

# Assignment Report

Student Name: Ziqi Pei

Student ID:33429472

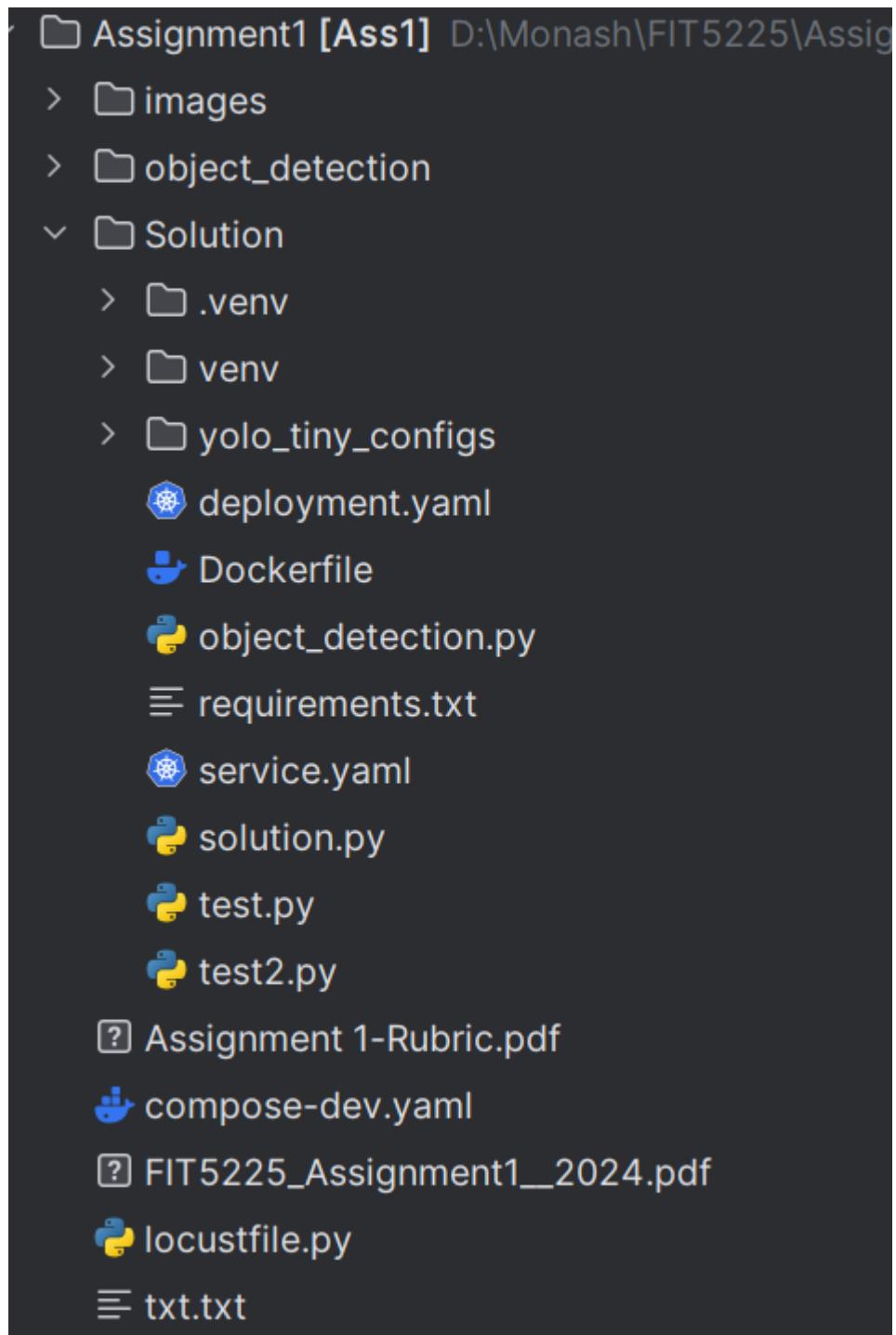
Tutor: Jay Zhao,**Jinchun Du (Goldi),Qifan Deng**

## Task 1: Prepare and Develop Web Service

The goal of this task is to write a Python web service that receives an image object in JSON format, processes the image using YOLO and OpenCV, and returns a JSON object containing a list of detected objects.

### 1 .Environmental preparation

First, create a PyCharm working folder and install all necessary dependent libraries such as numpy, Flask and OpenCV.



Install these libraries using the following commands:

```
pip install numpy
pip install opencv-python
pip install falsk
```

## Task2 Web Service

### 2.1 Modify object\_detection.py

Make the following modifications to object\_detection.py:

- (1) Import new modules: base64 and threading.
- (2) Add a thread lock in the do\_prediction function to ensure thread safety for concurrent requests.
- (3) Process detection results to produce JSON output containing label, accuracy, and bounding box information.
- (4) Add transfer\_base64 function for decoding base64 encoded images and calling do\_prediction function for object detection.
- (5) Add exception handling in main function.

```
# Import new modules
import base64
import threading

# Add thread lock to do_prediction function
def do_prediction(image, net, LABELS):
    with lock:
        (H, W) = image.shape[:2]
        # Code omitted...

        # Add handling of detection results to generate JSON
        res = []
        if len(idxs) > 0:
            for i in idxs.flatten():
                print("detected item:{}, accuracy:{}, X:{}, Y:{}".format(
                    classIDs[i], confidences[i], boundingBox[0][i],
                    boundingBox[1][i]))
                res.append({
                    "label": LABELS[classIDs[i]],
                    "accuracy": confidences[i],
                    "rectangle": {
```

```

        "height": boxes[i][3],
        "left": boxes[i][0],
        "top": boxes[i][1],
        "width": boxes[i][2]
    }
)
return res

# Add new function transfer_base64
def transfer_base64(image):
    img_npyarray = np.fromstring(base64.b64decode(image), np.u
    img_mat = cv2.imdecode(img_npyarray, cv2.IMREAD_COLOR)
    image = cv2.cvtColor(img_mat, cv2.COLOR_BGR2RGB)

    return do_prediction(image, nets, Lables)

# Add exception handling in main function
def main():
    try:
        imagefile = str(sys.argv[2])
        img = cv2.imread(imagefile)
        npimg=np.array(img)
        image=npimg.copy()
        image=cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
        nets = load_model(CFG, Weights)
        do_prediction(image, nets, Lables)

    except Exception as e:
        print("Exception  {}".format(e))

# Initialize thread lock
lock = threading.Lock()

# Update main function usage
if __name__ == '__main__':
    print(str(sys.argv))
    yolo_path  = str(sys.argv[1])
    if len(sys.argv) != 3:

```

```
    raise ValueError("Argument list is wrong. Please use  
                      format("python iWebLens_server.py",  
main()
```

## 2.2 Test object\_detection.py

Test the modified object\_detection.py using a local image file. Run the following command:

```
PS D:\Monash\FIT5225\Assignment1\object_detection> cd object_<br/>PS D:\Monash\FIT5225\Assignment1\object_detection\object_dete
```

Confirm that the image can be analyzed successfully.

```
C:\Users\pzqfr\AppData\Local\Programs\Python\Python312\python.exe D:\Monash\FIT5225\Assignment1\object_detection\object_detection\object_detection.py D:\Monash\FIT5225\Assignment1\Solution\yolo_t  
['D:\\Monash\\FIT5225\\Assignment1\\object_detection\\object_detection\\object_detection.py', 'D:\\Monash\\FIT5225\\Assignment1\\Solution\\yolo_tiny_configs', 'D:\\Monash\\FIT5225\\Assignment1\\i  
D:\\Monash\\FIT5225\\Assignment1\\Solution\\yolo_tiny_configs  
[INFO] loading YOLO from disk...  
[INFO] YOLO took 0.103213 seconds  
detected item:person, accuracy:0.9833722114562988, X:73, Y:13, width:121, height:144  
  
Process finished with exit code 0
```

## 2.3 Build Flask web service

Create a Flask web service named solution.py. Import the necessary libraries and define routes.

```
from flask import Flask  
  
app = Flask(__name__)  
  
@app.route("/")  
def hello_world():  
    return "<h1>This program is develop by Ziqi Pei for fit522</h1>"  
  
if __name__=='__main__':  
    app.run(debug=True, host='0.0.0.0', threaded=True, port=5000)
```

import flask library and know the relation

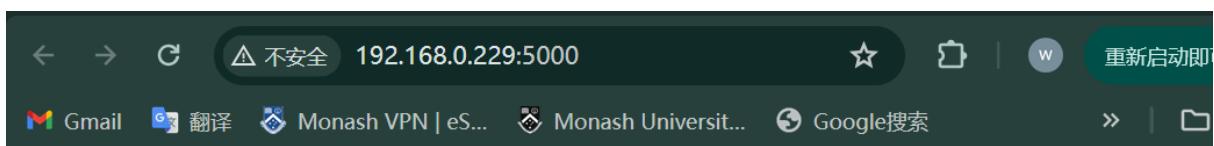
```
pip install flask  
pip freeze
```

```
● PS D:\Monash\FIT5225\Assignment1\demo> pip freeze  
blinker==1.7.0  
click==8.1.7  
colorama==0.4.6  
Flask==3.0.2  
itsdangerous==2.1.2  
Jinja2==3.1.3  
MarkupSafe==2.1.5  
Werkzeug==3.0.2
```

## 2.4 Test the web service locally

Run the Flask service and visit <http://192.168.0.229:5000/> in the browser to test.

```
[INFO] loading YOLO from disk...  
* Serving Flask app 'solution'  
* Debug mode: on  
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.  
* Running on all addresses (0.0.0.0)  
* Running on http://127.0.0.1:5000  
* Running on http://192.168.0.229:5000  
Press CTRL+C to quit  
* Restarting with stat  
yolo_tiny_configs\coco.names
```



# This program is developed by Ziqi Pei for FIT5225 2024 S1 Assignment 1!

## 2.5 Integration object\_detection.py

Modify solution.py, add the /api/transfer POST route, call the transfer\_base64 function of object\_detection.py in this route to detect the object, and return a JSON response containing the detection results.

```

from flask import Flask, request, jsonify
import object_detection

app = Flask(__name__)

@app.route("/")
def hello_world():
    return "<h1>This program is developed by Ziqi Pei for FIT</h1>"

# get user input for detection and return result
@app.route('/api/transfer', methods=['POST'])
def transfer():
    data = request.get_json()
    id = data['id']
    img_base64 = data["image"]
    res = object_detection.transfer_base64(img_base64)
    print("The process id: {}".format(id))
    return jsonify({"id": id, "objects": res})

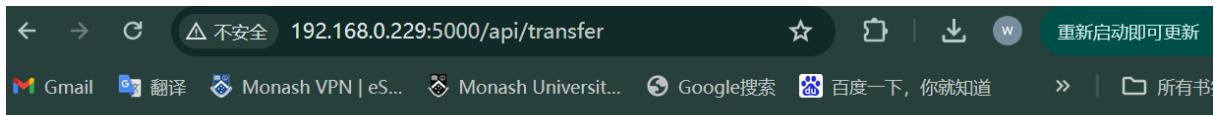
if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0', threaded=True, port=5000)

```

Run the command and get such result

```
http://192.168.0.229:5000/api/transfer
```

We can see that this web page exists on localhost, but this is a post command, so we need to use the postman test function.



## Method Not Allowed

The method is not allowed for the requested URL.

### 2.6 Test the integrated web service

Use a tool such as Postman to send a JSON request containing the base64-encoded image to /api/transfer and confirm that the service handles the request correctly and returns the expected JSON response.

Example request:

```
{
  "id": 123,
  "image": " /9j/4AAQSkZJRgABAQEASABIAAD/2wBDAEBAQEBAQEBAQI
  2wBDAQEBAQICAgQCAgQIBQQFCAgICAgICAgICAgICAgICAgICAgICAgIC
  8QAHwAAAAYDAQEBAAAAAAAAAAAAwQFBgcIAgkKAQAL
  /8QATRAAAgEDAwIFAgQDBAkDAgAPAQIDBAURBhIhAACIEyIxQRRRCTJhc
  y35r89m/0e1poV06Mk7eMcj3x+nvx1k61SDK9MYwEBMEUwacxuwCAndxk
  ...
}
```

Postman sed a post raw json format

The screenshot shows the Postman interface with a collection named "My first collection". A POST request is being made to `http://127.0.0.1:5000/api/transfer`. The request body is a JSON object with two entries:

```

{
  "id": 123,
  "objects": [
    {
      "accuracy": 0.9590027332305908,
      "label": "person",
      "rectangle": {
        "height": 277,
        "left": 300,
        "top": 211,
        "width": 134
      }
    },
    {
      "accuracy": 0.915320098400116,
      "label": "elephant",
      "rectangle": {
        "height": 243,
        "left": -6,
        "top": 219,
        "width": 250
      }
    }
  ]
}

```

Sample response:

```
{
  "id": 123,
  "objects": [
    {
      "accuracy": 0.9590027332305908,
      "label": "person",
      "rectangle": {
        "height": 277,
        "left": 300,
        "top": 211,
        "width": 134
      }
    },
    {
      "accuracy": 0.915320098400116,
      "label": "elephant",
      "rectangle": {
        "height": 243,
        "left": -6,
        "top": 219,
        "width": 250
      }
    }
  ]
}
```

```
        "width": 232
    }
}
]
}
```

## 2.7 Summarize

In this task, I successfully developed a web service using the Flask framework. The service is able to handle JSON requests containing base64 encoded images, create a thread for each request, use YOLO and OpenCV for object detection, and return a JSON response in the specified format. I used the yolov3-tiny model and pre-trained weights to implement a fast and reliable object detection service.

## Task 3: Dockerfile

The goal of this task is to build a Docker image for the object detection web service. A Dockerfile is a text file that contains all the ordered commands required to build a Docker image. Its role is to automate and simplify the image building process while ensuring the consistency and repeatability of the building process.

### 3.1 Dockerfile content

Below is the contents of our Dockerfile, with an explanation of each instruction:

```
# Use the specified Python version as the base image
#FROM python:3.12.2
FROM python:3.12.2

# Set the working directory for the builder stage
WORKDIR /link

# Copy the requirements.txt file to the working directory
COPY requirements.txt .

# Install necessary system dependencies
```

```

RUN apt-get update && apt-get install -y \
    libgl1-mesa-glx \
    libglib2.0-0 \
    libsm6 \
    libxrender1 \
    libxext6

# Install OpenCV dependencies
RUN apt-get install -y \
    libopencv-dev \
    python3-opencv

# Install Python packages
RUN pip install --no-cache-dir -r requirements.txt && \
    rm -rf ~/.cache/pip/*

# Set the working directory for the final stage
WORKDIR /link

# Copy additional files and directories
# Copy yolo_tiny_configs directory
COPY yolo_tiny_configs /opt/ass/yolo_tiny_configsdoc
COPY object_detection.py .
COPY solution.py .

# Expose the port
EXPOSE 5000

# Define the command to run the application
CMD ["python3", "solution.py", ""]

```

In order to optimize the Dockerfile, I took the following measures:

- (1)Install only necessary system dependencies and Python packages and avoid installing unnecessary packages.
- (2)Install Python packages using the --no-cache-dir option and delete the pip

cache after installation to reduce image size.

(3)Copy the application code as the last step to take advantage of Docker's caching mechanism and speed up the build.

