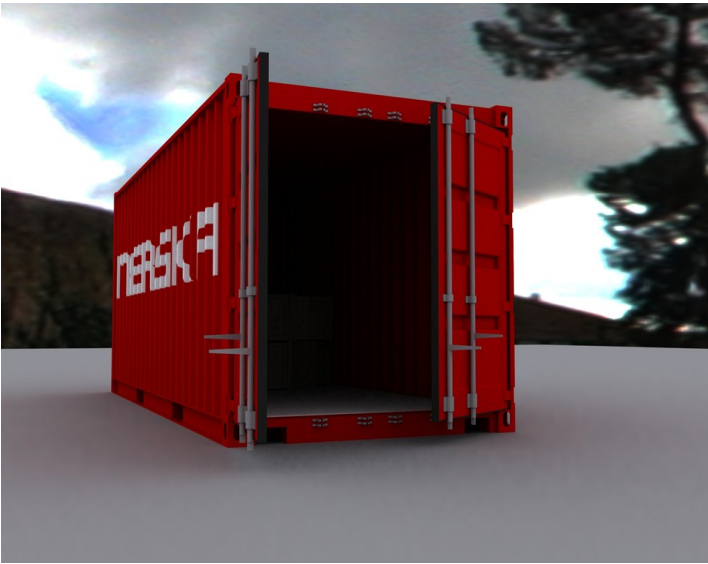
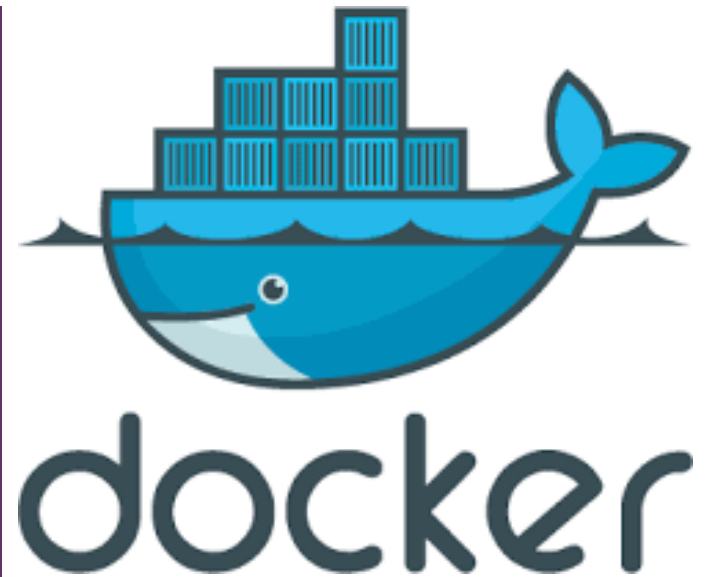


# Manual

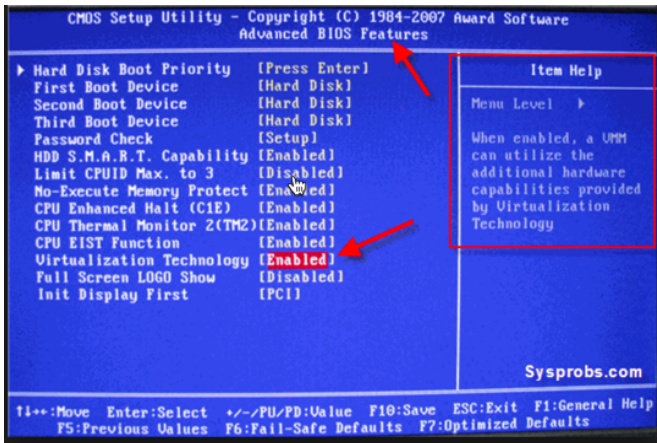


Install Docker Toolbox for  
Windows

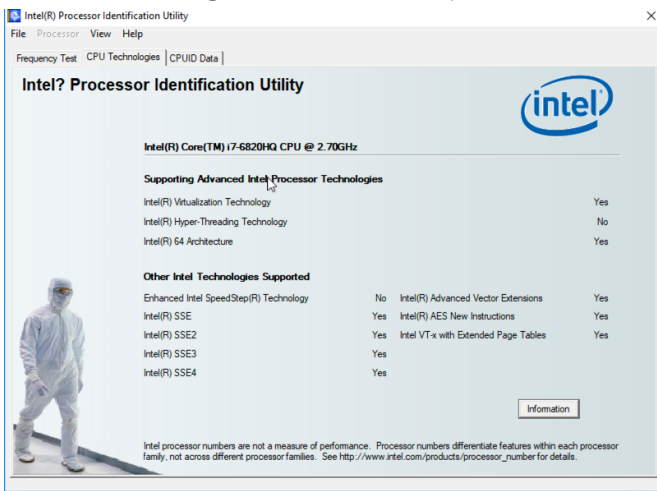
# Prerequisite

Enable Virtualization Technology (vt-x/AMD-v) on BIOS:

Normally virtualization technology should be enable by default.

