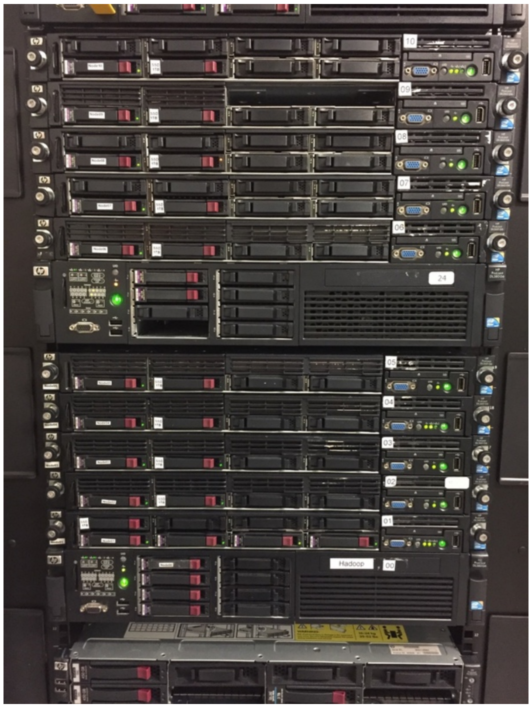
CSC570 Containerization for Big data

UNIVERSITY OF ILLINOIS SPRINGFIELD

Hands-On Exercise 4-1 Building a Kubernetes cluster

1 CSCluster at UIS

11-node Hadoop nodes



- CPU: Intel Xeon **4C/8T** per node
- Memory: **266 GB** in total
  - Master (head) node: 74GB RAM
  - 2nd Master node: 48GB RAM
  - 9 worker nodes: 16GB/node
- Storage: **10 TB SSD** in total
  - Master node: 400GB,
  - 10 worker nodes: 1TB SSD/node
- Hadoop: **Cloudera CDH 6.3**
  - HDFS, MapReduce
  - Spark 2, HBase, Hive
  - Pig, HUE, and etc.

VMs for Docker/K8S (You will use these VMs)

- CPU: Intel Xeon **8** core/node
- Memory: **16GB** RAM/node
- Storage: **100GB**

3-node Cassandra cluster (NoSQL-Column family DB)

- CPU: Intel Xeon **4C/8T** per node
- Memory: **48GB** RAM (16GB/node)
- Storage: **3TB SSD** (1TB/node)

PostgreSQL node (Relational DB)

- CPU: Intel Xeon **4C/8T** per node
- Memory: **18 GB RAM**
- Storage: **1TB SSD**

MongoDB node (NoSQL-Document DB)

- CPU: Intel Xeon **4C/8T** per node
- Memory: **16 GB RAM**
- Storage: **1TB SSD**

2 Accessing your VM for Docker and Kubernetes

2.1 Accessing campus network

If you are in UIS campus, you are fine. If you are not in UIS campus, you should install a Cisco VPN client software. The VPN client software gives remote users a secure and encrypted VPN (Virtual Private Network) connection to the UIS campus network. Please see below website, https://www.uis.edu/informationtechnologyservices/connect/vpn/

2.2 Accessing your VM using SSH

After you install the VPN client software and make a VPN connection to UIS network, you can access your VM using a terminal (Mac), PowerShell (Windows), or Putty (Windows).

- Check the IP address for your VM in the 'Course Information' under 'Modules' in Canvas

Type below command in your preferred SSH shell client:

```
ssh your-login@10.92.128.36
```

your-login: your UIS NetID (for example, **slee675** from slee675@uis.edu)

Initial password & IPs: See the Virtual Machine IPs page in the 'Course Information' under 'Modules' in Canvas.

After you logged in to your VM, you will see below prompt. The 'us2004lts' is a name of your VM and stands for 'Ubuntu Sever 20.04 LTS' that we are using as an OS.

```
your-login@us2004lts:~$
```

# Example1: using terminal in Mac

```
LeeMBP15:~ sslee777$ ssh sslee777@10.92.128.36
The authenticity of host '10.92.128.36 (10.92.128.36)' can't be established.
ECDSA key fingerprint is SHA256:dXhKfHsYIXe/53hvU+HOK2V6fVrTbz/QxmUhpnPXpzA.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.92.128.36' (ECDSA) to the list of known hosts.
sslee777@10.92.128.36's password: <== Use initial password until you change it
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-33-generic x86_64)
...

sslee777@us2004lts:~$
```

3 Building a Kubernetes Cluster

3.1 Overview

There are two popular learning environments for Kubernetes, Minukube and Kind. We will use Minukube in this exercise. See below and visit https://brennerm.github.io/posts/minikube-vs-kind-vs-k3s.html

Minikube is a Kubernetes SIGs project and has been started more than three years ago. It takes the approach of spawning a VM that is essentially a single node K8s cluster. Due to the support for a bunch of hypervisors it can be used on all of the major operating systems. This also allows you to create multiple instances in parallel.

