Lab5: Hosting a web service using Kubernetes cloud orchestration

*Please **fill in the report** and submit the **pdf** to NYUClasses*

Name: Eugene Wang ID: yjw259 Date:

1. Objectives

- Understand the concept of container and the reason for cloud orchestration
- Get familiar with the Kubernetes and how to use the Kubernetes orchestration tool.
- Experience with hosting a popular cloud service using Kubernetes
- Use Kubernetes to deploy your WordPress server with a trivial loadbalancer
- Understand that you can manage Kubernetes services on Google Cloud Platform (Google Kubernetes Engine)

2. Experiments Tasks

2.1 Basics

a. Go through the Kubernetes introduction to get the general idea of Kubernetes https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/

2.2 Install Tools

a. Install Docker: https://docs.docker.com/engine/install/ubuntu/
b. Install kubectl: https://kubernetes.io/docs/tasks/tools/install-kubectl/
c. Install Minikube: https://kubernetes.io/docs/tasks/tools/install-minikube/
d. Install Helm: https://github.com/helm/helm

2.3 Host a Wordpress server on Kubernetes Cluster

a. Make sure the installation is complete
b. Run a single-node Kubernetes cluster locally

c. Deploy your WordPress server using helm
   https://hub.helm.sh/charts/bitnami/wordpress
d. Use kubectl commands to interact with Kubernetes cluster
e. Verify your deployment of wordpress both on your browser and terminal
f. Monitor the status of your Kubernetes cluster

3. Reports

3.1 General

1. (a) What is a container from an operating systems perspective?
   (b) Why are cgroups important to containers?

(a) For (host) OS, the (running) containers are just another application software processes. For those applications inside the container, container is an (light-weighted) abstraction of the OS.

(b) Cgroups is a linux kernel feature to control resources (namespace,CPU,I/O,etc) for processes. Cgroups enables processes to run in an isolated space.

Ref: https://itnext.io/breaking-down-containers-part-0-system-architecture-37afe0e51770

2. In Docker what does the ENTRYPOINT command allow you to do?

We use ENTRYPOINT command in dockerfile to run the program. It will be the first program when we run an image.

3. (a) In Kubernetes, what is a pod and why is it useful?
   (b) What is a label and why is it useful?

A Pod is a group of containers, with shared storage/network resources, and the same namespace. Pod are useful because it enable these containers to efficiently communicate.

Labels are key/value pairs used to specify pods and manage/group/filter pods/resources. It is useful because we can use it to highlight the ownership of a pod, control the timing to run the pods, or limit the resources.

4. (a) In Kubernetes, what is a service and why is it useful?
   (b) What is a ReplicaSet and why is it useful?

A Kubernetes service is a logical abstraction for a group of pods (which perform as a whole to provide a function). Since pods are frequently created/deleted, a Kubernetes service enables a group of pods, which as a whole provide that function (web service, for example) to be assigned a name and unique IP address. It is useful because we can use function to divide/manage pods.

A ReplicaSet ensures that a specified number of pod replicas are running at any given time. It is useful because we can use it to ensure reliability, manage load balance, and do autoscaling.

