

LAB 5

CI/CD PIPELINE USING JENKINS, GITHUB AND DOCKER



Fullscreen: Nguyen Minh Nhut

Student ID: B2205896

- Note: screenshots need to be clear and good-looking; submissions must be in PDF format.

1. Manually dockerize a Flask project

1.1. Deploy a Flask application

- Create a sample Flask application:

```
$mkdir cicd_tutorial ; cd cicd_tutorial  
$nano flask_docker.py
```

flask_docker.py

```
from flask import Flask  
app = Flask(__name__)  
  
@app.route('/')  
def hello_world():  
    return 'Hello FOSS'  
  
if __name__ == '__main__':  
    app.run(debug=True,host='0.0.0.0')
```

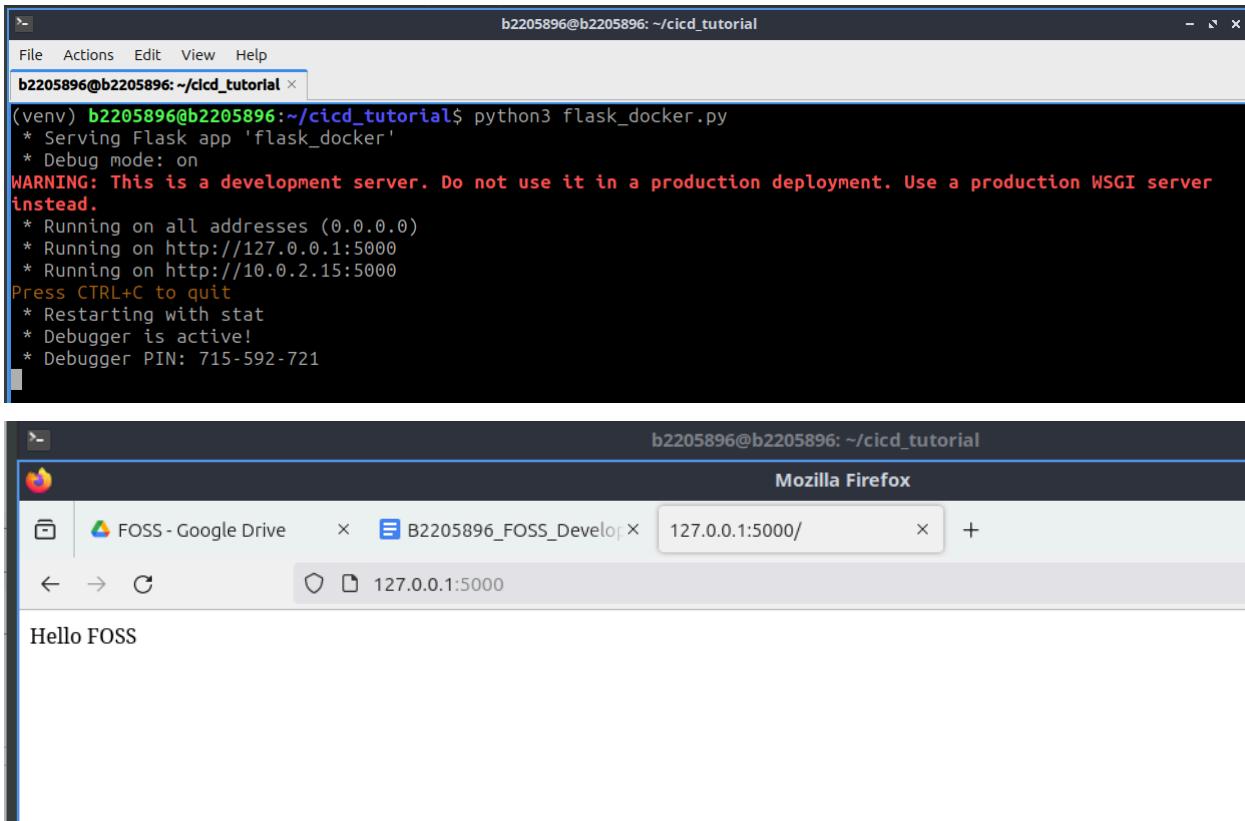
- Install pip (package installer for Python), and then the Flask framework

```
$sudo apt install python3-pip -y  
$pip3 install flask
```

- We can test it out by running:

```
$python3 flask_docker.py  
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)  
* Restarting with stat  
* Debugger is active!  
* Debugger PIN: 135-043-124
```

- Access the application from a browser (<http://localhost:5000>), (take a screenshot)



1.2. Dockerize a Flask application using Dockerfile

- Update the apt package index and install Docker

```
$sudo apt update
$sudo apt install docker.io -y
```

- Add current user to the docker group:

```
$sudo usermod -aG docker ${USER}
$su - ${USER}
```

- Check whether you can access and download images from Docker Hub

```
$docker run hello-world
```

The output will indicate that Docker is working correctly:

Hello from Docker!

This message shows that your installation appears to be working correctly.