Kind is another Kubernetes SIGs project but is quite different compared to minikube. As the name suggests it moves the cluster into Docker containers. This leads to a significantly faster startup speed compared to spawning VM.

Minikube also support to create nodes using Docker instead of hypervisor (virtual machine).

3.2 Install and Set Up kubectl

Official Doc for installing kubectl: https://kubernetes.io/docs/tasks/tools/install-kubectl/

Before we install the minikube, we will install a Kubernetes command-line tool, kubectl first. Kubectl allows you to run commands against Kubernetes clusters. You can use kubectl to deploy applications, inspect and manage cluster resources, and view logs

Create a directory and download the latest release.

```
sslee777@us2004lts:~$ mkdir myK8s
sslee777@us2004lts:~$ cd myK8s/
sslee777@us2004lts:~/myK8s$ curl -LO https://storage.googleapis.com/kubernetes-
release/release/`curl -s https://storage.googleapis.com/kubernetes-
release/release/stable.txt`/bin/linux/amd64/kubectl
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 41.9M  100 41.9M    0     0  43.8M      0 --:--:-- --:--:-- --:--:-- 43.8M
sslee777@us2004lts:~/myK8s$ ls
kubectl
sslee777@us2004lts:~/myK8s$
```

Make the kubectl binary executable, then move the binary in to local bin directory. Check the version of the kubectl.

```
sslee777@us2004lts:~/myK8s$ chmod +x ./kubectl

sslee777@us2004lts:~/myK8s$ sudo mv ./kubectl /usr/local/bin/kubectl
[sudo] password for sslee777:

sslee777@us2004lts:~/myK8s$ ls /usr/local/bin/kubectl
/usr/local/bin/kubectl

sslee777@us2004lts:~/myK8s$ kubectl version --client
Client Version: version.Info{Major:"1", Minor:"18", GitVersion:"v1.18.5",
GitCommit:"e6503f8d8f769ace2f338794c914a96fc335df0f", GitTreeState:"clean",
BuildDate:"2020-06-26T03:47:41Z", GoVersion:"go1.13.9", Compiler:"gc",
Platform:"linux/amd64"}
sslee777@us2004lts:~/myK8s$
```

(Optional) You may use apt or apt-get package manger on Ubuntu

```
sudo apt-get update && sudo apt-get install -y apt-transport-https gnupg2
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -
echo "deb https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee -a
/etc/apt/sources.list.d/kubernetes.list
sudo apt-get update
sudo apt-get install -y kubectl
```

3.3 Installing Minikube

Official Doc for installing miniKube: https://kubernetes.io/docs/tasks/tools/install-minikube/

Minikube is a tool that makes it easy to run Kubernetes locally. Minikube runs a single-node Kubernetes cluster inside your Virtual Machine (VM) for users looking to try out Kubernetes or develop with it day-to-day.

Minikube supports the following Kubernetes features:

- DNS
- NodePorts
- ConfigMaps and Secrets
- Dashboards
- Container Runtime: Docker, CRI-O, and containerd
- Enabling CNI (Container Network Interface)
- Ingress

To check if virtualization is supported on Linux, run the following command and verify that the output is non-empty:

```
sslee777@us2004lts:~$ grep -E --color 'vmx|svm' /proc/cpuinfo
flags        : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat
pse36 clflush mmx fxsr sse sse2 ss syscall nx rdtscp lm constant_tsc arch_perfmon
nopl xtopology tsc_reliable nonstop_tsc cpuid pni pclmulqdq vmx ssse3 cx16 pcid
sse4_1 sse4_2 x2apic popcnt tsc_deadline_timer aes xsave avx f16c rdrand hypervisor
lahf_lm cpuid_fault pti ssbd ibrs ibpb stibp tpr_shadow vnmi ept vpid fsgsbase
tsc_adjust smep arat md_clear flush_l1d arch_capabilities
...
```

Download the binary and also make it executable.

```
sslee777@us2004lts:~/myK8s$ curl -Lo minikube
https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64 \
>   && chmod +x minikube
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 52.8M  100 52.8M    0     0  48.5M      0  0:00:01  0:00:01 --:--:-- 48.5M
sslee777@us2004lts:~/myK8s$ ls
minikube
sslee777@us2004lts:~/myK8s$
```

Move the binary in to local bin directory

```
sslee777@us2004lts:~/myK8s$ sudo mv ./minikube /usr/local/bin/minikube
[sudo] password for sslee777:
sslee777@us2004lts:~/myK8s$ ls /usr/local/bin/minikube
/usr/local/bin/minikube
sslee777@us2004lts:~/myK8s$
```