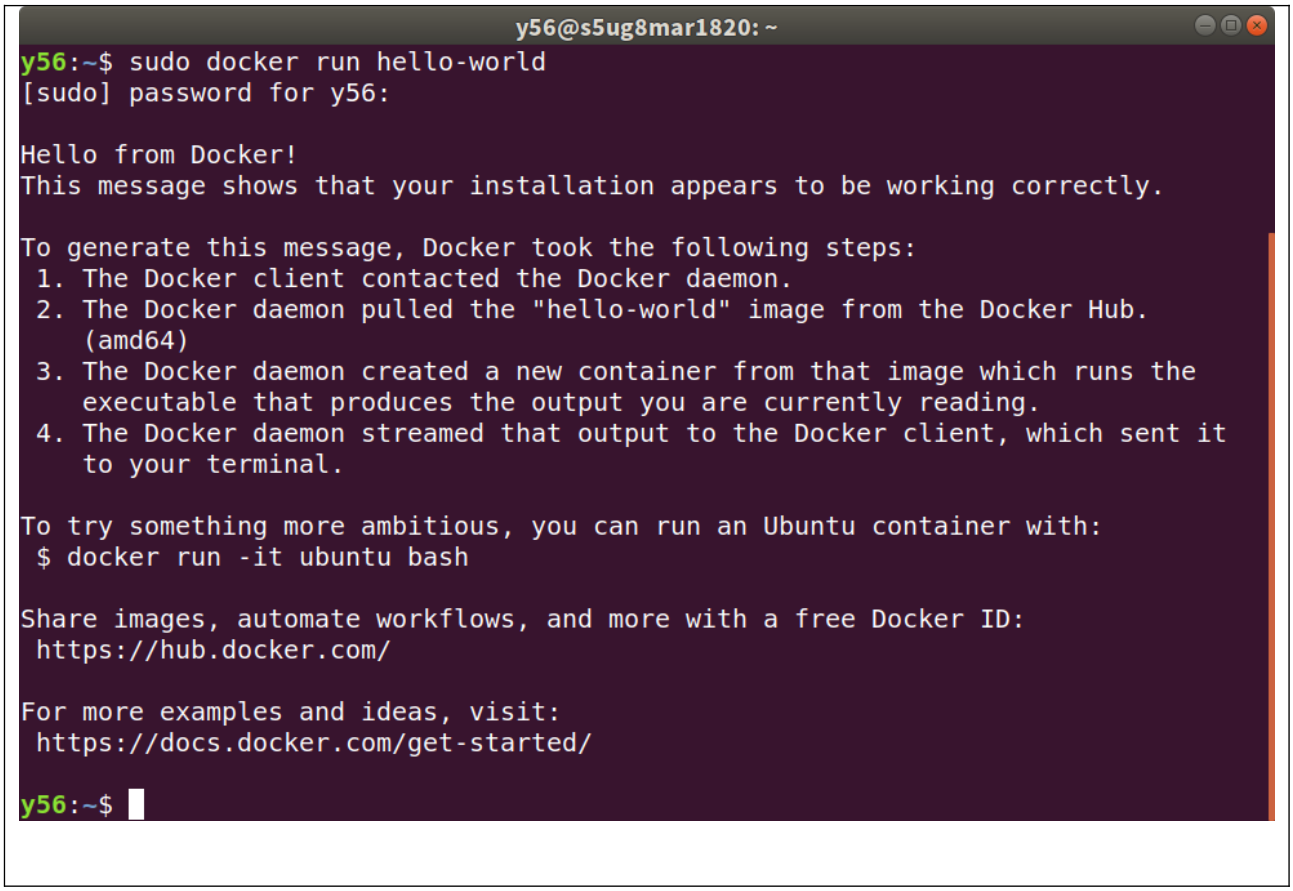
5. What is the difference between a pod and a node?

A Node is a worker machine in Kubernetes and may be either a virtual or a physical machine. A Pod always runs on a Node.

3.2 Verify your installation

6. Screenshots of running docker successfully

   `sudo docker run hello-world`



7. Screenshots showing that you can start minikube successfully

```
minikube start --driver=docker
minikube status
```

```
y56@s5ug8mar1820: ~
y56:~$ minikube start --driver=docker
😄  minikube v1.15.1 on Ubuntu 18.04
✨  Using the docker driver based on existing profile
👍  Starting control plane node minikube in cluster minikube
🔄  Restarting existing docker container for "minikube" ...

❗  Docker is nearly out of disk space, which may cause deployments to fail! (98%
of capacity)
💡  Suggestion:

    Try at least one of the following to free up space on the device:

    1. Run "docker system prune" to remove unused docker data
    2. Increase the amount of memory allocated to Docker for Desktop via
    Docker icon > Preferences > Resources > Disk Image Size
    3. Run "minikube ssh -- docker system prune" if using the docker container ru
ntime
🍿  Related issue: https://github.com/kubernetes/minikube/issues/9024

🐳  Preparing Kubernetes v1.19.4 on Docker 19.03.13 ...
🔎  Verifying Kubernetes components...
🌟  Enabled addons: storage-provisioner, default-storageclass, dashboard
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" names
pace by default
y56:~$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

y56:~$
```

8. Screenshots of verifying kubectl configuration
   `kubectl cluster-info`

```
y56@s5ug8mar1820: ~
y56:~$ kubectl cluster-info
Kubernetes master is running at https://192.168.49.2:8443
KubeDNS is running at https://192.168.49.2:8443/api/v1/namespaces/kube-system/se
rvices/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
y56:~$
```

3.3 Deploy your WordPress

9. Explain the commands

(a)

`kubectl get pods`: **List all pods**

`kubectl get services`: **List all services in the namespace**

(b)

