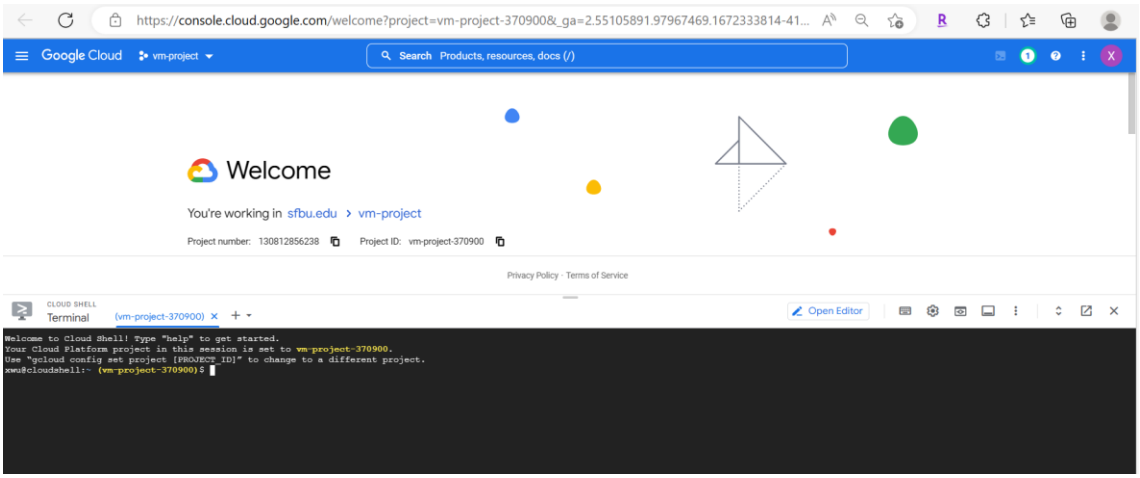
Using Kubernetes to send JSON data from node.js server

1. On your GCP, open a project, and then create a compute engine instance using command lines.

   Step1. Launch a Cloud Shell in Google Cloud Console, click the Activate Cloud Shell.
   Enter link in browser



   Step2. Check the
   $ gcloud compute zones describe us-west1-a

```
xwu@cloudshell:~ (vm-project-370900)$ gcloud compute zones describe us-west1-a
availableCpuPlatforms:
- Intel Ice Lake
- Intel Cascade Lake
- Intel Skylake
- Intel Broadwell
- Intel Haswell
- Intel Ivy Bridge
- Intel Sandy Bridge
- AMD Milan
- AMD Rome
creationTimestamp: '1969-12-31T16:00:00.000-08:00'
description: us-west1-a
id: '2210'
kind: compute#zone
name: us-west1-a
region: https://www.googleapis.com/compute/v1/projects/vm-project-370900/regions/us-west1
selfLink: https://www.googleapis.com/compute/v1/projects/vm-project-370900/zones/us-west1-a
status: UP
supportsPzs: false
xwu@cloudshell:~ (vm-project-370900)$
```

   Step3. Create a boot staging disk
   $ gcloud compute disks create stagingdisk --image-project ubuntu-os-cloud --image-family ubuntu-minimal-2004-lts --zone us-west1-a

```
xwu@cloudshell:~ (vm-project-370900)$ gcloud compute disks create stagingdisk --image-pro
ject ubuntu-os-cloud --image-family ubuntu-minimal-2004-lts --zone us-west1-a
Created [https://www.googleapis.com/compute/v1/projects/vm-project-370900/zones/us-west1-
a/disks/stagingdisk].
NAME: stagingdisk
ZONE: us-west1-a
SIZE_GB: 10
TYPE: pd-standard
STATUS: READY
xwu@cloudshell:~ (vm-project-370900)$
```

Step4. Create an image

$ gcloud compute images create nested-vm-image --source-disk=stagingdisk
--source-disk-zone=us-west1-a
--licenses=https://www.googleapis.com/compute/v1/projects/vm-options/glob
al/licenses/enable-vmx

