Manual

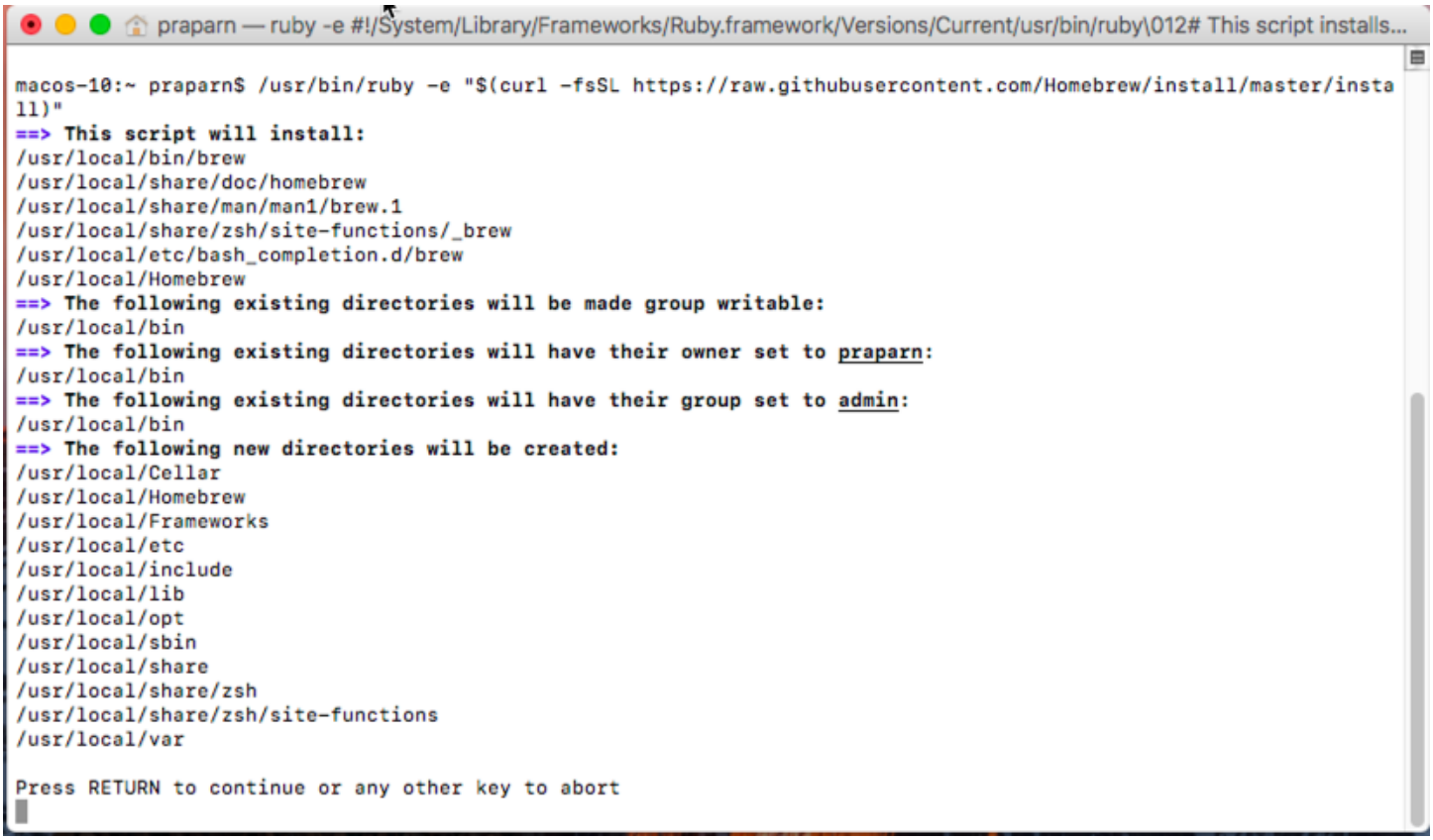
Install Minikube Software Set for OSX Platform

