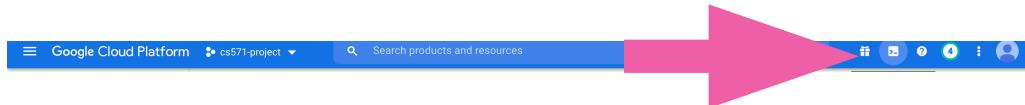


Maryam Taherzadeh:

Question2:

1. Login into your Google Cloud Platform. First you should create project and create Vm like we did for homework week one.
2. Activate Cloud Shell on Google Cloud Platform



3. Enter this command gcloud compute zones describe us-west2-a in the terminal and check your available CPU platform (please make sure Intel Haswell is in the list before you proceed)

```
Welcome to Cloud Shell! Type "help" to get started.  
Your Cloud Platform project in this session is set to cs571-project.  
Use "gcloud config set project [PROJECT_ID]" to change to a different project.  
taherzadeh19529@cloudshell:(cs571-project)$ gcloud compute zones describe us-west2-a  
availableCpuPlatforms:  
- Intel Cascade Lake  
- Intel Skylake  
- Intel Broadwell  
- Intel Haswell  
- Intel Ivy Bridge  
- Intel Sandy Bridge  
creationTimestamp: '1969-12-31T16:00:00.000-08:00'  
  
□ minikube-linux-amd... ▾ Show All X
```

4. Create a boot/staging disk by using below command in the terminal.

gcloud compute disks create stagingdisk --image-project ubuntu-os-cloud -image-family ubuntu-minimal-2004-lts --zone us-west2-a

You should see like this.

```
status: UP  
supportsPre: false  
taherzadeh19529@cloudshell:(cs571-project)$ gcloud compute disks create stagingdisk --image-project ubuntu-os-clou  
d --image-family ubuntu-minimal-2004-lts --zone us-west2-a  
  
Created [https://www.googleapis.com/compute/v1/projects/cs571-project/zones/us-west2-a/disks/stagingdisk].  
NAME ZONE SIZE_GB TYPE STATUS  
stagingdisk us-west2-a 10 pd-standard READY
```

5-Create a nested VM

```
gcloud compute images create nested-vm-image --source-disk=stagingdisk -source-disk-zone=us-west2-a -licenses=https://www.googleapis.com/compute/v1/projects/vm-options/glob al/licenses/enable-vmx
```

```
taherzadeh19529@cloudshell:~ (cs571-project)$ gcloud compute images create nested-vm-image --source-disk=stagingdisk --source-disk-zone=us-west2-a --licenses=https://www.googleapis.com/compute/v1/projects/vm-options/global/licenses/enable-vmx
Created [https://www.googleapis.com/compute/v1/projects/cs571-project/global/images/nested-vm-image].
NAME          PROJECT      FAMILY      DEPRECATED      STATUS
nested-vm-image  cs571-project      READY
```

6. Create VM Image

```
gcloud compute instances create nested-vm-image1 --zone us-west2-a --mincpu-platform "Intel Haswell" --machine-type n1-standard-4 --image nestedvm-image
```

In here you should see external ip, internal ip

```
taherzadeh19529@cloudshell:~ (cs571-project)$ gcloud compute instances create nested-vm-image1 --zone us-west2-a --mincpu-platform "Intel Haswell" --machine-type n1-standard-4 --image nested-vm-image
Created [https://www.googleapis.com/compute/v1/projects/cs571-project/zones/us-west2-a/instances/nested-vm-image1].
NAME          ZONE      MACHINE TYPE      PREEMPTIBLE      INTERNAL IP      EXTERNAL IP      STATUS
nested-vm-image1  us-west2-a  n1-standard-4      10.168.0.3      35.235.109.24      RUNNING
taherzadeh19529@cloudshell:~ (cs571-project)$
```

minikube-linux-amd... ^ Show All X

7. You should see the vm in here

Name	Zone	Recommendation	In use by	Internal IP	External IP	Conn
cs571-demo-project-new	us-west2-a			10.168.0.2 (nic0)	35.215.123.43	SSH
nested-vm-image1	us-west2-a			10.168.0.3 (nic0)	35.235.109.24	SSH