Check the version of the minikube.

```
sslee777@us2004lts:~/myK8s$ minikube version
minikube version: v1.11.0
commit: 57e2f55f47effe9ce396cea42a1e0eb4f611ebbd
sslee777@us2004lts:~/myK8s$
```

You can run the following command to start up a local Kubernetes cluster. We may use virtual machine for nodes like docker-machine in previous exercise, but we will use ‘docker’ or ‘none’ driver to speed up starting containers and save some resources by using Docker containers instead of virtual machines. ‘none’ driver requires ‘root’.

```
sslee777@us2004lts:~/myK8s$ minikube start --driver=none
😄  minikube v1.11.0 on Ubuntu 20.04
✨  Using the none driver based on user configuration

❗  'none' driver reported an issue: the 'none' driver must be run as the root user
💡  Suggestion: For non-root usage, try the newer 'docker' driver

💣  Failed to validate 'none' driver
```

So, we use a newer ‘docker’ driver.

```
sslee777@us2004lts:~/myK8s$ minikube start --driver=docker
😄  minikube v1.11.0 on Ubuntu 20.04
✨  Using the docker driver based on user configuration
👍  Starting control plane node minikube in cluster minikube
🚜  Pulling base image ...
💾  Downloading Kubernetes v1.18.3 preload ...
    > preloaded-images-k8s-v3-v1.18.3-docker-overlay2-amd64.tar.lz4: 526.01 MiB
🔥  Creating docker container (CPUs=2, Memory=4000MB) ...
🐳  Preparing Kubernetes v1.18.3 on Docker 19.03.2 ...
    ▪ kubeadm.pod-network-cidr=10.244.0.0/16
🔎  Verifying Kubernetes components...
🌟  Enabled addons: default-storageclass, storage-provisioner
🏄  Done! kubectl is now configured to use "minikube"
sslee777@us2004lts:~/myK8s$
```

Once minikube start finishes, run the command below to check the status of the cluster

```
sslee777@us2004lts:~/myK8s$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

sslee777@us2004lts:~/myK8s$
```

After you have confirmed whether Minikube is working, you can continue to use Minikube or you can stop your cluster. Stop your cluster and check the status of your cluster again.

```
sslee777@us2004lts:~/myK8s$ minikube stop
✋  Stopping "minikube" in docker ...
🛑  Powering off "minikube" via SSH ...
🛑  Node "minikube" stopped.

sslee777@us2004lts:~/myK8s$ minikube status
minikube
type: Control Plane
host: Stopped
kubelet: Stopped
apiserver: Stopped
kubeconfig: Stopped

sslee777@us2004lts:~/myK8s$
```

3.4 Installing Kubernetes with Minikube

Official documentation for installing Kubernetes: https://kubernetes.io/docs/setup/learning-environment/minikube/

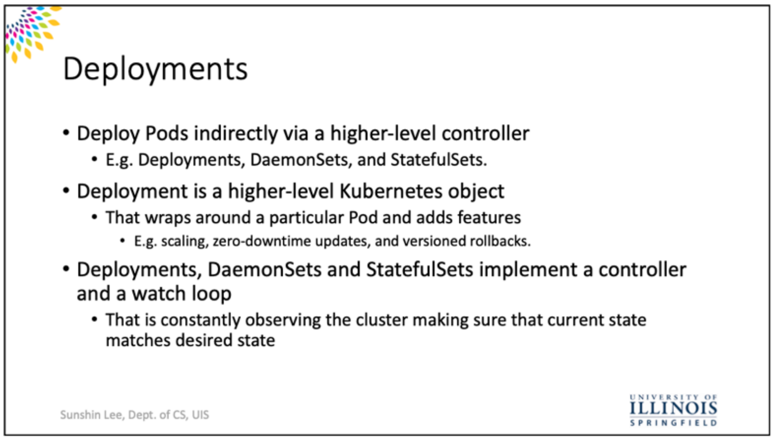
Start the cluster you created in previous section. They remember that you will use 'docker' driver. Check the status of the cluster.

```
sslee777@us2004lts:~/myK8s$ minikube start
😄  minikube v1.11.0 on Ubuntu 20.04
✨  Using the docker driver based on existing profile
👍  Starting control plane node minikube in cluster minikube
🔄  Restarting existing docker container for "minikube" ...
🐳  Preparing Kubernetes v1.18.3 on Docker 19.03.2 ...
    ▪ kubeadm.pod-network-cidr=10.244.0.0/16
🔎  Verifying Kubernetes components...
🌟  Enabled addons: default-storageclass, storage-provisioner
🏄  Done! kubectl is now configured to use "minikube"

sslee777@us2004lts:~/myK8s$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

sslee777@us2004lts:~/myK8s$
```

