KUBERNETES - LAB GUIDE

PREMKUMAR NATARAJAN

Table of Contents

1 Purpose

This document will lay out the details to setup the Kubernetes in window 7 64 bit OS environment and practice the Lab exercises.

2 Prerequisites

Kubernetes exercise are performed on Windows 7 Professional OS using minikube tool

2.1 Software Download

S.No	Software	Download Location	File Name	Version
1	Minikube	https://github.com/kubernetes/minikube/releases/tag/v0.25.0	minikube-windows-amd64	Windows 64 Bit
2	Kubectl	https://storage.googleapis.com/kubernetes-release/release/v1.11.0/bin/windows/amd64/kubectl.exe	Kubectl.exe	Kubectl CLI

3 LAB-01: Install Minikube in Windows

This LAB exercise shows you how to install Minikube.

Minikube is an open source tool that was developed to enable developers and system administrators to run a single cluster of Kubernetes on their local machine. Minikube starts a single node kubernetes cluster locally with small resource utilization.

3.1 Time to Complete

Approximately 0.30 Hr.

3.2 Environment

1. Windows 7 Professional

3.3 What You Need

3.3.1 Pre-setup

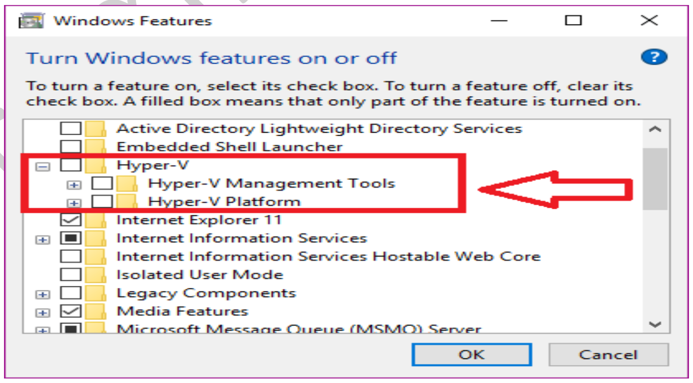
1. BIOS VT-x enabled for Virtual Machines to come up
2. Turn off the hyperv in windows

   <u>Using VirtualBox and not Hyper-V</u>

   VirtualBox and Hyperv (which is available on Windows 10) do not make a happy pair and you are bound to run into situations where the tools get confused. I preferred to use VirtualBox and avoid all esoteric command-line switches that we need to provide to enable creation of the underlying Docker hosts, etc.

   To disable Hyper-V, go to Turn Windows features on or off and you will see a dialog with list of Windows features as shown below. Navigate to the Hyper-V section and disable it completely.

   a. If hyperv is running
      i. Search for "turn windows feature on or off"
      ii. uncheck "Hyper-V"



Windows Features

Turn Windows features on or off

To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

- [ ] Active Directory Lightweight Directory Services
- [ ] Embedded Shell Launcher
- [ ] Hyper-V
  - [ ] Hyper-V Management Tools
  - [ ] Hyper-V Platform
- [x] Internet Explorer 11
- [■] Internet Information Services
- [ ] Internet Information Services Hostable Web Core
- [ ] Isolated User Mode
- [ ] Legacy Components
- [x] Media Features
- [■] Microsoft Message Queue (MSMQ) Server

3. Requires VirtualBox - If it is already installed then there is no need for a new setup. Else download and install from www.virtualbox.org

3.4 Minikube Installation

3.4.1 Minikube Software

1. Download minikube-windows-amd64 from https://github.com/kubernetes/minikube/releases/tag/v0.25.0
2. Rename the file to minikube.exe
3. Save file in the target directory (Ex: D:\Kubernetes\Minikube)
4. Add the Minikube directory in the PATH environment variable

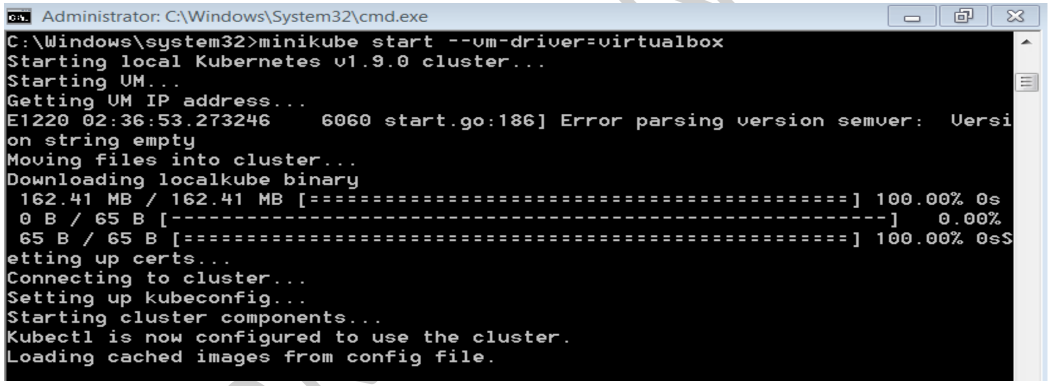
3.4.2 Kubectl Software

1. Download kubectl from this location https://storage.googleapis.com/kubernetes-release/release/v1.11.0/bin/windows/amd64/kubectl.exe
2. Save file in the target directory (Ex: D:\Kubernetes\Minikube)

3.4.3 Launch Kubernetes Cluster locally - Minikube start

Note: You might run into multiple issues while starting a cluster the first time. I have several of them and have created a section in this guide for Troubleshooting. Take a look at it, in case you run into any issues.