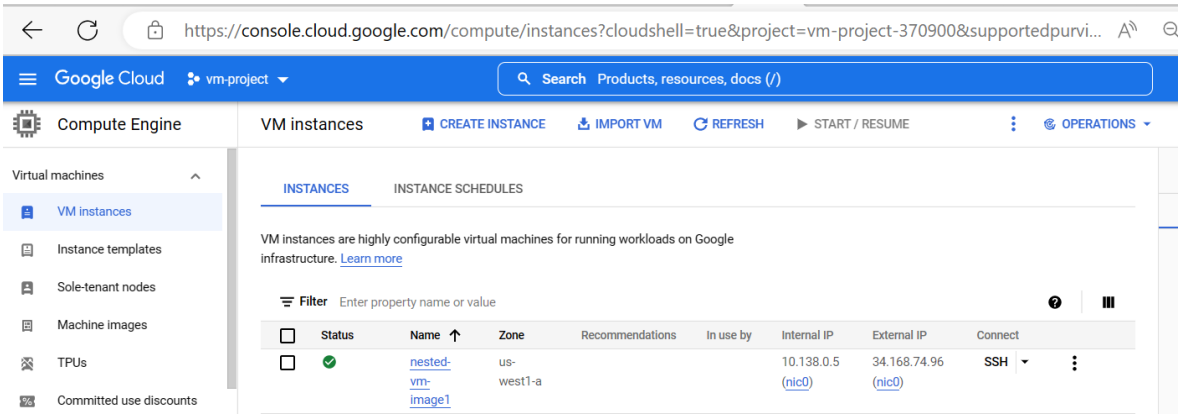
```
xwu@cloudshell:~ (vm-project-370900)$ gcloud compute images create nested-vm-image --sour
ce-disk=stagingdisk --source-disk-zone=us-west1-a --licenses=https://www.googleapis.com/c
ompute/v1/projects/vm-options/global/licenses/enable-vmx
Created [https://www.googleapis.com/compute/v1/projects/vm-project-370900/global/images/n
ested-vm-image].
NAME: nested-vm-image
PROJECT: vm-project-370900
FAMILY:
DEPRECATED:
STATUS: READY
xwu@cloudshell:~ (vm-project-370900)$
```

Step5. Create a vm instance

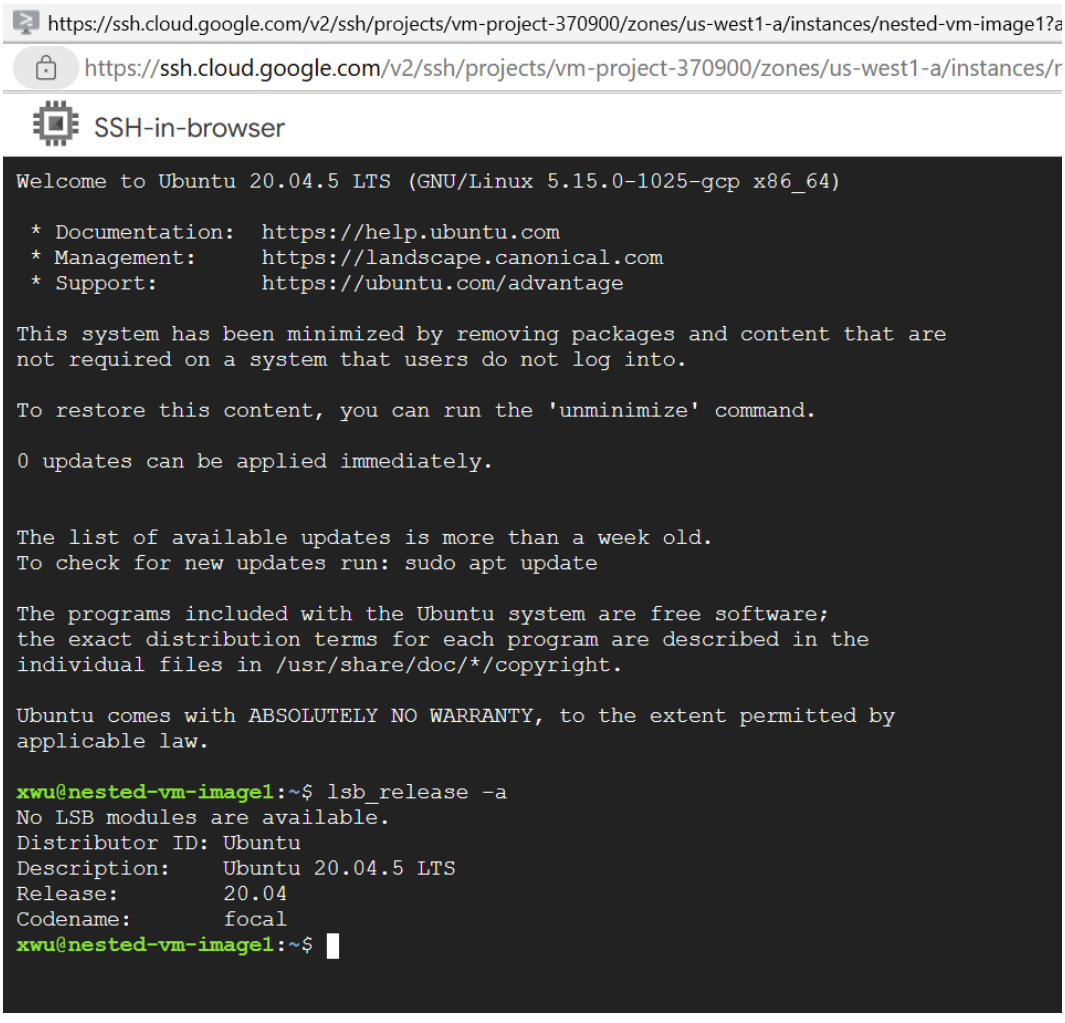
$ gcloud compute instances create nested-vm-image1 --zone us-west1-b
--min-cpu-platform "Intel Haswell" --machine-type n1-standard-4 --image
nested-vm-image

```
xwu@cloudshell:~ (vm-project-370900)$ gcloud compute instances create nested-vm-image1 --
zone us-west1-a --min-cpu-platform "Intel Haswell" --machine-type n1-standard-4 --image n
ested-vm-image
Created [https://www.googleapis.com/compute/v1/projects/vm-project-370900/zones/us-west1-
a/instances/nested-vm-image1].
NAME: nested-vm-image1
ZONE: us-west1-a
MACHINE_TYPE: n1-standard-4
PREEMPTIBLE:
INTERNAL_IP: 10.138.0.5
EXTERNAL_IP: 34.168.74.96
STATUS: RUNNING
xwu@cloudshell:~ (vm-project-370900)$
```