- Create a requirements.txt file

```
$nano requirements.txt
```

requirements.txt

Flask==2.2.2

- Create a Dockerfile file

```
$nano Dockerfile
```

Dockerfile

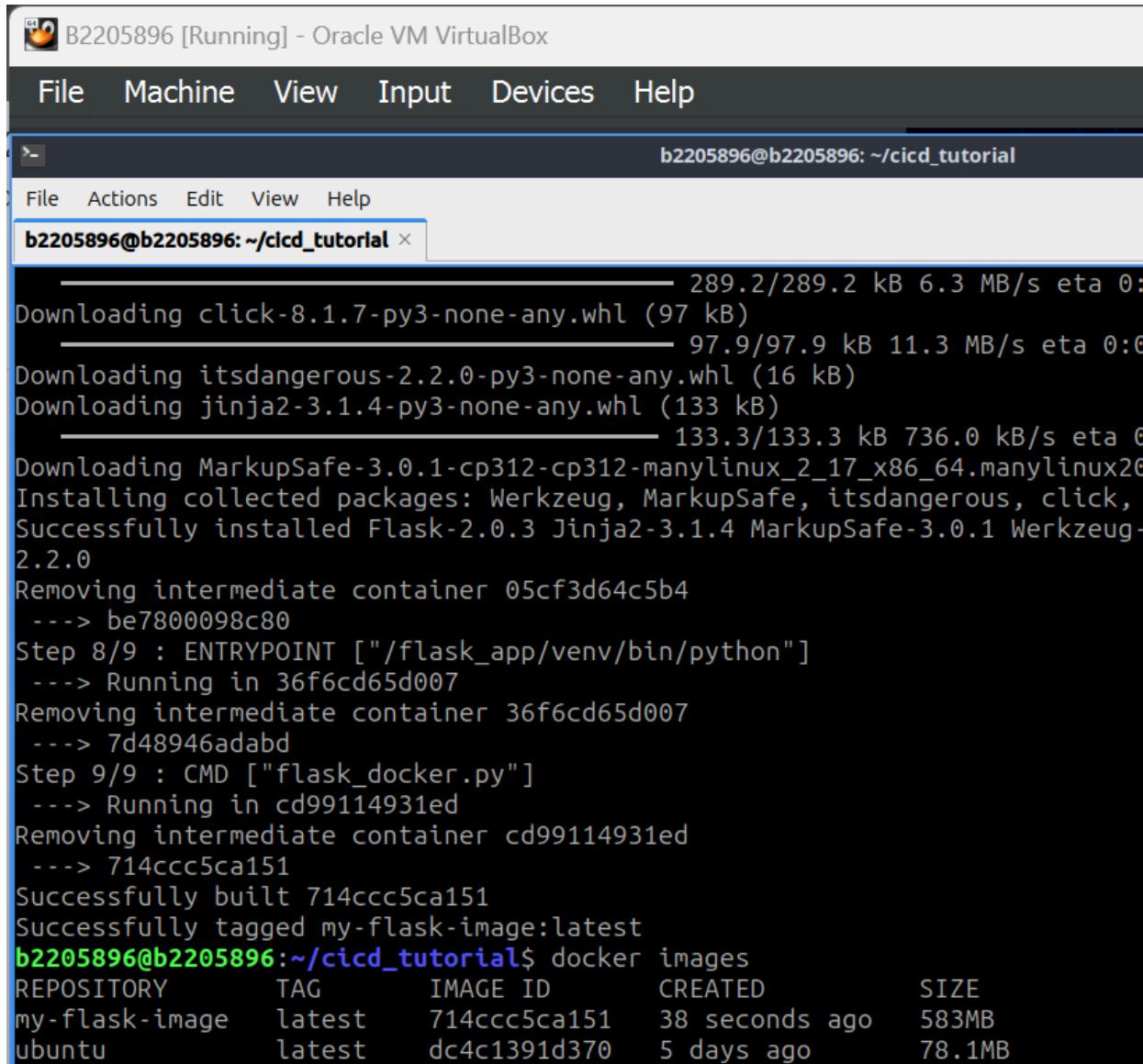
```
FROM ubuntu:latest
MAINTAINER Tuan Thai "tuanthai@example.com"
RUN apt update -y
RUN apt install -y python3-pip python3-dev build-essential
ADD . /flask_app
WORKDIR /flask_app
RUN pip3 install -r requirements.txt
ENTRYPOINT ["python3"]
CMD ["flask_docker.py"]
```

- Create a Docker image whose name is "my-flask-image:latest", using the Dockerfile

```
$docker build -t my-flask-image:latest .
```