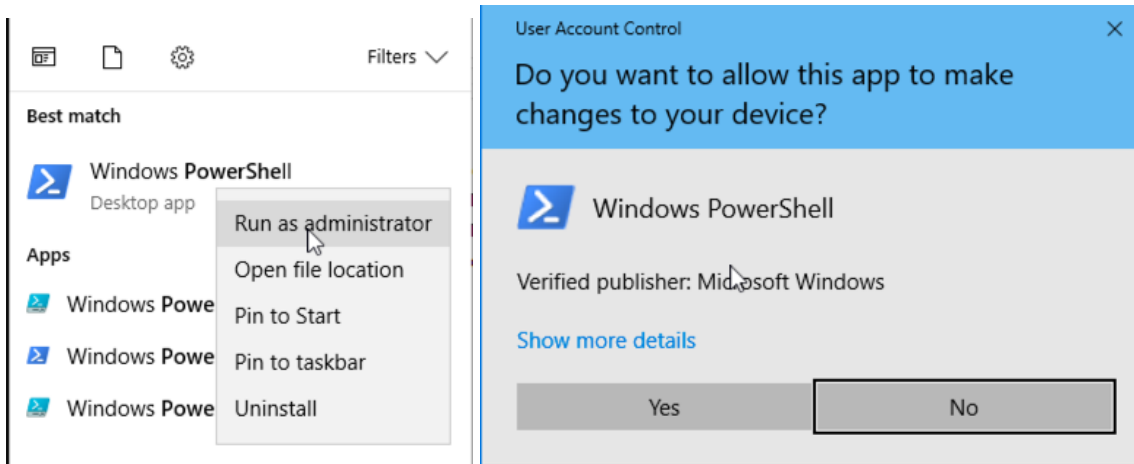


## Checking Feature via Utility Tools

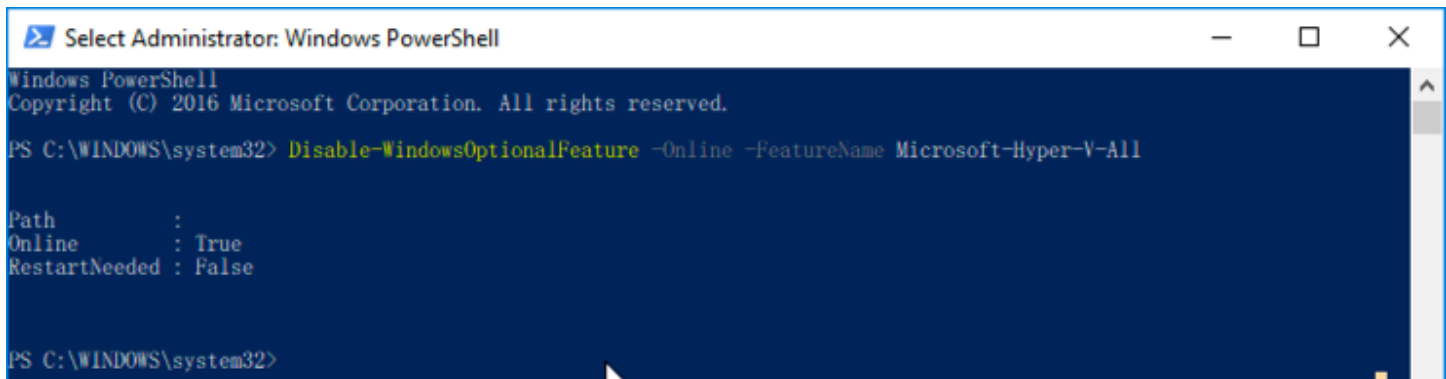


# Install Docker Toolbox

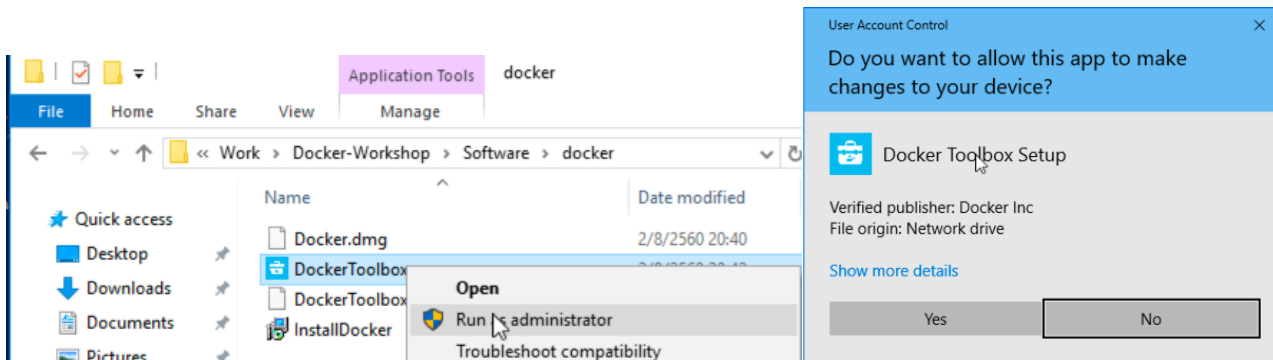
1. Open PowerShell by "select windows powershell" and run as administrator



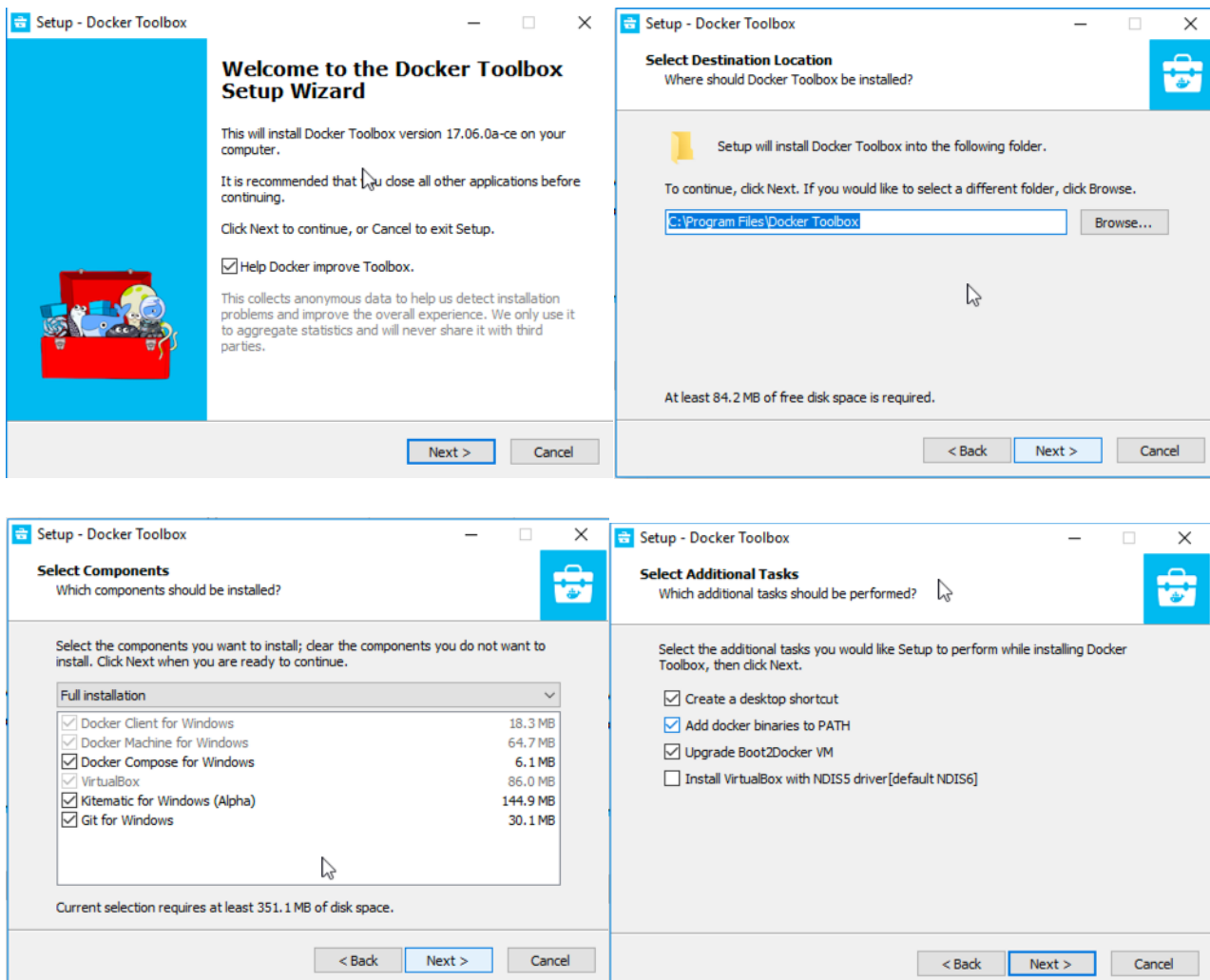
2. Run command "Disable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V-All"