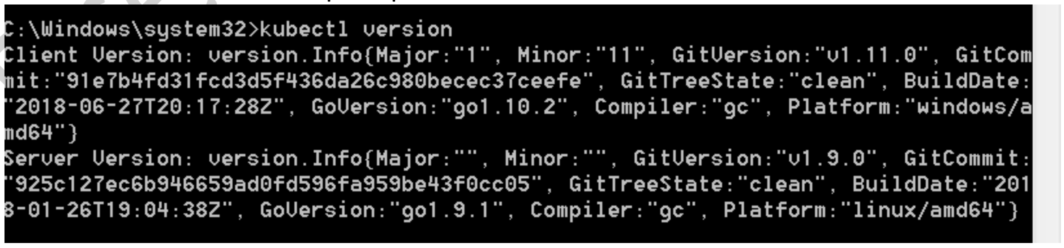
1. Open a command prompt as administrator
2. From the command prompt on windows execute "**minikube start --vm-driver=virtualbox**"
   Note: Run this command from C drive.



```
Administrator: C:\Windows\System32\cmd.exe
C:\Windows\system32>minikube start --vm-driver=virtualbox
Starting local Kubernetes v1.9.0 cluster...
Starting VM...
Getting VM IP address...
E1220 02:36:53.273246    6060 start.go:186] Error parsing version semver:  Versi
on string empty
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% 0sS
etting 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.
```

3.4.4 Kubernetes Client and Server version

1. Open a command prompt as administrator
2. From the above command prompt execute "**kubectl version**"



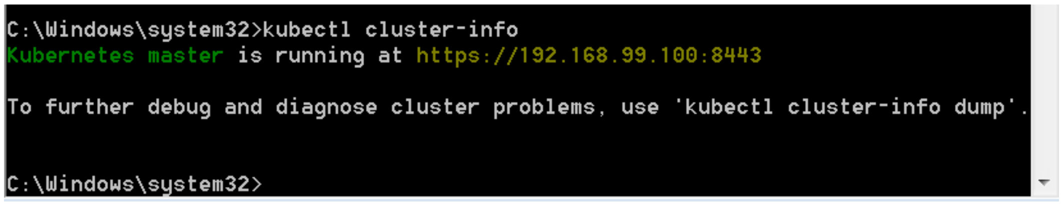
```
C:\Windows\system32>kubectl version
Client Version: version.Info{Major:"1", Minor:"11", GitVersion:"v1.11.0", GitCom
mit:"91e7b4fd31fcd3d5f436da26c980becec37ceefe", GitTreeState:"clean", BuildDate:
"2018-06-27T20:17:28Z", GoVersion:"go1.10.2", Compiler:"gc", Platform:"windows/a
md64"}
Server Version: version.Info{Major:"", Minor:"", GitVersion:"v1.9.0", GitCommit:
"925c127ec6b946659ad0fd596fa959be43f0cc05", GitTreeState:"clean", BuildDate:"201
8-01-26T19:04:38Z", GoVersion:"go1.9.1", Compiler:"gc", Platform:"linux/amd64"}
```

Output will show both client and server versions

3.4.5 Minikube commands version

1. Check the status of minikube using command "**minikube status**"

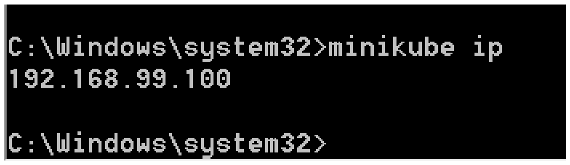
2. From the above command prompt execute "**kubectl version**"
3. Use the kubectl CLI to get the cluster information: “**kubectl cluster-info**”



```
C:\Windows\system32>kubectl cluster-info
Kubernetes master is running at https://192.168.99.100:8443

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.


C:\Windows\system32>
```

3.4.6 Cluster IP Address

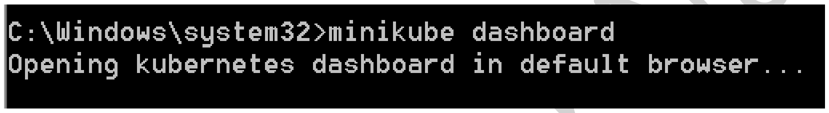
1. Get the IP address of the cluster via the ip command “**minikube ip**”