Prerequisite

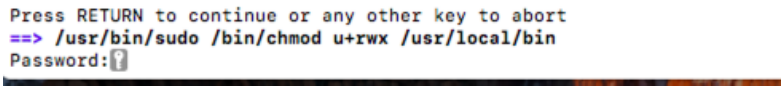
Install brew for MACOS

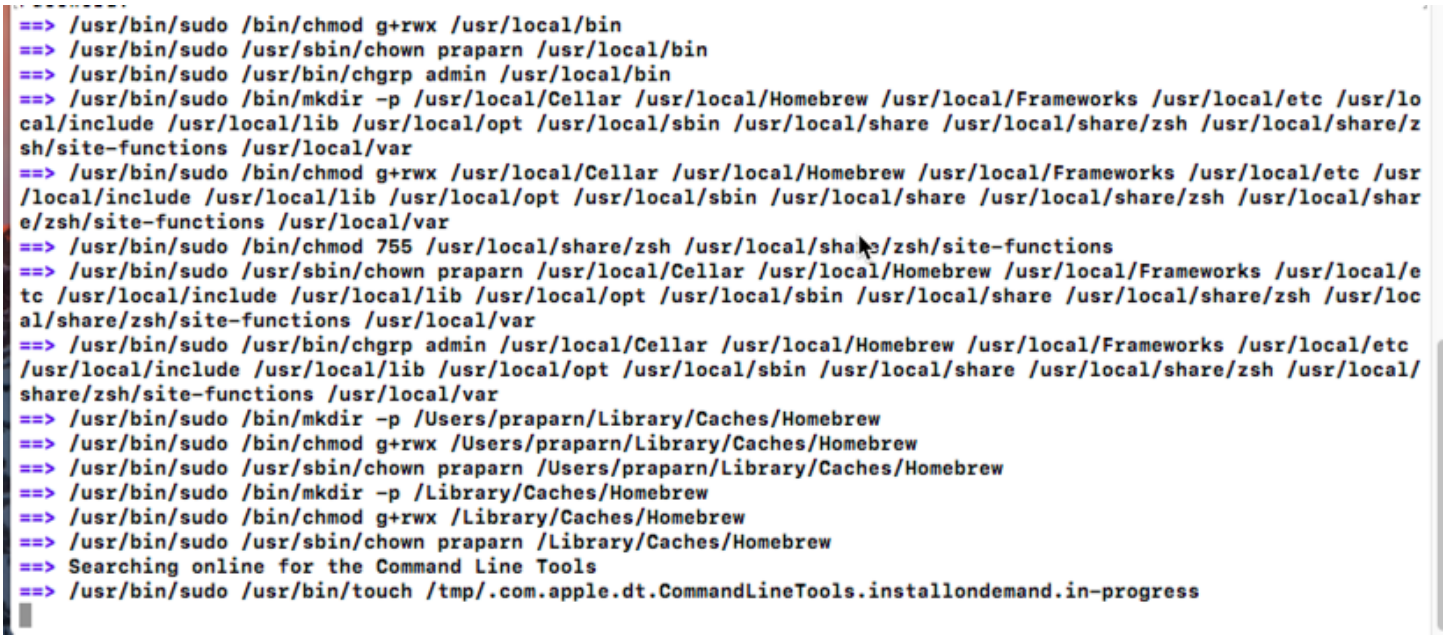
1. Install brew module by command:l.