### 3.2 Dockerfile requirement

The requirements.txt file lists all the Python dependencies required by the application. The contents of this file are as follows:

```
blinker==1.7.0
Brotli==1.1.0
certifi==2020.6.20
cffi==1.16.0
charset-normalizer==3.3.2
click==8.1.7
colorama==0.4.4
ConfigArgParse==1.7
Flask==3.0.3
Flask-Cors==4.0.0
Flask-Login==0.6.3
gevent==24.2.1
geventhttpclient==2.0.12
greenlet==3.0.3
idna==3.3
itsdangerous==2.1.2
Jinja2==3.1.3
locust==2.25.0
MarkupSafe==2.1.5
msgpack==1.0.8
numpy==1.26.4
opencv-python==4.7.0.72
opencv-python-headless==4.7.0.72
psutil==5.9.8
pycparser==2.22
pyzmq==25.1.2
roundrobin==0.0.4
setuptools==59.6.0
six==1.16.0
```

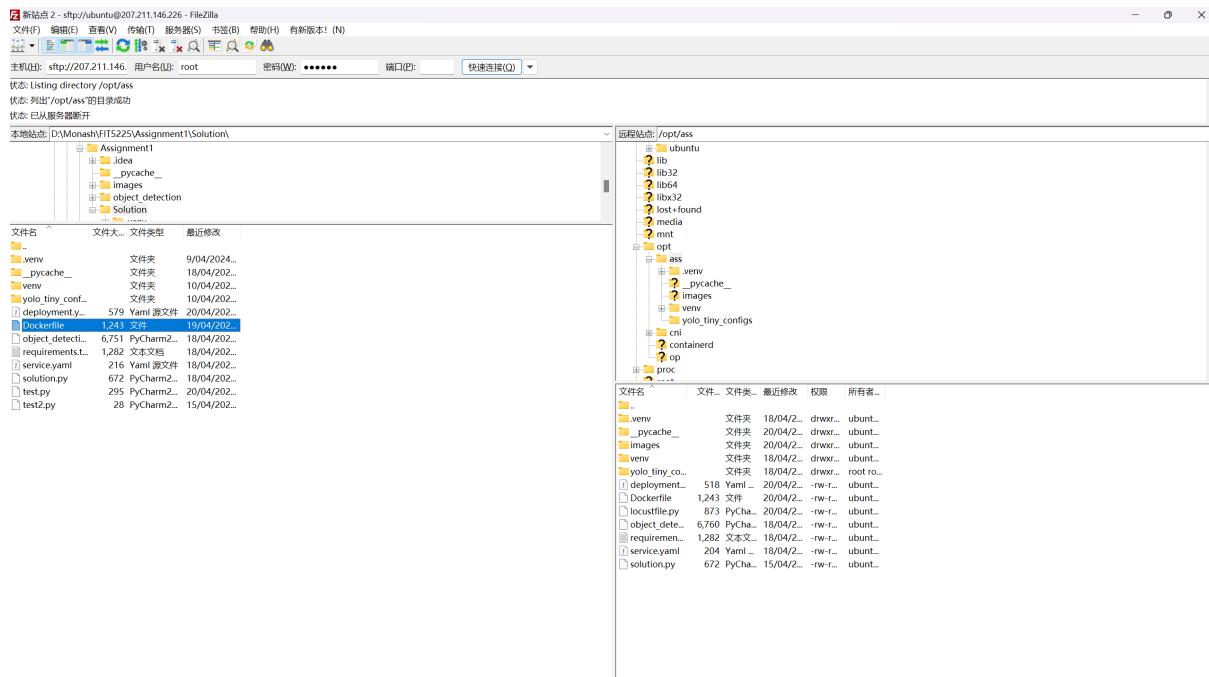
```

urllib3==1.26.5
Werkzeug==3.0.2
zope.event==5.0
zope.interface==5.4.0
importlib-metadata==6.1.0
Zipp==3.15.0

```

### 3.3 File upload(Filezilia)

I use FileZilla to upload the application files and Dockerfile to the Ubuntu server.



### 3.4 Build and push images

Create a new repository on Docker Hub to store the image. Then, on the Ubuntu server, use the following command to build the Docker image:

The screenshot shows a Docker Hub repository page for the user 'ziqipei' with the repository name 'my-link'. The 'General' tab is selected. At the top right, it says 'Using 0 of 1 private repositories. Get more'. The repository was updated about 7 hours ago. It has one tag, '1.0.0', which is an Image type. There are no pulls or pushes listed. Under 'Docker commands', there is a button to 'Push' with the command 'docker push ziqipei/my-link:tagname'. The 'Automated Builds' section indicates that manually pushing images to Hub? is possible by connecting to GitHub or Bitbucket. The 'Repository overview' section is incomplete.

### 3.3 Connect to Ubuntu server

To upload the application files and Dockerfile to the Ubuntu server, we need to connect to the server via SSH. Use the following command to establish a connection:

```
ubuntu ssh -i <private key> ubuntu@<k8s-master public IP>
```

Please replace <private\_key> with the path to your private key file and <k8s-master\_public\_IP> with the public IP address of your Kubernetes master node.

### 3.4 Build and push Docker images

After connecting to the Ubuntu server, use the following command to inspect the contents of the Dockerfile:

```
cat Dockerfile
```

Make sure the contents of the Dockerfile are as expected.

Next, build the Docker image using the following command:

```
docker image build -t ziqipei/my-link:1.0.0 .
```

```

CMD [ python3 , solution.py ]
ubuntu@k8s-master:/opt/ass$ docker image build -t ziqipei/my-link:1.0.0 .
[+] Building 85.4s (15/15) FINISHED
--> [internal] load build definition from Dockerfile
--> => transferring dockerfile: 1.28kB
--> [internal] load metadata for docker.io/library/python:3.12.2
--> [internal] load .dockerignore
--> => transferring context: 2B
--> [ 1/10] FROM docker.io/library/python:3.12.2@sha256:19973e1796237522ed1fcc1357c766770b47dc15854eafdda0
--> [internal] load build context
--> => transferring context: 305B
--> CACHED [ 2/10] WORKDIR /link
--> CACHED [ 3/10] COPY requirements.txt .
--> CACHED [ 4/10] RUN apt-get update && apt-get install -y libgl1-mesa-glx libglib2.0-0 libsm
--> [ 5/10] RUN apt-get install -y libopencv-dev python3-opencv
--> [ 6/10] RUN pip install --no-cache-dir -r requirements.txt && rm -rf ~/.cache/pip/*
--> [ 7/10] WORKDIR /link
--> [ 8/10] COPY yolo_tiny_configs /opt/ass/yolo_tiny_configs
--> [ 9/10] COPY object_detection.py .
--> [10/10] COPY solution.py .
--> => exporting to image
--> => exporting layers
--> => writing image sha256:b54f78a8b61e34ae9892476edb635ff9a5a1631c84e80544182c99ad2aed522f
--> => naming to docker.io/ziqipei/my-link:1.0.0

```

This command will build an image named ziqipei/my-link based on the Dockerfile in the current directory and label it 1.0.0. Make corresponding modifications according to the Docker Hub user name and image name.

### 3.4.1 Push the docker

(1)When dockerfile is truly it can success build docker images then should get pull the docker hub

```
docker push ziqipei/my-link:1.0.0
```

(2)View status

```

ubuntu@k8s-master:/opt/ass$ docker push ziqipei/my-link:1.0.0
The push refers to repository [docker.io/ziqipei/my-link]
1ecad8d9836d: Pushed
73c5ed5b6c16: Pushed
385ed348fbe6: Pushed
5f70bf18a086: Layer already exists
4888ceedb299: Pushed
2af948c73ceb: Pushed
376b13172a5c: Pushed
c89e966b1138: Pushed
eb2ff5dbfc0d: Pushed
9adbc4b1428d: Layer already exists
1193f41e6b14: Layer already exists
e077e19b6682: Layer already exists
21elc4948146: Layer already exists
68866beb2ed2: Layer already exists
e6e2ab10dba6: Layer already exists
0238a1790324: Layer already exists
1.0.0: digest: sha256:9d3063775f1960adca2dd9a6102192b2608bae5b5d9ac1fd989176d85a782d6a7 size: 3891
f52093e4f67d: Layer already exists

```

```
command docker images see docker document
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ziqipei/my-link	1.0.0	b54f78a8b61e	7 hours ago	2.62GB
ziqipei/my-link	<none>	cd5586df8fc7	46 hours ago	2.55GB
registry.k8s.io/kube-apiserver	v1.29.3	39f995c9f199	5 weeks ago	127MB
registry.k8s.io/kube-scheduler	v1.29.3	8c390d98f50c	5 weeks ago	59.6MB
registry.k8s.io/kube-controller-manager	v1.29.3	6052a25da3f9	5 weeks ago	122MB
registry.k8s.io/kube-proxy	v1.29.3	a1d263b5dc5b	5 weeks ago	82.4MB
registry.k8s.io/etcd	3.5.12-0	3861cfcd7c04	2 months ago	149MB
registry.k8s.io/coredns/coredns	v1.11.1	cbb01a7bd410	8 months ago	59.8MB
registry.k8s.io/pause	3.9	e6f181688397	18 months ago	744kB
weaveworks/weave-npc	latest	690c3345cc9c	3 years ago	39.3MB
weaveworks/weave-kube	latest	62fea85d6052	3 years ago	89MB

```
ubuntu@k8s-master:/opt/ass$ docker run cd5586df8fc7
```

```
ubuntu@k8s-master:/opt/ass$ docker run cd5586df8fc7
/opt/ass/yolo_tiny_configs/coco.names

[INFO] loading YOLO from disk...
 * Serving Flask app 'solution'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.17.0.3:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 139-097-368
```

Before pushing, make sure I'm logged into Docker Hub using the docker login command. After the push is completed, my image can be downloaded and used by me or others.

Through the above steps, I have successfully packaged the application as a Docker image and published it to Docker Hub. This approach greatly simplifies the application deployment and distribution process, making it more convenient and reliable.

#### Task 4: Kubernetes

The goal of this task is to create a Kubernetes cluster on a virtual machine (instance) of Oracle Cloud Infrastructure (OCI).

##### 4.1 Set up a Kubernetes cluster on OCI

Since my instance is limited to 2GB OCPU, if set to 4GB, it would exceed the plan cost. Therefore, I made appropriate adjustments based on available resources.

First, create a Kubernetes cluster with three nodes, one of which serves as the control plane node and the other two nodes serve as worker nodes. Install the Docker engine on each node and use Kubeadm to configure the Kubernetes cluster.

Make sure every node in the cluster can communicate with each other. Connectivity can be verified by pinging between nodes

Instances *in* MonashCloudLab compartment

An instance is a compute host. Choose between virtual machines (VMs) and bare metal instances. The image that you use to launch an instance determines its operating system and other software.

<input type="checkbox"/>	Name	State	Public IP	Private IP	Shape	OCPUs count	Memory (GB)	Availability domain	Fault domain	Created
<input type="checkbox"/>	k8s-master	Running	207.211.146.226	10.0.0.115	VM.Standard3.Flex	2	8	AD-1	FD-3	Sun, Apr 14, 2024, 12:06:48 UTC
<input type="checkbox"/>	k8s-worker1	Running	168.138.22.27	10.0.0.150	VM.Standard3.Flex	2	8	AD-1	FD-3	Sun, Apr 14, 2024, 12:08:35 UTC
<input type="checkbox"/>	k8s-worker2	Running	207.211.149.80	10.0.0.242	VM.Standard3.Flex	2	8	AD-1	FD-3	Sun, Apr 14, 2024, 12:10:44 UTC

0 selected      Showing 3 items < 1 of 1 >

## Work1 configuration environment

```
ubuntu@k8s-worker1:~$ sudo apt update
Hit:1 http://ap-melbourne-1-ad-1.clouds.archive.ubuntu.com/ub
Get:2 http://ap-melbourne-1-ad-1.clouds.archive.ubuntu.com/ub
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRele
Hit:4 http://ap-melbourne-1-ad-1.clouds.archive.ubuntu.com/ub
Fetched 229 kB in 2s (114 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
57 packages can be upgraded. Run 'apt list --upgradable' to see
ubuntu@k8s-worker1:~$ sudo apt install -y apt-transport-https
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
lsb-release is already the newest version (11.1.0ubuntu4).
lsb-release set to manually installed.
ca-certificates is already the newest version (20230311ubuntu1).
ca-certificates set to manually installed.
gnupg is already the newest version (2.2.27-3ubuntu2.1).
gnupg set to manually installed.
```

```
software-properties-common is already the newest version (0.9.10.1-0ubuntu1).
software-properties-common set to manually installed.
```

```
The following additional packages will be installed:
```

```
    libcurl4
```

```
The following NEW packages will be installed:
```

```
    apt-transport-https
```

```
The following packages will be upgraded:
```

```
    curl libcurl4
```

```
2 upgraded, 1 newly installed, 0 to remove and 55 not upgraded.
```

```
Need to get 486 kB of archives.
```

```
After this operation, 170 kB of additional disk space will be
```

```
Get:1 http://ap-melbourne-1-ad-1.clouds.archive.ubuntu.com/ubuntu/
```

```
Get:2 http://ap-melbourne-1-ad-1.clouds.archive.ubuntu.com/ubuntu/
```

```
Get:3 http://ap-melbourne-1-ad-1.clouds.archive.ubuntu.com/ubuntu/
```

```
Fetched 486 kB in 2s (243 kB/s)
```

```
Selecting previously unselected package apt-transport-https.
```

```
(Reading database ... 70814 files and directories currently installed)
```

```
Preparing to unpack .../apt-transport-https_2.4.12_all.deb ...
```

```
Unpacking apt-transport-https (2.4.12) ...
```

```
Preparing to unpack .../curl_7.81.0-1ubuntu1.16_amd64.deb ...
```

```
Unpacking curl (7.81.0-1ubuntu1.16) over (7.81.0-1ubuntu1.15)
```

```
Preparing to unpack .../libcurl4_7.81.0-1ubuntu1.16_amd64.deb ...
```

```
Unpacking libcurl4:amd64 (7.81.0-1ubuntu1.16) over (7.81.0-1ubuntu1.15)
```

```
Setting up apt-transport-https (2.4.12) ...
```

```
Setting up libcurl4:amd64 (7.81.0-1ubuntu1.16) ...
```

```
Setting up curl (7.81.0-1ubuntu1.16) ...
```

```
Processing triggers for man-db (2.10.2-1) ...
```

```
Processing triggers for libc-bin (2.35-0ubuntu3.6) ...
```

```
Scanning processes...
```

```
Scanning linux images...
```

```
Running kernel seems to be up-to-date.
```

```
No services need to be restarted.
```

```
No containers need to be restarted.
```

```
No user sessions are running outdated binaries.
```

```
No VM guests are running outdated hypervisor (qemu) binaries.  
ubuntu@k8s-worker1:~$ sudo mkdir -m 0755 -p /etc/apt/keyrings  
ubuntu@k8s-worker1:~$ curl -fsSL https://download.docker.com/linux/ubuntu/jammy/docker-ce.key | sudo tee /etc/apt/keyrings/docker-ce.key  
ubuntu@k8s-worker1:~$ sudo chmod a+r /etc/apt/keyrings/docker-ce.key  
ubuntu@k8s-worker1:~$ echo "deb [arch=$(dpkg --print-architecture)] https://download.docker.com/linux/ubuntu jammy InRelease" | sudo tee /etc/apt/sources.list.d/docker.list  
ubuntu@k8s-worker1:~$ sudo apt update  
Get:1 https://download.docker.com/linux/ubuntu jammy InRelease  
Get:2 https://download.docker.com/linux/ubuntu jammy/stable arm64 Packages  
Hit:3 http://security.ubuntu.com/ubuntu jammy-security InRelease  
Hit:4 http://ap-melbourne-1-ad-1.clouds.archive.ubuntu.com/ubuntu jammy-updates InRelease  
Hit:5 http://ap-melbourne-1-ad-1.clouds.archive.ubuntu.com/ubuntu jammy-backports InRelease  
Hit:6 http://ap-melbourne-1-ad-1.clouds.archive.ubuntu.com/ubuntu jammy InRelease  
Fetched 78.8 kB in 1s (84.8 kB/s)  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
55 packages can be upgraded. Run 'apt list --upgradable' to see them.  
ubuntu@k8s-worker1:~$ sudo apt install -y docker-ce docker-ce-rootless-extras  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  docker-ce-rootless-extras libltdl7 libslirp0 pigz slirp4netns  
Suggested packages:  
  aufs-tools cgroupfs-mount | cgroup-lite  
The following NEW packages will be installed:  
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras libslirp0 pigz slirp4netns  
0 upgraded, 10 newly installed, 0 to remove and 55 not upgraded.  
Need to get 120 MB of archives.  
After this operation, 430 MB of additional disk space will be used.  
Get:1 https://download.docker.com/linux/ubuntu jammy/stable arm64 Packages  
Get:2 https://download.docker.com/linux/ubuntu jammy/stable arm64 DEP-Status  
Get:3 http://ap-melbourne-1-ad-1.clouds.archive.ubuntu.com/ubuntu jammy-updates DEP-Status  
Get:4 https://download.docker.com/linux/ubuntu jammy/stable DEP-Status  
Get:5 https://download.docker.com/linux/ubuntu jammy/stable DEP-Status  
Get:6 https://download.docker.com/linux/ubuntu jammy/stable DEP-Status
```