Step6. Verify the compute engine vm instance is available on the Google Cloud Console.



SSH → Open in browser window
$ lsb_release -a



2. Install kubectl and Minikube (Please refer to this link)

   Install kubectl:

   $ sudo apt-get update

   $ sudo apt-get install -y ca-certificates curl

   $ curl https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key --keyring /usr/share/keyrings/cloud.google.gpg add -

   $ echo "deb [signed-by=/etc/apt/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list

   $ sudo snap install kubectl –classic

   $ kubectl version –client

```
xwu@nested-vm-image1:~$ sudo snap install kubectl --classic
kubectl 1.26.0 from Canonical✓ installed
xwu@nested-vm-image1:~$ kubectl version --client
WARNING: This version information is deprecated and will be replaced with the output fro
m kubectl version --short.  Use --output=yaml|json to get the full version.
Client Version: version.Info{Major:"1", Minor:"26", GitVersion:"v1.26.0", GitCommit:"b46
a3f887ca979b1a5d14fd39cb1af43e7e5d12d", GitTreeState:"clean", BuildDate:"2022-12-09T16:2
3:44Z", GoVersion:"go1.19.4", Compiler:"gc", Platform:"linux/amd64"}
Kustomize Version: v4.5.7
```

   Check that kubectl is properly configured by getting the cluster state:

   $ `kubectl cluster-info`

```
xwu@nested-vm-image1:~$ kubectl cluster-info
Kubernetes control plane is running at https://192.168.59.100:8443
CoreDNS is running at https://192.168.59.100:8443/api/v1/namespaces/kube-system/services
/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
xwu@nested-vm-image1:~$
```

   Install minikube:

   $ curl -Lo minikube https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64

   $ chmod +x minikube

   Move minikube to /usr/bin

   $ sudo cp minikube /usr/bin/

   Install Docker

   $ sudo apt-get install curl wget apt-transport-https virtualbox virtualbox-ext-pack -y

   $ sudo apt-get install docker.io -y

   $ docker –version

```
xwu@nested-vm-image1:~$ docker --version
Docker version 20.10.12, build 20.10.12-0ubuntu2~20.04.1
xwu@nested-vm-image1:~$
```

   Start Docker, install conntrack and set permission

   $ sudo service docker start

$ sudo apt-get install conntrack
$ sudo chown -R $USER $HOME/minikube; chmod -R u+wrx $HOME/minikube