8. From SSH and select open in browser window. Then wait to connect



9. Verify your VM with lsb_release -a

```
Connected, host fingerprint: ssh-rsa 0 F3:39:ED:92:B6:14:81:40:8C:20:D5:90:72:C0
:F2:11:A9:28:FE:67:3B:56:A8:C4:A2:60:67:E0:9A:F9:15:47
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-1036-gcp x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

2 updates can be installed immediately.
2 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Last login: Fri Feb 19 16:53:40 2021 from 35.235.241.210
taherzadeh19529@nested-vm-image1:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 20.04.2 LTS
Release:        20.04
Codename:       focal
taherzadeh19529@nested-vm-image1:~$
```

10. Install kubectl

<https://kubernetes.io/docs/tasks/tools/install-kubectl/#install-kubectl>

```
sudo apt-get update && sudo apt-get install -y apt-transport-https gnupg2 curl 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
```

11. Check kubectl installation is successful with **kubectl version -client**

```
taherzadeh19529@nested-vm-image1:~$ kubectl version -client
Client Version: version.Info{Major:"1", Minor:"20", GitVersion:"v1.20.4", GitCommit:"e87da0bd6e03ec3fea7933c4b5263d151aaf07c"
, GitTreeState:"clean", BuildDate:"2021-02-18T16:12:00Z", GoVersion:"go1.15.8", Compiler:"gc", Platform:"linux/amd64"}
Server Version: version.Info{Major:"1", Minor:"20", GitVersion:"v1.20.2", GitCommit:"faecb196815e248d3ecfb03c680a4507229c2a56"
, GitTreeState:"clean", BuildDate:"2021-01-13T13:20:00Z", GoVersion:"go1.15.5", Compiler:"gc", Platform:"linux/amd64"}
taherzadeh19529@nested-vm-image1:~$
```

12. Install Minikube

```
curl -Lo minikube https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64 && chmod +x minikube
```

```
taherzadeh19529@nested-vm-image1:~$ curl -Lo minikube https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64 && chmod +x minikube
% Total    % Received % Xferd  Average Speed   Time     Time      Current
          Dload  Upload Total Spent   Left Speed
100 53.1M  100 53.1M    0     0  77.8M      0 --:--:-- --:--:-- 77.9M
taherzadeh19529@nested-vm-image1:~$
```

13. Move Minikube to /usr/bin/: **sudo cp minikube /usr/bin/**

14. Install docker with

```
sudo apt-get install curl wget apt-transport-https virtualbox virtualbox-extpack -y
sudo apt-get install docker.io -y
```

```
taherzadeh19529@nested-vm-image1:~$ sudo apt-get install curl wget apt-transport-https virtualbox virtualbox-ext-pac
Reading package lists... Done
Building dependency tree
Reading state information... Done
wget is already the newest version (1.20.3-1ubuntu1).
curl is already the newest version (7.68.0-1ubuntu2.4).
apt-transport-https is already the newest version (2.0.4).
virtualbox is already the newest version (6.1.16-dfsg-6~ubuntu1.20.04.1).
virtualbox-ext-pac is already the newest version (6.1.16-1~ubuntu1.20.04.1).
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
taherzadeh19529@nested-vm-image1:~$ sudo apt-get install docker.io -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
docker.io is already the newest version (19.03.8-0ubuntu1.20.04.2).
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
taherzadeh19529@nested-vm-image1:~$ sudo apt-get install docker.io -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
docker.io is already the newest version (19.03.8-0ubuntu1.20.04.2).
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
taherzadeh19529@nested-vm-image1:~$ docker --version
Docker version 19.03.8, build afacb8b7f0
taherzadeh19529@nested-vm-image1:~$
```

15. Check docker installation is successful with docker –version

```
curl: () failed to connect to localhost port 8080. Connection refused
taherzadeh19529@nested-vm-image1:/usr/bin/dockerimage$ docker --version
Docker version 19.03.8, build afacb8b7f0
taherzadeh19529@nested-vm-image1:/usr/bin/dockerimage$
```

16. Start docker with sudo service docker start

17. Install conntrack with
sudo apt-get install conntrack

```
taherzadeh19529@nested-vm-image1:~$ sudo service docker start
taherzadeh19529@nested-vm-image1:~$ sudo apt-get install conntrack
Reading package lists... Done
Building dependency tree
Reading state information... Done
conntrack is already the newest version (1:1.4.5-2).
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
```