```
Get:7 https://download.docker.com/linux/ubuntu jammy/stable ar
Get:8 http://ap-melbourne-1-ad-1.clouds.archive.ubuntu.com/ub
Get:9 http://ap-melbourne-1-ad-1.clouds.archive.ubuntu.com/ub
Get:10 http://ap-melbourne-1-ad-1.clouds.archive.ubuntu.com/u
Fetched 120 MB in 2s (58.7 MB/s)
Selecting previously unselected package pigz.
(Reading database ... 70818 files and directories currently i
Preparing to unpack .../0-pigz_2.6-1_amd64.deb ...
Unpacking pigz (2.6-1) ...
Selecting previously unselected package containerd.io.
Preparing to unpack .../1-containerd.io_1.6.31-1_amd64.deb ..
Unpacking containerd.io (1.6.31-1) ...
Selecting previously unselected package docker-buildx-plugin.
Preparing to unpack .../2-docker-buildx-plugin_0.13.1-1-ubunt
Unpacking docker-buildx-plugin (0.13.1-1-ubuntu.22.04~jammy)
Selecting previously unselected package docker-ce-cli.
Preparing to unpack .../3-docker-ce-cli_5%3a26.0.1-1-ubuntu.2
Unpacking docker-ce-cli (5:26.0.1-1-ubuntu.22.04~jammy) ...
Selecting previously unselected package docker-ce.
Preparing to unpack .../4-docker-ce_5%3a26.0.1-1-ubuntu.22.04
Unpacking docker-ce (5:26.0.1-1-ubuntu.22.04~jammy) ...
Selecting previously unselected package docker-ce-rootless-ex
Preparing to unpack .../5-docker-ce-rootless-extras_5%3a26.0.
Unpacking docker-ce-rootless-extras (5:26.0.1-1-ubuntu.22.04~
Selecting previously unselected package docker-compose-plugin
Preparing to unpack .../6-docker-compose-plugin_2.26.1-1-ubun
Unpacking docker-compose-plugin (2.26.1-1-ubuntu.22.04~jammy)
Selecting previously unselected package libltdl7:amd64.
Preparing to unpack .../7-libltdl7_2.4.6-15build2_amd64.deb .
Unpacking libltdl7:amd64 (2.4.6-15build2) ...
Selecting previously unselected package libslirp0:amd64.
Preparing to unpack .../8-libslirp0_4.6.1-1build1_amd64.deb .
Unpacking libslirp0:amd64 (4.6.1-1build1) ...
Selecting previously unselected package slirp4netns.
Preparing to unpack .../9-slirp4netns_1.0.1-2_amd64.deb ...
Unpacking slirp4netns (1.0.1-2) ...
Setting up docker-buildx-plugin (0.13.1-1-ubuntu.22.04~jammy)
Setting up containerd.io (1.6.31-1) ...
```

```
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service
Setting up docker-compose-plugin (2.26.1-1~ubuntu.22.04~jammy)
Setting up libltdl7:amd64 (2.4.6-15build2) ...
Setting up docker-ce-cli (5:26.0.1-1~ubuntu.22.04~jammy) ...
Setting up libslirp0:amd64 (4.6.1-1build1) ...
Setting up pigz (2.6-1) ...
Setting up docker-ce-rootless-extras (5:26.0.1-1~ubuntu.22.04~jammy)
Setting up slirp4netns (1.0.1-2) ...
Setting up docker-ce (5:26.0.1-1~ubuntu.22.04~jammy) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker-ce-rootless-extras.service
Created symlink /etc/systemd/system/sockets.target.wants/docker-ce.socket
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.6) ...
Scanning processes...
Scanning linux images...
```

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

```
No VM guests are running outdated hypervisor (qemu) binaries
ubuntu@k8s-worker1:~$ sudo usermod -aG docker ${USER}
ubuntu@k8s-worker1:~$ wget https://github.com/Mirantis/cri-dockerd
--2024-04-14 12:19:24-- https://github.com/Mirantis/cri-dockerd/
Resolving github.com (github.com)... 20.248.137.48
Connecting to github.com (github.com)|20.248.137.48|:443...
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production/prerelease/cri-dockerd_0.3.10.3-0.ubuntu-jammy_amd64.deb
--2024-04-14 12:19:24-- https://objects.githubusercontent.com/github-production/prerelease/cri-dockerd_0.3.10.3-0.ubuntu-jammy_amd64.deb
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 20.248.137.48
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|20.248.137.48|:443...
HTTP request sent, awaiting response... 200 OK
Length: 11107240 (11M) [application/octet-stream]
Saving to: 'cri-dockerd_0.3.10.3-0.ubuntu-jammy_amd64.deb'
```

```
cri-dockerd_0.3.10.3-0.ubuntu 100%[=====]
2024-04-14 12:19:25 (345 MB/s) - 'cri-dockerd_0.3.10.3-0.ubuntu'
ubuntu@k8s-worker1:~$ sudo dpkg -i cri-dockerd_0.3.10.3-0.ubuntu
Selecting previously unselected package cri-dockerd.
(Reading database ... 71084 files and directories currently installed)
Preparing to unpack cri-dockerd_0.3.10.3-0.ubuntu-jammy_amd64 ...
Unpacking cri-dockerd (0.3.10~3-0~ubuntu-jammy) ...
Setting up cri-dockerd (0.3.10~3-0~ubuntu-jammy) ...
Created symlink /etc/systemd/system/multi-user.target.wants/cri-dockerd.service → /lib/systemd/system/cri-dockerd.service.
Created symlink /etc/systemd/system/sockets.target.wants/cri-dockerd.socket → /lib/systemd/system/cri-dockerd.socket.
ubuntu@k8s-worker1:~$ sudo apt update && sudo apt-get install cri-dockerd
Hit:1 https://download.docker.com/linux/ubuntu jammy InRelease
Hit:2 http://ap-melbourne-1-ad-1.clouds.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:4 http://ap-melbourne-1-ad-1.clouds.archive.ubuntu.com/ubuntu jammy InRelease
Hit:5 http://ap-melbourne-1-ad-1.clouds.archive.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
55 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20230311ubuntu1).
curl is already the newest version (7.81.0-1ubuntu1.16).
apt-transport-https is already the newest version (2.4.12).
0 upgraded, 0 newly installed, 0 to remove and 55 not upgraded.
ubuntu@k8s-worker1:~$ sudo curl -fSSL https://pkgs.k8s.io/core交代
ubuntu@k8s-worker1:~$ echo 'deb [signed-by=/etc/apt/keyrings/k8s.gpg] https://prod-cdn.packages.k8s.io/repositories/isv:/kubelet jammy main'
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://prod-cdn.packages.k8s.io/repositories/isv:/kubelet jammy main
ubuntu@k8s-worker1:~$ sudo apt update && sudo apt-get install kubelet
Hit:1 https://download.docker.com/linux/ubuntu jammy InRelease
Get:2 https://prod-cdn.packages.k8s.io/repositories/isv:/kubelet jammy InRelease
Get:3 https://prod-cdn.packages.k8s.io/repositories/isv:/kubelet jammy InRelease
Hit:4 http://ap-melbourne-1-ad-1.clouds.archive.ubuntu.com/ubuntu jammy InRelease
```

```
Hit:5 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:6 http://ap-melbourne-1-ad-1.clouds.archive.ubuntu.com/ubuntu InRelease
Hit:7 http://ap-melbourne-1-ad-1.clouds.archive.ubuntu.com/ubuntu jammy InRelease
Fetched 7697 B in 1s (6773 B/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
55 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  conntrack cri-tools ebtables kubernetes-cni socat
The following NEW packages will be installed:
  conntrack cri-tools ebtables kubeadm kubectl kubelet kubernetes-cni
0 upgraded, 8 newly installed, 0 to remove and 55 not upgraded.
Need to get 92.4 MB of archives.
After this operation, 346 MB of additional disk space will be freed.
Get:1 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes-cni/conntrack/1.1.0-1_amd64.deb [10.4 kB]
Get:2 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes-cni/cri-tools/1.29.0-1_amd64.deb [10.4 kB]
Get:3 http://ap-melbourne-1-ad-1.clouds.archive.ubuntu.com/ubuntu/pool/main/c/conntrack/conntrack_1%3a1.4.6-2build2_amd64.deb [10.4 kB]
Get:4 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes-cni/ebtables/2.0.11-4_amd64.deb [10.4 kB]
Get:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes-cni/kubelet/1.29.0-1_amd64.deb [10.4 kB]
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes-cni/kubectl/1.29.0-1_amd64.deb [10.4 kB]
Get:7 http://ap-melbourne-1-ad-1.clouds.archive.ubuntu.com/ubuntu/pool/main/k/kubernetes-cni/_socat_1.7.0-1_amd64.deb [10.4 kB]
Get:8 http://ap-melbourne-1-ad-1.clouds.archive.ubuntu.com/ubuntu/pool/main/k/kubernetes-cni/_kubernetes-cni_0.4.0-1_amd64.deb [10.4 kB]
Fetched 92.4 MB in 3s (37.0 MB/s)
Selecting previously unselected package conntrack.
(Reading database ... 71089 files and directories currently installed)
Preparing to unpack .../0-conntrack_1%3a1.4.6-2build2_amd64.deb ...
Unpacking conntrack (1:1.4.6-2build2) ...
Selecting previously unselected package cri-tools.
Preparing to unpack .../1-cri-tools_1.29.0-1.1_amd64.deb ...
Unpacking cri-tools (1.29.0-1.1) ...
Selecting previously unselected package ebtables.
Preparing to unpack .../2-ebtables_2.0.11-4build2_amd64.deb ...
Unpacking ebtables (2.0.11-4build2) ...
Selecting previously unselected package kubernetes-cni.
```

```
Preparing to unpack .../3-kubernetes-cni_1.3.0-1.1_amd64.deb
Unpacking kubernetes-cni (1.3.0-1.1) ...
Selecting previously unselected package socat.
Preparing to unpack .../4-socat_1.7.4.1-3ubuntu4_amd64.deb ...
Unpacking socat (1.7.4.1-3ubuntu4) ...
Selecting previously unselected package kubelet.
Preparing to unpack .../5-kubelet_1.29.3-1.1_amd64.deb ...
Unpacking kubelet (1.29.3-1.1) ...
Selecting previously unselected package kubectl.
Preparing to unpack .../6-kubectl_1.29.3-1.1_amd64.deb ...
Unpacking kubectl (1.29.3-1.1) ...
Selecting previously unselected package kubeadm.
Preparing to unpack .../7-kubeadm_1.29.3-1.1_amd64.deb ...
Unpacking kubeadm (1.29.3-1.1) ...
Setting up conntrack (1:1.4.6-2build2) ...
Setting up kubectl (1.29.3-1.1) ...
Setting up ebttables (2.0.11-4build2) ...
Setting up socat (1.7.4.1-3ubuntu4) ...
Setting up cri-tools (1.29.0-1.1) ...
Setting up kubernetes-cni (1.3.0-1.1) ...
Setting up kubelet (1.29.3-1.1) ...
Setting up kubeadm (1.29.3-1.1) ...
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...
```

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries
ubuntu@k8s-worker1:~\$ sudo apt-mark hold kubelet kubeadm kube
kubelet set on hold.
kubeadm set on hold.

```
kubectl set on hold.  
ubuntu@k8s-worker1:~$ kubeadm version  
kubeadm version: &version.Info{Major:"1", Minor:"29", GitVers  
  
ubuntu@k8s-worker1:~$ sudo iptables-restore < /etc/iptables/br0.iptables.save  
ubuntu@k8s-worker1:~$ sudo kubeadm join 10.0.0.115:6443 --token=473bf0594888b66d  
--discovery-token-ca-cert-hash sha256:473bf0594888b66d  
[preflight] Running pre-flight checks  
^C  
ubuntu@k8s-worker1:~$ sudo kubeadm join 10.0.0.115:6443 --token=473bf0594888b66d  
--discovery-token-ca-cert-hash sha256:473bf0594888b66d  
[preflight] Running pre-flight checks  
  
^C  
ubuntu@k8s-worker1:~$ sudo ufw disable  
Firewall stopped and disabled on system startup  
ubuntu@k8s-worker1:~$ sudo kubeadm join 10.0.0.115:6443 --token=473bf0594888b66d  
--discovery-token-ca-cert-hash sha256:473bf0594888b66d  
[preflight] Running pre-flight checks  
^C  
ubuntu@k8s-worker1:~$ sudo kubeadm join 10.0.0.115:6443 --token=473bf0594888b66d  
--discovery-token-ca-cert-hash sha256:473bf0594888b66d  
[preflight] Running pre-flight checks  
^C  
ubuntu@k8s-worker1:~$ sudo kubeadm join 10.0.0.115:6443 --token=473bf0594888b66d  
--discovery-token-ca-cert-hash sha256:473bf0594888b66d  
[preflight] Running pre-flight checks  
^C  
ubuntu@k8s-worker1:~$ sudo kubeadm join 10.0.0.115:6443 --token=473bf0594888b66d  
--discovery-token-ca-cert-hash sha256:473bf0594888b66d  
[preflight] Running pre-flight checks  
^C  
ubuntu@k8s-worker1:~$ sudo kubeadm join 10.0.0.115:6443 --token=473bf0594888b66d  
--discovery-token-ca-cert-hash sha256:473bf0594888b66d  
[preflight] Running pre-flight checks
```