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

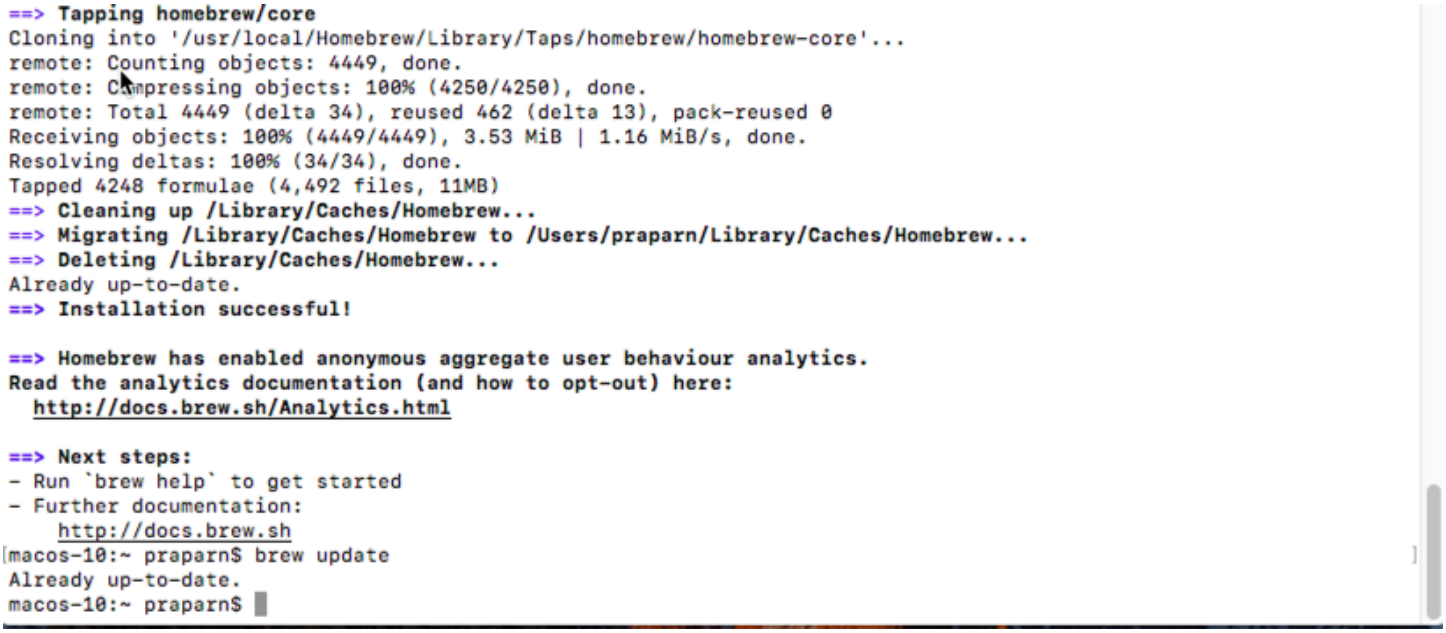


2. Input password for grant privilege for install and wait until all install process was done.



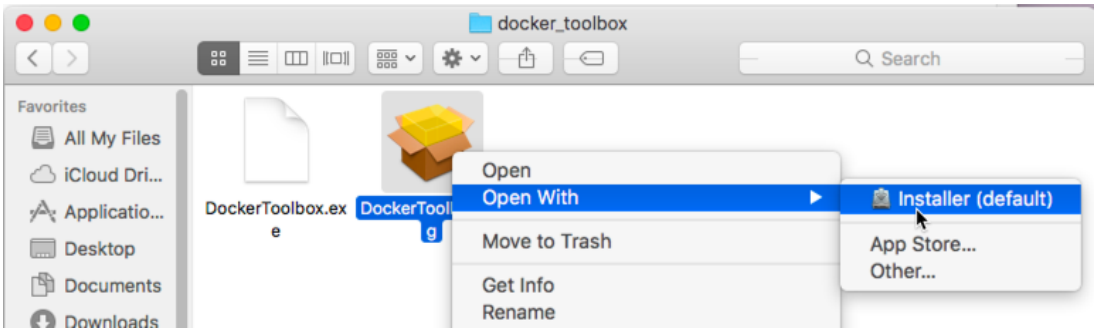


