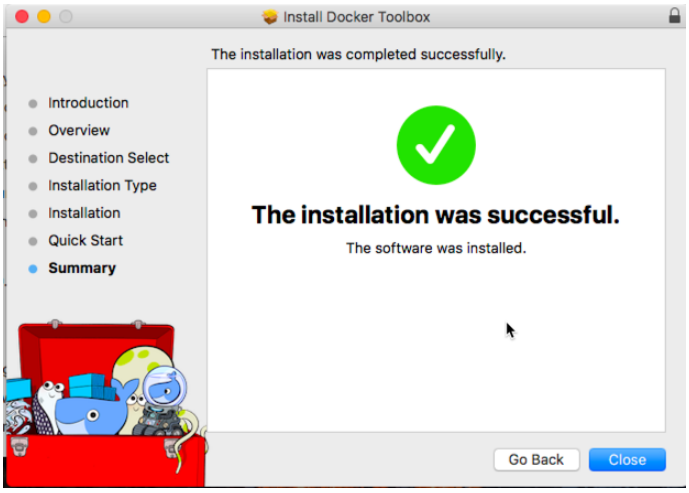







Install Docker Toolbox
Installing Docker Toolbox
Introduction
Overview
Destination Select
Installation Type
Installation
Quick Start
Summary
Writing files...
Go Back
Continue


Install Docker Toolbox
Quick Start
Introduction
Overview
Destination Select
Installation Type
Installation
Quick Start
Summary
Choose a tool to get started with Docker:
Docker Quickstart Terminal
Kitematic (Beta) Visual Management for Docker
You can find these later in the Docker directory under Applications.
Go Back
Continue


Install Docker Toolbox
The installation was completed successfully.
Introduction
Overview
Destination Select
Installation Type
Installation
Quick Start
Summary
The installation was successful.
The software was installed.
Go Back
Close

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

```
praparn — -bash — 114×32
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

```
praparn — -bash — 114×32
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$
```

2. Check minikube interactive command

```
praparn — -bash — 114×32
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

```
praparn — -bash — 114×32
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$
```

4. Check kubectl interactive command

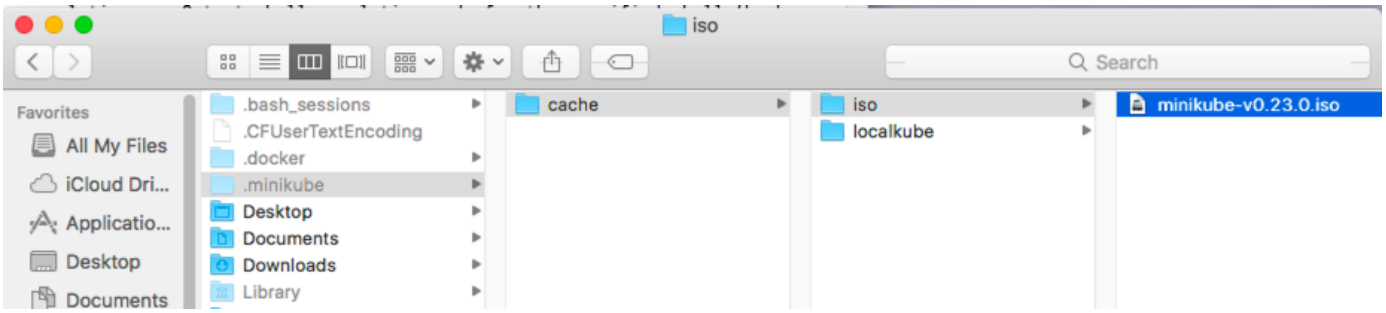
```
praparn — -bash — 114×32
macos-10:~ praparn$ kubectl
kubectl controls the Kubernetes cluster manager.

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. Create folder by command:

```
cd $Home
mkdir .minikube
mkdir .minikube/cache
mkdir .minikube/cache/iso
mkdir .minikube/cache/localkube
```

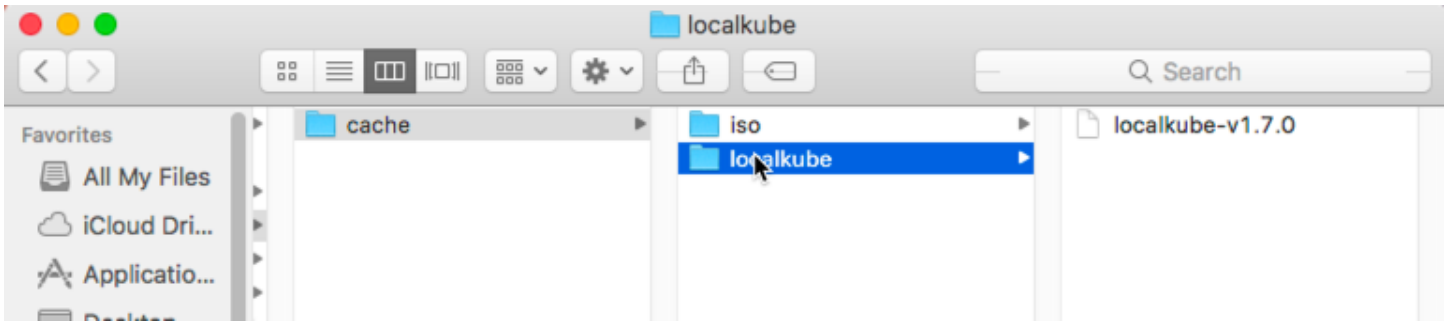
6. Copy file "minikube-v0.21.0iso" from Software_Package/kubenetes/ISO/minikube-v0.21.0.iso to

   /Users/<username>/.minikube/cache/iso