Now, you can interact with your cluster using kubectl.
Let's create a Kubernetes Deployment using an existing image named 'echoserver', which is a simple HTTP server and expose it on port 8080 using --port.

```
sslee777@us2004lts:~/myK8s$ kubectl create deployment myminikube --
image=k8s.gcr.io/echoserver:1.10
deployment.apps/myminikube created
sslee777@us2004lts:~/myK8s$
```

Let's review related concepts in the lecture slides



Deployments

- Deploy Pods indirectly via a higher-level controller
  - E.g. Deployments, DaemonSets, and StatefulSets.
- Deployment is a higher-level Kubernetes object
  - That wraps around a particular Pod and adds features
    - E.g. scaling, zero-downtime updates, and versioned rollbacks.
- Deployments, DaemonSets and StatefulSets implement a controller and a watch loop
  - That is constantly observing the cluster making sure that current state matches desired state

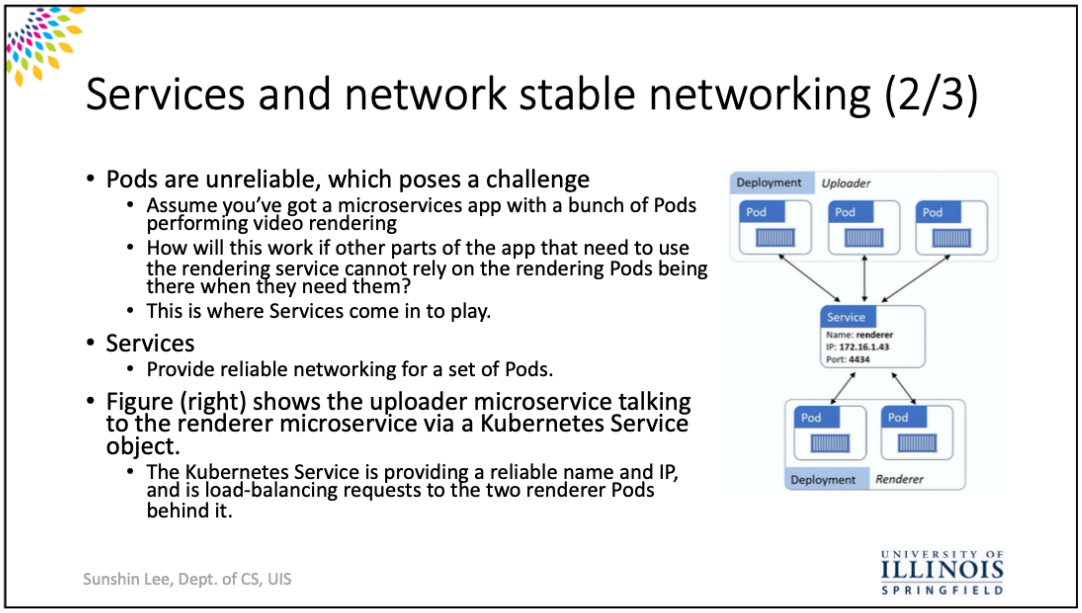
Sunshin Lee, Dept. of CS, UIS

To access the myminikube Deployment, expose it as a Service:

```
sslee777@us2004lts:~/myK8s$ kubectl expose deployment myminikube --type=NodePort --
port=8080
service/myminikube exposed
sslee777@us2004lts:~/myK8s$
```

The option --type=NodePort specifies the type of the Service.

Let's review related concepts in the lecture slides.



Services and network stable networking (2/3)

- Pods are unreliable, which poses a challenge
  - Assume you've got a microservices app with a bunch of Pods performing video rendering
  - How will this work if other parts of the app that need to use the rendering service cannot rely on the rendering Pods being there when they need them?
  - This is where Services come in to play.
- Services
  - Provide reliable networking for a set of Pods.
- Figure (right) shows the uploader microservice talking to the renderer microservice via a Kubernetes Service object.
  - The Kubernetes Service is providing a reliable name and IP, and is load-balancing requests to the two renderer Pods behind it.

Sunshin Lee, Dept. of CS, UIS

UNIVERSITY OF ILLINOIS SPRINGFIELD

The myminikube Pod is now launched but you have to wait until the Pod is up before accessing it via the exposed Service.