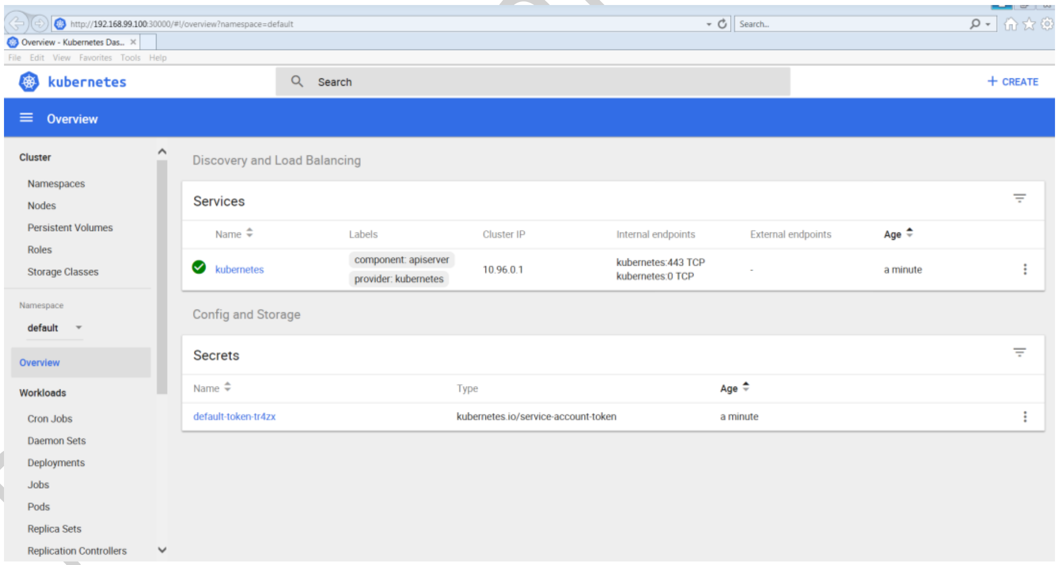
```
C:\Windows\system32>minikube ip
192.168.99.100

C:\Windows\system32>
```

3.4.7 Kubernetes Dashboard

1. Launch the Kubernetes Dashboard at any point via the dashboard command as shown below: “**minikube dashboard”**



```
C:\Windows\system32>minikube dashboard
Opening kubernetes dashboard in default browser...
```



Services

Name	Labels	Cluster IP	Internal endpoints	External endpoints	Age
✓ kubernetes	component: apiserver provider: kubernetes	10.96.0.1	kubernetes:443 TCP kubernetes:0 TCP	-	a minute

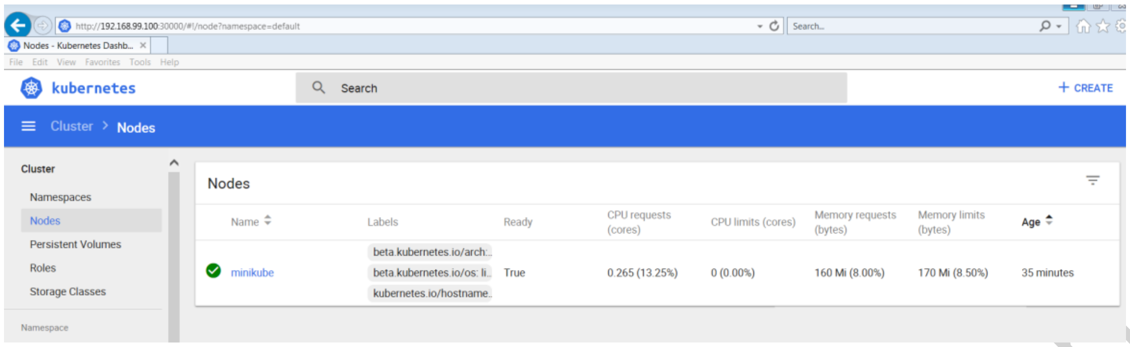
Secrets

Name	Type	Age
default-token-tr4zx	kubernetes.io/service-account-token	a minute

This will automatically launch the Dashboard in your local browser.

However if you just want to nab the Dashboard URL, you can use the following flag: “**minikube.exe dashboard --url=true** http://192.168.99.100:30000”

Click on the Node link and you will see that information:



Nodes

	Name	Labels	Ready	CPU requests (cores)	CPU limits (cores)	Memory requests (bytes)	Memory limits (bytes)	Age
✓	minikube	beta.kubernetes.io/arch:... beta.kubernetes.io/os: li... kubernetes.io/hostname...	True	0.265 (13.25%)	0 (0.00%)	160 Mi (8.00%)	170 Mi (8.50%)	35 minutes

The above node information can also be obtained by using the kubectl CLI to get the list of nodes. "**kubectl get nodes**"

```
C:\Windows\system32>kubectl get nodes
NAME       STATUS    ROLES     AGE       VERSION
minikube   Ready     <none>    38m       v1.9.0
```

3.4.8 Stop the Kubernetes Cluster

1. Run the below command to stop the Kubernetes cluster using command "**minikube stop**"

```
C:\Windows\system32>minikube stop
Stopping local Kubernetes cluster...
Machine stopped.
```

2. Check the status of cluster using minikube command "**minikube status**"

```
C:\Windows\system32>minikube status
minikube: Stopped
cluster:
kubectl:
```

<<<<<End of Lab>>>>>

4 LAB-02: Kubectl Commands

This LAB exercise is to practice the kubectl commands to work with Kubernetes deployment

4.1 Time to Complete

Approximately 0.30 Hr.

