Manual

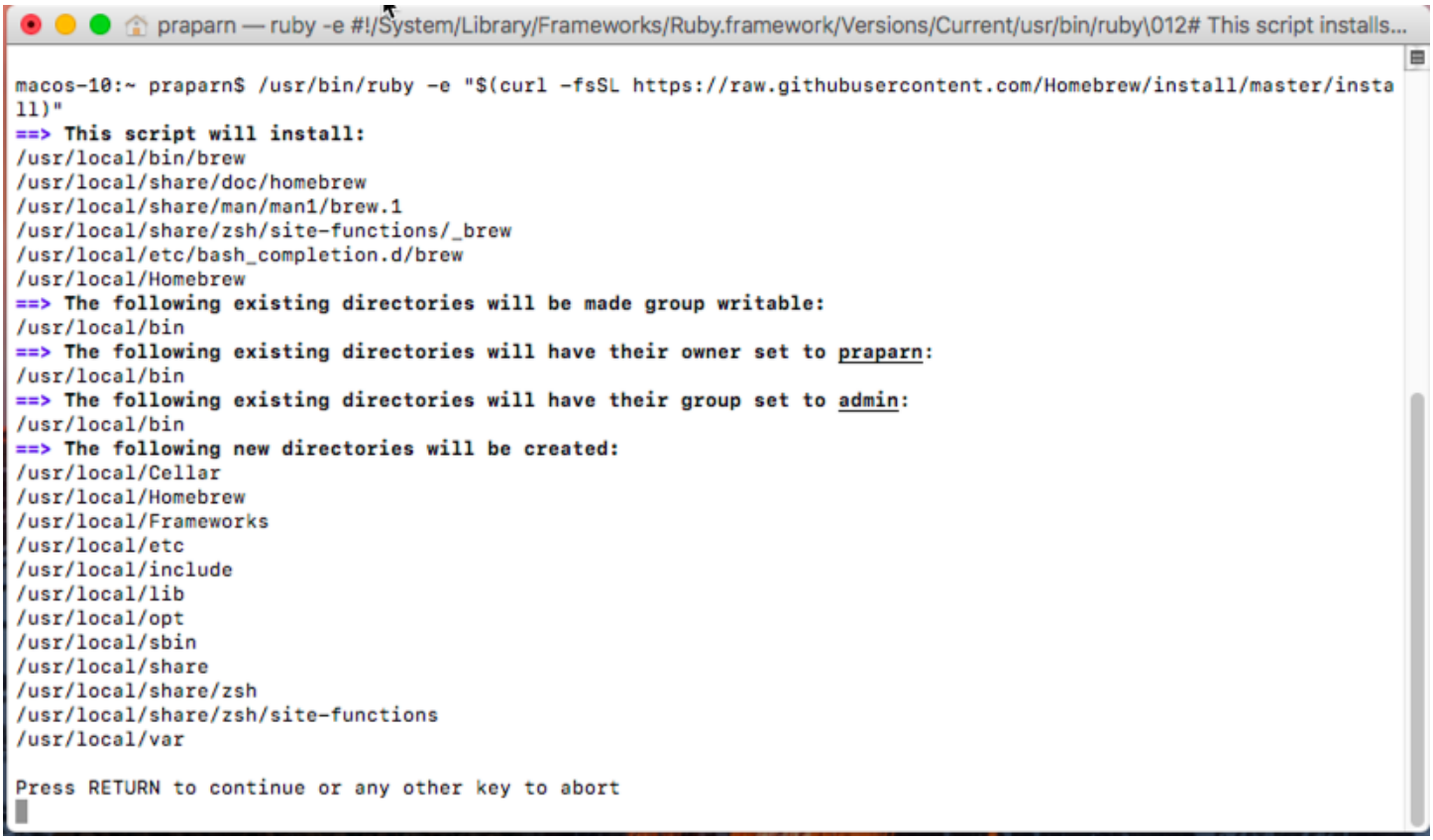
Install Minikube Software Set for OSX Platform

