Computer Science & Information Systems

**DevOps for Cloud - Lab Sheet 4 – Module 6**

**(M6: Docker and Kubernetes)**

This lab sheet needs to be administered along with Module 6: Docker and Kubernetes

**Notation used in the document**

* ‘>’ represents the terminal, where we type the commands.
* The text mentioned within ‘[‘ and ‘]’ brackets provides additional documentation for the step.

1. Objectives:
2. To demonstrate the steps to install and run Minikube
3. To perform experiments on Minikube
4. Pre-requisite:

* Basic knowledge of docker, kubernetes and minikube

1. Lab Exercise:

**Task 1: Install Minikube**

* Purpose: Install Minikube on any OS
* URL to download Minikube on various OS - https://minikube.sigs.k8s.io/docs/start/
* In Windows -> Install Minikube using ‘Windows Package Manager’
* [Settings -> OS: Windows; Architecture: x86-64; Release Type: Stable; Installer Type: Windows Package Manager]
* > winget install minikube
* [Winget is a windows package manager CLI available for Windows 10 and 11]

**Task 2: Install Kubectl**

* URL for installing Kubectl - <https://kubernetes.io/docs/tasks/tools/>
* [The URL provides information to download kubectl for Windows, Linux and MacOS]
* For Windows OS, command to install
* > winget install -e --id Kubernetes.kubectl

**Task 3: Install Kubernetes Cluster on Minikube and run basic commands**

* Prerequisite: Docker Desktop must be running and both Tasks 1 and 2 must be completed

**Steps** [to be executed on any terminal such as command prompt]

1. Create a Single Node Minikube Cluster on a Docker Container and start the control plane

> minikube start

[Once the cluster is installed, check the Docker Desktop and view the Minikube Image and Container]

1. Display the version of the kubectl installed

> kubectl version

1. Display the status of the minikube cluster

> minikube status

1. Display list of minikube services

> minikube service list

**Task 4: Steps to deploy a “Hello-Minikube” Application onto Minikube**

**Steps** [to be executed on any terminal such as command prompt]

1. Create a deployment inside the Kubernetes cluster. One pod will be created

> kubectl create deployment hello-minikube --image=kicbase/echo-server:1.0

**Usage / Syntax for creating deployment**

kubectl create deployment NAME --image=image [--dry-run] [options]

[We cannot directly create pods via Kubectl, which are the smallest units; we have to use an abstraction layer called ‘Deployment’ to create pods. Pods need to be created based on Container images]

1. Check pods created for ‘hello-minikube’ App

> kubectl get pod

1. View the application in minikube dashboard

> minikube dashboard

[Opens the minikube dashboard. See ‘Deployments’, ‘Pods’, ‘Replica Sets’, ‘services’, and view their metadata]

**Task 5: Create / Edit a deployment for nginx container image**

**Steps to create deployment**

1. Creates a pod using latest Nginx image from Docker Hub

> kubectl create deployment nginx-depl --image=nginx

1. Command to get deployment

> kubectl get deployment

1. Command to view status of all pods

> kubectl get pod

1. Command to view specific pod

> kubectl describe pod nginx-depl

1. Command to get replicaset

> kubectl get replicaset

**Steps to edit deployment**

1. Edit the deployment

> kubectl edit deployment nginx-depl

[We get auto generated yaml configuration file with default values]

1. Make the following changes to the config file

2a. Scroll to the spec: containers and change image to -> **image: nginx:1.16**

2b. progressDeadlineSeconds: 600

2c. replicas: 3

2d. Save the file

1. Review the changes to replicaset and pod

> kubectl get replicaset

> kubectl get pods

1. Command to delete deployment

> kubectl delete deployment nginx-depl

1. Command to delete minikube cluster

> minikube delete

1. Command to delete pod

> kubectl delete pod <pod-name>

[Self-healing property of Kubernetes cluster automatically creates a new pod, when you delete one. We can only delete a pod permanently, by deleting the deployment itself]

**Task 6: Create a Deployment with Custom Image pushed to DockerHub**

**Steps** [to be executed on any terminal such as command prompt]

1. > kubectl create deployment myimage-depl --image=6945/restappimage:latest

[Consider “6945/myimage:latest” as the custom image pushed to DockerHub]

1. > kubectl describe pod myimage-depl
2. > minikube dashboard

[View the deployments, pods, replicasets created]

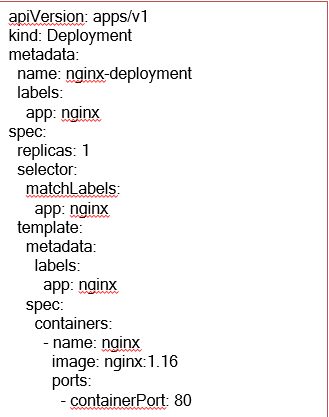
**Task 7: Create a Deployment File using the APPLY command**

[Creating Deployment file through commands is cumbersome. So we use the APPLY command to create a configuration file and mention all the resources to be created]

**Usage: kubectl apply -f [filename]**

**Steps**

1. Create the yaml file. The basic file is given below. Save the file in your directory as “nginx-basic-deployment.yml”



1. > kubectl apply –f nginx-basic-deployment.yml
2. > kubectl get deployment
3. > kubectl describe pod
4. If you want to edit the file, open the config file, make the changes and run the command again
5. > kubectl apply –f nginx-basic-deployment.yml [will re-deploy the changes]

4. Outputs/Results:

Students are expected to perform the tasks provided in the lab capsule, and thereby gain a practical understanding of the Container Orchestration platform, i.e. Kubernetes. Additionally, the students will be able to create a minikube cluster and run basic commands or experiments on the minikube cluster.

1. Observations:

* None

References:

* <https://minikube.sigs.k8s.io/docs/start/>
* <https://kubernetes.io/docs/tasks/tools/install-kubectl-windows/>
* <https://minikube.sigs.k8s.io/docs/start/>