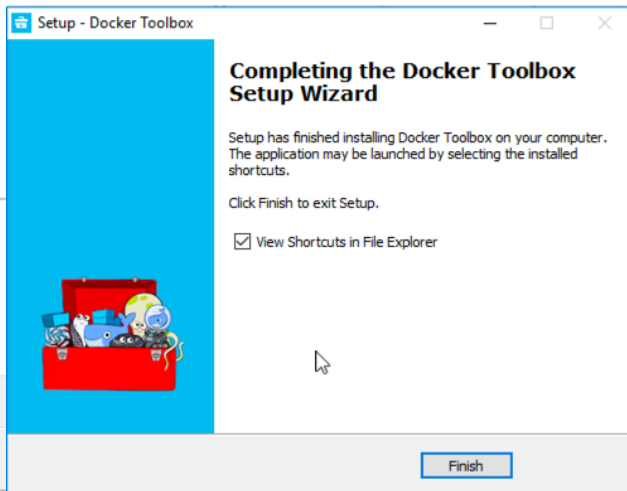
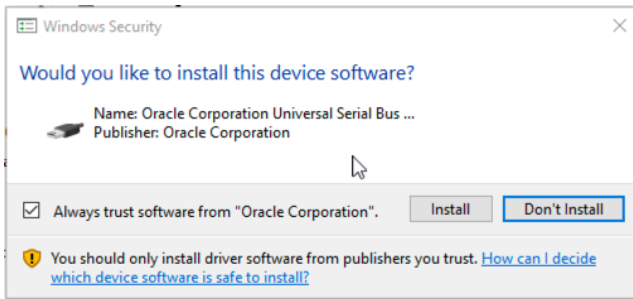
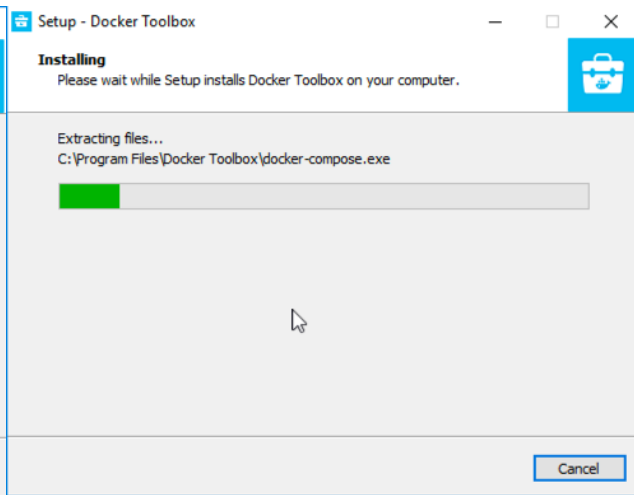
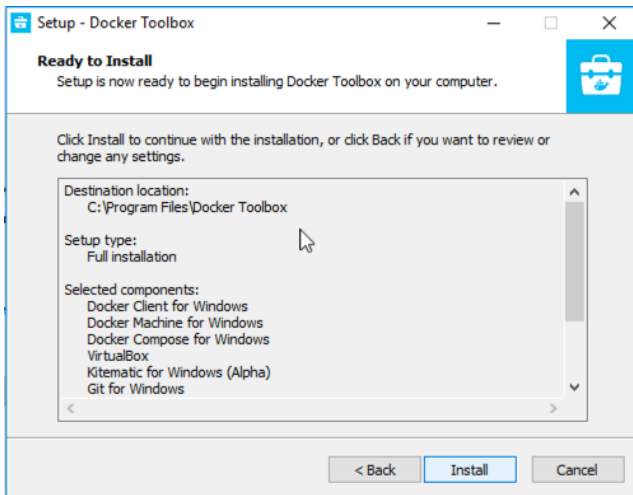


3. Right Click and "Run as administrator" on "DockerToolbox" and confirm "yes" on user account control.



4. Following screen for setup





5. Open new powershell (Run as Administrator) and check version of docker by "docker version"

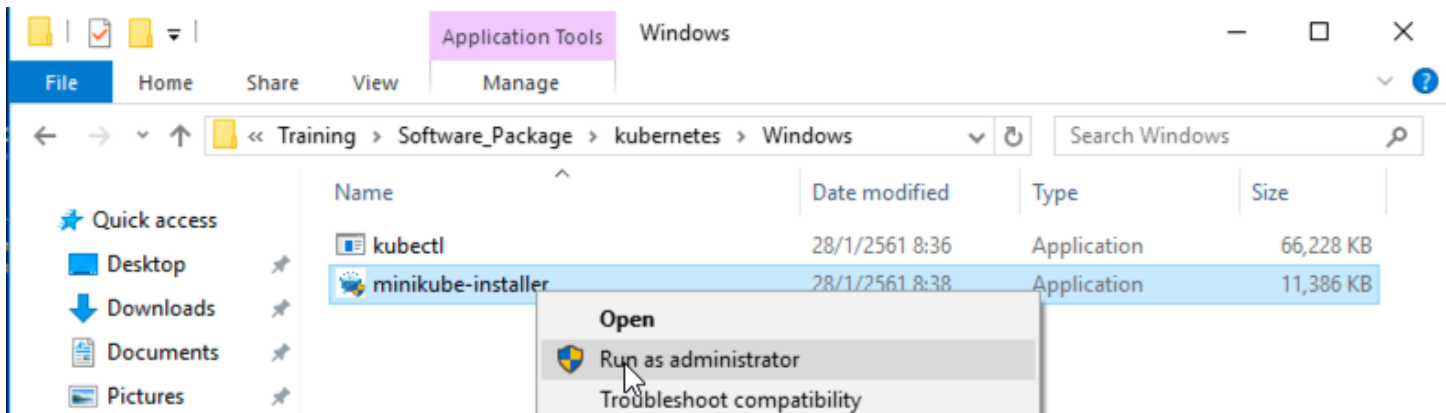


```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

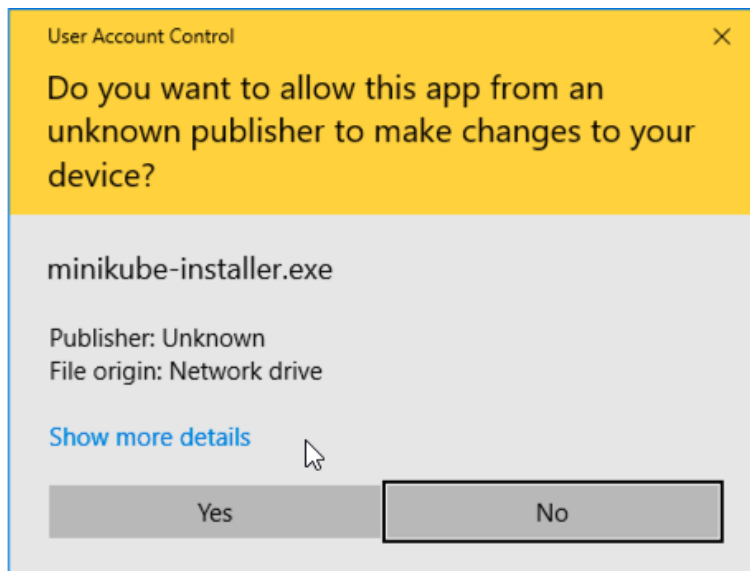
PS C:\WINDOWS\system32> docker version
Client:
 Version:      17.10.0-ce
 API version:  1.33
 Go version:   gol.8.3
 Git commit:   f4ffd25
 Built:        Tue Oct 17 19:00:02 2017
 OS/Arch:      windows/amd64
error during connect: Get http://%2F%2F.%2Fpipe%2Fdocker_engine/v1.33/version: open //.pipe/docker_engine: The system cannot find the file specified. In the default daemon configuration on Windows, the docker client must be run elevated to connect. This error may also indicate that the docker daemon is not running.
PS C:\WINDOWS\system32>
```