Prerequisite

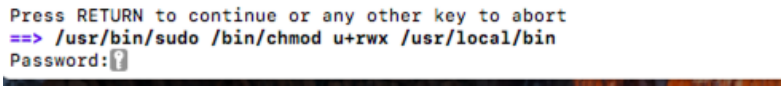
Install brew for MACOS

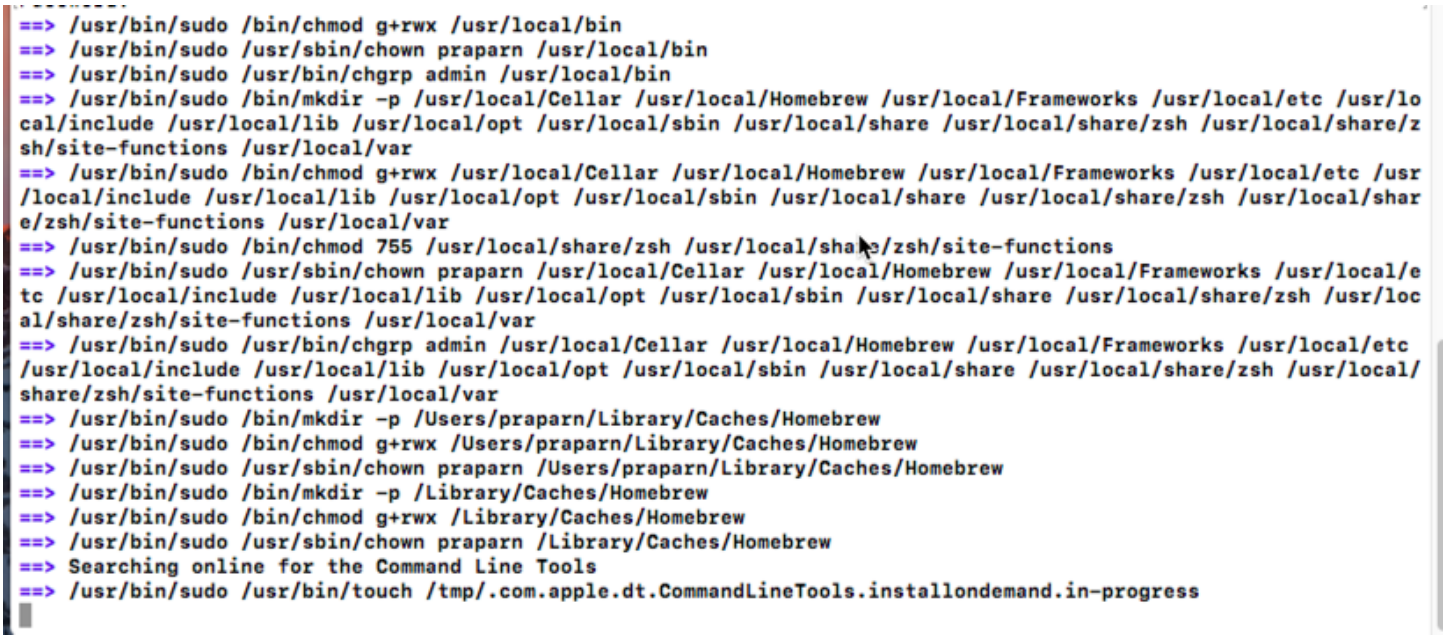
1. Install brew module by command:l.