3. After install brew have finished install. Check brew by command: brew update

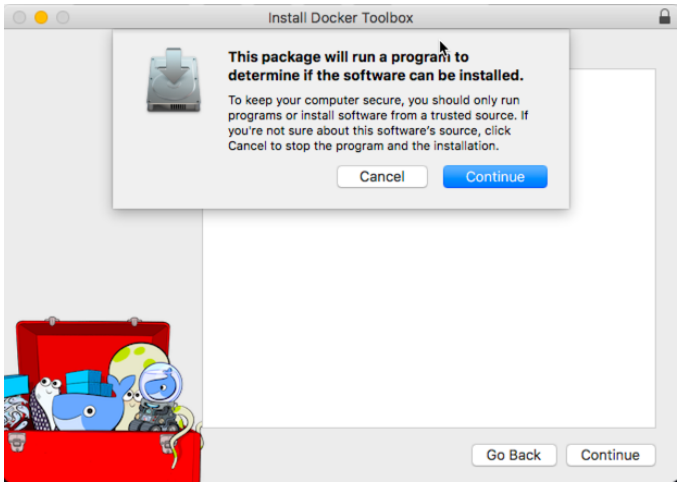


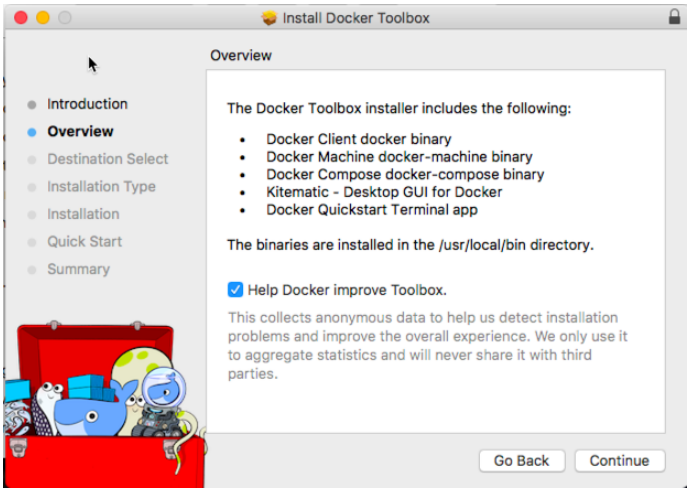
Install Docker Toolbox

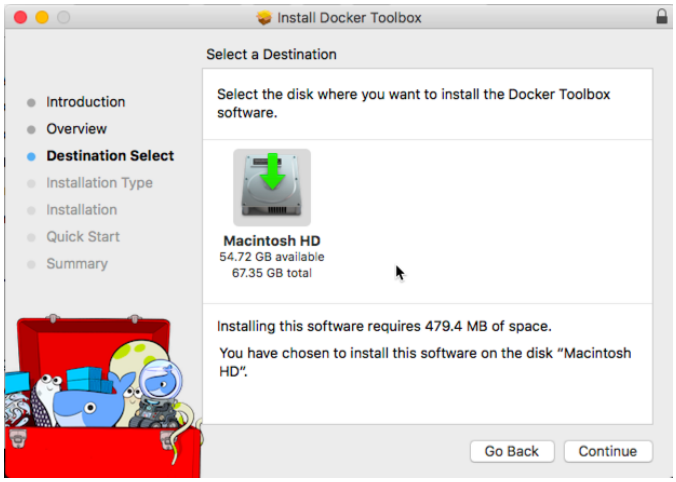
1. Right Click on "DockerToolbox.dmg" and select installer