# Install minikube

1. Right click on "minikube-installer" and choose "Run as administrator"

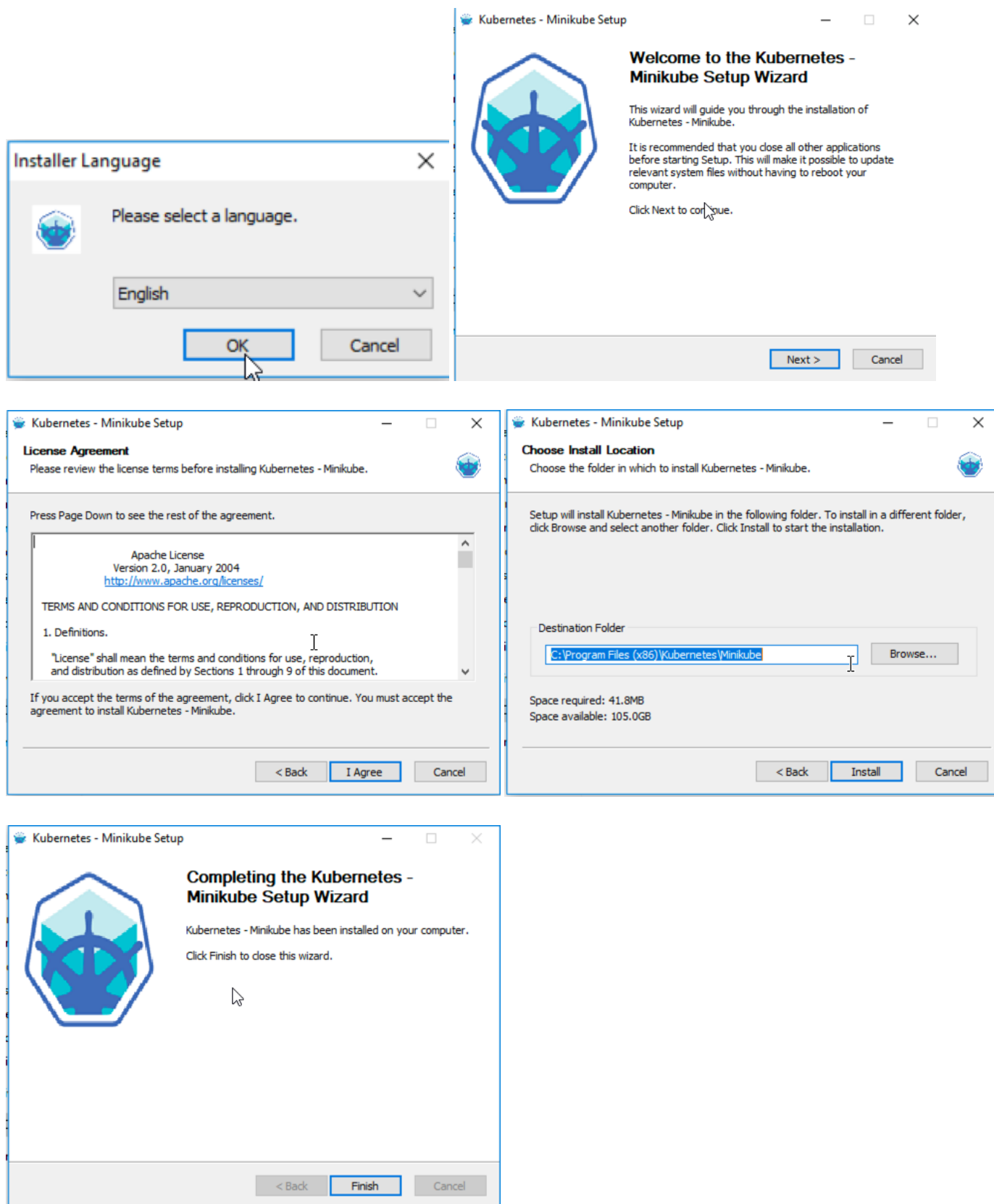


2. Click "Yes" for continue





3. Follow screen for continue installation





4. Test minikube command on powershell (Open new windows): minikube

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\WINDOWS\system32> minikube
Minikube is a CLI tool that provisions and manages single-node Kubernetes clusters optimized for development workflows.

Usage:
  minikube [command]

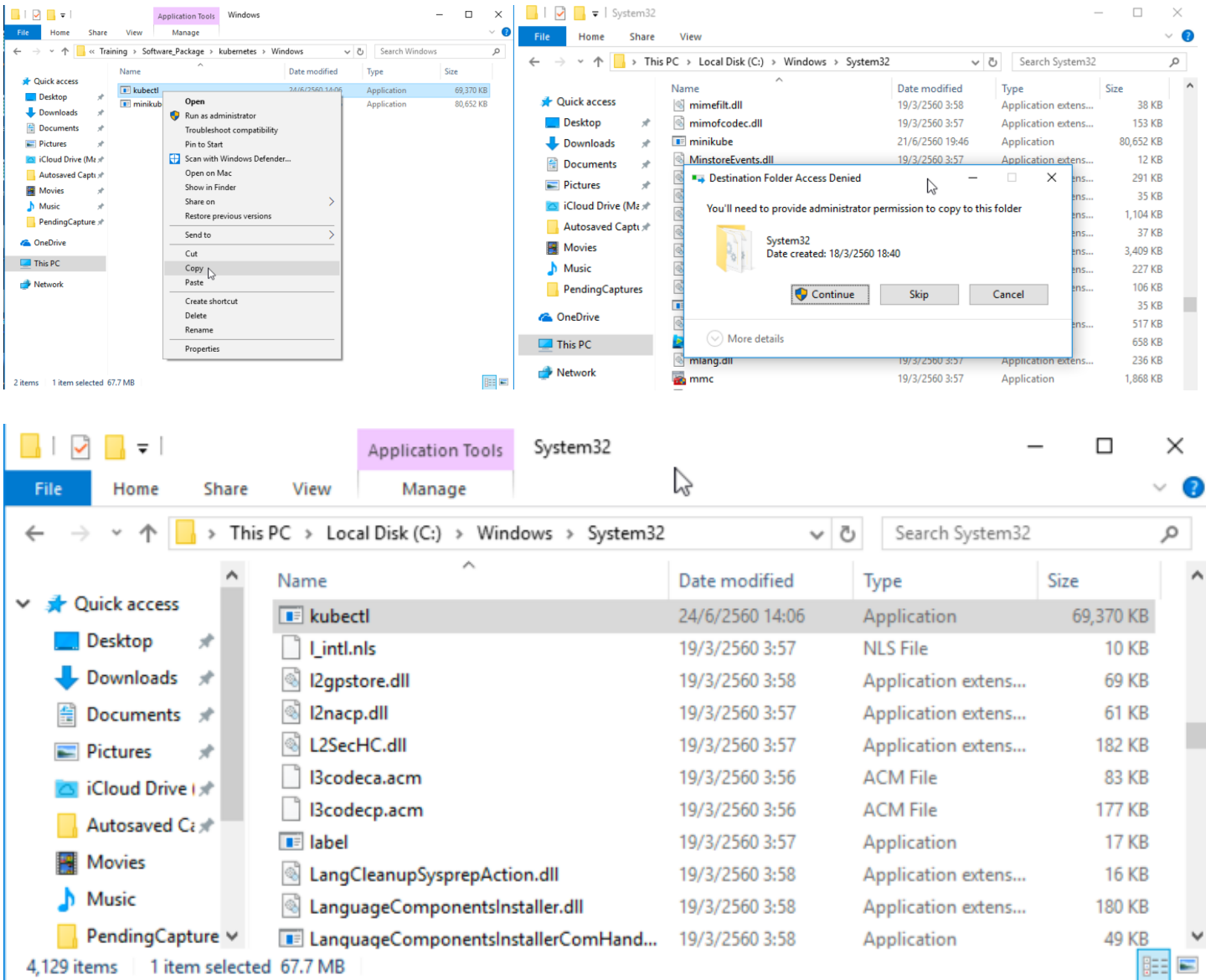
Available Commands:
  addons      Modify minikube's kubernetes addons
  cache       Add or delete an image from the local cache.
  completion  Outputs minikube shell completion for the given shell (bash or zsh)
  config      Modify minikube config
  dashboard   Opens/displays the kubernetes dashboard 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 Kubernetes versions available for minikube when using the localkube bootstrapper
  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-check  Print current and latest version number
  update-context  Verify the IP address of the running cluster in kubeconfig.
  version     Print the version of minikube