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

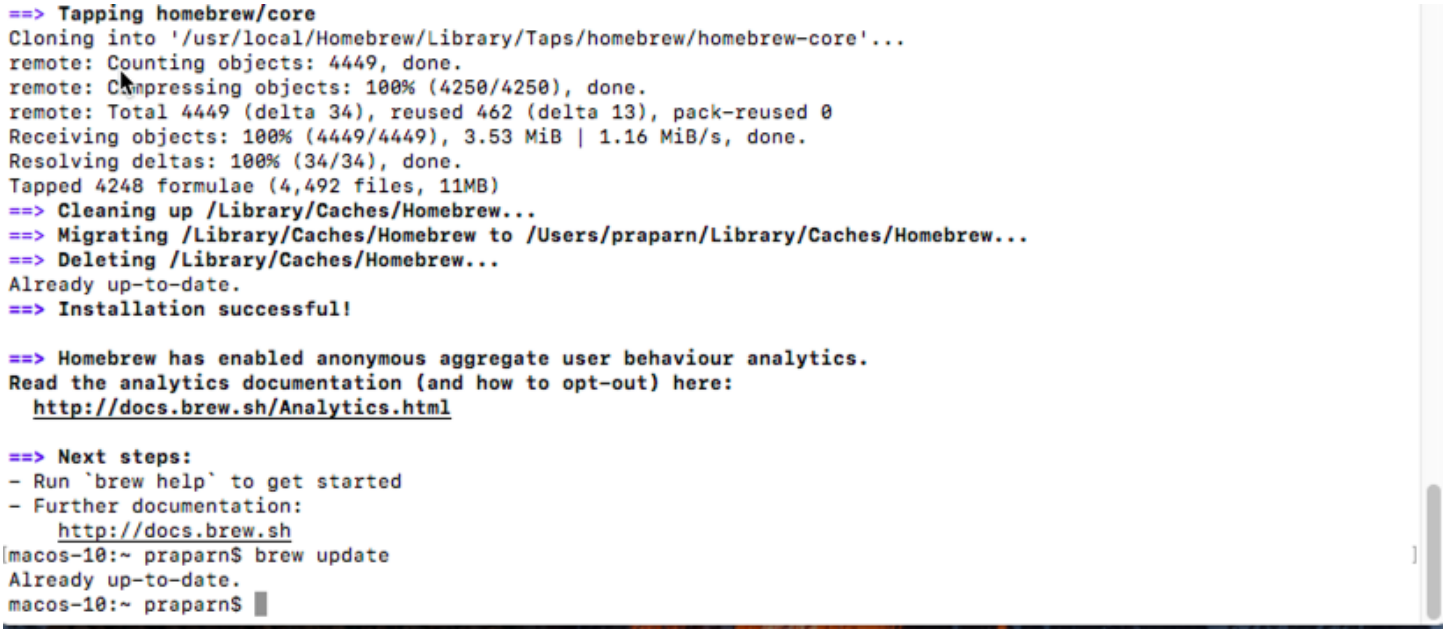


2. Input password for grant privilege for install and wait until all install process was done.



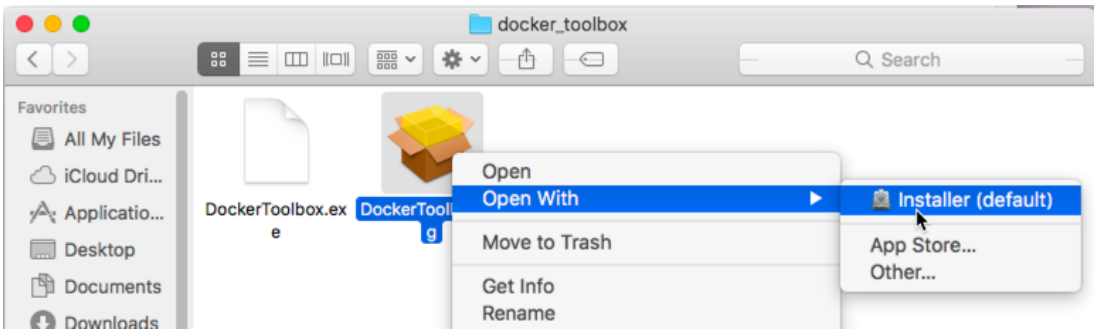


3. After install brew have finished install. Check brew by command: brew update

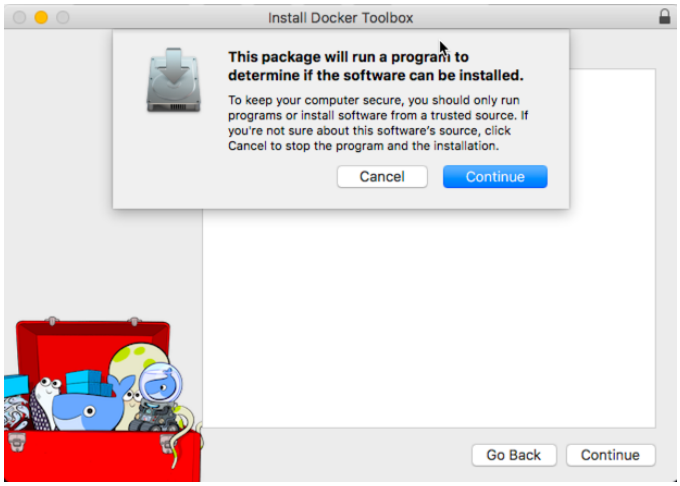


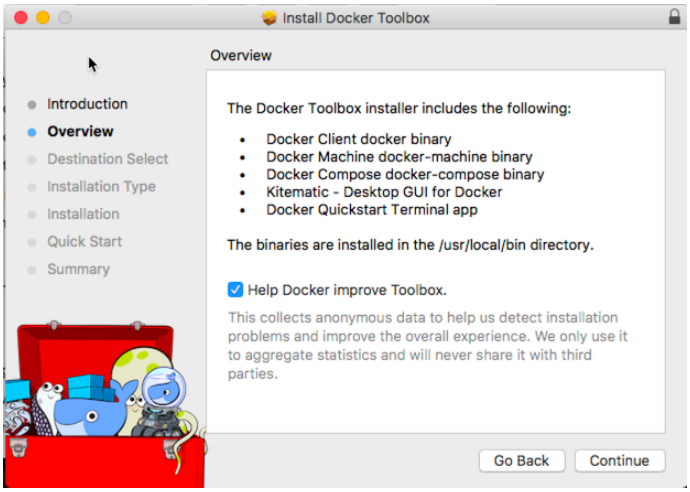
Install Docker Toolbox/DVM

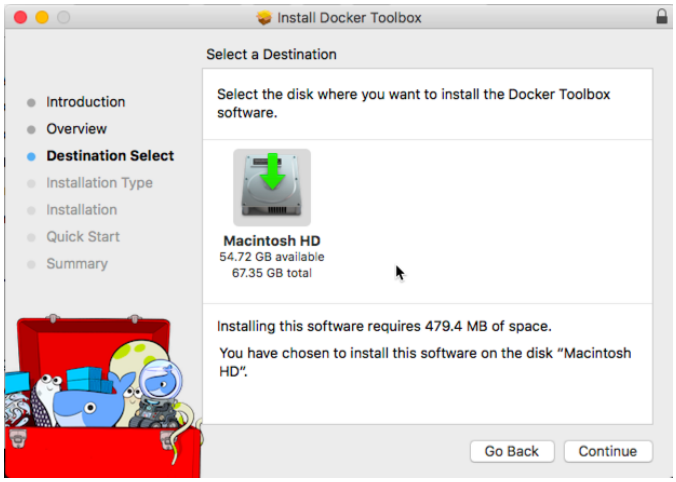
1. Right Click on "DockerToolbox.dmg" and open with installer.