```
^C
ubuntu@k8s-worker1:~$ sudo kubeadm join 10.0.0.115:6443 --token=473bf0594888b6654dc9ae0c2f3d4 --discovery-token-ca-cert-hash sha256:473bf0594888b6654dc9ae0c2f3d4
[preflight] Running pre-flight checks
error execution phase preflight: couldn't validate the identity of the peer
To see the stack trace of this error execute with --v=5 or higher
ubuntu@k8s-worker1:~$ sudo kubeadm join 10.0.0.115:6443 --token=473bf0594888b6654dc9ae0c2f3d4 --discovery-token-ca-cert-hash sha256:473bf0594888b6654dc9ae0c2f3d4
[preflight] Running pre-flight checks
error execution phase preflight: couldn't validate the identity of the peer
ubuntu@k8s-worker1:~$ sudo kubeadm join 10.0.0.115:6443 --token=473bf0594888b6654dc9ae0c2f3d4 --discovery-token-ca-cert-hash sha256:473bf0594888b6654dc9ae0c2f3d4
[preflight] Running pre-flight checks
^C
ubuntu@k8s-worker1:~$ sudo kubeadm join 10.0.0.115:6443 --token=473bf0594888b6654dc9ae0c2f3d4 --discovery-token-ca-cert-hash sha256:473bf0594888b6654dc9ae0c2f3d4
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl get node -o yaml | grep ^imagefs'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubelet.env"
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap
```

This node has joined the cluster:

- \* Certificate signing request was sent to apiserver and a response was received.
- \* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node

```
ubuntu@k8s-worker1:~$ sudo ufw disable
Firewall stopped and disabled on system startup
ubuntu@k8s-worker1:~$ sudo nano /etc/iptables/rules.v4
ubuntu@k8s-worker1:~$ curl 10.106.43.189:8080
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
```

```

<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
ubuntu@k8s-worker1:~$ python -V
Command 'python' not found, did you mean:
  command 'python3' from deb python3
  command 'python' from deb python-is-python3
ubuntu@k8s-worker1:~$ python3 -V
Python 3.10.12
ubuntu@k8s-worker1:~$ systemctl status docker.service
● docker.service - Docker Application Container Engine
    Loaded: loaded (/lib/systemd/system/docker.service; enabled;
              )
    Active: active (running) since Sun 2024-04-14 12:18:02 UTC
      Tasks: 11 (limit: 4655)
     Memory: 472.0M
        CPU: 5min 52.844s
      CGroup: /system.slice/docker.service
              └─3879 /usr/bin/dockerd -H fd:// --containerd=/r

```

```
Apr 14 12:18:02 k8s-worker1 dockerd[3879]: time="2024-04-14T1:  
Apr 14 14:13:26 k8s-worker1 dockerd[3879]: Started Docker Application  
Apr 14 14:13:29 k8s-worker1 dockerd[3879]: time="2024-04-14T1:  
Apr 15 06:49:13 k8s-worker1 dockerd[3879]: time="2024-04-15T0:  
Apr 15 06:49:13 k8s-worker1 dockerd[3879]: time="2024-04-15T0:
```

```
ubuntu@k8s-worker1:~$ client_loop: send disconnect: Connection to PS C:\Users\pzqfr> ssh -i D:\Monash\keypairoracle\k8s\master\s Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-1051-oracle x86_64)
```

- \* Documentation: <https://help.ubuntu.com>
- \* Management: <https://landscape.canonical.com>
- \* Support: <https://ubuntu.com/pro>

System information as of Tue Apr 16 12:36:41 UTC 2024

System load: 0.0 Users logged in:  
Usage of /: 10.3% of 44.96GB IPv4 address for docker0:  
Memory usage: 9% IPv4 address for ens3:  
Swap usage: 0% IPv4 address for weave:  
Processes: 150

\* Strictly confined Kubernetes makes edge and IoT secure. Let's just raise the bar for easy, resilient and secure K8s clusters.

<https://ubuntu.com/engage/secure-kubernetes-at-the-edge>

Expanded Security Maintenance for Applications is not enabled

14 updates can be applied immediately.

1 of these updates is a standard security update.

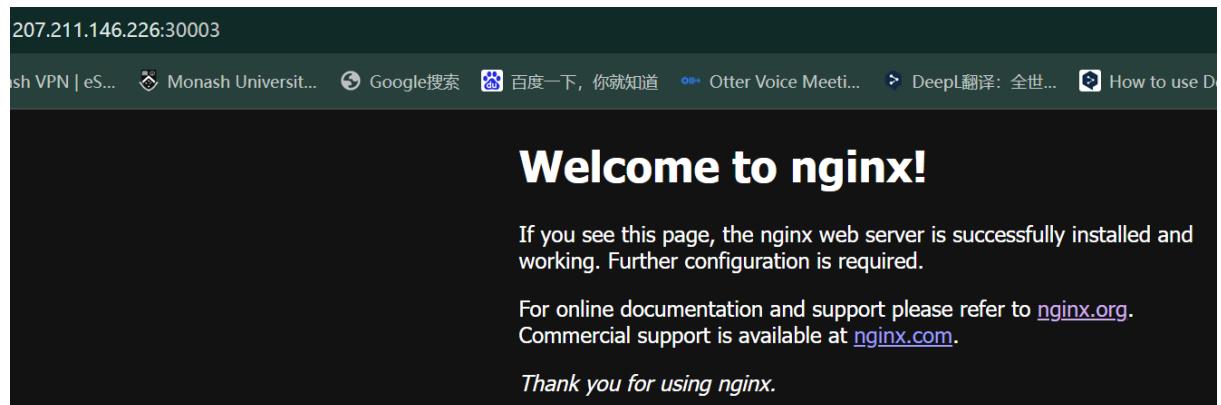
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates  
See <https://ubuntu.com/esm> or run: sudo pro status

It should be added Agreement and each pod should ping each other

```
^Cubuntu@k8s-master:/opt/ass$ kubectl get nodes
NAME           STATUS    ROLES      AGE   VERSION
k8s-master     Ready     control-plane   5d22h  v1.29.3
k8s-worker1    Ready     <none>        5d20h  v1.29.3
k8s-worker2    Ready     <none>        5d20h  v1.29.3
```

I needed to ensure the deployability of my network, so I initially utilized the code from week 4 to deploy an NGINX cluster. And find the need to add ingress rule.



## Task 5: Kubernetes Service

In this task, I will create a Kubernetes service for the deployed object detection web service and expose and access the service with appropriate configuration.

### 5.1 Create deployment files

Create a file named deployment.yaml with the following content:

This deployment file defines a deployment named my-link-deployment, which contains a replica. It uses the ziqipei/my-link:1.0.0 image and sets the container port to 5000. At the same time, it also sets resource limits and requests for memory and CPU.

memory "512Mi" and CPU "0.5"

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-link-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: my-link
  template:
    metadata:
      labels:
        app: my-link
    spec:
      containers:
        - name: ziqipei
          image: ziqipei/my-link:1.0.0
          ports:
            - containerPort: 5000
          resources:
            limits:
              memory: "512Mi"
              cpu: "0.5"
            requests:
              memory: "512Mi"
              cpu: "0.5"
```

use kubectl get pods to check

use `kubectl apply -f service.yaml` to run it Add VM → Security List → ingress rule

```
ubuntu@k8s-master:/opt/ass$ kubectl apply -f deployment.yaml
deployment.apps/my-link-deployment configured
ubuntu@k8s-master:/opt/ass$
```

use kubectl get pods to check

```
ubuntu@k8s-master:/opt/ass$ kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
my-link-deployment-68678f49ff-klndc 1/1     Running   2 (8h ago)  8h
```

use `kubectl get pods,deployment,svc -owide` to check out

```
ubuntu@k8s-master:/opt/ass$ kubectl get pods,deployment,svc -owide
NAME                               NOMINATED-NODE  READINESS   GATES   READY   STATUS    RESTARTS   AGE   IP           NODE
pod/my-link-deployment-68678f49ff-klndc <none>        <none>      1/1     Running   2 (8h ago)  8h   10.40.0.1   k8s-worker1
NAME                               SELECTOR
deployment.apps/my-link-deployment   app=my-link:1.0.0
NAME          TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE   SELECTOR
service/kubernetes   ClusterIP   10.96.0.1      <none>        443/TCP      4d22h
service/my-link     NodePort   10.108.142.210  <none>        5000:32009/TCP 46h   app=my-link
ubuntu@k8s-master:/opt/ass$ |
```

5.2 Create a file named service.yaml with the following content:

This service configuration file defines a service named my-link-service of type NodePort. It is linked to the previously created deployment via the tag selector app: my-link. The service maps the container's port 5000 to the node's port 32009.

```
apiVersion: v1
kind: Service
metadata:
  name: my-link
spec:
  ports:
  - name: api
    nodePort: 32009
    port: 5000
    protocol: TCP
    targetPort: 5000
  selector:
    app: my-link
  type: NodePort
~
```

5.21 Application service configuration

Apply the service configuration using the following command:

```
kubectl apply -f service.yaml
```

### 5.3 Configure security groups and firewalls

In order to allow external access to Kubernetes services, we need to properly configure OCI's Virtual Cloud Network (VCN) and the virtual machine's security group.

In the OCI virtual machine instance, enter the Security List settings and add an Ingress Rule to allow access to node port 32009.

In addition, a virtual machine-level firewall (such as firewall or iptables) needs to be configured to ensure that the corresponding ports are open.

Add VM → Security List → ingress rule

Add Ingress Rules								
	Stateless ▾	Source	IP Protocol	Source Port Range	Destination Port Range	Type and Code	Allows	Description
<input type="checkbox"/>	No	10.0.0.0/16	ICMP		3		ICMP traffic for: 3 Destination Unreachable	⋮
<input type="checkbox"/>	No	0.0.0.0/0	TCP	All	22		TCP traffic for ports: 22 SSH Remote Login Protocol	⋮
<input type="checkbox"/>	No	0.0.0.0/0	TCP	All	6443		TCP traffic for ports: 6443	⋮
<input type="checkbox"/>	No	10.0.0.0/16	ICMP		All		ICMP traffic for: All	⋮
<input type="checkbox"/>	No	10.0.0.0/16	All Protocols				All traffic for all ports	⋮
<input type="checkbox"/>	No	10.0.0.0/16	TCP	All	6443		TCP traffic for ports: 6443	⋮
<input type="checkbox"/>	No	0.0.0.0/0	TCP	All	80		TCP traffic for ports: 80 Allow HTTP and HTTPS traffic to webserver	⋮
<input type="checkbox"/>	No	0.0.0.0/0	TCP	All	443		TCP traffic for ports: 443 HTTPS Allow HTTP and HTTPS traffic to webserver	⋮
<input type="checkbox"/>	No	0.0.0.0/0	TCP	All	30003		TCP traffic for ports: 30003	⋮
<input type="checkbox"/>	No	0.0.0.0/0	TCP	All	6443		TCP traffic for ports: 6443	⋮
<input type="checkbox"/>	No	0.0.0.0/0	TCP	All	32009		TCP traffic for ports: 32009	⋮
<input type="checkbox"/>	No	0.0.0.0/0	TCP	All	8089		TCP traffic for ports: 8089	⋮

### 5.4 Check service status and logs

Check the status of the service using the following command:

```
kubectl get pods
```

```
use kubectl logs <pod name> to check out logs
```

```

deployment.apps/my-link-deployment configured
ubuntu@k8s-master:/opt/ass$ kubectl get pods
NAME                      READY   STATUS    RESTARTS   AGE
my-link-deployment-68678f49ff-klndc  1/1     Running   2 (8h ago)  8h
ubuntu@k8s-master:/opt/ass$ kubectl logs my-link-deployment-68678f49ff-klndc
/opt/ass/yolo_tiny_configs/coco.names

[INFO] loading YOLO from disk...
 * Serving Flask app 'solution'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use
a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://10.40.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 304-847-082
10.40.0.0 - - [20/Apr/2024 02:29:26] "POST /api/transfer HTTP/1.1" 200 -
10.40.0.0 - - [20/Apr/2024 02:29:27] "POST /api/transfer HTTP/1.1" 200 -

```

## 5.5 Accessing web services

Now, I can access the Web service through the public IP address of the worker node and the exposed node port. I chose the public IP of k8s-worker1 plus 32009, which is my nodeport node. I tested whether it can be accessed.

Lastly need use k8s-worker1 public address : 32009 to access

```

k8s-worker1 http://168.138.22.27:32009/
k8s-master http://207.211.146.226:32009/

```

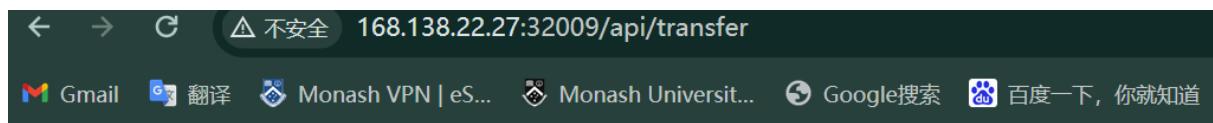
instance



**This program is developed by Ziqi Pei for FIT5225 2024 S1 Assignment 1!**

```
k8s-worker1 http://168.138.22.27:32009/api/transfer  
k8s-master http://207.211.146.226:32009/api/transfer
```

Beacause its the post function,so can use post man to access this function



## Method Not Allowed

The method is not allowed for the requested URL.

Through the above steps, I successfully created a Kubernetes service

This allows us to communicate with web services running in Pods from outside the cluster. At the same time, we have also configured necessary security measures, such as security groups and firewall rules, to ensure the security of the service.

```
sudo ufw allow 8089/tcp  
sudo apt install ufw
```

Utilize Postman to send a POST request to the public web service

```

{
  "id": 123,
  "objects": [
    {
      "accuracy": 0.9590020179748535,
      "label": "person",
      "rectangle": {
        "height": 277,
        "left": 300,
        "top": 211,
        "width": 134
      }
    },
    {
      "accuracy": 0.9153203368186951,
      "label": "elephant",
      "rectangle": {
        "height": 243,
        "left": -6,
        "top": 219,
        "width": 232
      }
    }
  ]
}

```

## Task 6 Locust load generation

In this task, I will use the Locust tool to simulate concurrent users accessing our RESTful API and test the performance of the API under heavy load.

### 6.1 Install Locust

First, make sure you have Locust installed. If it is not installed yet , we can install it using the following command:

```
pip install locust
```

### 6.2 Writing Locust scripts

Create a file named locustfile.py with the following content:

```

from locust import HttpUser, task, between

class HelloWorldUser(HttpUser):
    wait_time = between(5, 10)

```

```

@task
def hello_world(self):
    import os
    folder_path = "./images/inputfolder/"
    files = os.listdir(folder_path)
    for file in files:
        import uuid
        # Generate UUID
        id = str(uuid.uuid4())
        base64_image = self.read_img_base64(folder_path + file)
        res = self.client.post("/api/transfer", json={"id": id})
        print(res.text)

def read_img_base64(self, image):
    import base64
    # Read the image file
    with open(image, 'rb') as image_file:
        # Convert the image file content to base64 format
        base64_image = base64.b64encode(image_file.read())
    return base64_image

```

This Locust script defines a user class named HelloWorldUser. In the hello\_world task, it will traverse all image files in the ./images/inputfolder/ directory and perform the following operations on each image:

1. Generate a unique UUID.
2. Convert image files to base64 encoding.
3. Encapsulate UUID and base64-encoded images into a JSON-formatted request body.
4. Send a POST request to the /api/transfer endpoint.

The read\_img\_base64 function is used to read and convert image files to base64 encoding.

## Task7 Performance test results

The following are the results of performance testing in different environments

### 7.1 Running Locust(try localhost Locust)

Run Locust using the following command:

```
locust -f locustfile.py
```

```
ubuntu@k8s-master:/opt/ass$ locust -f locustfile.py
[2024-04-20 12:31:16,218] k8s-master/INFO/locust.main: Starting web interface at http://0.0.0.0:8089
[2024-04-20 12:31:16,226] k8s-master/INFO/locust.main: Starting Locust 2.25.0
```

## 7.21

In Locust's web interface, you can configure the number of concurrent users and startup rate, and then start testing.