Flags:
  --alsologtostderr    log to standard error as well as files
  -b, --bootstrapper string  The name of the cluster bootstrapper that will set up the kubernetes cluster. (
default "localkube")
  -h, --help            help for minikube
  --log_backtrace_at traceLocation  when logging hits line file:N, emit a stack trace (default :0)
  --log_dir string      If non-empty, write log files in this directory
  --logtostderr         log to standard error instead of files
  -p, --profile string  The name of the minikube VM being used.
  This can be modified to allow for multiple minikube instances to be run independently (default "minikube")
  --stderrthreshold severity  logs at or above this threshold go to stderr (default 2)
  -v, --v Level         log level for V logs
  --vmodule moduleSpec  comma-separated list of pattern=N settings for file-filtered logging

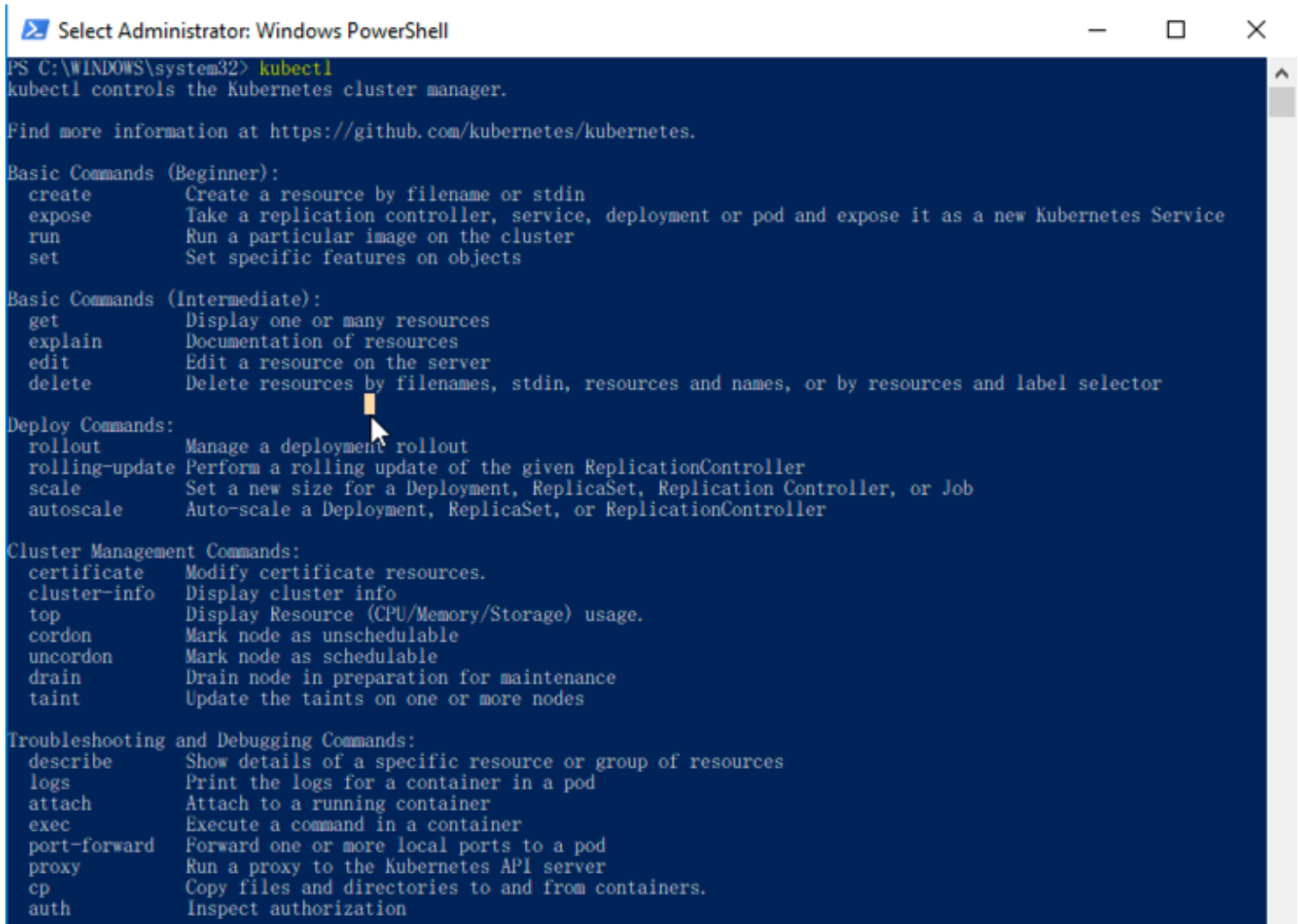
Use "minikube [command] --help" for more information about a command.
PS C:\WINDOWS\system32>
```

# Install kubectl

1. Right Click and Copy "kubectl" to c:\windows\system32



2. Check kubectl active command from powershell



```
PS C:\WINDOWS\system32> 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