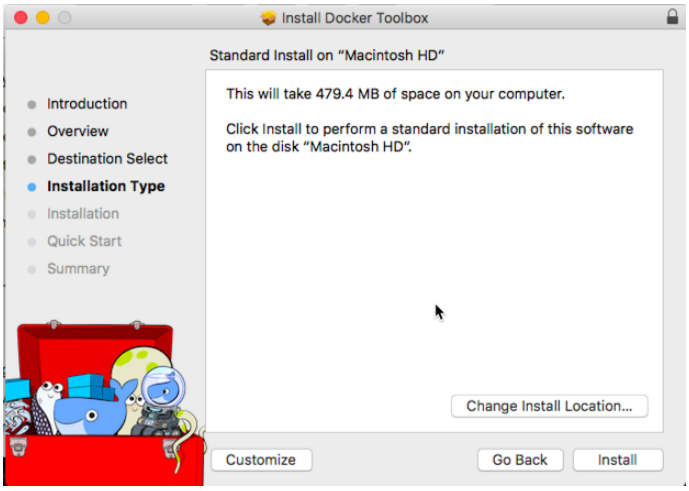


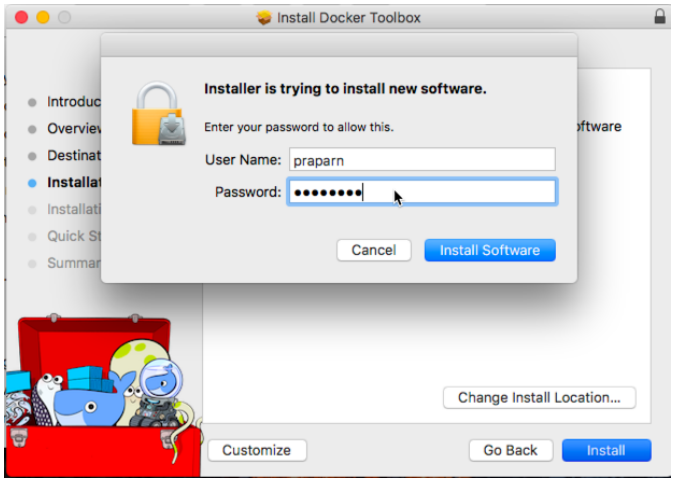
2. Following screen for setup

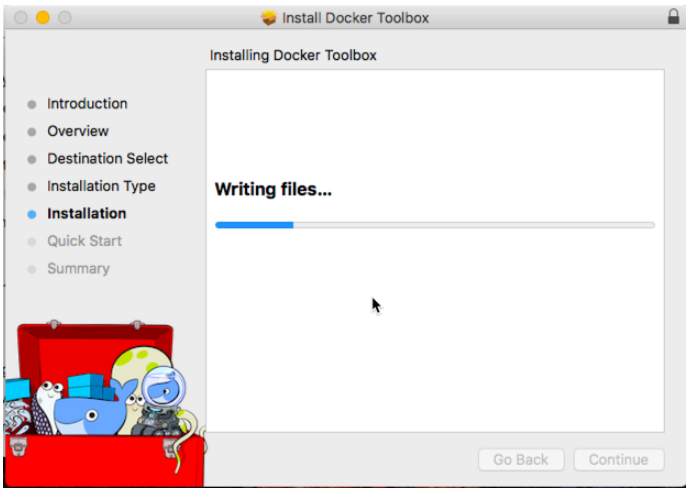




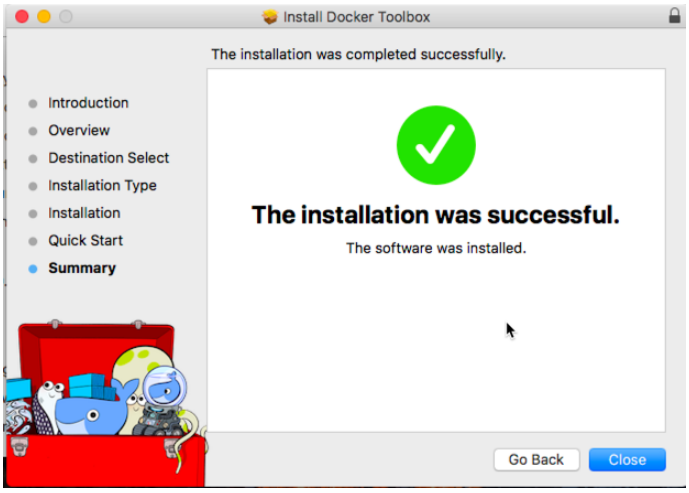




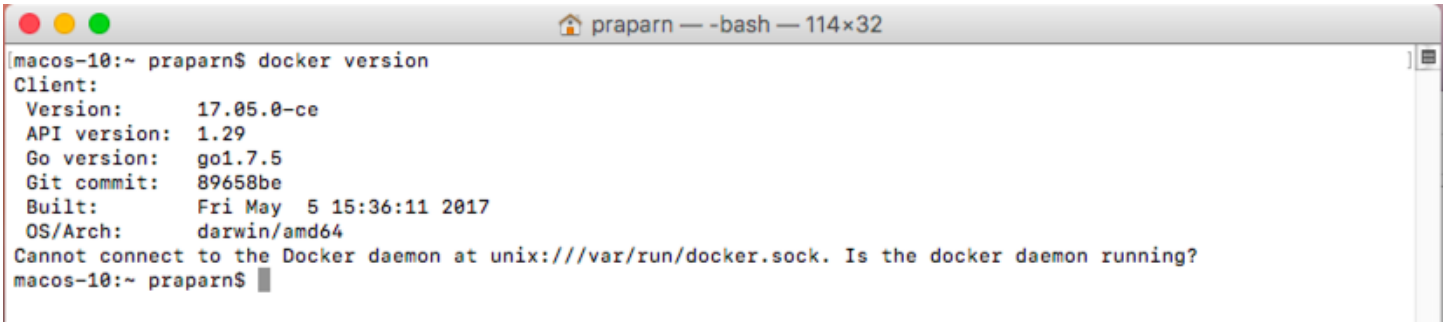



Install Docker Toolbox
Installing Docker Toolbox
Introduction
Overview
Destination Select
Installation Type
Installation
Quick Start
Summary
Writing files...
Go Back
Continue