4.2 What You Need

1. Lab 01 to completed successfully.
2. Kubernetes Cluster should be running. If not start the cluster using the command "minikube start"

4.3 Kubectl Commands

1. Open a command prompt as administrator.
2. Check the Kubernetes cluster status using **minikube status** command.
3. If the cluster is not running, start the cluster using command **minikube start**.
4. Show merged kube config settings using the command **kubectl config view**.
5. Start a single instance of nginx using the command **kubectl run nginx --image=nginx**
6. Get the POD documents using the command **kubectl explain pods**
7. View and find resources using below commands

   - # List all services in the namespaces
     kubectl get services

   - # List all pods in all namespaces
     kubectl get pods --all-namespaces

   - # List all pods in the namespace, with more details
     kubectl get pods -o wide

   - #List a particular deployment
     kubectl get deployment nginx

8. Print the supported versions of API on the cluster using the command **kubectl api-versions**
9. Displays the cluster Info using the command **kubectl cluster-info**
10. Display the current context of the cluster using the command : **kubectl config current-context**
11. Describes any particular resource in kubernetes using the command: **kubectl describe pod nginx-8586cf59-kmtx9**
12. Execute a command in the container using the command : **kubectl exec nginx-8586cf59-kmtx9 ls**
13. Run command to run an image on the Kubernetes cluster. **kubectl run -i -t busybox --image=busybox --restart=Never**

<<<<<End of Lab>>>>>

5 LAB-03: Deploy Application Install Minikube in Windows

This LAB exercise shows you how to deploy a sample application in kubernetes.

5.1 Time to Complete

Approximately 0.30 Hr.

5.2 What You Need

1. Lab 01 to completed successfully.
2. Kubernetes Cluster should be running. If not start the cluster using the command “minikube start”

5.3 Deploy the Application

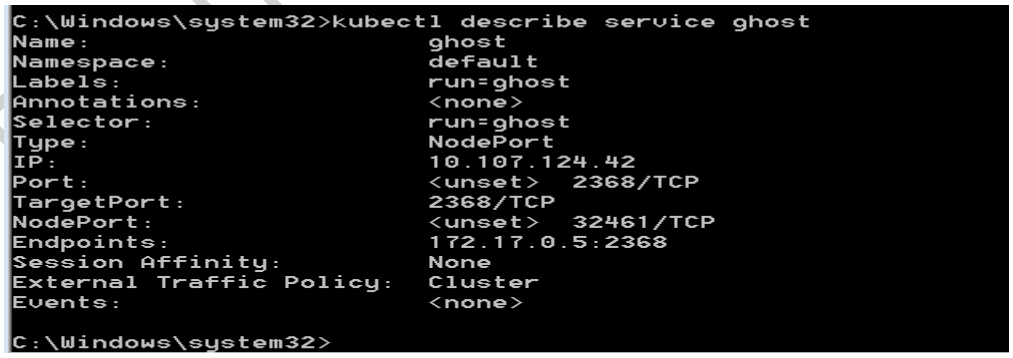
With Kubernetes cluster ready, start deploying application containers. The application container deploying will be an instance of Ghost.

Ghost is a popular JavaScript-based blogging platform, and with its official Docker image.

1. Open the command prompt as administrator.
2. Use the below command to start Ghost container.
   kubectl run ghost --image=ghost --port=2368
3. Verify that the container is running using the below command.
   kubectl get pods
4. To make **Ghost** application accessible outside the cluster, the deployment just created needs to be exposed as a Kubernetes Service.
   kubectl expose deployment ghost --type="NodePort"

   - **NodePort** service type will set all nodes to listen on the specified port.
   - **ClusterIP** is to only expose service to other Pods within this cluster
   - **LoadBalancer** service type is designed to provision an external IP to act as a Load Balancer for the service.