Set up a locust web service as 1 user 1replicas

The screenshot shows the Locust web interface. At the top, it displays the URL `207.211.146.226:8089`. Below the header, there's a navigation bar with links to Gmail, 翻译, Monash VPN | eS..., Monash Universit..., Google搜索, 百度一下, 你就知道, and 所有书签. The main content area has a title "Locust" with a green dragonfly icon, followed by "HOST" and the URL "http://168.138.22.27:32009". To the right, there are status metrics: STATUS READY, RPS 3.2, and FAILURES 0%. A gear icon for settings is also present. Below this, there are input fields for "Start new load test": "Number of users (peak concurrency)" set to 1, "Ramp up (users started/second)" set to 1, and "Host" set to `http://168.138.22.27:32009`. An "Advanced options" dropdown is partially visible. At the bottom is a large green "START" button.

That monitoring and record relevant performance metrics ,response time, querry persecond

## return Json file

The screenshot shows a web browser with two tabs open. The left tab is a terminal window showing a JSON array of objects, each representing a detected person in an image. The right tab is the Locust web interface at <http://168.138.22.27:32009>. The Locust interface displays the following statistics:

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
POST	/api/transfer	273	0	300	390	390	308.81	217	396	387.13	0.7	0
Aggregated		273	0	300	390	390	308.81	217	396	387.13	0.7	0

The screenshot shows a web browser with two tabs open. The left tab is a terminal window showing a JSON array of objects, each representing a detected car in an image. The right tab is the Locust web interface at <http://168.138.22.27:32009>. The Locust interface displays the following statistics:

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)
POST	/api/transfer	184	0	300	380	380	304.97	205	386
Aggregated		184	0	300	380	380	304.97	205	386

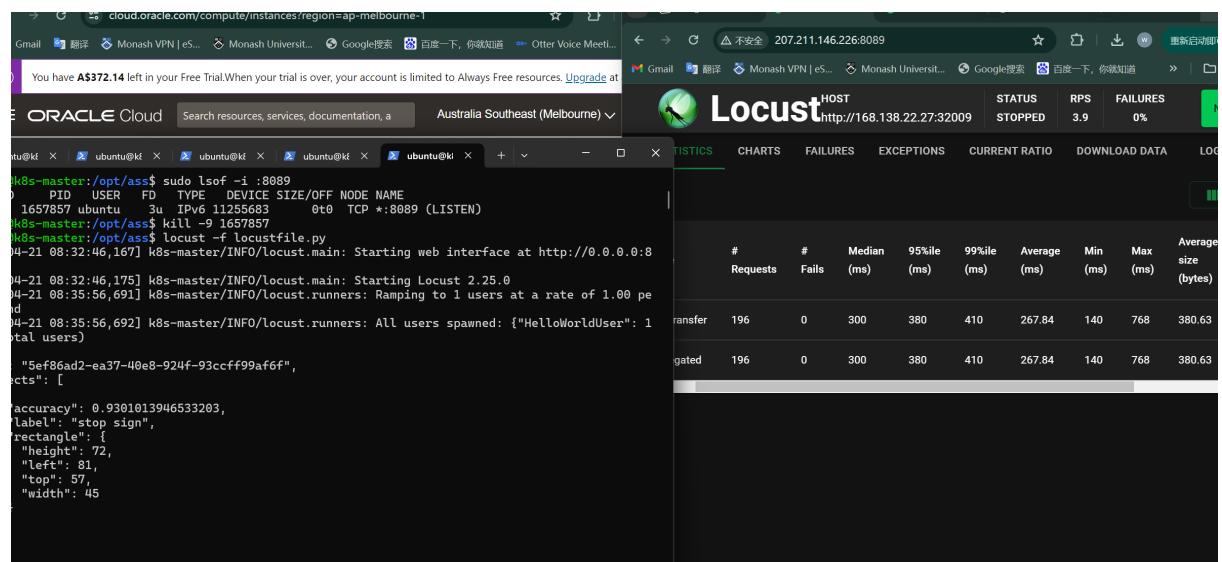
update yaml replicas 1 → 2 then apply locust again

```

ubuntu@k8s-master:/opt/ass/
ubuntu@k8s-master:/opt/ass$ ls
Dockerfile      images          requirements.txt  venv
__pycache__     locustfile.py    service.yaml    yolo_tiny_configs
deployment.yaml object_detection.py solution.py
ubuntu@k8s-master:/opt/ass$ kubectl apply -f deployment.yaml
deployment.apps/my-link-deployment configured
ubuntu@k8s-master:/opt/ass$ kubectl get pods
NAME                  READY   STATUS    RESTARTS   AGE
my-link-deployment-68678f49ff-frr6x  1/1     Running   0          14s
my-link-deployment-68678f49ff-klndc  1/1     Running   8 (19h ago)  30h
ubuntu@k8s-master:/opt/ass$ ls
Dockerfile      images          requirements.txt  venv
__pycache__     locustfile.py    service.yaml    yolo_tiny_configs
deployment.yaml object_detection.py solution.py
ubuntu@k8s-master:/opt/ass$ locust -f locustfile.py
[2024-04-21 08:31:36,578] k8s-master/INFO/locust.main: Starting web interface at http://0.0.0.0:899
[2024-04-21 08:31:36,586] k8s-master/INFO/locust.main: Starting Locust 2.25.0

```

## Use 2 pods first time



## Use 2 pods second time

The screenshot shows two browser tabs. The left tab is an Oracle Cloud Compute instance log for 'ubuntu@k8s' showing command-line activity. The right tab is a Locust performance monitoring interface showing statistics for a POST request to '/api/transfer'.

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)
POST	//api/transfer	129	0	300	380	390	269.4	140	390
	Aggregated	129	0	300	380	390	269.4	140	390

update yaml replicas 2 → 4 then apply locust again first time

The screenshot shows two browser tabs. The left tab is an Oracle Cloud Compute instance log for 'ubuntu@k8s' showing command-line activity. The right tab is a Locust performance monitoring interface showing statistics for a POST request to '/api/transfer'.

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)
POST	//api/transfer	192	0	260	310	740	258.82	142	795
	Aggregated	192	0	260	310	740	258.82	142	795

second times

The screenshot shows two browser tabs. The left tab is an Oracle Cloud Compute instance log for 'ubuntu@k8s' showing command-line activity. The right tab is a Locust performance monitoring interface showing statistics for a POST request to '/api/transfer'.

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)
POST	//api/transfer	128	0	260	310	390	254.07	142	389
	Aggregated	128	0	260	310	390	254.07	142	389

update yaml replicas 4 → 8 then apply locust again first time

The screenshot shows two browser windows. The left window is a terminal session on an Oracle Cloud instance with multiple tabs open, showing command-line output related to Kubernetes deployment and Locust configuration. The right window is the Locust host interface at <http://168.138.22.27:32009/>, displaying a summary table of performance metrics.

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)
POST	//api/transfer	128	0	280	380	810	279.47	141	814
	Aggregated	128	0	280	380	810	279.47	141	814

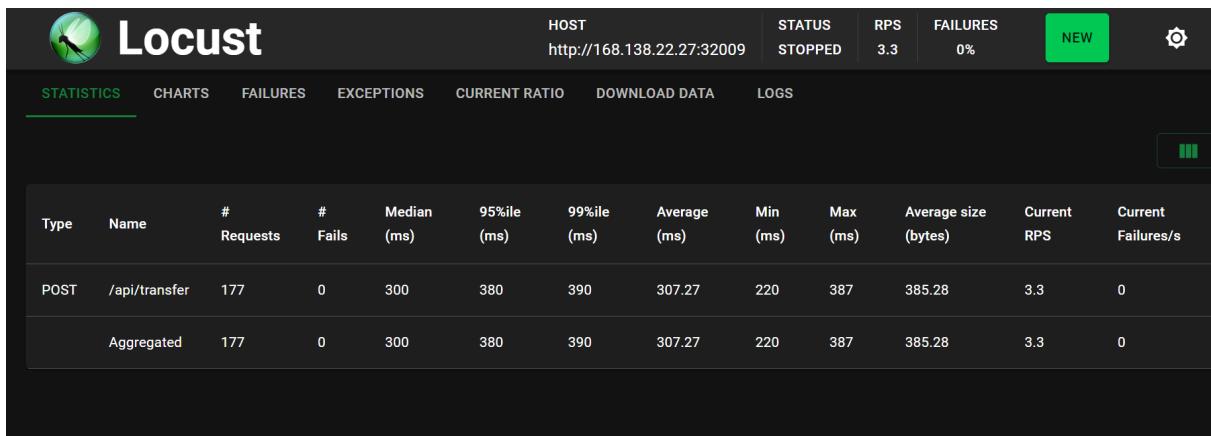
second times

Oracle Host

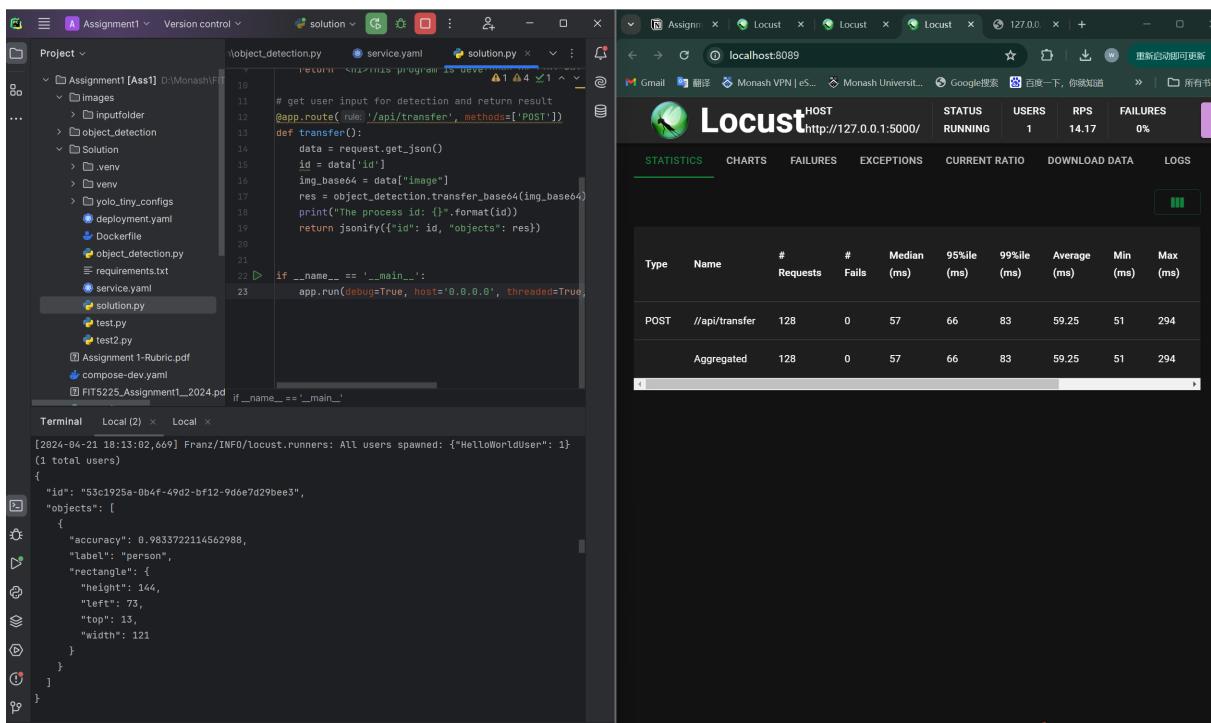
Number of pods	Max Users	Average Response Time 1st (second)	Average Response Time 2nd (Second)
1	1	0.30881	0.30497
2	1	0.26784	0.26940
4	1	0.25882	0.25407
8	1	0.27947	0.25502

7.22

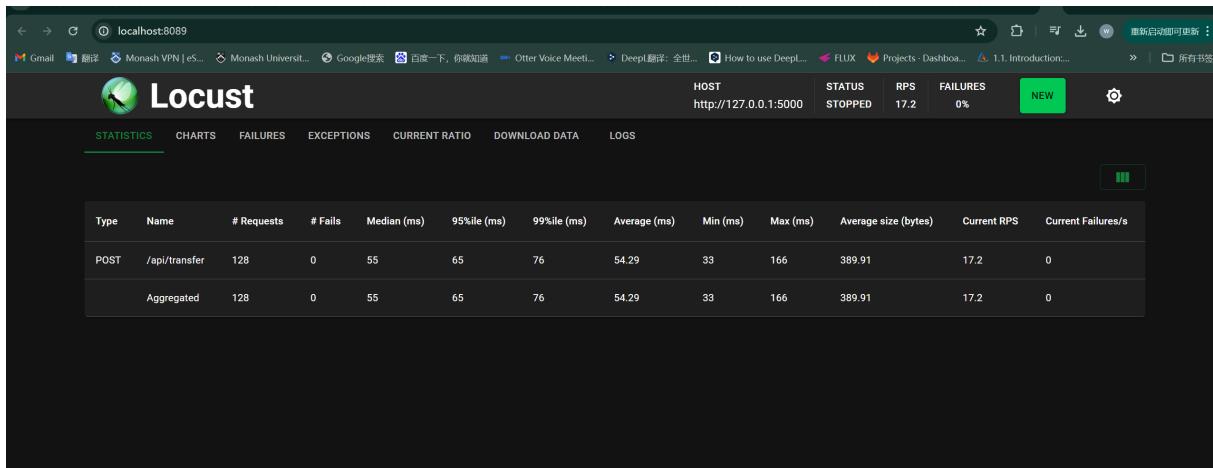
In Locust's Local interface, i can configure the number of connect public api Locust access



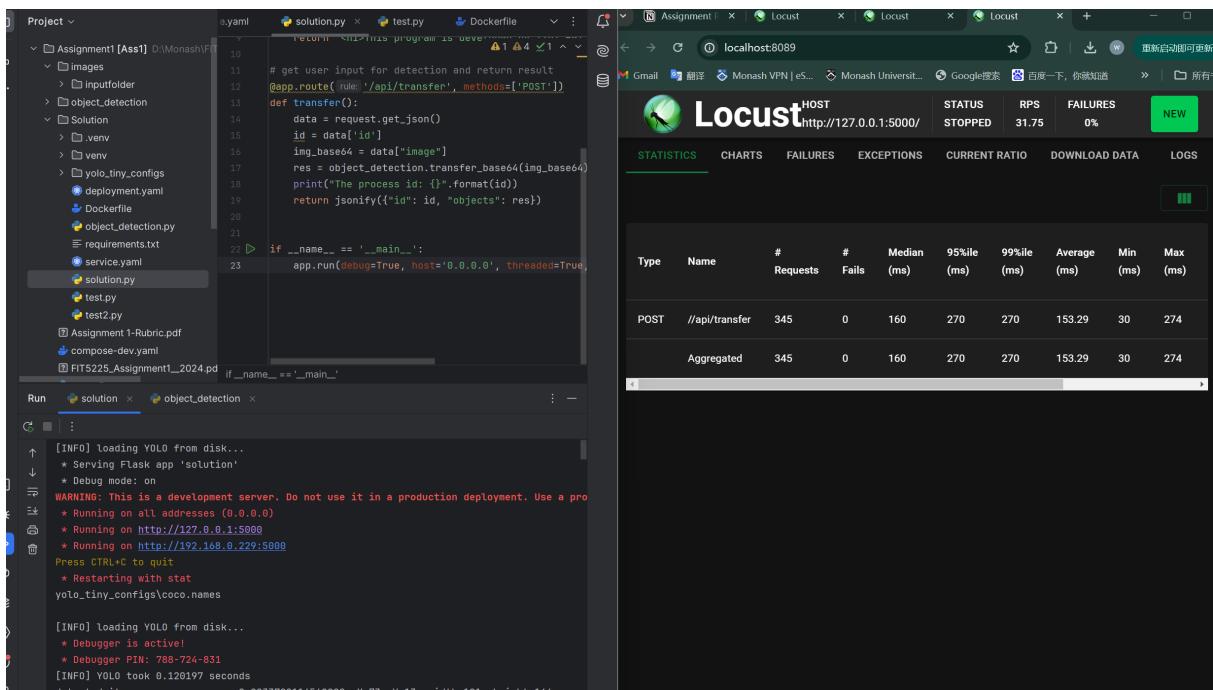
## Local hostlocust



## Second time test



## 10 Users Locust first time



## 10 Users Locust Second time

```

project v
Assignment1 [Ass1] D:\MonashF1T
  - images
  - inputfolder
  - object_detection
  - Solution
  - .venv
  - yolo_tiny_configs
    - deployment.yaml
    - Dockerfile
    - object_detection.py
  - requirements.txt
  - service.yaml
  - solution.py
  - test.py
  - test2.py
  - Assignment 1-Rubric.pdf
  - compose-dev.yaml
  - FIT5225_Assignment1_2024.pdf
if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0', threaded=True)