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

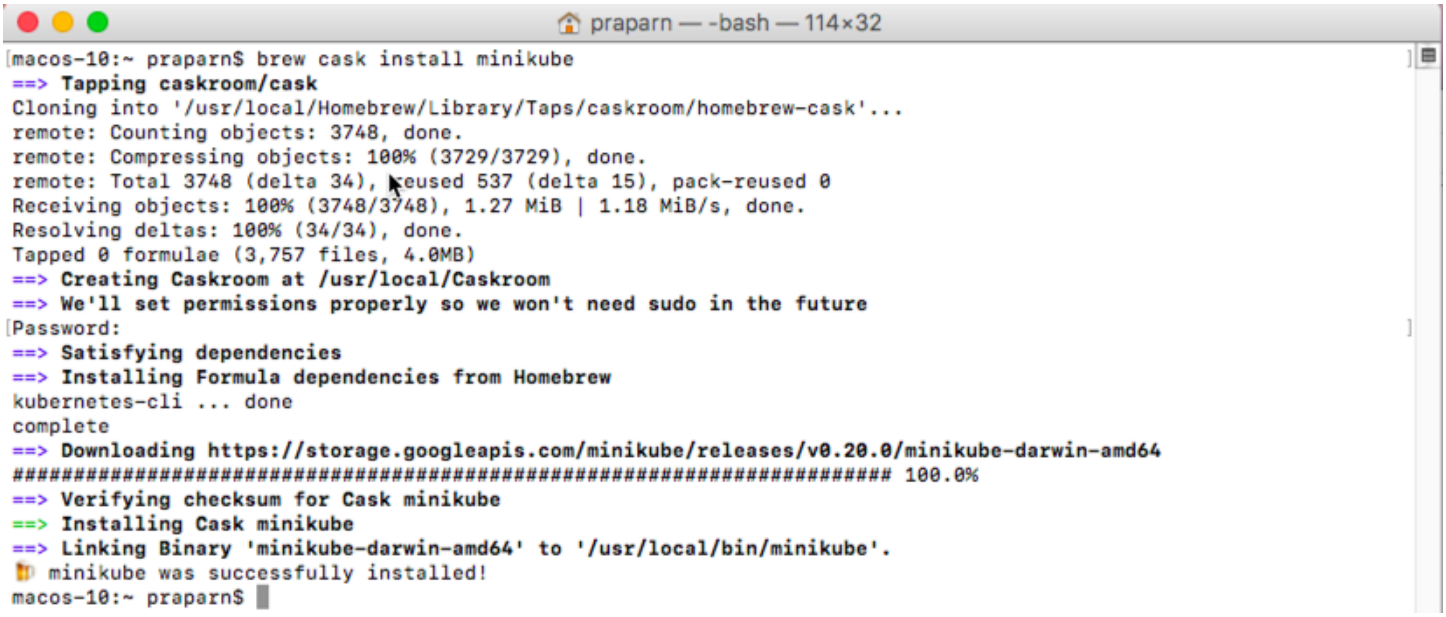
3. Check version of Docker Tools by command: docker version