Start minikube
$ cd /usr/bin
$ minikube start

```
xwu@nested-vm-image1:/usr/bin$ minikube start
😄  minikube v1.28.0 on Ubuntu 20.04 (kvm/amd64)
✨  Automatically selected the virtualbox driver. Other choices: ssh, none
💿  Downloading VM boot image ...
    > minikube-v1.28.0-amd64.iso....:  65 B / 65 B [---------] 100.00% ? p/s 0s
    > minikube-v1.28.0-amd64.iso:  274.45 MiB / 274.45 MiB  100.00% 288.81 MiB
👍  Starting control plane node minikube in cluster minikube
💾  Downloading Kubernetes v1.25.3 preload ...
    > preloaded-images-k8s-v18-v1...:  385.44 MiB / 385.44 MiB  100.00% 264.79
🔥  Creating virtualbox VM (CPUs=2, Memory=3700MB, Disk=20000MB) ...
🐳  Preparing Kubernetes v1.25.3 on Docker 20.10.20 ...
    ▪ Generating certificates and keys ...
    ▪ Booting up control plane ...
    ▪ Configuring RBAC rules ...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: default-storageclass, storage-provisioner
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by
default
xwu@nested-vm-image1:/usr/bin$
```

3. Run application
   Install vim editor,
   $ cd
   $ sudo apt-get install vim

```
xwu@nested-vm-image1:~$ sudo apt-get install vim
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libpolkit-gobject-1-0
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libcanberra0 libltdl7 libtdb1 libvorbisfile3 vim-common vim-runtime
```

Create application files
$ cd
$ mkdir dockerimg
$ vi student_info.js

```
const os = require ( 'os' );
var http = require('http');
//console.log( "Kubia server starting..." );

function parseStudentInfo(id) {
  if (id == "11111")
    return {"id": 11111, "name":"Bruce Lee", "score":84}
  else if (id == "22222")
    return {"id": 22222, "name":"Jackie Chen", "score":93}
  else if (id == "33333")
    return {"id": 33333, "name":"Jet Li", "score":88}
  else
    return {}
}
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/json'});
  let myURL = req.url;
  const urlArray = myURL.split("=");
  let stdId = urlArray[1];
  let stdInfo = parseStudentInfo(stdId);
  res.write(JSON.stringify(stdInfo));
  res.end("\nYou've hit " + os.hostname() + "\n");
}).listen(8000);
```

$ vi Dockerfile

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

$ sudo docker build -t kubia .

```
xwu@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
3245b5a1c52c: Pull complete
afa075743392: Pull complete
9fb9f21641cd: Pull complete
3f40ad2666bc: Pull complete
49c0ed396b49: Pull complete
Digest: sha256:af5c2c6ac8bc3fa372ac031ef60c45a285eeba7bce9ee9ed66dad3a01e29ab8d
Status: Downloaded newer image for node:7
 ---> d9aed20b68a4
Step 2/3 : ADD student_info.js /student_info.js
 ---> 03f6b01a8cc2
Step 3/3 : ENTRYPOINT ["node", "student_info.js"]
 ---> Running in db1d2c1443c0
Removing intermediate container db1d2c1443c0
 ---> 0638ff2345a0
Successfully built 0638ff2345a0
Successfully tagged kubia:latest
```

$ sudo docker run --name kubia-container -p 8000:8000 -d kubia

```
xwu@nested-vm-image1:~/dockerimg$ sudo docker run --name kubia-container -p 8000:8000 -d
 kubia
30a8576113c0ef25cef5f1818a6fccd492a6938570069b52b8c41807fae06281
```

Check if it's running correctly

$ curl localhost:8000/api/score?student_id=11111

```
xwu@nested-vm-image1:~/dockerimg$ curl localhost:8000/api/score?student_id=11111
{"id":11111,"name":"Bruce Lee","score":84}
You've hit 1a28eb35ca9e
xwu@nested-vm-image1:~/dockerimg$ curl localhost:8000/api/score?student_id=33333
{"id":33333,"name":"Jet Li","score":88}
You've hit 1a28eb35ca9e
xwu@nested-vm-image1:~/dockerimg$ curl localhost:8000/api/score?student_id=22222
{"id":22222,"name":"Jackie Chen","score":93}
You've hit 1a28eb35ca9e
xwu@nested-vm-image1:~/dockerimg$ curl localhost:8000/api/score?student_id=00000
{}
You've hit 1a28eb35ca9e
```