```

The process id: e950d3bf-73bf-46a6-b337-7c3ec3421274  
127.0.0.1 - - [21/Apr/2024 19:23:19] "POST /api/transfer HTTP/1.1" 200 -  
[INFO] YOLO took 0.021001 seconds  
detected item:car, accuracy:0.935228880773621, X:332, Y:265, width:177, height:159  
detected item:traffic light, accuracy:0.648381233215332, X:561, Y:66, width:23, height:44  
detected item:person, accuracy:0.38100364804267883, X:55, Y:309, width:30, height:53  
detected item:person, accuracy:0.37914738058963884, X:214, Y:341, width:36, height:51  
detected item:car, accuracy:0.36754417419433594, X:488, Y:239, width:40, height:24  
detected item:car, accuracy:0.33403199911117554, X:2, Y:309, width:242, height:15  
detected item:person, accuracy:0.311599817838668, X:518, Y:273, width:83, height:150  
The process id: b770c05e-52a5-4bd4-af2a-23f868ca4af9  
127.0.0.1 - - [21/Apr/2024 19:23:19] "POST /api/transfer HTTP/1.1" 200 -  
[INFO] YOLO took 0.020815 seconds  
The process id: abbb4fec-4acc-45cc-8900-34dcf4ee72cd  
127.0.0.1 - - [21/Apr/2024 19:23:19] "POST /api/transfer HTTP/1.1" 200 -

## 25 Users Locust first time

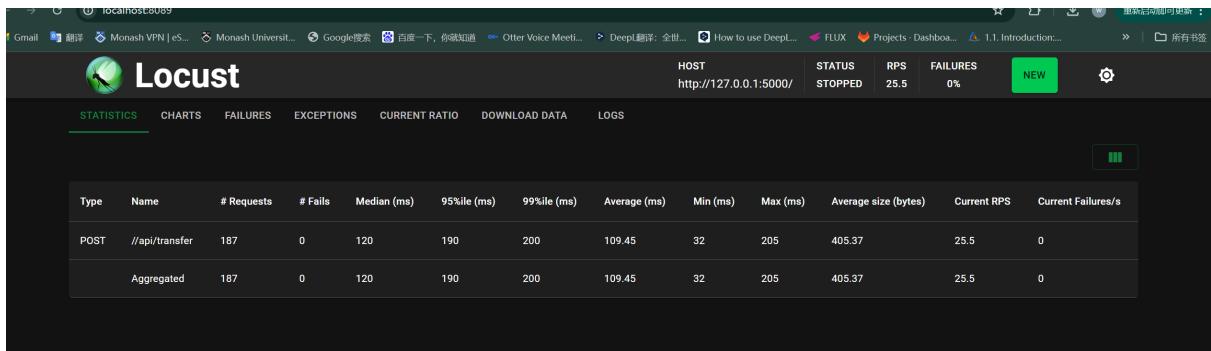
```

project v
Assignment1 [Ass1] D:\MonashF1T
  - images
  - inputfolder
  - object_detection
  - Solution
  - .venv
  - yolo_tiny_configs
    - deployment.yaml
    - Dockerfile
    - object_detection.py
  - requirements.txt
  - service.yaml
  - solution.py
  - test.py
  - test2.py
  - Assignment 1-Rubric.pdf
  - compose-dev.yaml
  - FIT5225_Assignment1_2024.pdf
if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0', threaded=True)

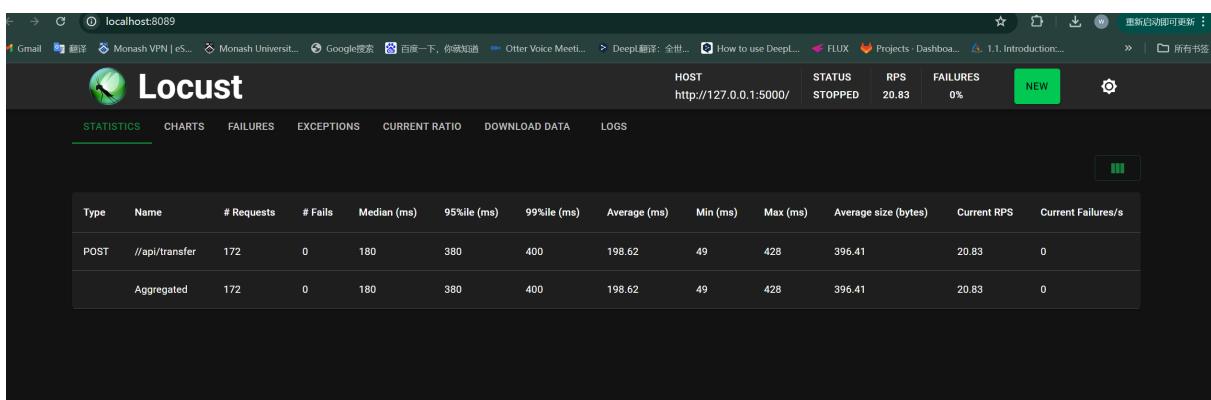
```

detected item:orange, accuracy:0.5968321561813354, X:249, Y:17, width:231, height:238  
The process id: 07c78e9c-6f40-4dc1-a827-3b211d8b5885  
127.0.0.1 - - [21/Apr/2024 19:26:12] "POST /api/transfer HTTP/1.1" 200 -  
[INFO] YOLO took 0.018523 seconds  
detected item:bird, accuracy:0.6463877558708191, X:376, Y:311, width:27, height:35  
The process id: 17596482-1d5c-48d0-928c-e2e50aa45f41  
127.0.0.1 - - [21/Apr/2024 19:26:12] "POST /api/transfer HTTP/1.1" 200 -  
[INFO] YOLO took 0.019002 seconds  
detected item:person, accuracy:0.8396459221839985, X:259, Y:60, width:122, height:244  
detected item:person, accuracy:0.7573572208595276, X:492, Y:18, width:113, height:32  
detected item:person, accuracy:0.687190175054575, X:54, Y:152, width:17, height:32  
detected item:person, accuracy:0.535307692527771, X:94, Y:146, width:20, height:33  
detected item:person, accuracy:0.5293374061584473, X:198, Y:140, width:29, height:61

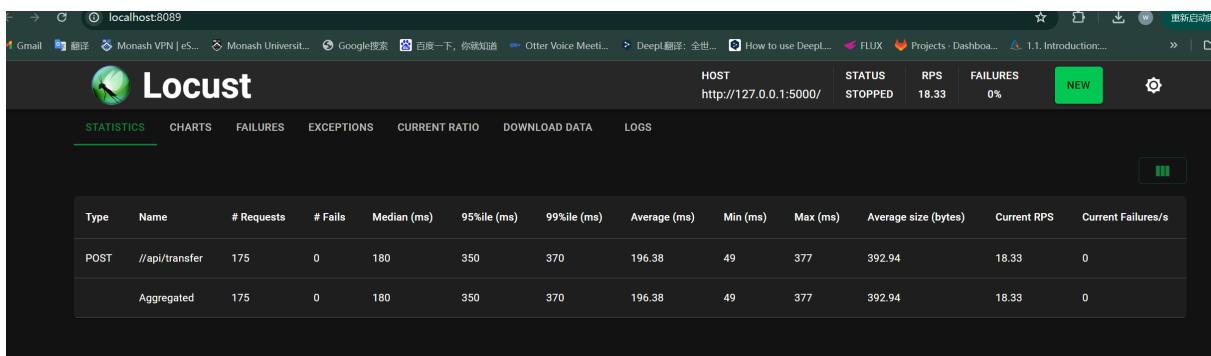
## 25 User Locust second time



## 100 User



## 100 User second time



## Local Host

Number of pods	Max Users	Average Response Time 1st (second)	Average Response Time 2nd (Second)
1	1	0.05925	0.05429
1	10	0.15329	0.13616
1	25	0.13258	0.10945

1	100	0.198.62	0.19638
---	-----	----------	---------

### Use matplotlib analyse Data

```

import matplotlib.pyplot as plt
import numpy as np

data1 = [
    [1, 1, 0.30881, 0.30497],
    [1, 2, 0.26784, 0.26940],
    [1, 4, 0.25882, 0.25407],
    [1, 8, 0.27947, 0.25502]
]

data2 = [
    [1, 1, 0.05925, 0.05429],
    [1, 10, 0.15329, 0.13616],
    [1, 250, 0.13258, 0.10945],
    [1, 1000, 0.19862, 0.19638]
]

data1 = np.array(data1)
data2 = np.array(data2)

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10, 4))

ax1.plot(data1[:, 1], data1[:, 2], color="#0072BD", label='1s')
ax1.plot(data1[:, 1], data1[:, 3], color="#D95319", label='2n')
ax1.set_xlabel('Max Users')
ax1.set_ylabel('Average Response Time (s)')
ax1.set_title('Table 1')
ax1.legend()
ax1.set_xticks(data1[:, 1])

ax2.plot(data2[:, 1], data2[:, 2], color="#77AC30", label='1s')
ax2.plot(data2[:, 1], data2[:, 3], color="#A2142F", label='2n')
ax2.set_xlabel('Max Users')
ax2.set_ylabel('Average Response Time (s)')
ax2.set_title('Table 2')

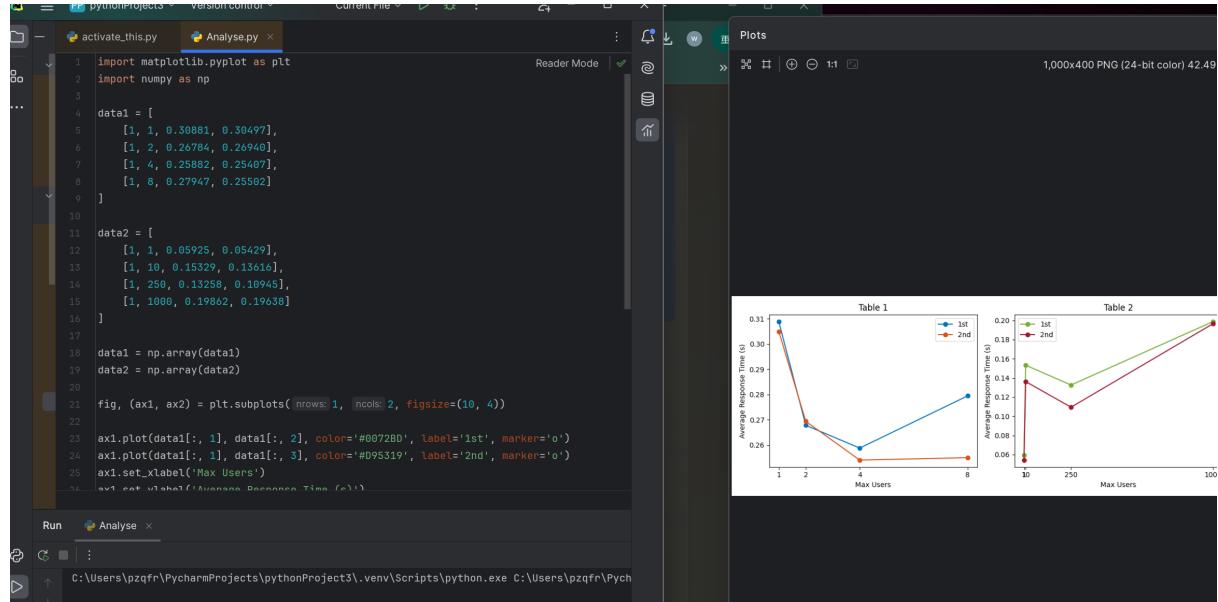
```

```

ax2.legend()
ax2.set_xticks(data2[:, 1])

plt.tight_layout()
plt.show()

```



I performed the following analysis on the performance test results of Table 1 (Web) and Table 2 (Local):

Table 1 (Web):

- (1) Table 1 shows the results of testing in a Web environment.
- (2) The test used 1 Pod and gradually increased the maximum number of users (Max Users) from 1 to 8.
- (3) As the maximum number of users increases, the average response time generally shows a downward trend, from about 0.31 seconds for 1 user to about 0.26 seconds for 8 users.
- (4) The results of the two tests (1st and 2nd) are very close, indicating that the performance of the system in the Web environment is relatively stable.
- (5) Within the range of the number of users tested, the system can handle the increase in concurrent users well, and the average response time remains low.

Table 2 (Local):

Table 2 shows the results of testing in the local environment (Local).

- (1) The test also used 1 Pod, but the maximum number of users (Max Users) increased even more, from 1 to 1,000.

(2)The average response time fluctuates as the maximum number of users increases. From about 0.06 seconds with 1 user, it rises to about 0.15 seconds with 10 users, then drops slightly with 250 users, and finally rises again to about 0.20 seconds with 1000 users.

(3)The results of the two tests (1st and 2nd) are also relatively close, indicating that the performance of the system in the local environment is relatively stable.

(4)Although the response time fluctuates, overall, the average response time is still within an acceptable range even with a high number of concurrent users (1000).

Compare Table 1 and Table 2:

Different

(1)The tests of Table 1 were conducted in the Web environment, while the tests of Table 2 were conducted in the local environment.

(2)Table 1 has a smaller maximum number of users (1 to 8), while Table 2 has a larger range (1 to 1000).

(3)The average response times in Table 1 are generally higher than those in Table 2. This may be due to network latency and communication overhead in a web environment.

(4)Although the environment and user number range are different, both tables demonstrate the stability and scalability of the system under different concurrent loads.

(5)Overall, these test results show that the system can handle different numbers of concurrent users well and maintain relatively stable performance, whether in a Web environment or a local environment. This provides important reference data for system deployment and expansion. Of course, in order to more comprehensively evaluate the performance of the system, more tests are needed, such as increasing the number of Pods, adjusting resource configuration, etc.