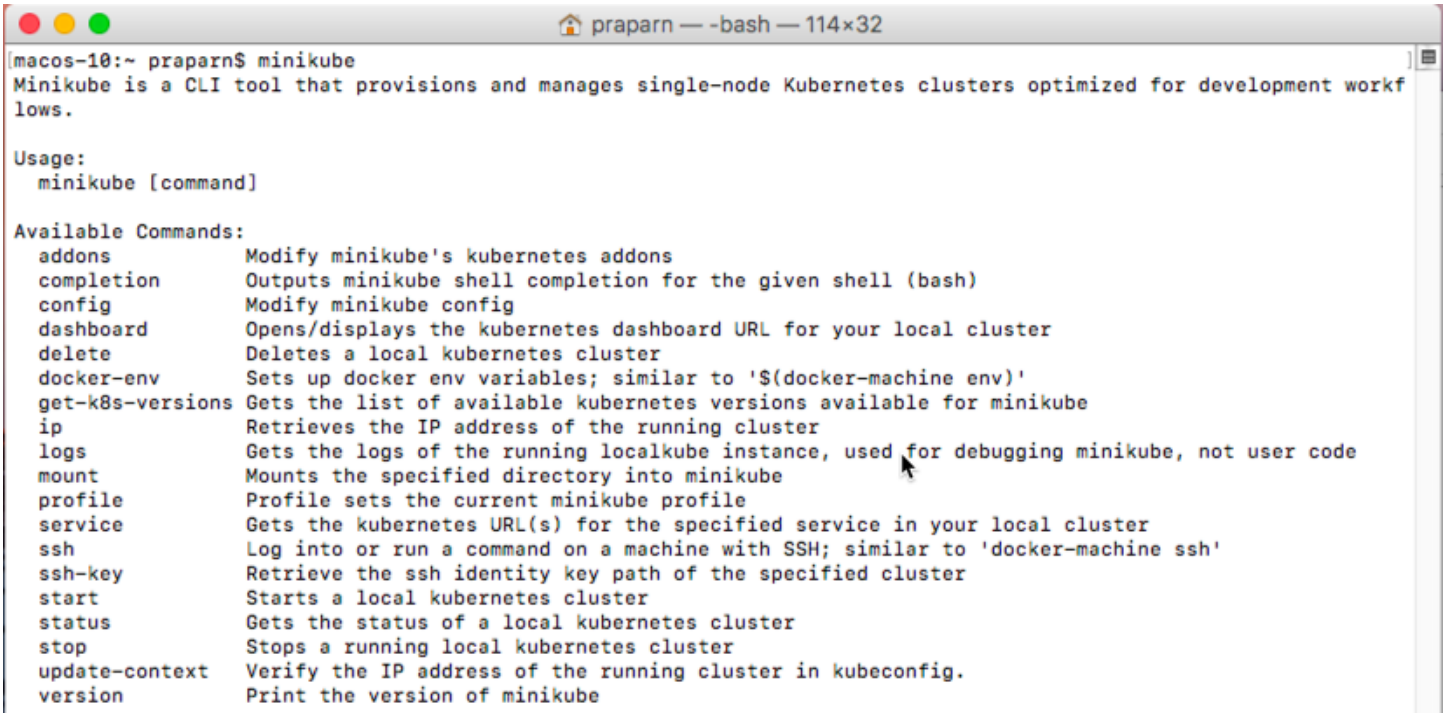
4. Install dvm (Docker Version Manager) by command
    a. brew update
    b. brew install dvm

Install minikube / Initial minikube machine

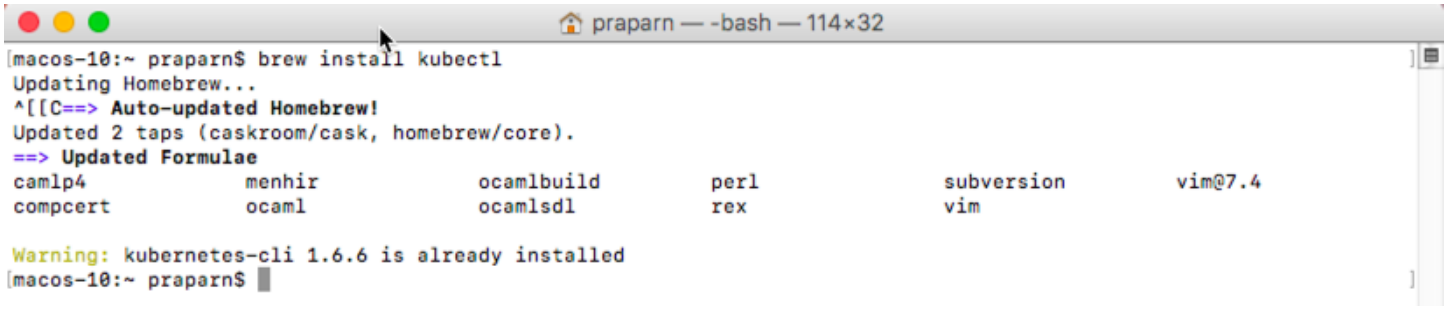
1. Install minikube by command: brew cask install minikube