`kubectl describe deployment`: **Get details of your Deployment**

`kubectl logs <pod-name>`: **Print the logs for a pod**

10. Screenshots of your terminals showing that your pod is up (Hint: use a command in 9a)

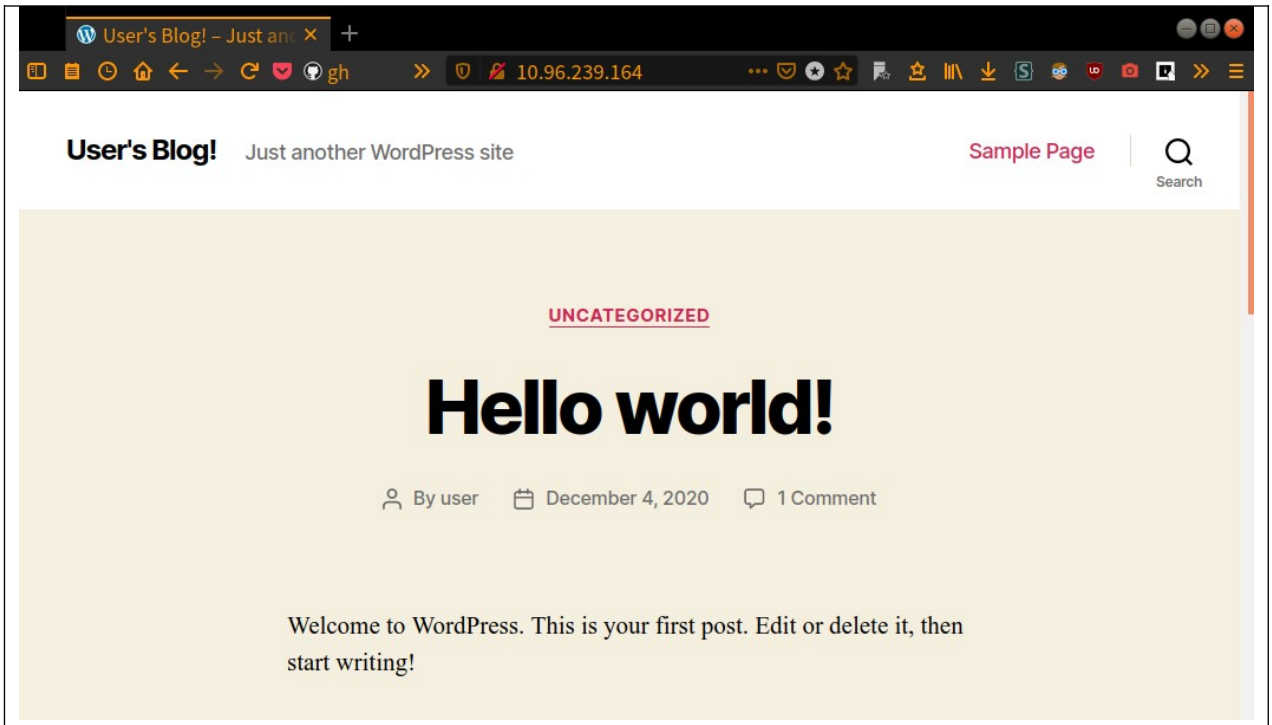
```
y56@s5ug8mar1820: ~
y56:~$ kubectl get pods
NAME                                    READY   STATUS    RESTARTS   AGE
my-release-mariadb-0                    1/1     Running   1          10d
my-release-wordpress-7b8666479f-pw4tn   1/1     Running   1          10d
y56:~$ kubectl get services
NAME                   TYPE           CLUSTER-IP      EXTERNAL-IP     PORT(S)                      AGE
kubernetes             ClusterIP      10.96.0.1       <none>          443/TCP                      10d
my-release-mariadb     ClusterIP      10.111.92.155   <none>          3306/TCP                     10d
my-release-wordpress   LoadBalancer   10.96.239.164   10.96.239.164   80:32608/TCP,443:30739/TCP   10d
y56:~$
```

11. Screenshots of your terminals showing your running service along with the external IP address that can be used to access your pod. (Hint: use commands in 9a)