Command dictionary k8s-Master history

```
1 sudo apt update
2 sudo mkdir -m 0755 -p /etc/apt/keyrings
3 sudo chmod a+r /etc/apt/keyrings/docker.gpg
4 echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/a
```

```
pt/sources.list.d/docker.list > /dev/null
      5  sudo apt update
      6  sudo dpkg -i cri-dockerd_0.3.10.3-0.ubuntu-jammy_amd
64.deb
      7  sudo curl -fsSL https://pkgs.k8s.io/core:/stable:/v
1.29/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyri
ngs/kubernetes-apt-keyring.gpg
      8  echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-ap
t-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.29/deb/
/' | sudo tee /etc/apt/sources.list.d/kubernetes.list
      9  sudo apt update && sudo apt-get install -y kubelet k
ubeadm kubectl
     10 sudo apt-mark hold kubelet kubeadm kubectl
     11 kubeadm version
     12 sudo swapoff -a
     13 sudo nano /etc/iptables/rules.v4
     14 sudo kubeadm init --pod-network-cidr=10.244.0.0/16 -
-cri-socket=unix:///var/run/cri-dockerd.sock --apiserver-a
dvertise-address=10.0.0.115
     15 mkdir -p $HOME/.kube
     16 sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/co
nfig
     17 sudo chown $(id -u):$(id -g) $HOME/.kube/config
     18 sudo kubeadm init --pod-network-cidr=10.244.0.0/16 -
-cri-socket=unix:///var/run/cri-dockerd.sock --apiserver-a
dvertise-address=10.0.0.115
     19 sudo ufw disable
     20 sudo iptables -F
     21 sudo iptables -X
     22 sudo iptables -P INPUT ACCEPT
     23 sudo systemctl stop firewalld
     24 kubectl get nodes
     25 kubectl get pods --all-namespaces
     26 kubectl get nodes -o wide
     27 kubectl get nodes
     28 kubectl get pods --all-namespaces
     29 kubectl get pods --all-namespaces -o wide
     30 kubectl cluster-info
```

```
31 kubectl get nodes
32 kubectl describe node k8s-master
33 kubectl get namespace
34 kubectl create namespace ziqipei
35 kubectl get nodes,deployments,pods -o wide
36 kubectl get nodes,deployments,pods --namespace=farza
d -o wide
37 kubectl get nodes,deployments,pods --namespace=ziqipei -o wide
38 kubectl get nodes,deployments,pods --namespace=ziqipei -o wide
39 kubectl get deployments --namespace=ziqipei
40 kubectl get pods --namespace=ziqipei
41 mkdir ziqipei
42 cd ziqipei/
43 nano my-deployment.yaml
44 nano my-deployment.yml
45 kubectl apply -f my-deployment.yaml --namespace=ziqipei
46 kubectl get deployments --namespace=ziqipei
47 kubectl get pods --namespace=ziqipei
48 kubectl logs nginx-deployment-7c79c4bf97-2lb58 --name
space=ziqipei
49 kubectl logs nginx-deployment-7c79c4bf97-t82pp --name
space=ziqipei
50 kubectl logs nginx-deployment-7c79c4bf97-2lb58 --name
space=ziqipei
51 kubectl get pods -o wide --namespace=ziqipei
52 kubectl logs nginx-deployment-7c79c4bf97-2lb58 --name
space=ziqipei
53 kubectl logs nginx-deployment-7c79c4bf97-t82pp --name
space=ziqipei
54 kubectl logs nginx-deployment-7c79c4bf97-t82pp --name
space=ziqipei -- /bin/bash
55 kubectl logs nginx-deployment-7c79c4bf97-t82pp --name
space=ziqipei
56 sudo systemctl status kubelet
57 sudo systemctl restart kubelet
```

```
58 kubectl logs nginx-deployment-7c79c4bf97-t82pp --name
espace=ziqipei
59 sudo systemctl status kubelet
60 kubectl get nodes
61 kubectl logs nginx-deployment-7c79c4bf97-t82pp --name
espace=ziqipei
62 ping 10.0.0.242
63 telnet 10.0.0.242 10250
64 kubectl get nodes
65 kubectl logs nginx-deployment-7c79c4bf97-t82pp --name
espace=ziqipei
66 telnet 10.0.0.242 10250
67 kubectl get nodes
68 telnet 10.0.0.242 10250
69 telnet 10.0.0.242 10250
70 kubectl logs nginx-deployment-7c79c4bf97-t82pp --name
espace=ziqipei
71 kubectl exec -it nginx-deployment-7c79c4bf97-t82pp -
-namespace=ziqipei -- /bin/bash
72 kubectl get pods -o wide --namespace=ziqipei
73 kubectl scale deployment nginx-deployment --replicas
=2 --namespace=ziqipei
74 kubectl get pods -o wide --namespace=ziqipei
75 curl 10.40.0.2:80
76 curl 10.46.0.1:80
77 nano my-service.yml
78 kubetal get svc --namespace=ziqipei -o wide
79 kubectl get svc --namespace=ziqipei -o wide
80 curl 10.106.43.189:8080
81 ping 207.211.146.226
82 sudo iptables -F
83 sudo apt install python3.12.2
84 sudo apt install python3.12
85 cd /
86 apt-get install libssl-dev openssl make gcc libffi-d
ev zlib1g-dev libbz2-dev zlibc wget
87 ls
88 cd opt/
```

```
89 ls
90 mkdir software
91 mkdir -f software
92 cd ~
93 ls
94 cd ziqipei/
95 ls
96 mkdir software
97 ls
98 cd software/
99 mkdir python
100 ls
101 cd python/
102 wget https://registry.npmmirror.com/-/binary/python/
3.12.2/Python-3.12.2.tgz
103 ls
104 remove -r Python-3.12.2.tgz
105 remove Python-3.12.2.tgz
106 rm Python-3.12.2.tgz
107 ls
108 cd ../..
109 rm -r software/
110 ls
111 cat my-
112 cat my-service.yml
113 kubectl get nodes
114 docker ps -a
115 systemctl status docker.service
116 ls
117 mkdir ass
118 ls
119 cd ass/
120 nano test.py
121 nano test2.py
122 ls
123 python test
124 python3 test.py
125 rm -r test*
```

```
126 ls
127 ls
128 nano object_detection.py
129 sudo passwd ziqipei
130 ls
131 cat /etc/passwd
132 sudo passwd lxd
133 chmod 777 /opt/
134 mkdir /upload
135 su root
136 sudo passwd root
137 su root
138 sudo systemctl restart kubelet
139 kubectl get nodes
140 cd /opt/
141 ls
142 cd ass/
143 ls
144 kubectl apply -f deployment.yaml
145 cat deployment.yaml
146 kubectl apply -f deployment.yaml
147 kubectl delete all --all -n ziqipei
148 kubectl delete all --all -n default
149 kubectl apply -f deployment.yaml --namespace=ziqipei
150 kubectl get pods -o wide --namespace=ziqipei
151 kubectl delete all --all -n ziqipei
152 kubectl apply -f deployment.yaml --namespace=ziqipei
153 kubectl get pods -o wide --namespace=ziqipei
154 sudo systemctl restart kubelet
155 kubectl get pods -o wide --namespace=ziqipei
156 kubectl get nodes
157 kubectl get pods -o wide --namespace=ziqipei
158 kubectl logs my-link-deployment-7c84755d64-4mrqz --namespace=ziqipei
159 kubectl delete all --all -n ziqipei
160 docker login
161 docker push pinq/my-link:latest
162 docker tag my-python-app:latest pinq/my-link:latest
```

```
163 docker tag my-link:latest pjq/my-link:latest
164 sudo docker tag my-link:latest 5225solution/my-link:
latest
165 sudo docker tag my-link:latest pjq/my-link:latest
166 sudo docker push pjq/my-link:latest
167 sudo docker login
168 sudo docker push pjq/my-link:latest
169 sudo docker tag my-link:latest 5225solution/my-link:
latest
170 sudo docker push 5225solution/my-link:latest
171 docker push ziqipei/my-link:tagname
172 sudo docker push ziqipei/my-link:tagname
173 kubectl get nodes
174 kubectl apply -f deployment.yaml --namespace=ziqipei
175 kubectl get pods -o wide --namespace=ziqipei
176 kubectl delete all --all -n ziqipei
177 sudo usermod -aG docker ${USER}
178 docker run hello-world
179 sudo docker run hello-world
180 sudo docker ps-a
181 sudo docker ps -a
182 sudo docker image list
183 sudo docker rm -f $(docker ps -aq)
184 sudo docker ps -a
185 sudo docker ps -aq
186 sudo docker ps -a
187 sudo docker rm -f $(docker ps -aq)
188 sudo docker rm -f d4dfbeb10645
189 sudo docker rm -f 060e412fd390
190 sudo docker rm -f d7b436effc5d
191 sudo docker rm -f 63d21a91314c
192 sudo docker ps -a
193 su root
194 sudo docker ps -a
195 sudo apt install -y apt-transport-https ca-certifica
tes curl gnupg software-properties-common lsb-release
196 sudo mkdir -m 0755 -p /etc/apt/keyrings
197 curl -fsSL https://download.docker.com/linux/ubuntu/
```

```
gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
198 echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
199 sudo apt update
200 sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
201 docker COMMAND --help
202 docker help COMMAND
203 docker run hello-world
204 sudo docker ps -a
205 sudo docker image lis
206 dudo docker image list
207 sudo docker image list
208 docker pull ubuntu
209 sudo docker pull ubuntu
210 sudo docker run ubuntu echo "Hello from ubuntu!"
211 sudo docker run ubuntu sh -c "echo Hello FIT5225 from ubuntu; ls; echo bye guys"
212 sudo docker run -it ubuntu bash
213 docker run -it --rm ubuntu bash
214 sudo docker run -it --rm ubuntu bash
215 kubectl apply -f deployment.yaml --namespace=ziqipei
216 kubectl get pods -o wide --namespace=ziqipei
217 kubectl delete all --all -n ziqipei
218 kubectl get pods -o wide --namespace=ziqipei
219 kubectl apply -f deployment.yaml --namespace=ziqipei
220 kubectl get pods -o wide --namespace=ziqipei
221 kubectl delete all --all -n ziqipei
222 kubectl apply -f deployment.yaml --namespace=ziqipei
223 kubectl get pods -o wide --namespace=ziqipei
224 sudo images
225 sudo docker images
226 sudo docker -h
227 sudo docker container
228 sudo docker ps
229 docker build -t my-link:latest .
```

```
230 sudo docker build -t my-link:latest .
231 kubectl apply -f deployment.yaml --namespace=ziqipei
232 kubectl get pods -o wide --namespace=ziqipei
233 kubectl delete all --all -n ziqipei
234 kubectl apply -f deployment.yaml --namespace=ziqipei
235 kubectl get pods -o wide --namespace=ziqipei
236 kubectl delete all --all -n ziqipei
237 sudo docker save -o my-link.tar my-link:latest
238 ls
239 scp my-link.tar ubuntu@k8s-master:/path/to/my-link.t
ar
240 ssh ubuntu@k8s-master
241 sudo ssh ubuntu@k8s-master
242 sudo docker load -i /path/to/my-link.tar
243 sudo scp my-link.tar ubuntu@k8s-master:/path/to/my-l
ink.tar
244 sudo scp -o PreferredAuthentications=password my-lin
k.tar ubuntu@k8s-master:/path/to/my-link.tar
245 sudo scp my-link.tar ubuntu@k8s-master:/path/to/my-l
ink.tar
246 sudo scp my-link.tar root@k8s-master:/path/to/my-lin
k.tar
247 sudo docker run -d -p 5000:5000 my-link:latest
248 sudo docker build -t my-link:latest .
249 sudo docker run -d -p 5000:5000 my-link:latest
250 sudo systemctl restart docker
251 sudo docker stop $(sudo docker ps -aq)
252 sudo docker network prune
253 sudo docker run -d -p 5000:5000 my-link:latest
254 kubectl delete all --all -n ziqipei
255 kubectl apply -f deployment.yaml --namespace=ziqipei
256 kubectl get pods -o wide --namespace=ziqipei
257 kubectl get nodes
258 sudo docker network prune
259 sudo docker stop $(sudo docker ps -aq)
260 sudo docker build -t my-link:latest .
261 cd /opt/ass/
262 ls
```

```
263 sudo docker build -t my-link:latest .
264 exit
265 cd /opt/ass/
266 docker images
267 docker rmi 5225solution/my-link:latest
268 docker images
269 docker rmi pzq/my-link:latest
270 docker rmi my-link:latest
271 docker images
272 docker ps
273 docker rm -f $(docker ps -aq)
274 docker ps
275 docker rmi my-link:latest
276 docker images
277 docker rmi hello-world:latest
278 docker rmi ubuntu:latest
279 docker images
280 ls
281 rm -r my-link.tar
282 ls
283 docker image build -t ziqipei/my-link:1.0.0 .
284 docker image
285 docker images
286 docker push ziqipei/my-link:1.0.0
287 docker images
288 docker ps
289 cat requirements.txt
290 docker image
291 docker images
292 docker rmi ziqipei/my-link:1.0.0
293 docker images
294 docker image build -t ziqipei/my-link:1.0.0 .
295 docker push ziqipei/my-link:1.0.0
296 docker images
297 docker ps -a --filter ancestor=4ee8a4619bd3
298 kubectl apply -f deployment.yaml --namespace=ziqipei
299 kubectl get pods -o wide --namespace=ziqipei
300 kubectl delete all --all -n ziqipei
```

```
301 kubectl apply -f deployment.yaml --namespace=ziqipei
302 kubectl get pods -o wide --namespace=ziqipei
303 kubectl delete all --all -n ziqipei
304 kubectl get pods -o wide --namespace=ziqipei
305 kubectl apply -f deployment.yaml --namespace=ziqipei
306 kubectl get pods -o wide --namespace=ziqipei
307 kubectl describe pod my-link-deployment-68678f49ff-7
45n5 --namespace=your-namespace
308 kubectl describe pod my-link-deployment-68678f49ff-7
45n5 --namespace=ziqipei
309 kubectl get pods -o wide --namespace=ziqipei
310 kubectl logs my-link-deployment-68678f49ff-745n5 --n
amespace=ziqipei
311 kubectl delete all --all -n ziqipei
312 docker image build -t ziqipei/my-link:1.0.0 .
313 docker push ziqipei/my-link:1.0.0
314 kubectl apply -f deployment.yaml --namespace=ziqipei
315 kubectl get pods -o wide --namespace=ziqipei
316 kubectl logs my-link-deployment-68678f49ff-745n5 --n
amespace=ziqipei
317 kubectl logs my-link-deployment-68678f49ff-7rdsz --na
mespace=ziqipei
318 kubectl logs my-link-deployment-68678f49ff-7rdsz --n
amespace=ziqipei
319 kubectl delete all --all -n ziqipei
320 su root
321 kubectl apply -f deployment.yaml --namespace=ziqipei
322 kubectl get pods -o wide --namespace=ziqipei
323 kubectl logs my-link-deployment-68678f49ff-7rdsz --n
amespace=ziqipei
324 kubectl logs my-link-deployment-68678f49ff-h5hx7 --n
amespace=ziqipei
325 apt-get update
326 sudo apt-get update
327 cd /usr/local/
328 ls
329 cd lib/
330 ls
```

```
331 cd python 3.12
332 cd python3.12
333 ls
334 ls site-packages/
335 python3 -V
336 cd /opt/ass/
337 ls
338 kubectl delete all --all -n ziqipei
339 docker image build -t ziqipei/my-link:1.0.0 .
340 ls /usr/local/lib/python3.12/site-packages/
341 docker image build -t ziqipei/my-link:1.0.0 .
342 ls /usr/local/lib/python3.10/
343 ls
344 ls /usr/local/lib/python3.10/dist-packages/
345 kubectl delete all --all -n ziqipei
346 docker image build -t ziqipei/my-link:1.0.0 .