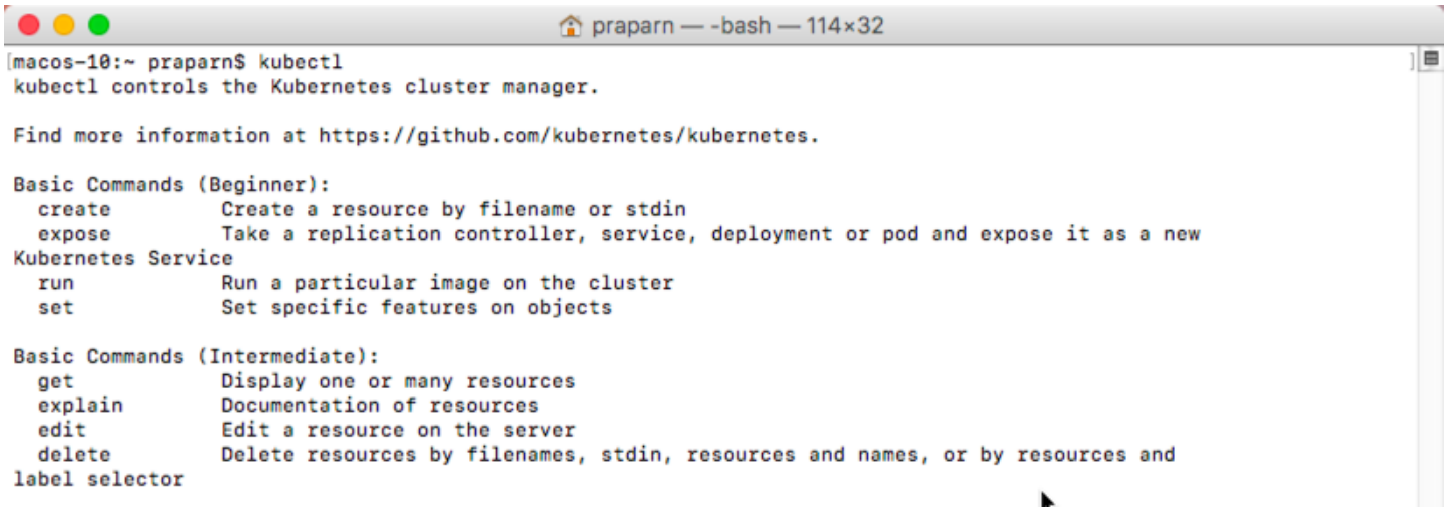
2. Check minikube interactive command



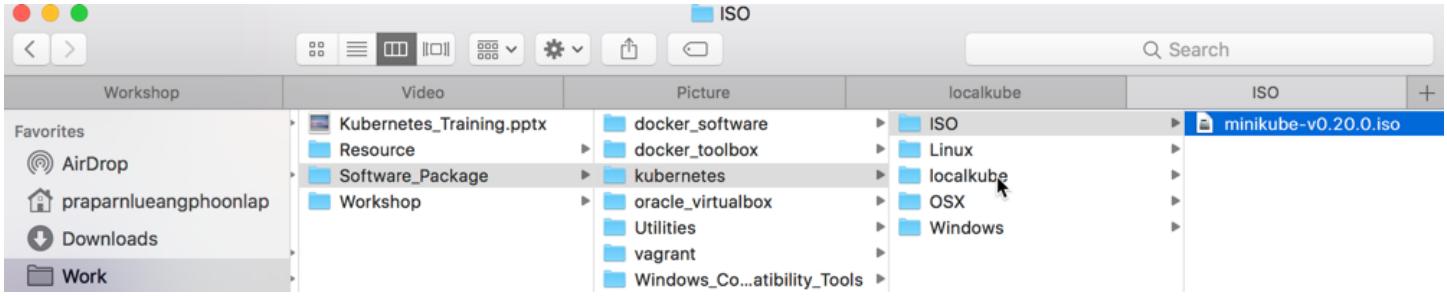
3. Install kubectl by command: brew install kubectl