5. To get the port assigned, use the kubectl command, with the describe service option.
   kubectl describe service ghost



```
C:\Windows\system32>kubectl describe service ghost
Name:                     ghost
Namespace:                default
Labels:                   run=ghost
Annotations:              <none>
Selector:                 run=ghost
Type:                     NodePort
IP:                       10.107.124.42
Port:                     <unset>  2368/TCP
TargetPort:               2368/TCP
NodePort:                 <unset>  32461/TCP
Endpoints:                172.17.0.5:2368
Session Affinity:         None
External Traffic Policy:  Cluster
Events:                   <none>

C:\Windows\system32>
```

6. Scale the deployment using the below command :
   kubectl scale deployment ghost --replicas=4

7. Get the status of the deployment using the below command :
   kubectl get deployment
8. Open the dashboard using the command **minikube dashboard**
9. From Dashboard, go to the Services section, check services entry.
10. Open the service using the command : **minikube service ghost**

<<<<<End of Lab>>>>>

6 LAB-04: Deploy Single Container POD in Kubernetes

This LAB exercise shows you how to deploy a sample application in kubernetes.

6.1 Time to Complete

Approximately 0.30 Hr.

6.2 What You Need

1. Lab 01 to completed successfully.
2. Kubernetes Cluster should be running. If not start the cluster using the command “minikube start”

6.3 Deploy the Application

1. Save this below **db-pod.yml** file in local storage.

```
apiVersion: "v1"
kind: Pod
metadata:
  name: mysql
  labels:
    name: mysql
    app: demo
spec:
  containers:
    - name: mysql
      image: mysql:latest
      ports:
        - containerPort: 3306
          protocol: TCP
      env:
        -
          name: "MYSQL_ROOT_PASSWORD"
          value: "password"
```

2. Create a POD with single conatiner using the command :
   C:\Windows\system32>kubectl create -f D:\Kubernetes\Labs\Pods\Single-Container\db-pod.yml
   pod/mysql created
3. Check the PODs using the below command
   C:\Windows\system32>kubectl get pods
4. Login to the Kubernetes UI and analyze the POD
5. Get the complete details of POD using below command/
   kubectl describe pod mysql

<<<<<End of Lab>>>>>

7 LAB-05: Deploy multiple POD and communication between POD in Kubernetes

This LAB exercise demonstrate the concept of packaging containers into a pod and communication between pods.

7.1 Time to Complete

Approximately 0.30 Hr.

7.2 What You Need

1. Lab 01 to completed successfully.
2. Kubernetes Cluster should be running. If not start the cluster using the command "minikube start"

7.3 Build a Docker image and push it to docker hub

1. Login to the Docker machine.
2. Copy all the file from the Folder PODLab from git Location : https://github.com/premkumarmlp/KubernetesExercises.git
3. Edit the below files in the Docker folder as per your docker hub username.
   - **build.sh**
   - **docker-compose.yml**

   Ex: Replace the username/repository from premkumarmlp/web to your username/<repository> in docker hub.
4. From the docker directory, build the image using below commnd.
```
dockeruser@dockeruser-VirtualBox:~/K8sPODLab/Docker$ docker build -t
<DOCKER_HUB_USERNAME>/web .
```
5. Once the build is successful, push the image to your docker hub.
```
dockeruser@dockeruser-VirtualBox:~/K8sPODLab/Docker$ docker push
<DOCKER_HUB_USERNAME>/web
```
6. To check the build image is working properly, use the docker compose file to create the container using below command.
```
dockeruser@dockeruser-VirtualBox:~/K8sPODLab/Docker$ docker-compose up -d
```
7. Check the container using below command.
```
dockeruser@dockeruser-VirtualBox:~/K8sPODLab/Docker$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
PORTS               NAMES
5f8e8f025b66        premkumarmlp/web    "python app.py"     27 seconds ago      Up 24 seconds
0.0.0.0:3000->5000/tcp   docker_web_1
```
8. Test the application using the url http://localhost:3000/ and the output will be as :
   Hello Container World! I have been seen 1 times.
   Now you build image is ready to deploy in Kubernetes.

7.4 Deploy the Application in Kubernetes

1. Login to the kubernetes machine
2. Copy all the file from the Folder PODLab from git Location and store it locally. (example : D:\PODLab): https://github.com/premkumarmlp/KubernetesExercises.git
3. Edit the below files in the Kubernetes folder as per your docker hub username.
   web-pod.yml
   web-rc.yml
   Ex: Replace the username/repository from premkumarmlp/web to your username/<repository> in docker hub.
4. Open the command prompt as administrator and create POD service as below

```
C:\Windows\system32>kubectl create -f D:\PODLab\db-pod.yml
pod/redis created
C:\Windows\system32>kubectl create -f D:\PODLab\db-svc.yml
service/redis created
C:\Windows\system32>kubectl create -f D:\PODLab\web-pod.yml
pod/web created
C:\Windows\system32>kubectl create -f D:\PODLab\web-svc.yml
service/web created
C:\Windows\system32>kubectl create -f D:\PODLab\web-rc.yml
replicationcontroller/web created
```

5. Get the list of PODs and verify all are running.

```
C:\Windows\system32>kubectl get pods
NAME       READY   STATUS    RESTARTS   AGE
redis      1/1     Running   0          32s
web        1/1     Running   0          18s
web-v5x9l  1/1     Running   0          4s
```

6. Get the list of exposed services.

```
C:\Windows\system32>kubectl get svc

NAME         TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
kubernetes   ClusterIP   10.96.0.1        <none>        443/TCP        16h
redis        ClusterIP   10.102.49.56     <none>        6379/TCP       49s
web          NodePort    10.101.202.109   <none>        80:31017/TCP   33s
```

7. Get service url for web application using below command.

```
C:\Windows\system32>minikube service web --url
http://192.168.99.102:31017
```

8. Access the url from the browser and verify the result.
   Hello Container World! I have been seen 4 times.

<<<<<End of Lab>>>>>

8 LAB-06: Deployment in Kubernetes

This LAB exercise shows you how to use the service deployment and expose it.