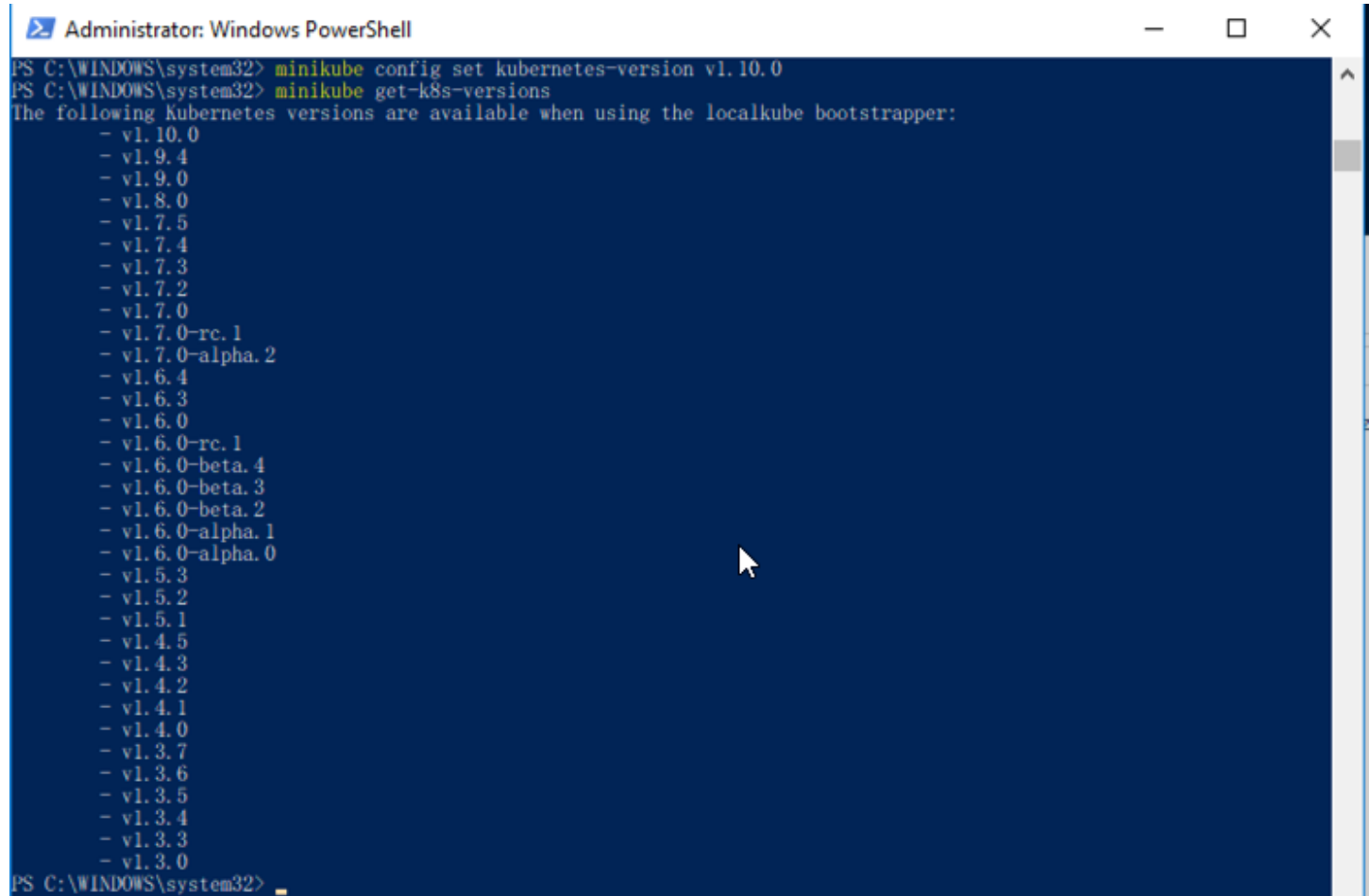
Deploy Commands:
  rollout     Manage a deployment rollout
  rolling-update Perform a rolling update of the given ReplicationController
  scale       Set a new size for a Deployment, ReplicaSet, Replication Controller, or Job
  autoscale   Auto-scale a Deployment, ReplicaSet, or ReplicationController

Cluster Management Commands:
  certificate  Modify certificate resources.
  cluster-info Display cluster info
  top          Display Resource (CPU/Memory/Storage) usage.
  cordon       Mark node as unschedulable
  uncordon     Mark node as schedulable
  drain        Drain node in preparation for maintenance
  taint        Update the taints on one or more nodes

Troubleshooting and Debugging Commands:
  describe    Show details of a specific resource or group of resources
  logs        Print the logs for a container in a pod
  attach      Attach to a running container
  exec        Execute a command in a container
  port-forward Forward one or more local ports to a pod
  proxy        Run a proxy to the Kubernetes API server
  cp          Copy files and directories to and from containers.
  auth        Inspect authorization
```

3. Configure minikube for use kubernetes version 1.9.0 on Powershell by command:

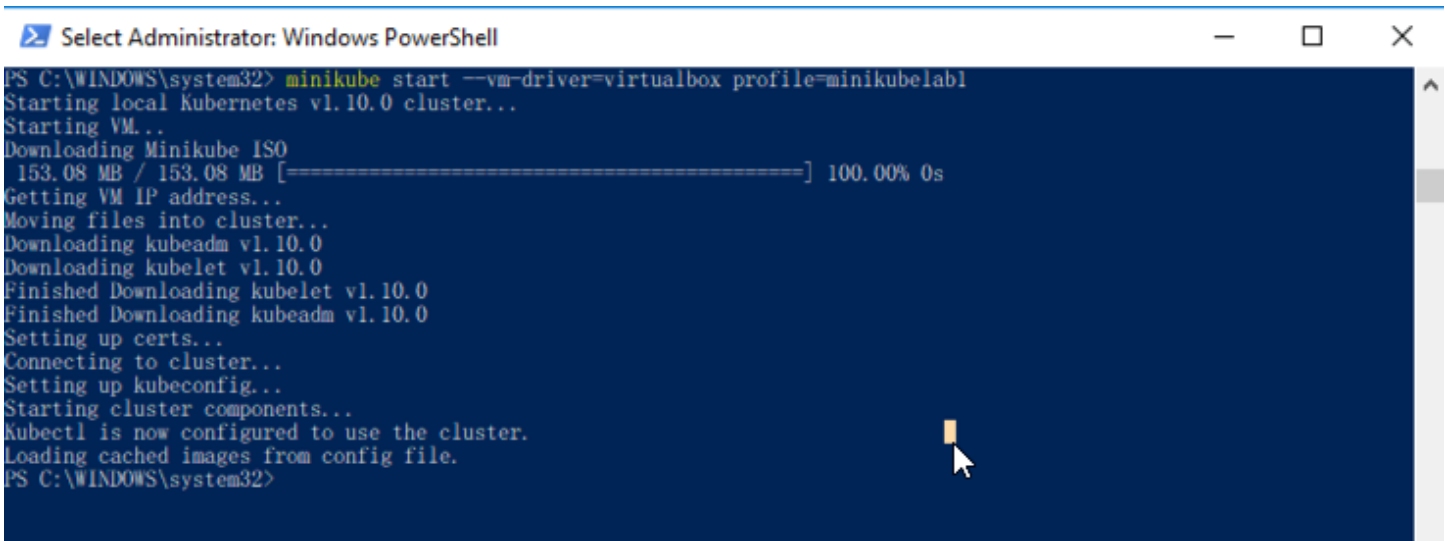
"minikube config set kubernetes-version v1.9.0"



```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> minikube config set kubernetes-version v1.10.0
PS C:\WINDOWS\system32> 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
PS C:\WINDOWS\system32>
```

4. Create minikube virtual machine on Powershell by command:

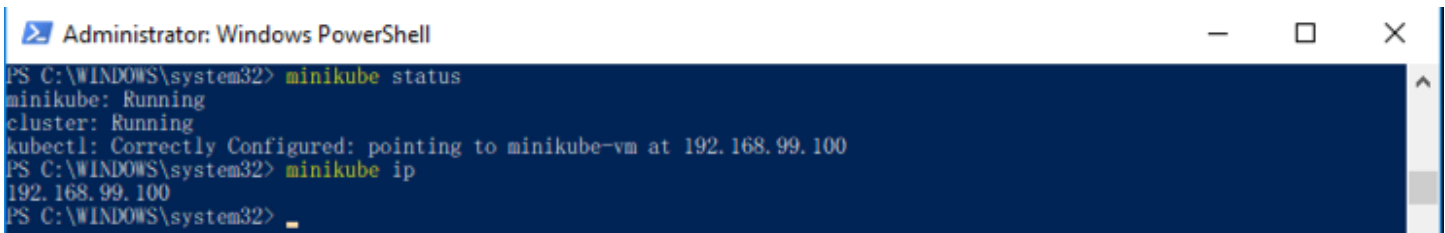
"minikube start --vm-driver=virtualbox profile=minikubelab1"