347 docker system prune --all --force --volumes
348 docker image build -t ziqipei/my-link:1.0.0 .
349 docker system prune --all --force --volumes
350 docker image build -t ziqipei/my-link:1.0.0 .
351 docker push ziqipei/my-link:1.0.0
352 kubectl apply -f deployment.yaml --namespace=ziqipei
353 kubectl get pods -o wide --namespace=ziqipei
354 kubectl logs my-link-deployment-68678f49ff-glz57 --n
amespace=ziqipei
355 kubectl delete all --all -n ziqipei
356 docker system prune --all --force --volumes
357 docker image build -t ziqipei/my-link:1.0.0 .
358 docker push ziqipei/my-link:1.0.0
359 kubectl apply -f deployment.yaml --namespace=ziqipei
360 kubectl get pods -o wide --namespace=ziqipei
361 kubectl logs my-link-deployment-68678f49ff-glz57 --n
amespace=ziqipei
362 kubectl logs my-link-deployment-68678f49ff-4lxqw --n
amespace=ziqipei
363 kubectl delete all --all -n ziqipei
364 docker system prune --all --force --volumes
365 cat Dockerfile
```

```
366 docker system prune --all --force --volumes
367 docker image build -t ziqipei/my-link:1.0.0 .
368 docker system prune --all --force --volumes
369 docker image build -t ziqipei/my-link:1.0.0 .
370 docker push ziqipei/my-link:1.0.0
371 kubectl apply -f deployment.yaml --namespace=ziqipei
372 kubectl get pods -o wide --namespace=ziqipei
373 kubectl logs my-link-deployment-68678f49ff-w65qg --n
amespace=ziqipei
374 sudo apt-get install libgl1-mesa-glx
375 docker system prune --all --force --volumes
376 docker image build -t ziqipei/my-link:1.0.0 .
377 docker push ziqipei/my-link:1.0.0
378 docker images
379 docker exec -it 4b400bd76b6e /bin/bash
380 docker exec -it ziqipei /bin/bash
381 docker run -it my-link /bin/bash
382 docker run -it ziqipei/my-link /bin/bash
383 docker login
384 docker run -it ziqipei/my-link /bin/bash
385 kubectl apply -f deployment.yaml --namespace=ziqipei
386 kubectl get pods -o wide --namespace=ziqipei
387 kubectl logs my-link-deployment-68678f49ff-w65qg --n
amespace=ziqipei
388 kubectl delete all --all -n ziqipei
389 docker system prune --all --force --volumes
390 docker image build -t ziqipei/my-link:1.0.0 .
391 docker push ziqipei/my-link:1.0.0
392 kubectl apply -f deployment.yaml --namespace=ziqipei
393 kubectl get pods -o wide --namespace=ziqipei
394 kubectl logs my-link-deployment-68678f49ff-62h7z --n
amespace=ziqipei
395 kubectl delete all --all -n ziqipei
396 docker system prune --all --force --volumes
397 docker image build -t ziqipei/my-link:1.0.0 .
398 kubectl apply -f deployment.yaml --namespace=ziqipei
399 kubectl get pods -o wide --namespace=ziqipei
400 kubectl logs my-link-deployment-68678f49ff-jxtcz --n
```

```
amespace=ziqipei
401 cd/opt/ass
402 ls
403 cd /opt/ass
404 cd /opt/ass
405 docker system prune --all --force --volumes
406 docker image build -t ziqipei/my-link:1.0.0 .
407 docker system prune --all --force --volumes
408 docker image build -t ziqipei/my-link:1.0.0 .
409 kubectl apply -f deployment.yaml --namespace=ziqipei
410 kubectl get pods -o wide --namespace=ziqipei
411 kubectl logs my-link-deployment-68678f49ff-jxtcz --n
amespace=ziqipei
412 ls
413 docker build ,
414 docker build .
415 cat Dockerfile
416 docker image ls
417 docker run ab2
418 docker build .
419 cat Dockerfile
420 docker image ls
421 docker run ab2
422 docker build .
423 docker image ls
424 docker run 987
425 docker build .
426 docker image ls
427 docker run 81d
428 ls
429 ls yolo_tiny_configs
430 ls -l yolo_tiny_configs/coco.names
431 docker build .
432 docker image ls
433 ducker build bd5
434 docker build bd5e
435 docker build bd5ee943a917
436 docker image build -t ziqipei/my-link:1.0.0 .
```

```
437 kubectl apply -f deployment.yaml --namespace=ziqipei
438 kubectl get pods -o wide --namespace=ziqipei
439 docker system prune --all --force --volumes
440 docker image build -t ziqipei/my-link:1.0.0 .
441 kubectl apply -f deployment.yaml --namespace=ziqipei
442 docker image ls
443 docker run ziqipei/my-link
444 kubectl get pods -o wide --namespace=ziqipei
445 docker run 8bf544c1098c
446 docker image build -t ziqipei/my-link:1.0.0 .
447 docker image ls
448 docker run 71da27c59378
449 curl 207.211.146.226:80
450 curl 207.211.146.226
451 kubectl logs k8s-master --namespace=ziqipei
452 kubectl get pods -o wide --namesapce=ziqipei
453 kubectl get pods -o wide --namespace=ziqipei
454 kubectl logs -f k8s-master --namespace=ziqipei
455 kubectl logs -f my-link-deployment-68678f49ff-jxtcz
--namespace=ziqipei
456 ls
457 cat deployment.yaml
458 docker image
459 docker images
460 docker build -t my-link:latest
461 docker push ziqipei/my-link:tagname
462 sudo docker build -t my-link:latest .
463 docker login
464 docker image build -t ziqipei/my-link:1.0.0 .
465 cat dockerfile
466 cd /opt/ass
467 cat dockerfile
468 ls
469 cat Dockerfile
470 docker image build -t ziqipei/my-link:1.0.0 .
471 docker push ziqipei/my-link:tagname
472 docker push ziqipei/my-link:1.0.0
473 kubectl cluster-info
```

```
474 kubectl get nodes
475 kubectl apply -f deployment.yaml --namespace=ziqipei
476 kubectl get pods -o wide --namespace=ziqipei
477 kubectl logs my-link-deployment-68678f49ff-jxtcz --n
ame=space=<your-namespace>
478 kubectl logs my-link-deployment-68678f49ff-jxtcz --n
ame=space=ziqipei
479 ls
480 Cat Dockerfile
481 cat Dockerfile
482 docker image
483 docker images
484 docker push ziqipei/my-link:tagname
485 docker image build -t ziqipei/my-link:1.0.0 .
486 docker push ziqipei/my-link:tagname
487 docker push ziqipei/my-link
488 docker images
489 docker system prune --all --force --volumes
490 docker login
491 docker push ziqipei/my-link:1.0.0
492 docker image build -t ziqipei/my-link:1.0.0 .
493 docker push ziqipei/my-link:1.0.0
494 docker image
495 docker images
496 docker build -t ziqipei/my-link
497 docker build -t ziqipei/my-link .
498 docker login
499 docker push ziqipei/my-link:1.0.0
500 kubectl create deployment --image deployment --namesp
ace=ziqipei
501 kubectl get deployment
502 kubectl get deployment.yaml
503 kubectl get deployment -n ziqipei
504 kubectl get pods -n ziqipei
505 kubectl apply -f deployment.yaml --namespace=ziqipei
506 kubectl get pods -o wide --namespace=ziqipei
507 docker image build -t ziqipei/my-link:1.0.0 .
508 docker push ziqipei/my-link:1.0.0
```

```
509 docker image build -t ziqipei/my-link:1.0.0 .
510 docker push ziqipei/my-link:1.0.0
511 kubectl apply -f deployment.yaml --namespace=ziqipei
512 kubectl get pods -o wide --namespace=ziqipei
513 kubectl get logs my-link-deployment-68678f49ff-jxtcz
514 docker images
515 docker rmi ziqipei/link
516 docker rmi ziqipei/link:latest
517 docker rmi ziqipei/my-link:latest
518 docker images
519 docker rmi ziqipei/my-link:1.0.0
520 docker images
521 kubectl get pods -o wide --namespace=ziqipei
522 kubectl delete namespace ziqipei
523 kubectl get pods -o wide --namespace=ziqipei
524 docker image build -t ziqipei/my-link:1.0.0 .
525 docker push ziqipei/my-link:1.0.0
526 docker pull ziqipei/my-link:1.0.0
527 docker images
528 docker run eac2c9137f9b
529 kubectl apply -f deployment.yaml --namespace=ziqipei
530 cat deployment.yaml
531 kubectl create namespace ziqipei
532 kubectl apply -f deployment.yaml --namespace=ziqipei
533 kubectl get pods -o wide --namespace=ziqipei
534 kubectl logs my-link-deployment-68678f49ff-hqrj4 --n
amespace=ziqipei
535 docker push ziqipei/my-link:1.0.0
536 docker system prune --all --force --volumes
537 docker images
538 kubectl delete namespace ziqipei
539 kubectl create namespace ziqipei
540 docker image build -t ziqipei/my-link:1.0.0 .
541 cat Dockerfile
542 docker push ziqipei/my-link:1.0.0
543 kubectl apply -f deployment.yaml --namespace=ziqipei
544 kubectl get pods -o wide --namespace=ziqipei
545 kubectl logs my-link-deployment-68678f49ff-ldnkg --n
```

```
amespace=ziqipei
546 cat docker
547 cat Dockerfile
548 docker images
549 run cd5586df8fc7
550 docker run cd5586df8fc7
551 kubectl l logs my-link-deployment-68678f49ff-ldnkg --
namespace=ziqipei
552 kubectl get pods -o wide --namespace=ziqipei
553 kubectl logs my-link-deployment-68678f49ff-ldnkg --n
amespace=ziqipei
554 docker pull ziqipei/my-link:1.0.0
555 kubectl delete namespace ziqipei
556 kubectl create namespace ziqipei
557 kubectl apply -f deployment.yaml --namespace=ziqipei
558 kubectl get pods -o wide --namespace=ziqipei
559 kubectl logs my-link-deployment-68678f49ff-fz5vj --n
amespace=ziqipei
560 kubectl get nodes
561 kubectl get pods -o wide --namespace=ziqipei
562 cat Dockerfile
563 cat requirements.txt
564 kubectl delete namespace ziqipei
565 kubectl create namespace ziqipei
566 docker image prune --all --force
567 docker system prune --all --force --volumes
568 docker images
569 docker image build -t ziqipei/my-link:1.0.0 .
570 docker images
571 docker run 747d542e48be
572 docker image build -t ziqipei/my-link:1.0.0 .
573 vi Dockerfile
574 docker image build -t ziqipei/my-link:1.0.0 .
575 vi Dockerfile
576 docker image build -t ziqipei/my-link:1.0.0 .
577 kubectl images
578 kubectl get pods
579 kubectl describe pods my-link-deployment-68678f49ff
```

```
-knms6
580 kubectl logs -f my-link-deployment-68678f49ff-knms6
581 docker images
582 docker rmi ziqipei/my-link:1.0.0
583 docker rmi ziqipei/my-link:1.0.0 -f
584 docker pull ziqipei/my-link:1.0.0
585 docker run -tid --name test ziqipei/my-link:1.0.0
586 docker logs -f test
587 kubectl get pods
588 kubectl delete pods my-link-deployment-68678f49ff-k
nms6
589 kubectl get pods
590 docker logs -f my-link-deployment-68678f49ff-59h9v
591 docker describe pods my-link-deployment-68678f49ff-
59h9v
592 kubectl describe pods my-link-deployment-68678f49f
f-59h9v
593 kubectl logs -f my-link-deployment-68678f49ff-59h9
v
594 history
595 kubectl get pods
596*
597 kubectl get pods
598 kubectl get pods -w
599 kubectl get pods
600 kubectl edit deployment my-link-deployment
601 kubectl get pods
602 kubectl get svc
603 ls
604 vi service.yaml
605*
606 vi service.yaml
607 kubectl apply -f service.yaml
608 kubectl version
609 vi service.yaml
610 kubectl apply -f service.yaml
611 kubectl get svc
612 kubectl get ep
```

```
613 cat service.yaml
614 curl 10.108.142.210:5000
615 curl 168.138.22.27:32009
616 curl 127.0.0.1:32009
617 ls
618 vi Dockerfile2
619 kubectl get node
620 cat service.yaml
621 kubectl apply -f deployment.yaml
622 kubectl edit deployment my-link-deployment
623 kubectl get pods
624 kubectl get pods -o wide --namespace=ziqipei
625 cat
626 cd /opt/ass/
627 ls
628 cat deployment.yaml
629 locust -V
630 kubectl get pods
631 kubectl delete pods my-link-deployment-68678f49ff-w
rg5m
632 kubectl apply -f deployment.yaml
633 kubectl get pods
634 kubectl logs -f my-link-deployment-775c7754f7-s5fh
1
635 kubectl delete pods my-link-deployment-775c7754f7-s
5fh1
636 kubectl apply -f deployment.yaml
637 kubectl get pods
638 kubectl logs -f my-link-deployment-775c7754f7-s5fh
1
639 kubectl logs -f my-link-deployment-775c7754f7-5rx9x
640 kubectl get pods
641 cat deployment.yaml
642 kubectl delete pods my-link-deployment-775c7754f7-s
5fh1
643 kubectl get pods
644 kubectl delete pods my-link-deployment-775c7754f7-5
rx9x
```

```
645 kubectl apply -f deployment.yaml
646 kubectl get pods
647 kubectl delete pods my-link-deployment-5769f584df-s
t9q7
648 kubectl apply -f deployment.yaml
649 kubectl get pods
650 kubectl logs -f my-link-deployment-5769f584df-s5phw
651 cat /opt/ass/locustfile.py
652 kubectl get pods
653 ls
654 kubectl exec -it my-link-deployment-5769f584df-s5phw
-- vi /opt/ass/locustfile.py
655 kubectl delete pods my-link-deployment-5769f584df-s
5phw
656 cat /opt/ass/locustfile.py
657 cat /opt/ass/deployment.yaml
658 kubectl apply -f deployment.yaml
659 kubectl get pods
660 locust -f locustfile.py
661 sudo ufw allow 8089/tcp
662 sudo apt install ufw
663 sudo ufw allow 8089/tcp
664 locust -f locustfile.py
665 clear
666 locust -f locustfile.py
667 kubectl get pods
668 kubectl logs -f my-link-deployment-68678f49ff-xhdpr
669 kubectl get pods
670*
671 kubectl apply -f deployment.yaml
672 kubectl get pods
673 kubectl logs -f my-link-deployment-68678f49ff-klndc
674 kubectl get pods
675*
676 history | grep "ls"
677 ls
678 sudo apt update
679 sudo mkdir -m 0755 -p /etc/apt/keyrings
```

```
680 sudo chmod a+r /etc/apt/keyrings/docker.gpg
681 echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
682 sudo apt update
683 sudo dpkg -i cri-dockerd_0.3.10.3-0.ubuntu-jammy_amd64.deb
684 sudo curl -fsSL https://pkgs.k8s.io/core:/stable:v1.29/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
685 kubectl get pods
686 ls
687 history
688* kubectl get node
689 docker image build -t ziqipei/my-link:1.0.0 .
690 cat Dockerfile
691 docker image build -t ziqipei/my-link:1.0.0 .
692 history
693 docker push ziqipei/my-link:1.0.0
694 kubectl get pods
695 docker images
696 docker run b54f78a8b61e
697 docker run cd5586df8fc7
698 kubectl get pods
699 kubectl apply -f deployment.yaml
700 kubectl get pods
701 kubectl logs my-link-deployment-68678f49ff-k1ndc
702 kubectl get pods,deployment,svc -owide
703 kubectl get pods,deployment,svc -owide
704 vi service.yaml
705 locust -f locustfile.py
706 history
```