8.1 Time to Complete

Approximately 0.30 Hr.

8.2 What You Need

1. Lab 01 to completed successfully.
2. Kubernetes Cluster should be running. If not start the cluster using the command "minikube start"

8.3 Deploy the Application

1. Save this below **deployment.yaml** file in local storage.

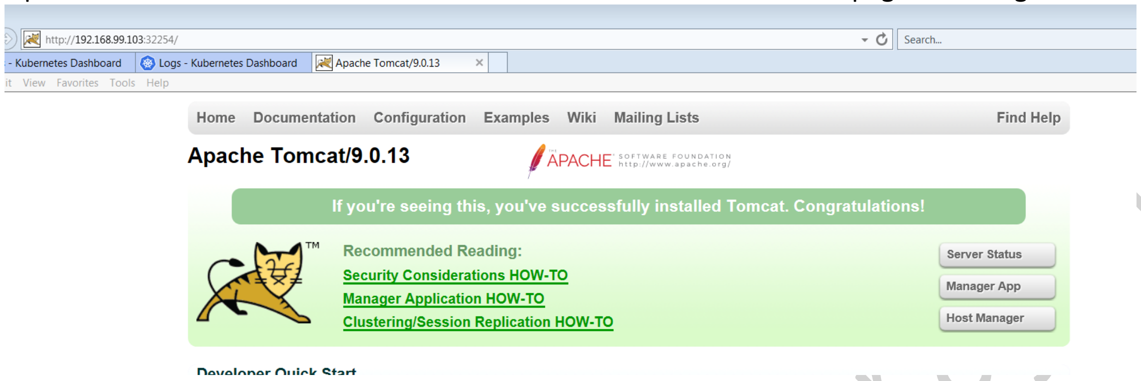
```
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: tomcat-deployment
spec:
  selector:
    matchLabels:
      app: tomcat
  replicas: 2
  template:
    metadata:
      labels:
        app: tomcat
    spec:
      containers:
      - name: tomcat
        image: tomcat:9.0
        ports:
        - containerPort: 8080
```

2. Create a POD service using the command :
   C:\Windows\system32>kubectl create -f D:\VM\Deployment\deployment.yaml
   deployment.apps/tomcat-deployment created

3. Check the PODs using the below command
   C:\Windows\system32>kubectl get pods
   Once the tomcat POD is running, then expose the service.

4. Expose the service using the command.
   C:\Windows\system32>kubectl expose deployment tomcat-deployment --type=NodePort

5. Get the minikube service url for this tomcat service using the command :
   C:\Windows\system32>minikube service tomcat-deployment --url
   http://192.168.99.102:30893
6. Open the browser and access the service URL. Check the tomcat home page is loading and version.



Apache Tomcat/9.0.13

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:
Security Considerations HOW-TO
Manager Application HOW-TO
Clustering/Session Replication HOW-TO

Developer Quick Start

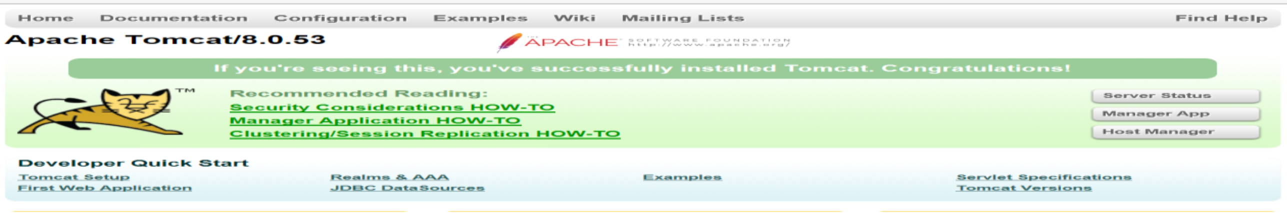
8.4 Upgrade the Application with different tomcat version

1. Update deployment of object tomcat to version 8.0 using the command :
   C:\Windows\system32>kubectl set image deployment/tomcat-deployment tomcat= tomcat:8.0
   deployment.extensions/tomcat-deployment image updated

2. Check the PODs are running using the below command
   C:\Windows\system32>kubectl get pods
   It will take some time, since 8.0 version of tomcat need to be pulled.
   Once the tomcat POD is running, then expose the service.