Verify if the Pod is up and running

```
sslee777@us2004lts:~/myK8s$ kubectl get pod
NAME                          READY   STATUS    RESTARTS   AGE
myminikube-65bbb4cc68-nlj2m   1/1     Running   0          3m18s
sslee777@us2004lts:~/myK8s$
```

Get the URL of the exposed Service to view the Service details

```
sslee777@us2004lts:~/myK8s$ minikube service myminikube --url
http://172.17.0.2:32345
sslee777@us2004lts:~/myK8s$
```

To view the details of your local cluster, copy and paste the URL you got as the output, on your browser. You may not be able to see the output in your browser, because you opened a browser in your laptop instead of in your VM. So, you may need to use SSH Port Forwarding. SSH port forwarding is a mechanism in SSH for tunneling application ports from the client machine to the server machine, or vice versa. See https://www.ssh.com/ssh/tunneling/example

Open another NEW terminal and run ssh command with some options like below:
(Note: Recommended to use two SSH terminals at the same time: one for normal command-line connection, the other for SSH port forwarding.)

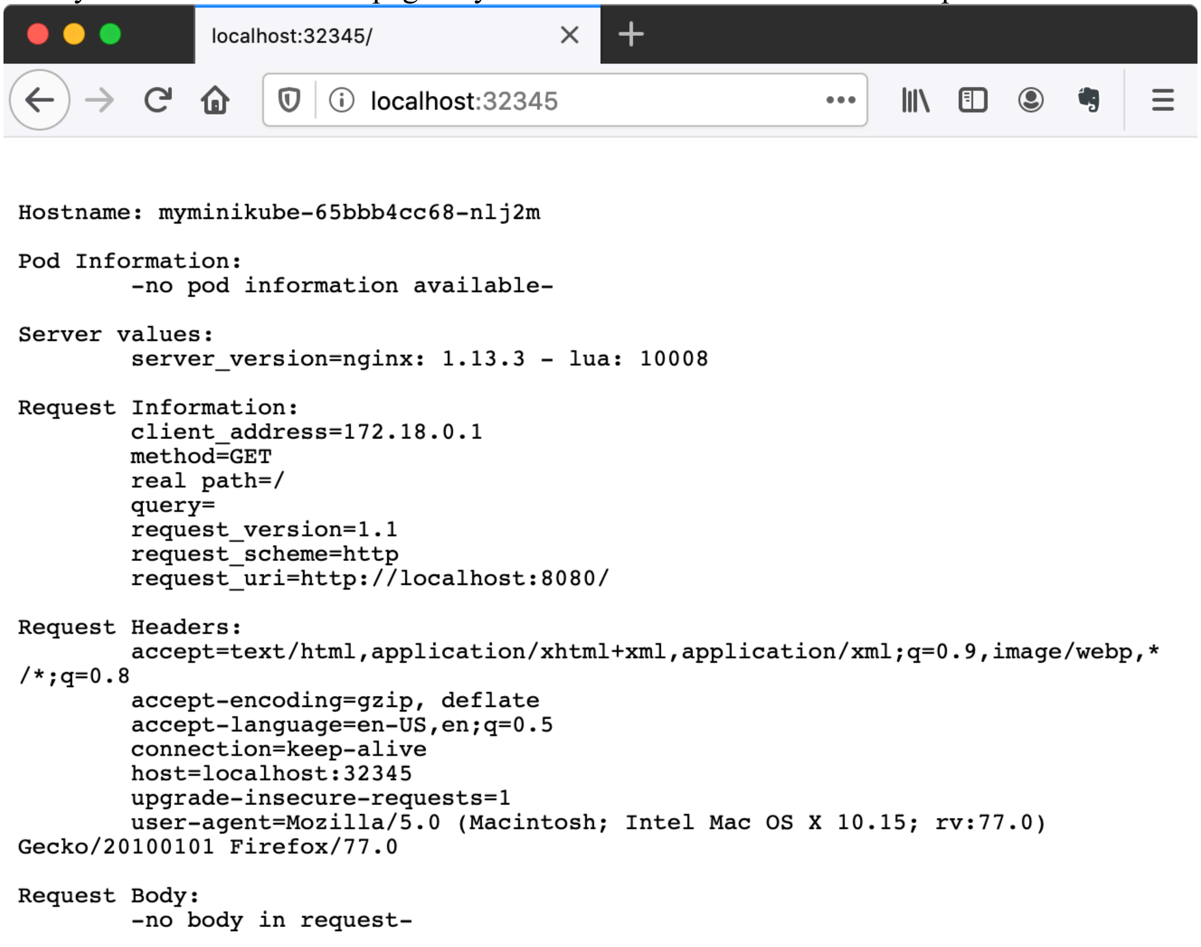
- SSH port forwarding option: -L
- Your laptop's local port: 32345
- Your cluster's IP and it's port inside your host/docker VM: 172.17.0.2:32345
- Your host/docker VM's login and IP: sslee777@10.92.128.46

LeeMBP15 is my laptop.

```
LeeMBP15:~ sslee777$ ssh -L 32345:172.17.0.2:32345 sslee777@10.92.128.46
Welcome to Ubuntu 20.04 LTS

...

Last login...
sslee777@us2004lts:~$
```

Note: You need to keep the SSH terminal during you are using your browser.