18. Set permission with

sudo chown -R \$USER \$HOME/minikube; chmod -R u+wrw \$HOME/minikube

19. Go to /usr/bin: cd /usr/bin

20. Start Minikube: minikube start

```
taherzadeh19529@nested-vm-image1:~$ sudo chown -R $USER $HOME/minikube; chmod -R u+wrw $HOME/minikube
taherzadeh19529@nested-vm-image1:~$ minikube start
🌟 minikube v1.17.1 on Ubuntu 20.04 (kvm/amd64)
💡 Automatically selected the virtualbox driver. Other choices: none, ssh
⬇️ Downloading VM boot image ...
  > minikube-v1.17.0.iso.sha256: 65 B / 65 B [=====] 100.00% ? p/s 0s
  > minikube-v1.17.0.iso: 212.69 MiB / 212.69 MiB [ 100.00% 289.06 MiB p/s ] s
👍 Starting control plane node minikube in cluster minikube
⬇️ Downloading Kubernetes v1.20.2 preload ...
  > preloaded-images-k8s-v8-v1...: 491.22 MiB / 491.22 MiB 100.00% 217.72 M
🔥 Creating virtualbox VM (CPUs=2, Memory=3700MB, Disk=20000MB) ...
👍 Preparing Kubernetes v1.20.2 on Docker 20.10.2 ...
  • Generating certificates and keys ...
  • Booting up control plane .../
```

21. Install code editor: sudo apt-get install vim

22. From the home directory, create a new directory: mkdir dockerimage

23. Go to dockerimg directory: cd dockerimage/

24. Create app.js: vim app.js or sudo vim app.js

* if vim app.js does not work use this command:sudo vim app.js
* for exit use :wq!
* Pay attention about the correct code

```
const http = require('http');
const os = require('os');

console.log("Kubia server starting...");

var handler = function(request, response) {
    console.log("Received request from " +
        + request.connection.remoteAddress);
    response.writeHead(200);
    response.end("You've hit " + os.hostname() + "\n");
};

var www = http.createServer(handler);
www.listen(8080);
```

```
const http = require('http');
const os = require('os');

console.log("Kubia server starting...");

var handler = function(request, response) {
    console.log("Received request from " +
        + request.connection.remoteAddress);
    response.writeHead(200);
    response.end("You've hit " + os.hostname() + "\n");
};

var www = http.createServer(handler);
www.listen(8080);~  
~  
~  
~  
~  
~  
~
```

25. Create Dockerfile: vim Dockerfile or sudo vim Dockerfile

```
FROM node:7
ADD app.js /app.js
ENTRYPOINT ["node", "app.js"]
```

```
FROM node:7
ADD app.js /app.js
ENTRYPOINT ["node", "app.js"]
~
```

26. Build docker image: sudo docker build -t kubia .

```
taherzadeh19529@nested-vm-image1:~/dockerimg$ vim Dockerfile
taherzadeh19529@nested-vm-image1:~/dockerimg$ sudo docker build -t kubia .
Sending build context to Docker daemon 3.072kB
Step 1/3 : FROM node:7
7: Pulling from library/node
ad74af05f5a2: Pull complete
2b032b8bbe8b: Pull complete
a9a5b35f6ead: Pull complete
3245b5alc52c: Pull complete
afa075743392: Pull complete
9fb9f21641cd: Pull complete
3f40ad2666bc: Pull complete
49c0ed396649: Pull complete
Digest: sha256:a5c2c6ac8bc3fa372ac031ef60c45a285eeba7bce9ee9ed66dad3a01e29ab8d
Status: Downloaded newer image for node:7
--> d9aed20b68a4
Step 2/3 : ADD app.js /app.js
--> c4fd53ec8902
Step 3/3 : ENTRYPOINT ["node", "app.js"]
--> Running in 717469f894d7
Removing intermediate container 717469f894d7
--> 97f2d6bfbb3d
Successfully built 97f2d6bfbb3d
Successfully tagged kubia:latest
```

27. Run image on a container on localhost:

```
sudo docker run --name kubia-container -p 8080:8080 -d kubia
```

```
taherzadeh19529@nested-vm-image1:/usr/bin/dockerimage$ sudo docker run --name kubia-container -p 8080:8080 -d kubia
9806fce7e00225677f0dc84cae2cc77fcdbccc0c7d49a0a9210bc0f9ceb936df4
taherzadeh19529@nested-vm-image1:/usr/bin/dockerimage$
```