3. Expose the service using the command.
   C:\Windows\system32>kubectl expose deployment tomcat-deployment --type=NodePort

4. Get the minikube service url for this tomcat service using the command :
   C:\Windows\system32>minikube service tomcat-deployment –url
   http://192.168.99.102:30893
   Open the browser and access the service URL. Check the tomcat home page is loading.
5. Get the minikube service url for this tomcat service using the command :
   C:\Windows\system32>minikube service tomcat-deployment --url
   http://192.168.99.102:30893
6. Open the browser and access the service URL. Check the tomcat home page is loading and version.



Apache Tomcat/8.0.53

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:
Security Considerations HOW-TO
Manager Application HOW-TO
Clustering/Session Replication HOW-TO

Developer Quick Start
Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Examples
Servlet Specifications
Tomcat Versions

8.5 Rollout the previous deployed version

1. Rollout the changes applied to the deployment object using below command.

kubectl rollout undo deployment/tomcat-deployment
deployment.extensions/tomcat-deployment
It will take some time, since 8.0 version of tomcat need to be pulled.
Once the tomcat POD is running, then expose the service.

2. Get the minikube service url for this tomcat service using the command :
   C:\Windows\system32>**minikube service tomcat-deployment --url**
   http://192.168.99.102:30893

3. Open the browser and access the service URL. Check the tomcat home page is loading and version.

<<<<<End of Lab>>>>>

9 LAB-07: Healthcheck in Kubernetes

This LAB exercise shows you how to use the apply readiness and liveness probe health check in kubernetes.

9.1 Time to Complete

Approximately 0.30 Hr.

9.2 What You Need

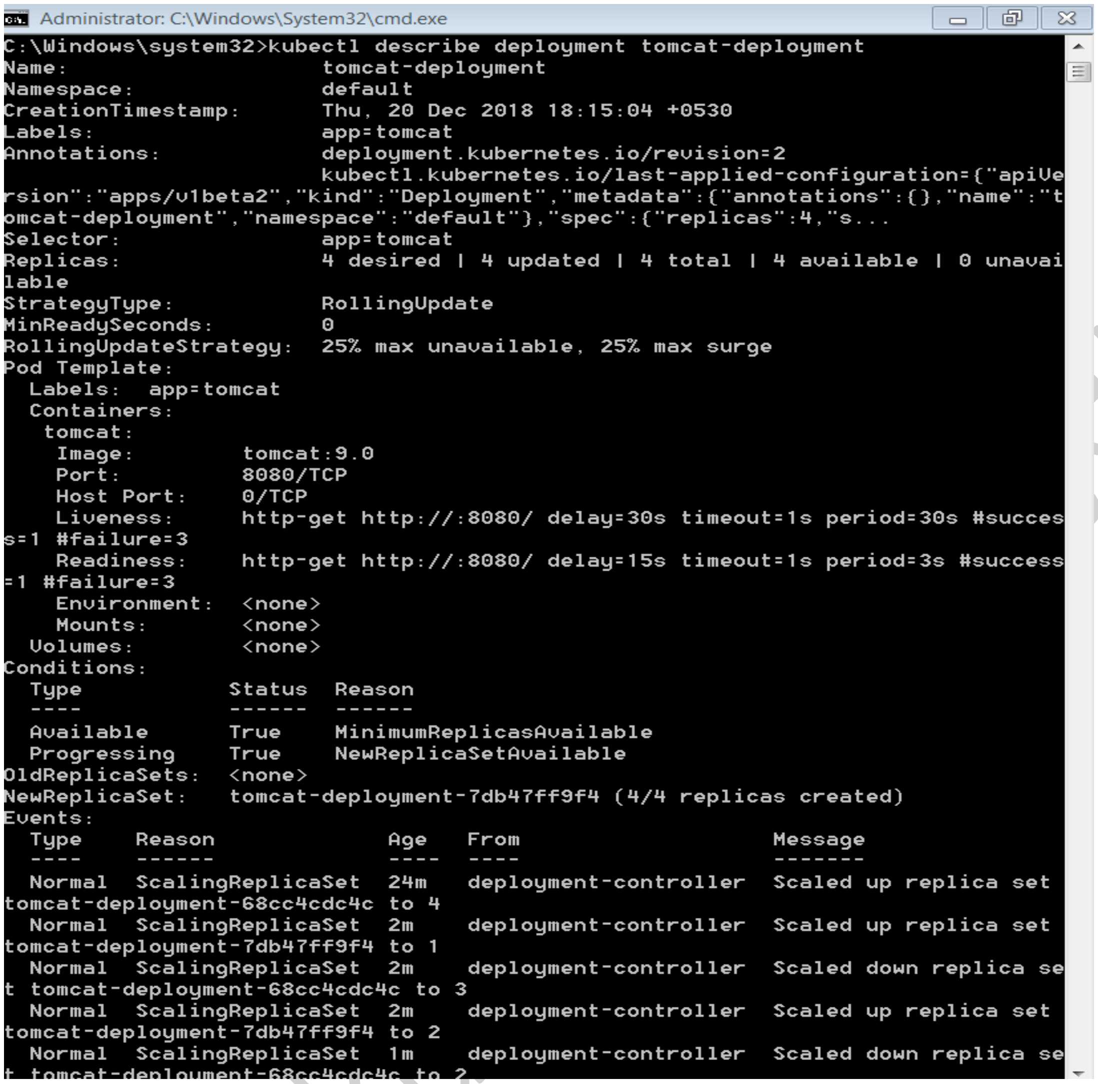
1. Lab 01 to completed successfully.
2. Kubernetes Cluster should be running. If not start the cluster using the command “minikube start”