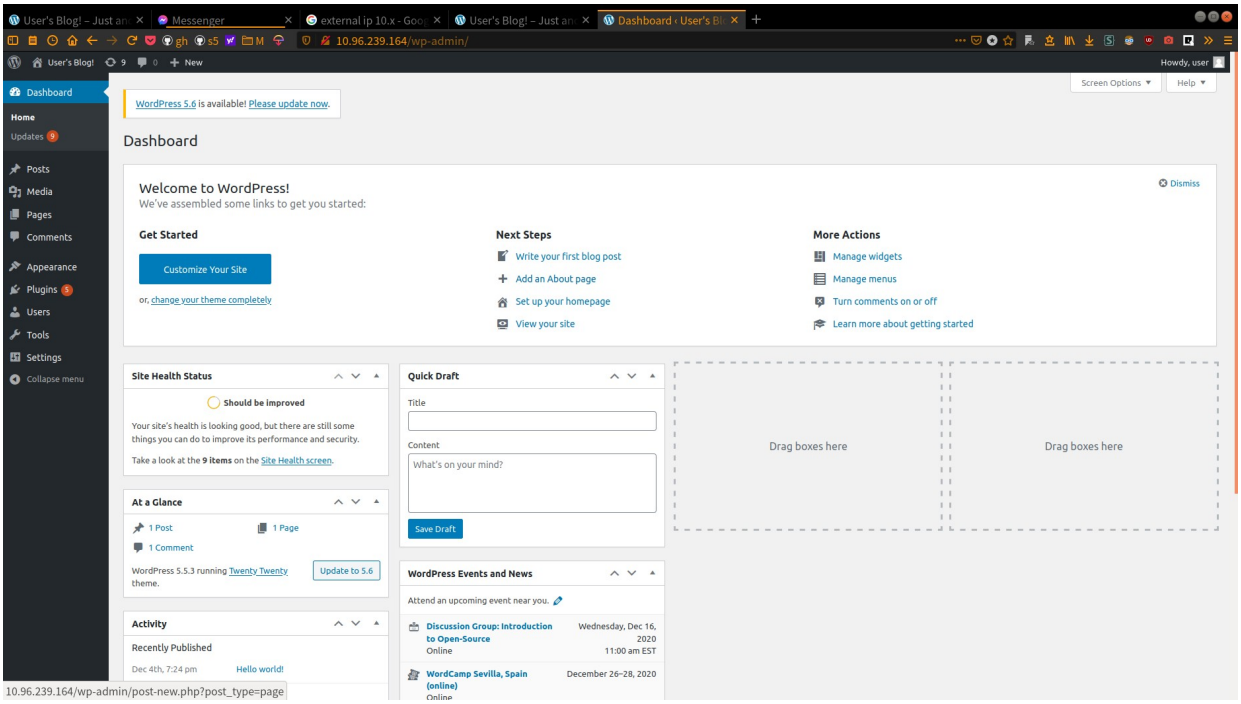
```
y56@s5ug8mar1820: ~
y56:~$ kubectl cluster-info
Kubernetes master is running at https://192.168.49.2:8443
KubeDNS is running at https://192.168.49.2:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/pro
xy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
y56:~$ kubectl get pods
NAME                                    READY   STATUS    RESTARTS   AGE
my-release-mariadb-0                    1/1     Running   1          10d
my-release-wordpress-7b8666479f-pw4tn   1/1     Running   1          10d
y56:~$ kubectl get services
NAME                   TYPE           CLUSTER-IP      EXTERNAL-IP     PORT(S)                      AGE
kubernetes             ClusterIP      10.96.0.1       <none>          443/TCP                      10d
my-release-mariadb     ClusterIP      10.111.92.155   <none>          3306/TCP                     10d
my-release-wordpress   LoadBalancer   10.96.239.164   10.96.239.164   80:32608/TCP,443:30739/TCP   10d
y56:~$ export SERVICE_IP=$(kubectl get svc --namespace default my-release-wordpress --template "{{ rang
e (index .status.loadBalancer.ingress 0) }}{{.}}{{ end }}")
y56:~$ echo "WordPress URL: http://$SERVICE_IP/"
WordPress URL: http://10.96.239.164/
y56:~$ echo "WordPress Admin URL: http://$SERVICE_IP/admin"
WordPress Admin URL: http://10.96.239.164/admin
y56:~$ echo Password: $(kubectl get secret --namespace default my-release-wordpress -o jsonpath="{.data
.wordpress-password}" | base64 --decode)
Password: Nq8mkvf3hp
y56:~$ ^C
y56:~$
```

12. (a)Screenshots of your browser running your wordpress. What is your wordpress URL?
(b)Screenshots of login to your wordpress as admin

http://10.96.239.164/


User's Blog! – Just an
10.96.239.164
User's Blog!
Just another WordPress site
Sample Page
Search
UNCATEGORIZED
Hello world!
By user
December 4, 2020
1 Comment
Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