Stop the container

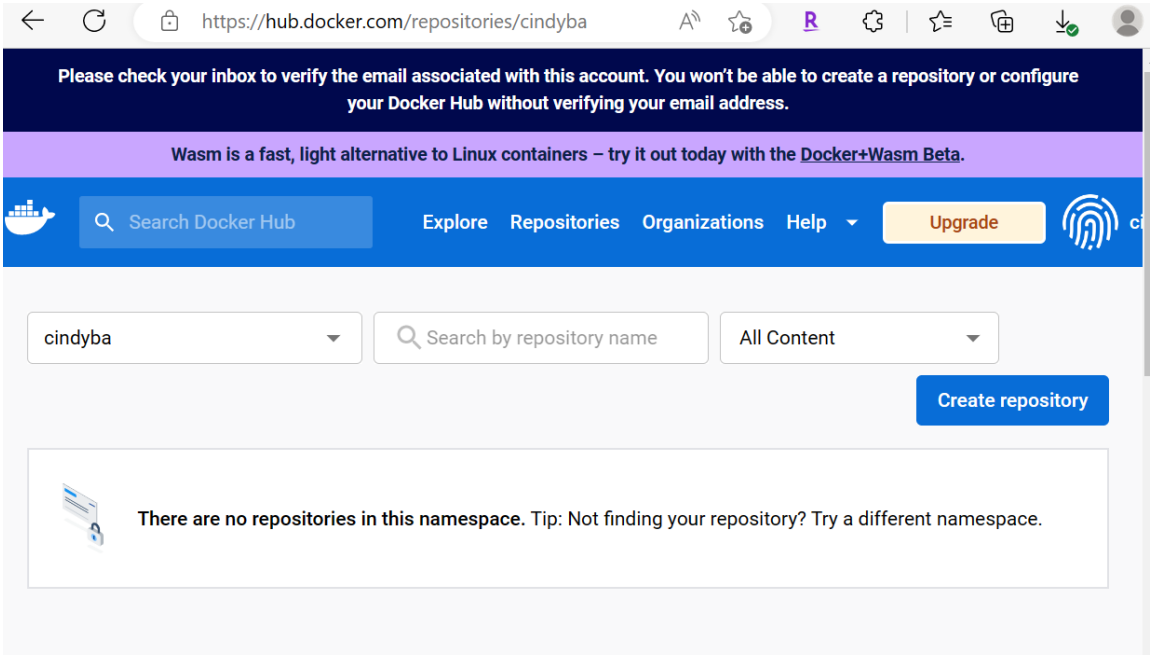
$ sudo docker stop kubia-container

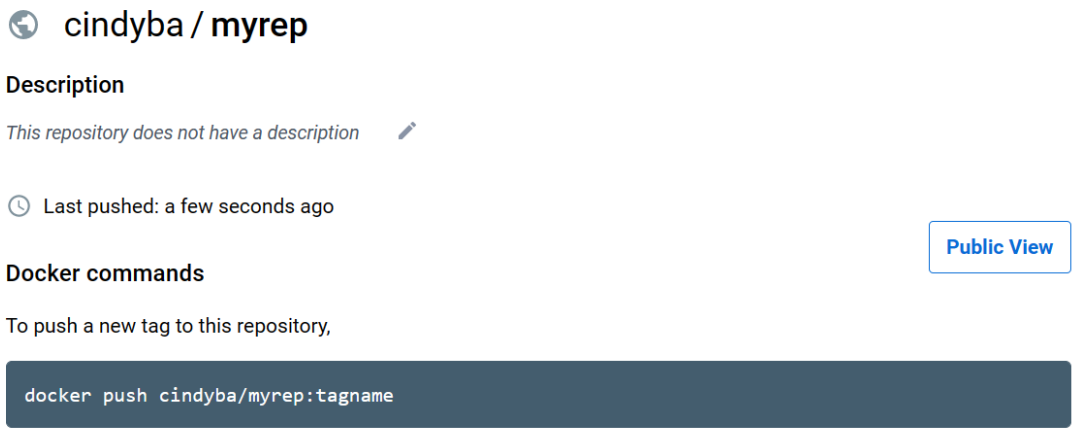
Other possible useful commands you may need:

$ sudo docker rm /kubia-container

$ sudo docker build --no-cache -t kubia .