Now you can access the webpage in your browser with localhost and the port number.



```
Hostname: myminikube-65bbb4cc68-nlj2m

Pod Information:
        -no pod information available-

Server values:
        server_version=nginx: 1.13.3 - lua: 10008

Request Information:
        client_address=172.18.0.1
        method=GET
        real path=/
        query=
        request_version=1.1
        request_scheme=http
        request_uri=http://localhost:8080/

Request Headers:
        accept=text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*
/*;q=0.8
        accept-encoding=gzip, deflate
        accept-language=en-US,en;q=0.5
        connection=keep-alive
        host=localhost:32345
        upgrade-insecure-requests=1
        user-agent=Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:77.0)
Gecko/20100101 Firefox/77.0

Request Body:
        -no body in request-
```

It is working great!

3.5 Kubernetes Dashboard

Finally, let’s try to create Kubernetes Dashboard, which is a web-based Kubernetes user interface.

You can use Dashboard to deploy containerized applications to a Kubernetes cluster, troubleshoot your containerized application, and manage the cluster resources. You can use Dashboard to get an overview of applications running on your cluster, as well as for creating or modifying individual Kubernetes resources (such as Deployments, Jobs, DaemonSets, etc). For example, you can scale a Deployment, initiate a rolling update, restart a pod or deploy new applications using a deploy wizard.

Dashboard also provides information on the state of Kubernetes resources in your cluster and on any errors that may have occurred. See https://kubernetes.io/docs/tasks/access-application-cluster/web-ui-dashboard/

```
sslee777@us2004lts:~/myK8s$ minikube dashboard
🔌  Enabling dashboard ...
🤔  Verifying dashboard health ...
🚀  Launching proxy ...
🤔  Verifying proxy health ...
🎉  Opening http://127.0.0.1:39071/api/v1/namespaces/kubernetes-
dashboard/services/http:kubernetes-dashboard:/proxy/ in your default browser...
👉  http://127.0.0.1:39071/api/v1/namespaces/kubernetes-
dashboard/services/http:kubernetes-dashboard:/proxy/
```

Note: Do not cancel the process by pressing ‘ctrl + c’

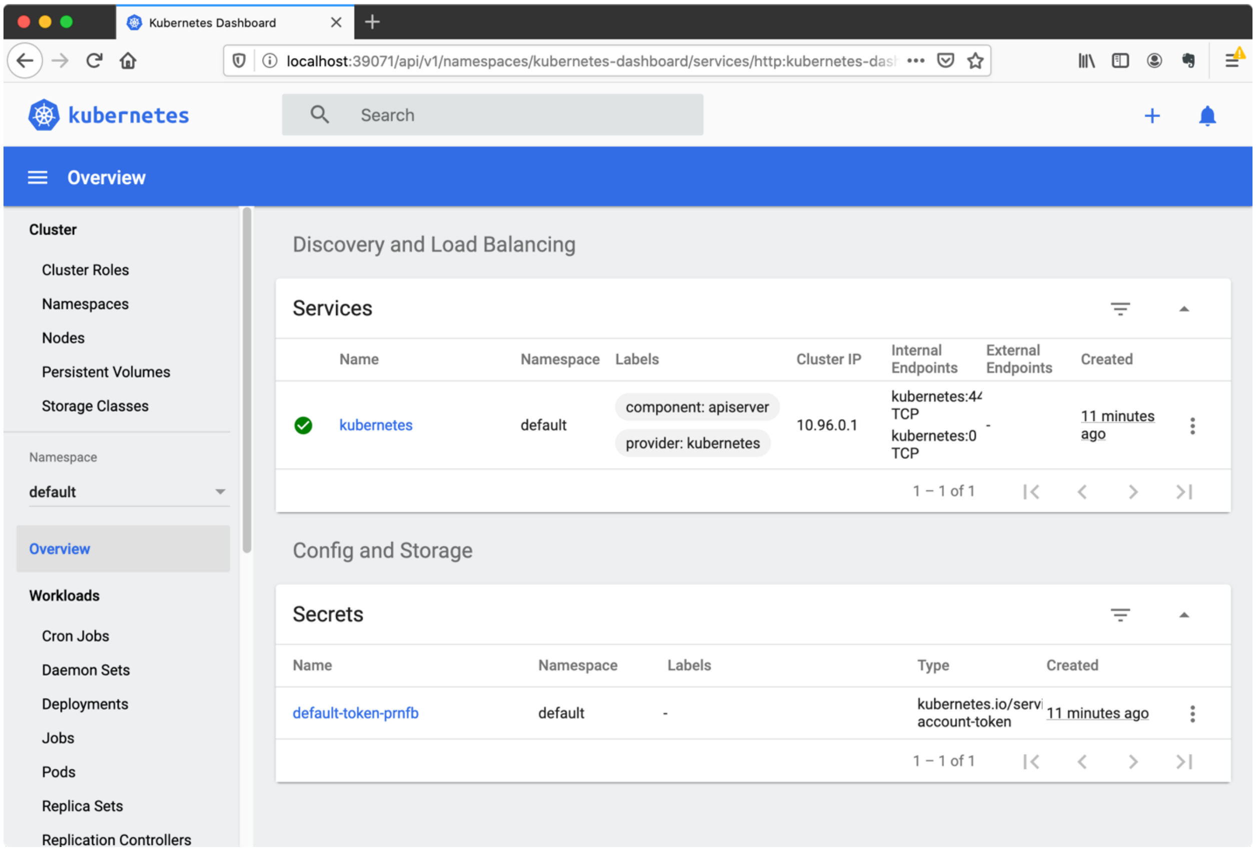
To access the K8s Dashboard, you need the SSH port forwarding again. Open another NEW terminal and run ssh command with some options like below:

- SSH port forwarding option: -L
- Your laptop’s local port: 39071
- Your dashboard’s proxy IP and it’s port inside your host/docker VM: 127.0.0.1:39071 or localhost:39071
- Your host/docker VM’s login and IP: sslee777@10.92.128.46

LeeMBP15 is my laptop.

```
LeeMBP15:~ sslee777$ ssh -L 39071:localhost:39071 sslee777@10.92.128.46
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-39-generic x86_64)
...
Last login: ...
sslee777@us2004lts:~$
```

Now you can access the webpage in your browser. Note: Use your port number http://localhost:39071/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/#/overview?namespace=default



Discovery and Load Balancing

Services

	Name	Namespace	Labels	Cluster IP	Internal Endpoints	External Endpoints	Created
✓	kubernetes	default	component: apiserver provider: kubernetes	10.96.0.1	kubernetes:44 TCP kubernetes:0 TCP	-	11 minutes ago

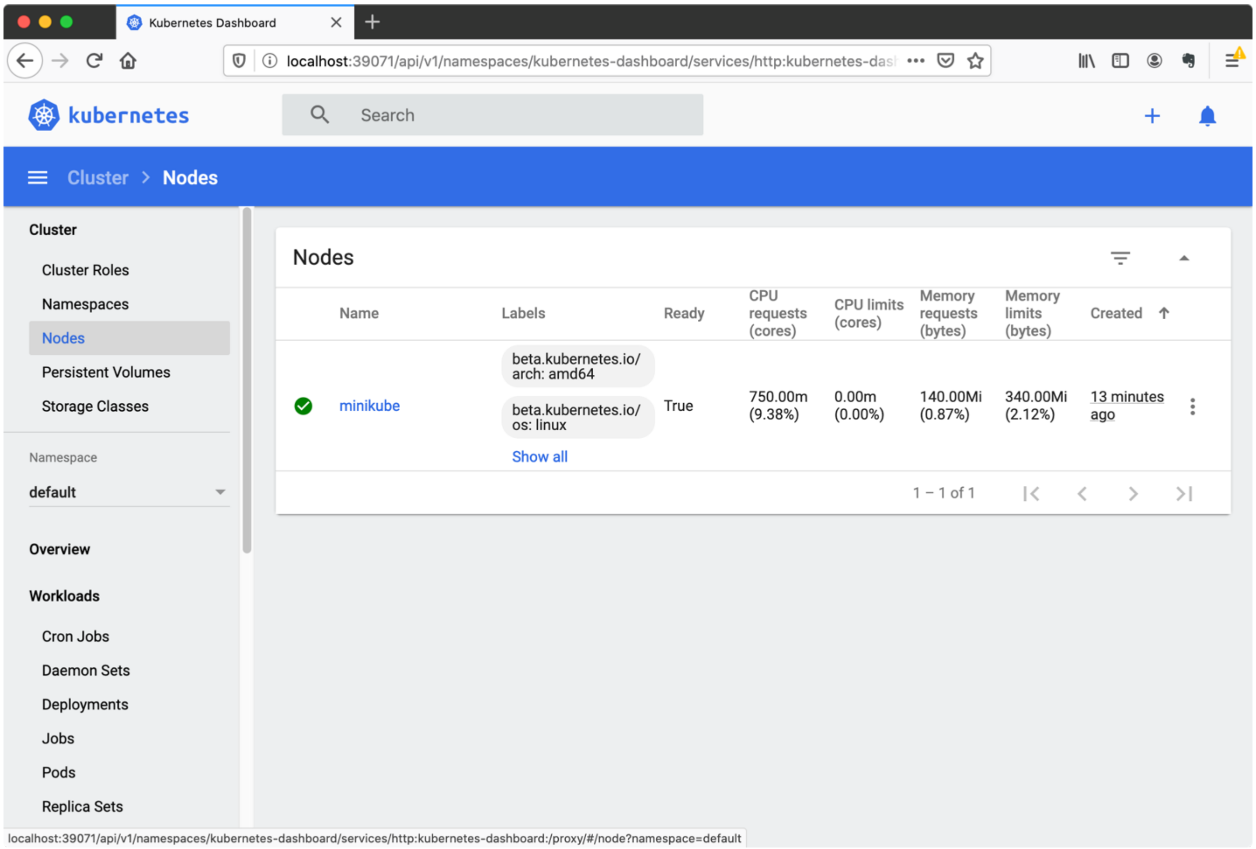
1 – 1 of 1

Config and Storage

Secrets

Name	Namespace	Labels	Type	Created
default-token-prnfb	default	-	kubernetes.io/servi account-token	11 minutes ago

1 – 1 of 1


Kubernetes Dashboard
localhost:39071/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-das
kubernetes
Search
Cluster > Nodes
Cluster
Cluster Roles
Namespaces
Nodes
Persistent Volumes
Storage Classes
Namespace
default
Overview
Workloads
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
localhost:39071/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/#/node?namespace=default

Nodes

	Name	Labels	Ready	CPU requests (cores)	CPU limits (cores)	Memory requests (bytes)	Memory limits (bytes)	Created ↑
✓	minikube	beta.kubernetes.io/ arch: amd64 beta.kubernetes.io/ os: linux Show all	True	750.00m (9.38%)	0.00m (0.00%)	140.00Mi (0.87%)	340.00Mi (2.12%)	13 minutes ago

1 – 1 of 1

To delete the ‘myminikube’ Service,

```
sslee777@us2004lts:~/myK8s$ kubectl delete services myminikube
service "myminikube" deleted
sslee777@us2004lts:~/myK8s$
```

To delete the ‘myminikube’ Deployment,

```
sslee777@us2004lts:~/myK8s$ kubectl delete deployment myminikube
deployment.apps "myminikube" deleted
sslee777@us2004lts:~/myK8s$