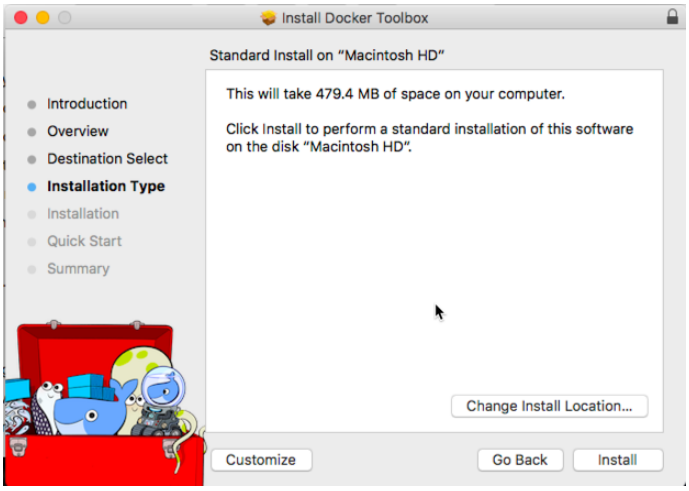


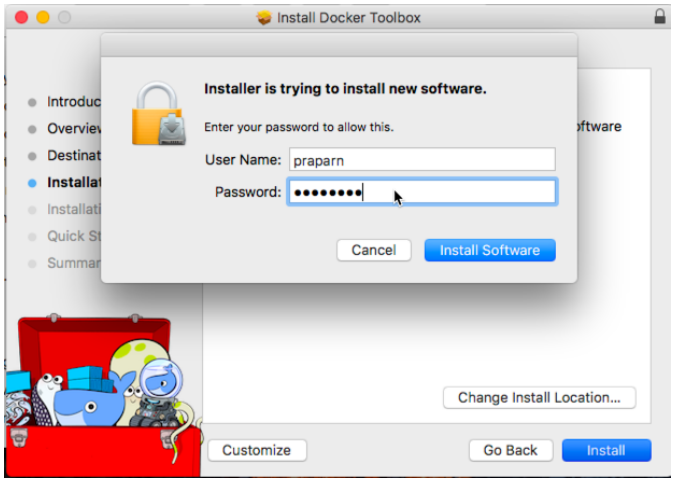
2. Following screen for setup

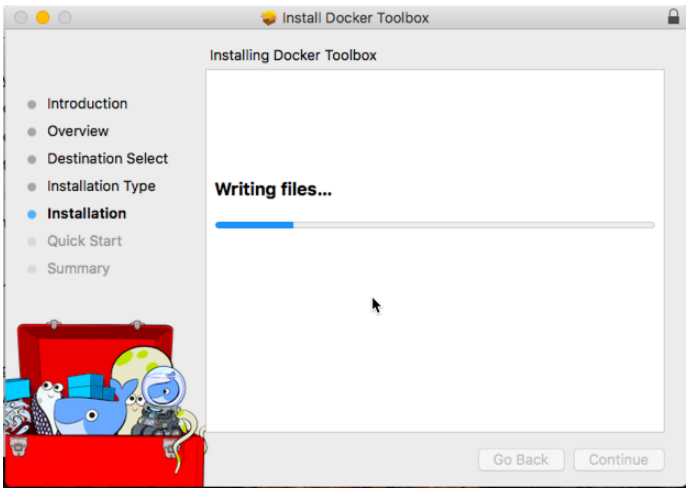




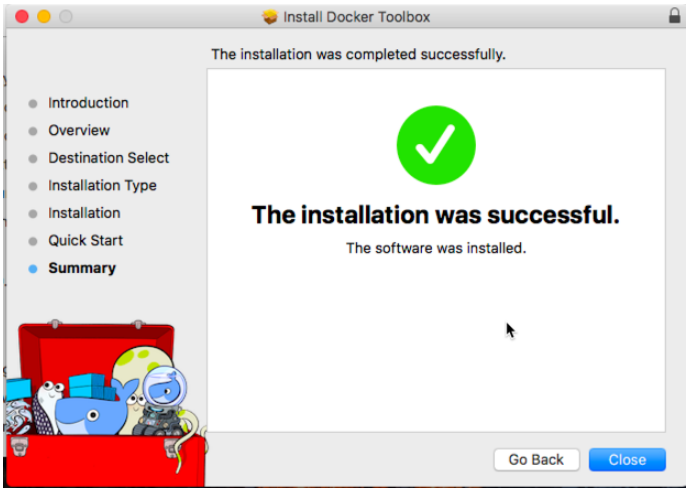




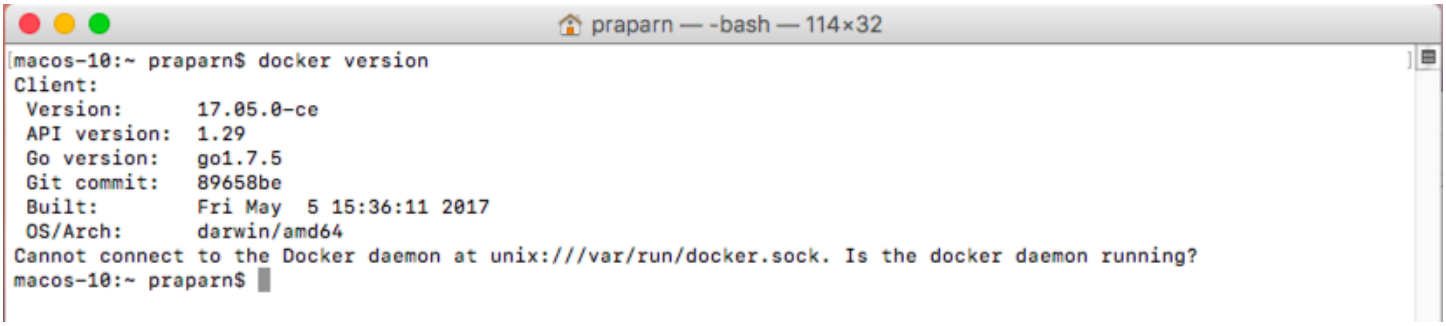



Install Docker Toolbox
Installing Docker Toolbox
Introduction
Overview
Destination Select
Installation Type
Installation
Quick Start
Summary
Writing files...
Go Back
Continue