–


10.96.239.164/wp-admin/
WordPress 5.6 is available! Please update now.
Dashboard
Welcome to WordPress!
We've assembled some links to get you started:
Dismiss
Get Started
Customize Your Site
or, change your theme completely
Next Steps
Write your first blog post
Add an About page
Set up your homepage
View your site
More Actions
Manage widgets
Manage menus
Turn comments on or off
Learn more about getting started
Site Health Status
Should be improved
Quick Draft
Title
Content
What's on your mind?
Save Draft
Drag boxes here
Drag boxes here
At a Glance
1 Post
1 Page
1 Comment
Update to 5.6
WordPress Events and News
Activity
Recently Published
Hello world!
10.96.239.164/wp-admin/post-new.php?post_type=page

13. What happens after you manually delete a pod using the following command?

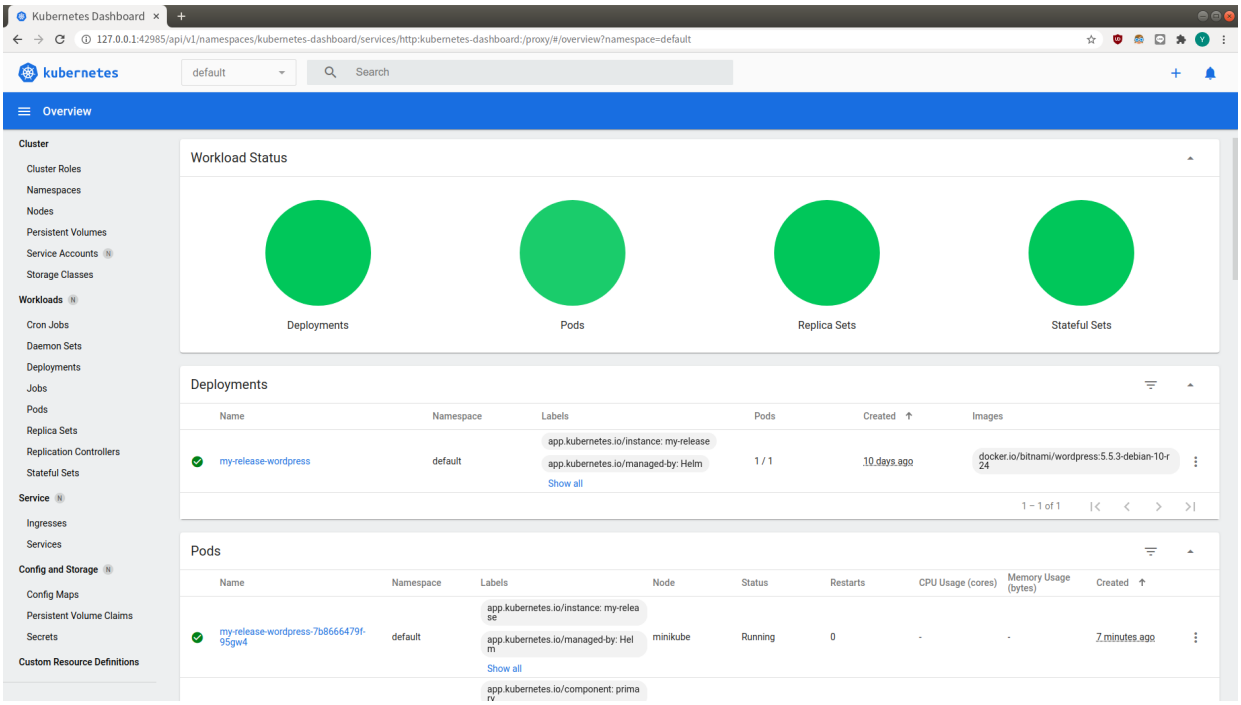
```
kubectl delete pods <your-pod-name>
```

```
y56:~$ kubectl delete pods my-release-wordpress-7b8666479f-pw4tn
pod "my-release-wordpress-7b8666479f-pw4tn" deleted
y56:~$ ^C
y56:~$ kubectl get pods
NAME                                    READY   STATUS    RESTARTS   AGE
my-release-mariadb-0                    1/1     Running   1          10d
my-release-wordpress-7b8666479f-95gw4   1/1     Running   0          66s
y56:~$
```

It restart a new pod. You can see the age is 66 sec.

14. Screenshots of your Kubernetes Dashboard. What is the health status of your workloads?

All healthy for all workload.



15. Challenges you've encountered while doing this experiment and explain how you manage to solve them. If you do not experience any problem, simply say no problem.

1. permission problem, solved by
sudo usermod -aG docker $USER && newgrp docker
2. external ip problem, solved by
minikube tunnel

I am using native ubuntu 18.04 and Internet of the US.

We have zero tolerance to forged or fabricated data!! A single piece of forged/fabricated data would bring the total score down to zero.