```

To stop your cluster,

```
sslee777@us2004lts:~/myK8s$ minikube stop
✋  Stopping "minikube" in docker ...
🛑  Powering off "minikube" via SSH ...
🛑  Node "minikube" stopped.
sslee777@us2004lts:~/myK8s$
```

To stop your cluster,

```
sslee777@us2004lts:~/myK8s$ minikube delete
🔥  Deleting "minikube" in docker ...
🔥  Deleting container "minikube" ...
🔥  Removing /home/sslee777/.minikube/machines/minikube ...
💀  Removed all traces of the "minikube" cluster.
sslee777@us2004lts:~/myK8s$
```

If you want to practice more in managing K8S cluster, alternative container runtimes, configuring kubernetes, Kubectl, networking, and volumes, see https://kubernetes.io/docs/setup/learning-environment/minikube/

4 Submit

Submit a word document (docx, doc, or PDF) to Canvas.
Assignments

- Run your code/scripts, take screenshots, and explain about it.
    a. Section 3.3 Installing Minikube
    b. Section 3.4 Installing Kubernetes with Minikube
    c. Section 3.5 Kubernetes Dashboard

If you have any problems or questions regarding this exercise, post messages in the 'Discussions' in Canvas. Posting exact answer codes for your assignment or links to an article that have exact answer codes are not allowed.

5 Reference (Note: Links may not work in PDF, try to copy&paste it)

1. Minikube vs. kind vs. k3s - What should I use?, https://brennerm.github.io/posts/minikube-vs-kind-vs-k3s.html
2. Install and Set Up kubectl, https://kubernetes.io/docs/tasks/tools/install-kubectl/
3. Install Minikube, https://kubernetes.io/docs/tasks/tools/install-minikube/
4. Installing Kubernetes with Minikube, https://kubernetes.io/docs/setup/learning-environment/minikube/
5. Simulating a data center with Kubernetes using virtual machines in your computer, https://medium.com/@shalithasuranga/simulating-a-data-center-with-virtual-machines-in-your-computer-76a434c02585
6. What Is SSH Port Forwarding, aka SSH Tunneling?, https://www.ssh.com/ssh/tunneling/example
7. How to install Kubernetes on Ubuntu 20.04 Focal Fossa Linux, https://linuxconfig.org/how-to-install-kubernetes-on-ubuntu-20-04-focal-fossa-linux