4. Upload the docker image into the docker Hub Container Image Library
   Create an account by link [Docker Hub Container Image Library | App Containerization](), and create repository.



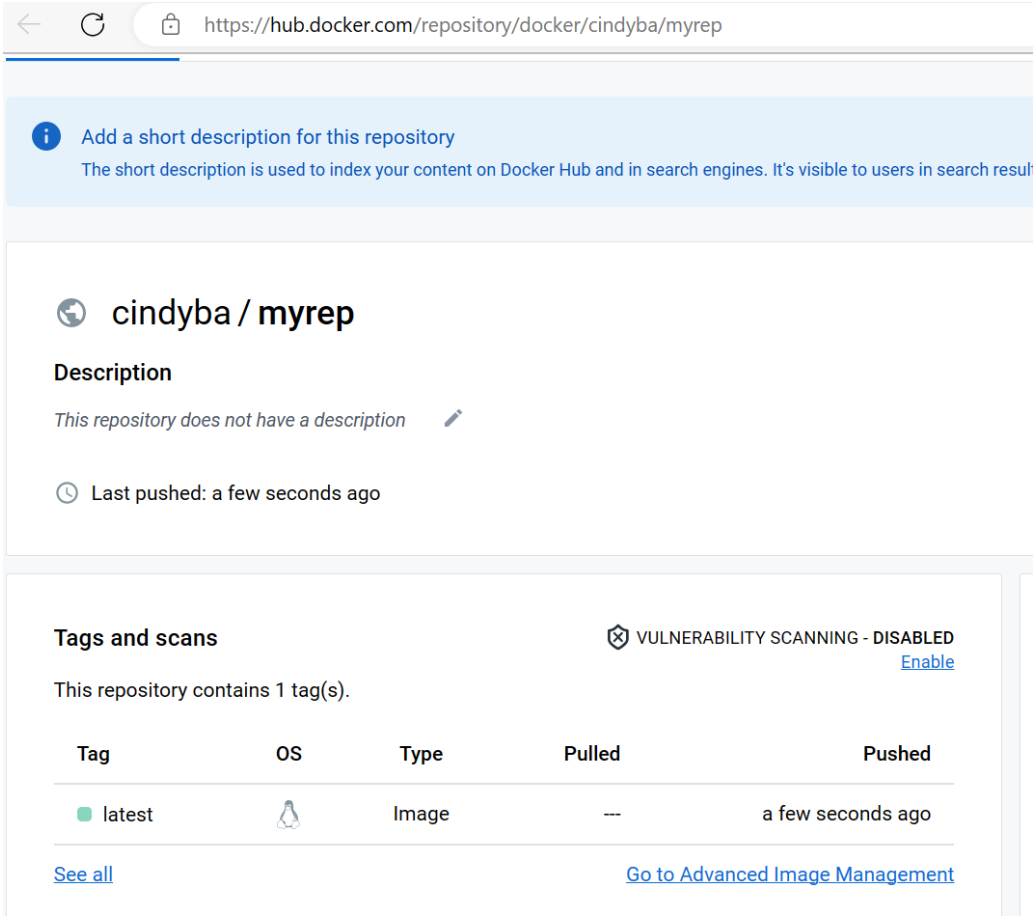


Tag the kubia docker image we just created
$ sudo docker tag kubia cindyba/myrep
$ sudo docker login -u=cindyba -p=123456789
$ sudo docker push cindyba/myrep

```
xwu@nested-vm-image1:~/dockerimg$ sudo docker tag kubia cindyba/myrep
xwu@nested-vm-image1:~/dockerimg$ sudo docker login -u=cindyba -p=1234Qwer!
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
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
xwu@nested-vm-image1:~/dockerimg$ sudo docker push cindyba/myrep
Using default tag: latest
The push refers to repository [docker.io/cindyba/myrep]
0c7989f6ec09: Pushed
ab90d83fa34a: Mounted from library/node
8ee318e54723: Mounted from library/node
e6695624484e: Mounted from library/node
da59b99bbd3b: Mounted from library/node
5616a6292c16: Mounted from library/node
f3ed6cb59ab0: Mounted from library/node
654f45ecb7e3: Mounted from library/node
2c40c66f7667: Mounted from library/node
latest: digest: sha256:240afdd86b75640ec1e294e13b3c62db5f0f44fee5334eea4c8d89ab7876a17f
size: 2213
xwu@nested-vm-image1:~/dockerimg$
```

Back to the browser



DONE!!!