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

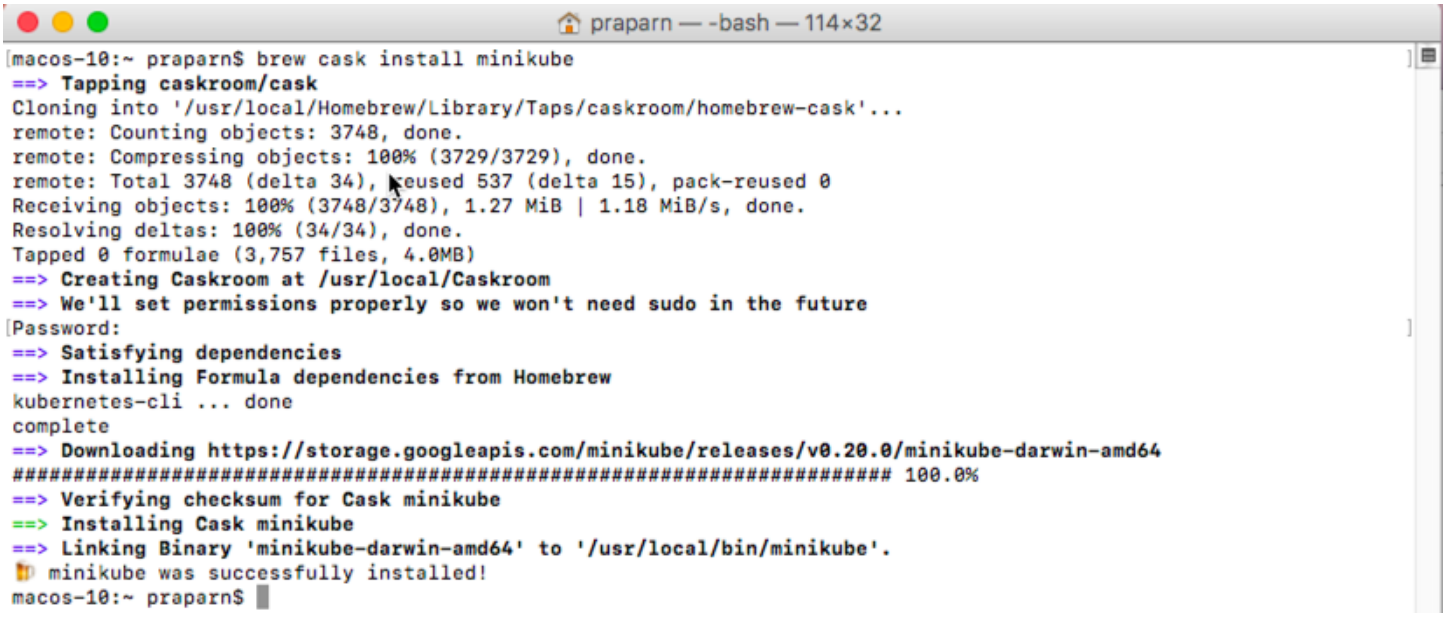


Install minikube / Initial minikube machine

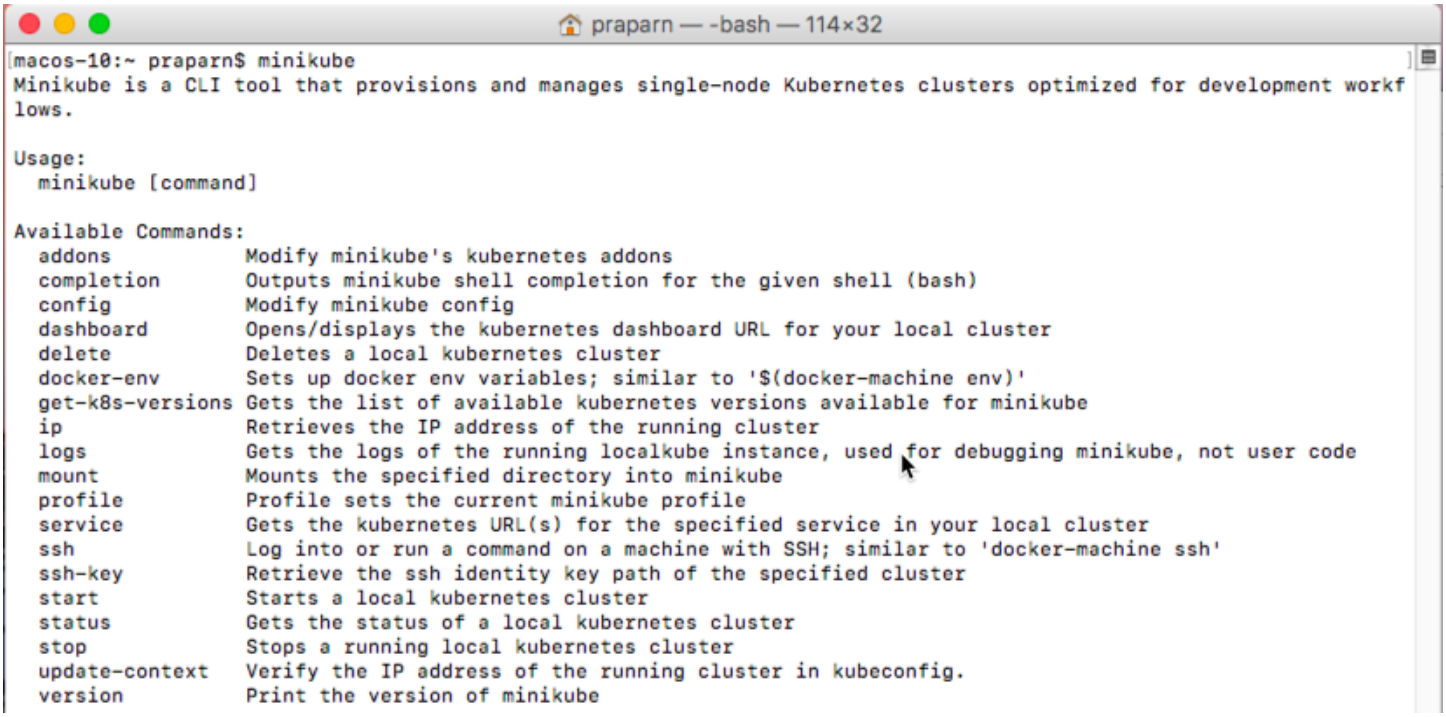
1. Install minikube by command: brew cask install minikube

   Or

curl -Lo minikube https://storage.googleapis.com/minikube/releases/vo.28.o/minikube-darwin-amd64 && chmod +x minikube && sudo mv minikube /usr/local/bin/