```
Select Administrator: Windows PowerShell

PS C:\WINDOWS\system32> minikube start --vm-driver=virtualbox profile=minikubelab1
Starting local Kubernetes v1.10.0 cluster...
Starting VM...
Downloading Minikube ISO
 153.08 MB / 153.08 MB [=====] 100.00% 0s
Getting VM IP address...
Moving files into cluster...
Downloading kubeadm v1.10.0
Downloading kubelet v1.10.0
Finished Downloading kubelet v1.10.0
Finished Downloading kubeadm v1.10.0
Setting 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.
PS C:\WINDOWS\system32>
```

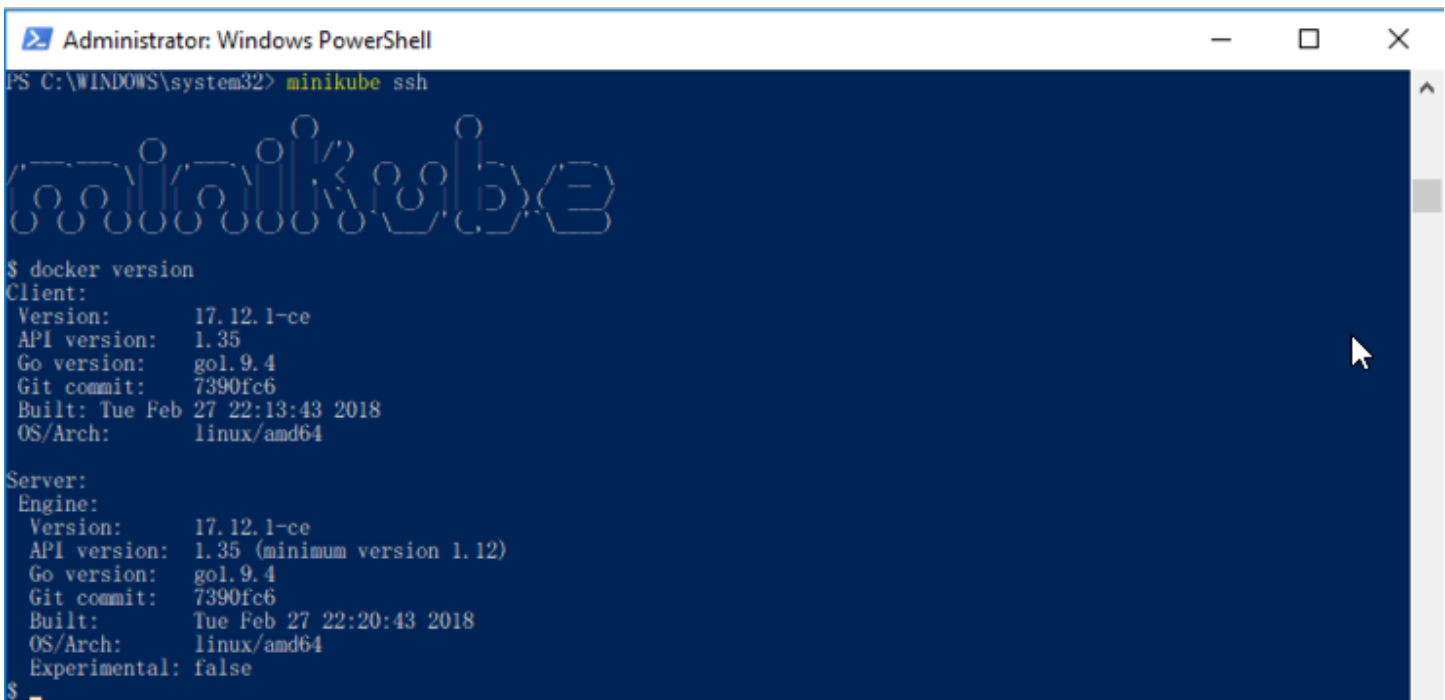
5. Check status of minikube's machine by command: "minikube status", "minikube ip"



```
Administrator: Windows PowerShell

PS C:\WINDOWS\system32> minikube status
minikube: Running
cluster: Running
kubectl: Correctly Configured: pointing to minikube-vm at 192.168.99.100
PS C:\WINDOWS\system32> minikube ip
192.168.99.100
PS C:\WINDOWS\system32>
```

6. Test ssh to minikube's machine by command: minikube ssh, docker version



```
Administrator: Windows PowerShell

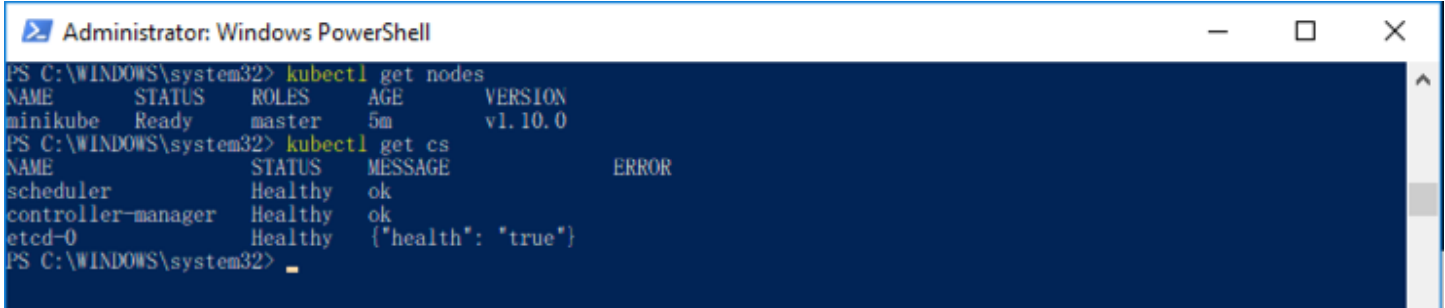
PS C:\WINDOWS\system32> minikube ssh

minikube

$ docker version
Client:
 Version:      17.12.1-ce
 API version:  1.35
 Go version:   gol.9.4
 Git commit:   7390fc6
 Built: Tue Feb 27 22:13:43 2018
 OS/Arch:     linux/amd64