```
praparns-MacBook-Pro% cp minikube-v0.23.0.iso /Users/praparnlueangphoonlap/.minikube/cache/iso
praparns-MacBook-Pro%
```

7. Copy file "localkube-v1.7.0" Software_Package/kubenetes/localkube/localkube-v.1.7.0" to /User/<username>/.minikube/cache/




```
praparns-MacBook-Pro% cp localkube-v1.7.0 /Users/praparnlueangphoonlap/.minikube/cache/localkube
praparns-MacBook-Pro%
```

8. Configure minikube for use kubernetes version 1.7.0 by command: minikube config set kubernetes-version v1.7.0

```
praparns-MacBook-Pro% minikube config set kubernetes-version v1.7.0
```

9. Create minikube machine by command:

"minikube start --vm-driver=virtualbox profile=minikubelab1 --iso-url=https://storage.googleapis.com/minikube/iso/minikube-v0.23.0.iso"

```
praparns-MacBook-Pro% minikube start --vm-driver=virtualbox profile=minikubelab1 --iso-url=https://storage.googleap
is.com/minikube/iso/minikube-v0.21.0.iso
Starting local Kubernetes v1.7.0 cluster...
Starting VM...
SSH-ing files into VM...
Setting up certs...
Starting cluster components...
Connecting to cluster...
Setting up kubeconfig...
Kubectl is now configured to use the cluster.
praparns-MacBook-Pro%
```

10. Check status of minikube's machine by command: "minikube status","minikube ip"

```
praparns-MacBook-Pro:localkube praparn$ minikube status
minikubeVM: Running
localkube: Running
praparns-MacBook-Pro:localkube praparn$ minikube ip
192.168.99.104
praparns-MacBook-Pro:localkube praparn$
```

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

```
praparns-MacBook-Pro% minikube ssh
$ docker version
Client:
 Version:      1.12.6
 API version:  1.24
 Go version:   go1.6.4
 Git commit:   78d1802
 Built:        Wed Jan 11 00:23:16 2017
 OS/Arch:      linux/amd64

Server:
 Version:      1.12.6
 API version:  1.24
 Go version:   go1.6.4
 Git commit:   78d1802
 Built:        Wed Jan 11 00:23:16 2017
 OS/Arch:      linux/amd64
$
```

12. Check health of kubenetes cluster by command
    a. kubectl get nodes ➔ check node status
    b. kubectl get cs ➔ check cluster status

```
praparns-MacBook-Pro% kubectl get node
NAME       STATUS    AGE       VERSION
minikube   Ready     1m        v1.7.0
praparns-MacBook-Pro% kubectl get cs
NAME                 STATUS    MESSAGE              ERROR
scheduler            Healthy   ok
controller-manager   Healthy   ok
etcd-0               Healthy   {"health": "true"}
praparns-MacBook-Pro%
```

13. 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:localkube praparn$ kubectl get pods
No resources found.
praparns-MacBook-Pro:localkube praparn$ kubectl get deployment
No resources found.
praparns-MacBook-Pro:localkube praparn$ kubectl get svc
NAME         CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   10.0.0.1     <none>        443/TCP   4m
praparns-MacBook-Pro:localkube praparn$ kubectl describe svc
Name:                   kubernetes
Namespace:              default
Labels:                 component=apiserver
                        provider=kubernetes
Annotations:            <none>
Selector:               <none>
Type:                   ClusterIP
IP:                     10.0.0.1
Port:                   https   443/TCP
Endpoints:              10.0.2.15:8443
Session Affinity:       ClientIP
Events:                 <none>
praparns-MacBook-Pro:localkube praparn$
```

14. 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:localkube praparn$ kubectl run webtest --image=labdocker/nginx:latest --port=80  deployment nginx
deployment "webtest" created
praparns-MacBook-Pro:localkube praparn$ kubectl expose deployment webtest --target-port=80 --type=NodePort
service "webtest" exposed
praparns-MacBook-Pro:localkube praparn$
```

15. 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:localkube praparn$ kubectl get svc
NAME         CLUSTER-IP   EXTERNAL-IP   PORT(S)        AGE
kubernetes   10.0.0.1     <none>        443/TCP        13m
webtest      10.0.0.83    <nodes>       80:30891/TCP   8s
praparns-MacBook-Pro:localkube praparn$ kubectl describe svc webtest
Name:                     webtest
Namespace:                default
Labels:                   run=webtest
Annotations:              <none>
Selector:                 run=webtest
Type:                     NodePort
IP:                       10.0.0.83
Port:                     <unset> 80/TCP
NodePort:                 <unset> 30891/TCP
Endpoints:                172.17.0.4:80
Session Affinity:         None
Events:                   <none>
praparns-MacBook-Pro:localkube praparn$
```

16. Test open webpage with port describe on command above (This example: http://192.168.99.104:30891)

17. 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$
```