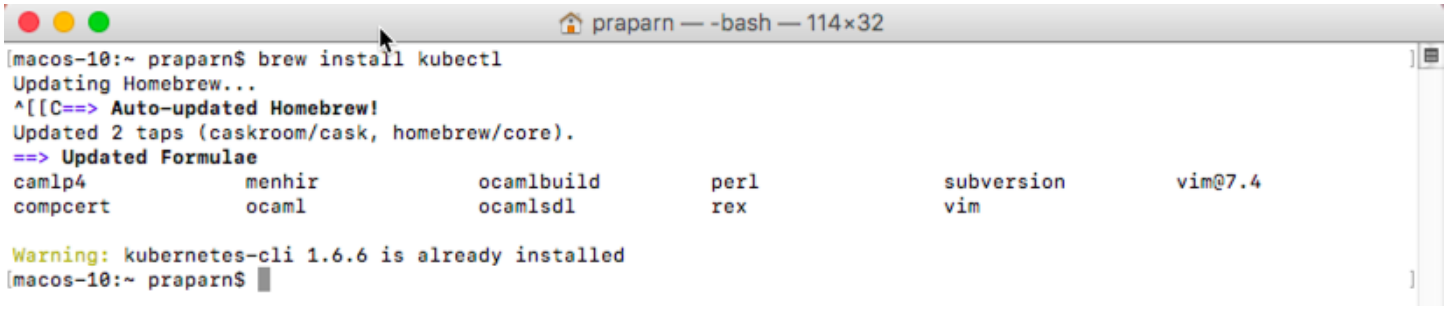
2. Check minikube interactive command

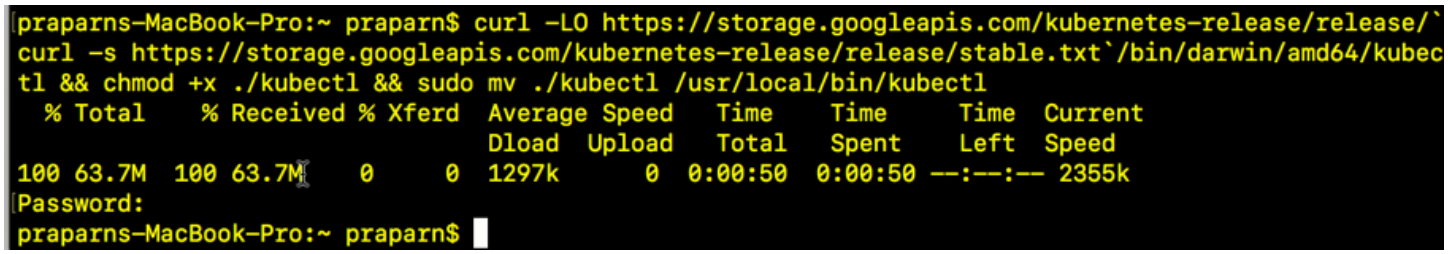


3. Install kubectl by command: brew install kubectl

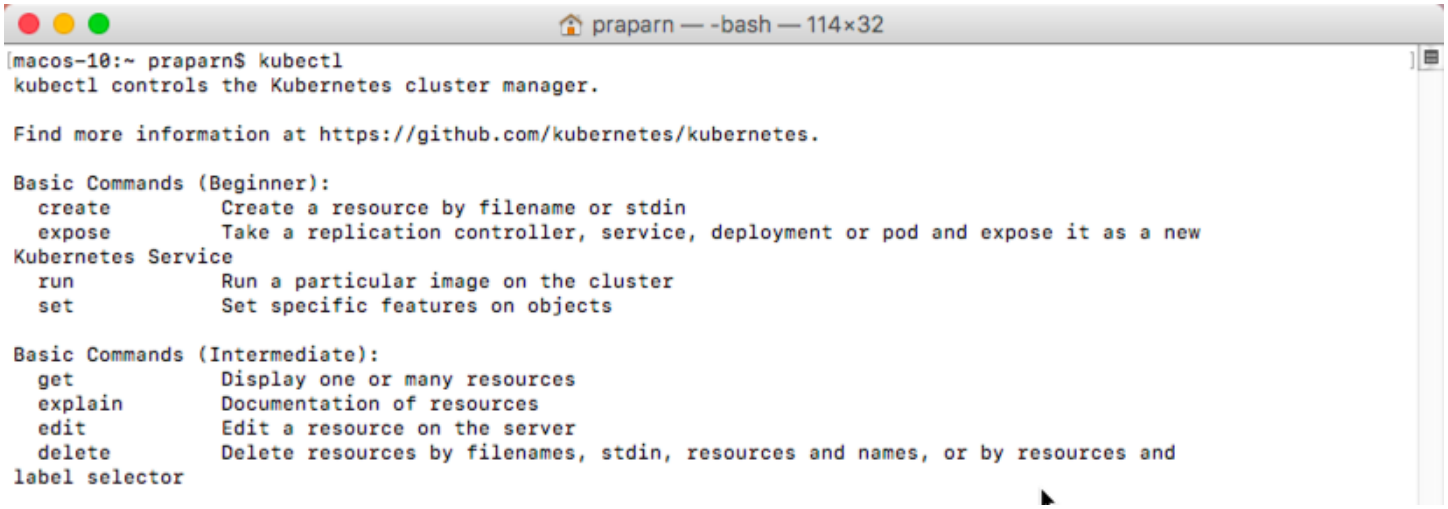
   Or

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

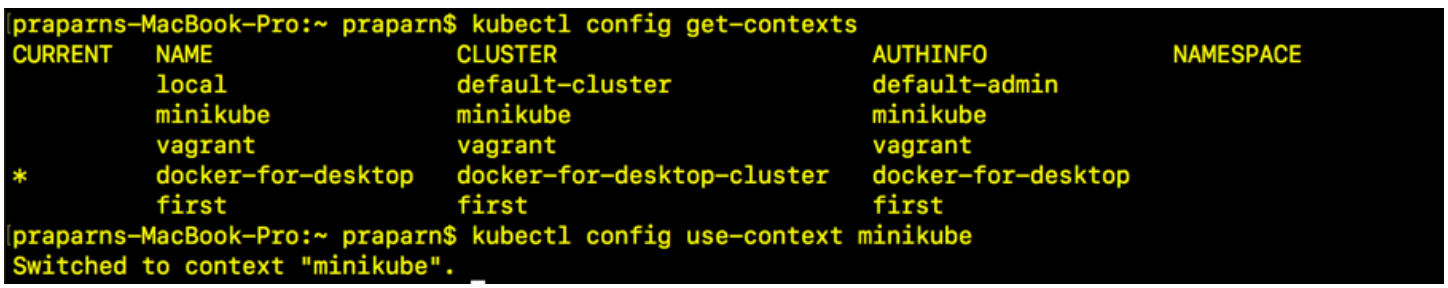




4. Check kubectl interactive command



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



6. Configure minikube for use kubernetes version 1.9.4 by command: minikube config set kubernetes-version v1.9.4

```
praparns-MacBook-Pro:~ praparn$ minikube get-k8s-versions
The following Kubernetes versions are available when using the localkube bootstrapper:
        - v1.10.0
        - v1.9.4
        - v1.9.0
        - v1.8.0
        - v1.7.5
        - v1.7.4
        - v1.7.3
        - v1.7.2
        - v1.7.0
        - v1.7.0-rc.1
        - v1.7.0-alpha.2
        - v1.6.4
        - v1.6.3
        - v1.6.0
        - v1.6.0-rc.1
        - v1.6.0-beta.4
        - v1.6.0-beta.3
        - v1.6.0-beta.2
        - v1.6.0-alpha.1
        - v1.6.0-alpha.0
        - v1.5.3
        - v1.5.2
        - v1.5.1
        - v1.4.5
        - v1.4.3
        - v1.4.2
        - v1.4.1
        - v1.4.0
        - v1.3.7
        - v1.3.6
        - v1.3.5
        - v1.3.4
        - v1.3.3
        - v1.3.0
praparns-MacBook-Pro:~ praparn$ minikube config set kubernetes-version v1.9.4
praparns-MacBook-Pro:~ praparn$
```

7. Create minikube machine by command:

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

```
praparns-MacBook-Pro:~ praparn$ minikube start --vm-driver=virtualbox profile=minikubelab1
Starting local Kubernetes v1.9.4 cluster...
Starting VM...
Getting VM IP address...
Moving files into cluster...
^[[C^[[CSetting 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.
praparns-MacBook-Pro:~ praparn$ minikube status
minikube: Running
cluster: Running
kubectl: Correctly Configured: pointing to minikube-vm at 192.168.99.100