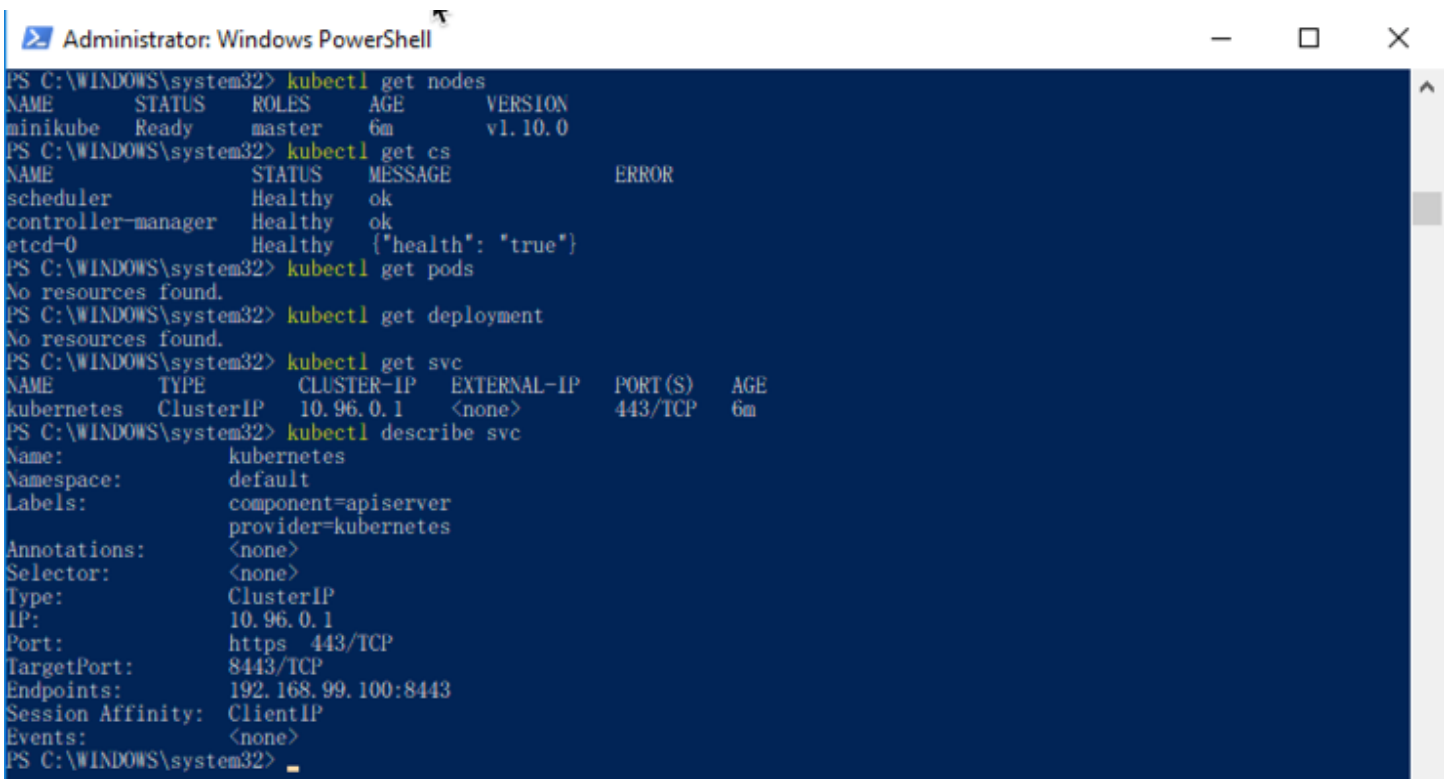
Server:
 Engine:
  Version:      17.12.1-ce
  API version:  1.35 (minimum version 1.12)
  Go version:   gol.9.4
  Git commit:   7390fc6
  Built: Tue Feb 27 22:20:43 2018
  OS/Arch:     linux/amd64
  Experimental: false
$
```

7. Check health of kubernetes cluster by command
  - a. `kubectl get nodes` → check node status
  - b. `kubectl get cs` → check cluster status



```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl get nodes
NAME        STATUS    ROLES    AGE      VERSION
minikube    Ready    master   5m       v1.10.0
PS C:\WINDOWS\system32> kubectl get cs
NAME                STATUS    MESSAGE           ERROR
scheduler           Healthy   ok                   
controller-manager   Healthy   ok                   
etcd-0              Healthy   {"health": "true"}
```

8. Check status of kubernetes'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 kubernetes
  - d. `kubectl describe svc` → check service description on kubernetes



```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl get nodes
NAME        STATUS    ROLES    AGE      VERSION
minikube    Ready    master   6m       v1.10.0
PS C:\WINDOWS\system32> kubectl get cs
NAME                STATUS    MESSAGE           ERROR
scheduler           Healthy   ok                   
controller-manager   Healthy   ok                   
etcd-0              Healthy   {"health": "true"}
PS C:\WINDOWS\system32> kubectl get pods
No resources found.
PS C:\WINDOWS\system32> kubectl get deployment
No resources found.
PS C:\WINDOWS\system32> kubectl get svc
NAME         TYPE        CLUSTER-IP    EXTERNAL-IP  PORT(S)    AGE
kubernetes   ClusterIP   10.96.0.1     <none>       443/TCP    6m
PS C:\WINDOWS\system32> 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:    192.168.99.100:8443
Session Affinity: ClientIP
Events:       <none>
```



9. 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)

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl run webtest --image=labdocker/nginx:latest --port=80
deployment.apps/webtest created
PS C:\WINDOWS\system32> kubectl expose deployment webtest --target-port=80 --type=NodePort
service/webtest exposed
PS C:\WINDOWS\system32>
```

10. 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 (Check port for access. This example is "3211

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl get svc webtest
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
webtest   NodePort  10.102.255.183   <none>           80:31355/TCP  45s
PS C:\WINDOWS\system32> kubectl describe svc webtest
Name:      webtest
Namespace: default
Labels:    run=webtest
Annotations: <none>
Selector:  run=webtest
Type:      NodePort
IP:        10.102.255.183
Port:      <unset> 80/TCP
TargetPort: 80/TCP
NodePort:   <unset> 31355/TCP
Endpoints:  <none>
Session Affinity: None
External Traffic Policy: Cluster
Events:     <none>
PS C:\WINDOWS\system32>
```

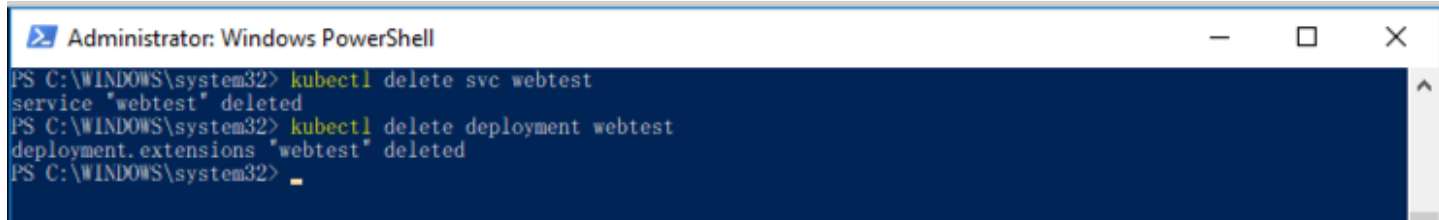
11. Open browser for access test (Default IP: 192.168.99.100)





12. Stop deployment by command and recheck again

- a. `kubectl delete svc webtest`
- b. `kubectl delete deployment webtest`



```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl delete svc webtest
service "webtest" deleted
PS C:\WINDOWS\system32> kubectl delete deployment webtest
deployment.extensions "webtest" deleted
PS C:\WINDOWS\system32> _
```