28. Check if it is running: curl localhost:8080

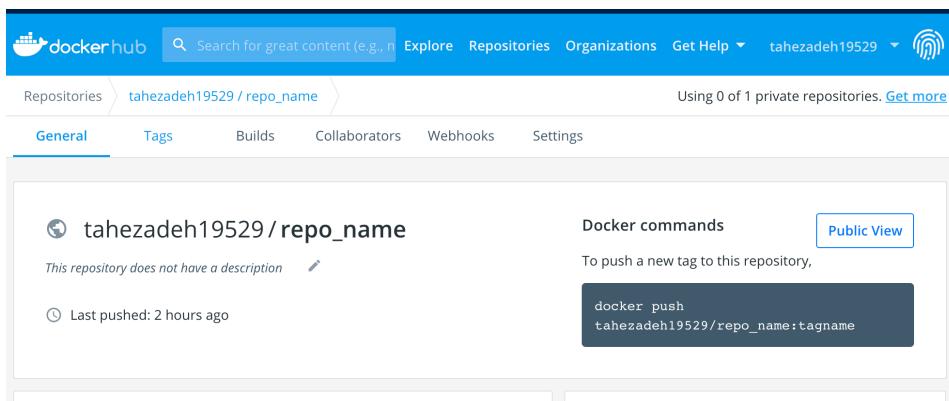
```
taherzadeh19529@nested-vm-image1:/usr/bin/dockerimage$ curl localhost:8080
You've hit 9806fce7e002
```

29. Stop container: sudo docker stop kubia-container

```
taherzadeh19529@nested-vm-image1:/usr/bin/dockerimage$ sudo docker stop kubia-container
```

30. Go to hub.docker.com and create an account

after creating the account go to hub.docker.com
then go to Repositories > Create repository



31. Login to docker hub:

sudo docker login

```
taherzadeh19529@nested-vm-image1:/usr/bin/dockerimage$ sudo docker stop kubia-container
kubia-container
taherzadeh19529@nested-vm-image1:/usr/bin/dockerimage$ sudo docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
Login Succeeded
```

32. Tag kubia docker image: **sudo docker tag kubia username/repository name**
33-Push docker image to repository: **sudo docker push username/ repository**

```
taherzadeh19529@nested-vm-image1:/usr/bin/dockerimage$ sudo docker tag kubia tahezadeh19529/repo_name
taherzadeh19529@nested-vm-image1:/usr/bin/dockerimage$ sudo docker push tahezadeh19529/repo_name
The push refers to repository [docker.io/tahezadeh19529/repo_name]
dc8fb37c749c: Pushed
ab90d83fa34a: Layer already exists
8ee318e54723: Layer already exists
e6695624484e: Layer already exists
da59b99bb03b: Layer already exists
5616a6292c16: Layer already exists
f3ed6cb59a0: Layer already exists
654f45ecb7e3: Layer already exists
2c40c66f7667: Layer already exists
latest: digest: sha256:05e62f57f5ff1c1cadab1f32fdf5b80555503df330a441a57d15f08ff7dda3d size: 2213
taherzadeh19529@nested-vm-image1:/usr/bin/dockerimage$
```

34. Go to hub.docker.com/repositories to see the result

The screenshot shows the Docker Hub interface. At the top, there's a search bar and navigation links for Explore, Repositories, Organizations, Get Help, and a user account. Below that, the user's repositories are listed, with 'tahezadeh19529 / repo_name' selected. The 'Tags' tab is active, showing a single tag named 'latest'. The tag details are as follows:

TAG	DIGEST	OS/ARCH	LAST PULL	COMPRESSED SIZE
latest	8be51b1e79b0	linux/amd64	2 hours ago	251.45 MB

A 'docker pull' button is located next to the tag details.

*Note :when you see the error like connection refuse at point 27 : curl localhost:8080

Try below steps :

1. sudo vi app.js
2. Use this code

```
const http = require('http');
const os = require('os');

console.log("Kubia server starting...");

var handler = function(request, response) {
    console.log("Received request from "
        + request.connection.remoteAddress);
    response.writeHead(200);
    response.end("You've hit " + os.hostname() + "\n");
};

var www = http.createServer(handler);
www.listen(8080);
```

3. save the file
4. sudo docker rmi --force kubia
5. sudo docker rm kubia-container
6. sudo docker run --name kubia-container -p 8080:8080 -d kubia
7. curl localhost:8080