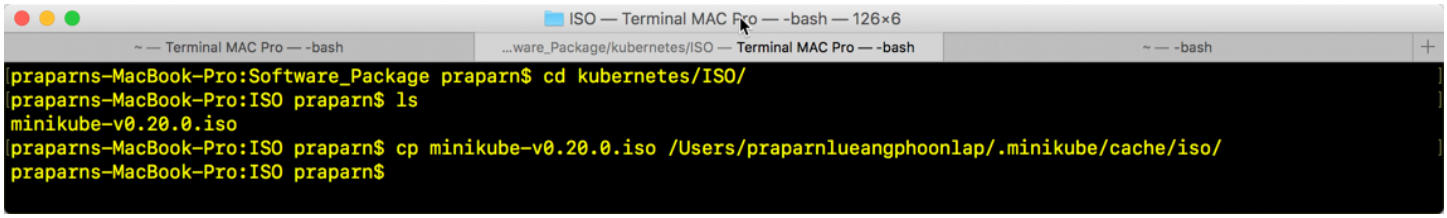


4. Check kubectl interactive command

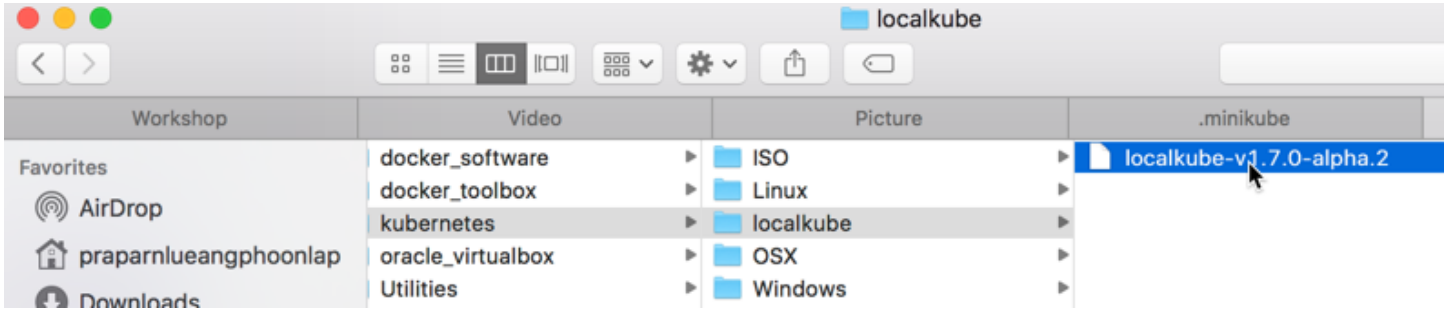


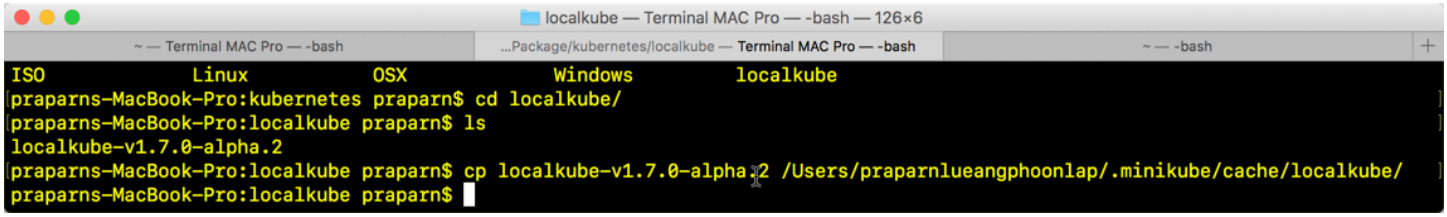
5. Copy file "minikube-v0.20.0iso" from Software_Package/kubenetes/ISO/minikube-v0.20.0.iso to /Users/<username>/.minikube/cache/iso



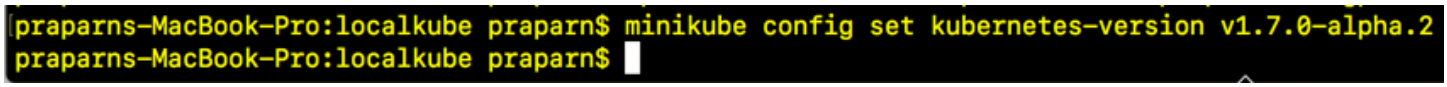


6. Copy file "localkube-v1.7.0-alpha.2" Software_Package/kubenetes/localkube/localkube-v.1.7.0-alpha.2" to /User/<username>/.minikube/cache/

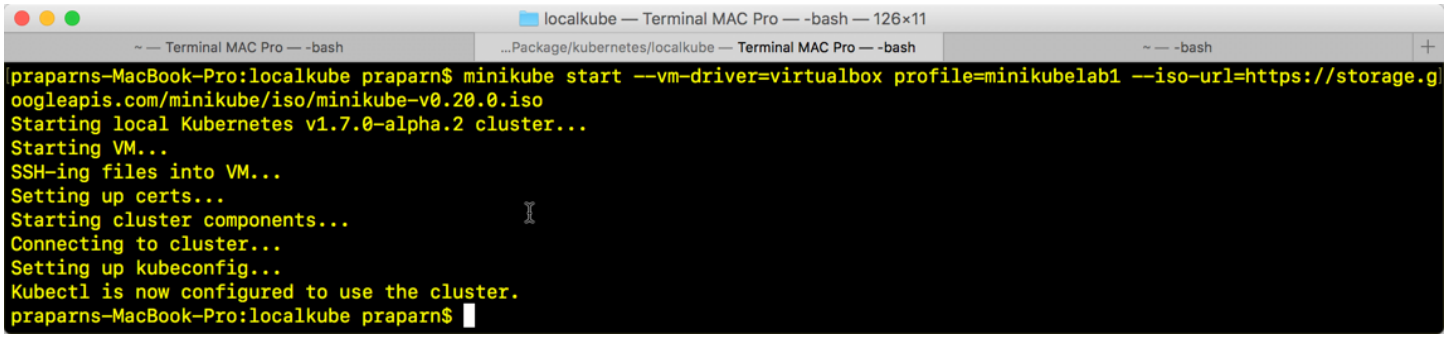




7. Configure minikube for use kubernetes version 1.7.0 by command: minikube config set kubernetes-version v1.7.0-alpha.2



8. Create minikube machine by command: "minikube start --vm-driver=virtualbox profile=minikubelab1 --iso-url=https://storage.googleapis.com/minikube/iso/minikube-v0.20.0.iso"



9. 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$
```

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

```
praparns-MacBook-Pro:localkube praparn$ minikube ssh
$ docker version
Client:
 Version:      1.11.1
 API version:  1.23
 Go version:   go1.5.4
 Git commit:   5604cbe
 Built:        Wed Apr 27 00:34:20 2016
 OS/Arch:      linux/amd64

Server:
 Version:      1.11.1
 API version:  1.23
 Go version:   go1.5.4
 Git commit:   5604cbe
 Built:        Wed Apr 27 00:34:20 2016
 OS/Arch:      linux/amd64
$
```

11. From previous screen the minikube's machine will use docker version 1.11.1. So we would like to switch current docker version by command
    a. dvm ls check current version of docker on machine
    b. dvm use 1.11.1

```
praparns-MacBook-Pro:localkube praparn$ dvm ls
        1.11.1
->      system (17.06.0-ce-rc5)
praparns-MacBook-Pro:localkube praparn$ dvm use 1.11.1
Now using Docker 1.11.1
praparns-MacBook-Pro:localkube praparn$ dvm ls
->      1.11.1
        system (17.06.0-ce-rc5)
```

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:localkube praparn$ dvm ls
        1.11.1
->      system (17.06.0-ce-rc5)
praparns-MacBook-Pro:localkube praparn$ dvm use 1.11.1
Now using Docker 1.11.1
praparns-MacBook-Pro:localkube praparn$ dvm ls
->      1.11.1
        system (17.06.0-ce-rc5)
```

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$
```