praparns-MacBook-Pro:~ praparn$ minikube ip
192.168.99.100
```

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

```
praparns-MacBook-Pro% minikube status
minikube: Running
cluster: Running
kubectl: Correctly Configured: pointing to minikube-vm at 192.168.99.100
praparns-MacBook-Pro% minikube ip
192.168.99.100
praparns-MacBook-Pro%
```

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

```
praparns-MacBook-Pro% 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% kubectl get nodes
NAME       STATUS    ROLES     AGE       VERSION
minikube   Ready     <none>    1m        v1.10.0
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%
```

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   2m
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%
```

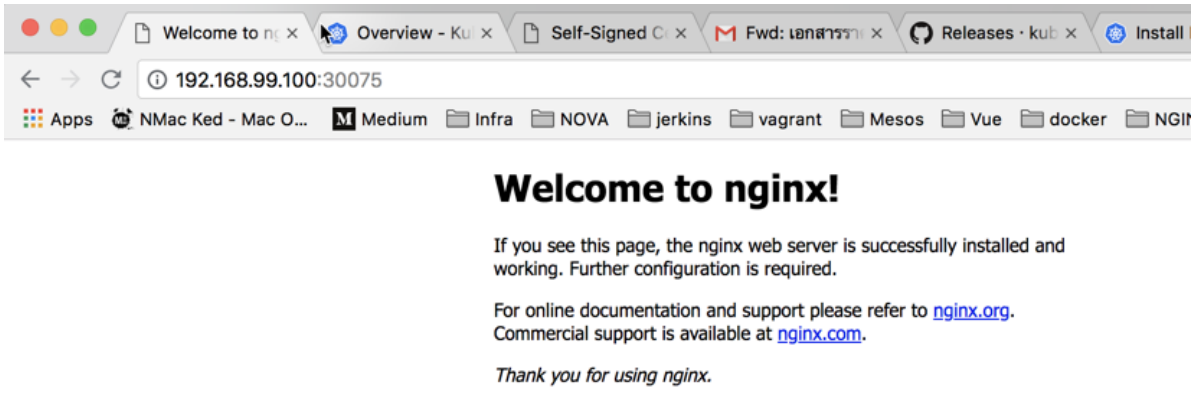
12. Test deployment "nginx" web server by command:
    a. kubectl run webtest --image=labdocker/nginx:latest --port=80 ➔ deployment nginx (image: labdocker/nginx:latest) with port 80 service
    b. kubectl expose deployment webtest --target-port=80 --type=NodePort ➔ expose pods with service 80 (http)

```
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
praparns-MacBook-Pro%
```

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   10.102.248.172   <none>        80:30075/TCP   27s
praparns-MacBook-Pro% kubectl describe svc webtest
Name:                     webtest
Namespace:                default
Labels:                   run=webtest
Annotations:              <none>
Selector:                 run=webtest
Type:                     NodePort
IP:                       10.102.248.172
Port:                     <unset>  80/TCP
TargetPort:               80/TCP
NodePort:                 <unset>  30075/TCP
Endpoints:                <none>
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://192.168.99.101:30075)



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$
```