9.3 Deploy the Application

1. Download the YAML file from the Healthchecks folder of the Git location : https://github.com/premkumarmlp/KubernetesExercises.git

2. Apply the changes in the deployment using the command
   C:\Windows\system32>kubectl apply -f D:\VM\Healthchecks\deployment.yaml
   deployment.apps/tomcat-deployment configured

3. Describe the deployment to view the healthcheck probes
   C:\Windows\system32>kubectl describe deployment tomcat-deployment



```
Administrator: C:\Windows\System32\cmd.exe
C:\Windows\system32>kubectl describe deployment tomcat-deployment
Name:                   tomcat-deployment
Namespace:              default
CreationTimestamp:      Thu, 20 Dec 2018 18:15:04 +0530
Labels:                 app=tomcat
Annotations:            deployment.kubernetes.io/revision=2
                        kubectl.kubernetes.io/last-applied-configuration={"apiVe
rsion":"apps/v1beta2","kind":"Deployment","metadata":{"annotations":{},"name":"t
omcat-deployment","namespace":"default"},"spec":{"replicas":4,"s...
Selector:               app=tomcat
Replicas:               4 desired | 4 updated | 4 total | 4 available | 0 unavai
lable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=tomcat
  Containers:
   tomcat:
    Image:        tomcat:9.0
    Port:         8080/TCP
    Host Port:    0/TCP
    Liveness:     http-get http://:8080/ delay=30s timeout=1s period=30s #succes
s=1 #failure=3
    Readiness:    http-get http://:8080/ delay=15s timeout=1s period=3s #success
=1 #failure=3
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets:  <none>
NewReplicaSet:   tomcat-deployment-7db47ff9f4 (4/4 replicas created)
Events:
  Type    Reason             Age   From                   Message
  ----    ------             ----  ----                   -------
  Normal  ScalingReplicaSet  24m   deployment-controller  Scaled up replica set
tomcat-deployment-68cc4cdc4c to 4
  Normal  ScalingReplicaSet  2m    deployment-controller  Scaled up replica set
tomcat-deployment-7db47ff9f4 to 1
  Normal  ScalingReplicaSet  2m    deployment-controller  Scaled down replica se
t tomcat-deployment-68cc4cdc4c to 3
  Normal  ScalingReplicaSet  2m    deployment-controller  Scaled up replica set
tomcat-deployment-7db47ff9f4 to 2
  Normal  ScalingReplicaSet  1m    deployment-controller  Scaled down replica se
t tomcat-deployment-68cc4cdc4c to 2
```

<<<<<End of Lab>>>>>

10 LAB-08: ConfigMaps in Kubernetes

This LAB exercise shows you how to apply ConfigMaps in kubernetes.

10.1 Time to Complete

Approximately 0.30 Hr.

10.2 What You Need

1. Lab 01 to completed successfully.
2. Kubernetes Cluster should be running. If not start the cluster using the command “minikube start”

10.3 Deploy the Application

Download the files from folder ConfigMap of Git location:
https://github.com/premkumarmlp/KubernetesExercises.git

1. Create a generic secret from YAML file

kubectl create -f my-secret.yml

2. Create the POD

kubectl create -f secret-env-pod.yml

3. Access the Secret in the POD

kubectl exec -it secret-env-pod /bin/sh

# env

4. Clean up

kubectl delete -f my-secret.yml -f secret-env-pod.yml

<<<<<End of Lab>>>>>

11 LAB-09: Secrets in Kubernetes

This LAB exercise shows you how to apply secrets in kubernetes.

11.1 Time to Complete

Approximately 0.30 Hr.

11.2 What You Need

3. Lab 01 to completed successfully.
4. Kubernetes Cluster should be running. If not start the cluster using the command “minikube start”

11.3 Deploy the Application

Download the files from folder Secret of Git location:
https://github.com/premkumarmlp/KubernetesExercises.git

5. Create a generic secret from YAML file

kubectl create -f my-secret.yml

6. Create the POD

kubectl create -f secret-env-pod.yml

7. Access the Secret in the POD

kubectl exec -it secret-env-pod /bin/sh

```
# env
```

8. Clean up

kubectl delete -f my-secret.yml -f secret-env-pod.yml

<<<<<End of Lab>>>>>