- Then see if your image is in Docker (**take a screenshot**)

```
$docker images
```



```

B2205896 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
File Actions Edit View Help
b2205896@b2205896: ~/cicd_tutorial x
----- 289.2/289.2 kB 6.3 MB/s eta 0:00:00
Downloading click-8.1.7-py3-none-any.whl (97 kB)
----- 97.9/97.9 kB 11.3 MB/s eta 0:00:00
Downloading itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Downloading jinja2-3.1.4-py3-none-any.whl (133 kB)
----- 133.3/133.3 kB 736.0 kB/s eta 0:00:00
Downloading MarkupSafe-3.0.1-cp312-cp312-manylinux_2_17_x86_64.manylinux2014
Installing collected packages: Werkzeug, MarkupSafe, itsdangerous, click, Jinja2, Flask, MarkupSafe
Successfully installed Flask-2.0.3 Jinja2-3.1.4 MarkupSafe-3.0.1 Werkzeug-2.2.0
Removing intermediate container 05cf3d64c5b4
--> be7800098c80
Step 8/9 : ENTRYPOINT ["/flask_app/venv/bin/python"]
--> Running in 36f6cd65d007
Removing intermediate container 36f6cd65d007
--> 7d48946adabd
Step 9/9 : CMD ["flask_docker.py"]
--> Running in cd99114931ed
Removing intermediate container cd99114931ed
--> 714ccc5ca151
Successfully built 714ccc5ca151
Successfully tagged my-flask-image:latest
b2205896@b2205896:~/cicd_tutorial$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
my-flask-image      latest   714ccc5ca151  38 seconds ago  583MB
ubuntu              latest   dc4c1391d370  5 days ago    78.1MB

```

- Run your image (take a screenshot)

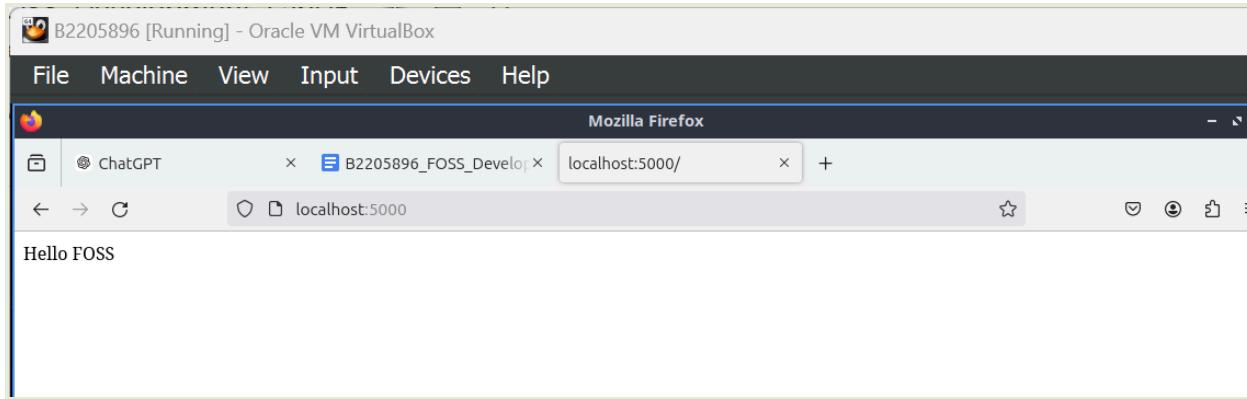
```
$docker run -d -p 5000:5000 my-flask-image
$docker ps
```

- Access the application from a browser (<http://localhost:5000>)

```

b2205896@b2205896:~/cicd_tutorial$ docker run -d -p 5000:5000 my-flask-image
a900b5ff15774d78915b2fe7685c70dd7caf5dad37efba48c11a79e42e879e3e
b2205896@b2205896:~/cicd_tutorial$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
NAMES
a900b5ff1577        my-flask-image     "/flask_app/venv/bin..."   19 seconds ago   Up 17 seconds   0.0.0.0:5000->5000/tcp, :::5000->5000/tcp
wonderful_pasteur
b2205896@b2205896:~/cicd_tutorial$ 

```



2. Automatically dockerize a Flask project using Jenkins

2.1. Push your code to a Github repository

- Create an account (or login) to GitHub at <https://github.com>
- Create a new repository, name it as "cicd_tutorial". Get the repository URL (for example: https://github.com/TuanThai/cicd_tutorial.git)
- Install and setup git on your computer (remember to set your name/email)

```
$sudo apt update ; sudo apt install git -y  
$git config --global user.name "Firstname Lastname"  
$git config --global user.email "example@ctu.edu.vn"
```

- Initialize git, commit and push your flask project files to Github

```
$mv ~/cicd_tutorial  
$git init  
$git add .  
$git commit -m "first commit"  
$git remote add origin <your repository URL>  
$git push -u origin master
```

2.2. Install and configure Jenkins

- Install Java and Jenkins

```
$sudo apt install openjdk-11-jdk -y  
$wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -  
$sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'  
$sudo apt update ; sudo apt install jenkins -y
```

- Launch Jenkins

```
$sudo usermod -aG docker jenkins  
$sudo systemctl restart jenkins.service
```

- Access Jenkins using a web browser (<http://localhost:8080>). Unlock Jenkins, install suggested plugins, create the first admin user.

2.3. Using Jenkins to automatically dockerize your application

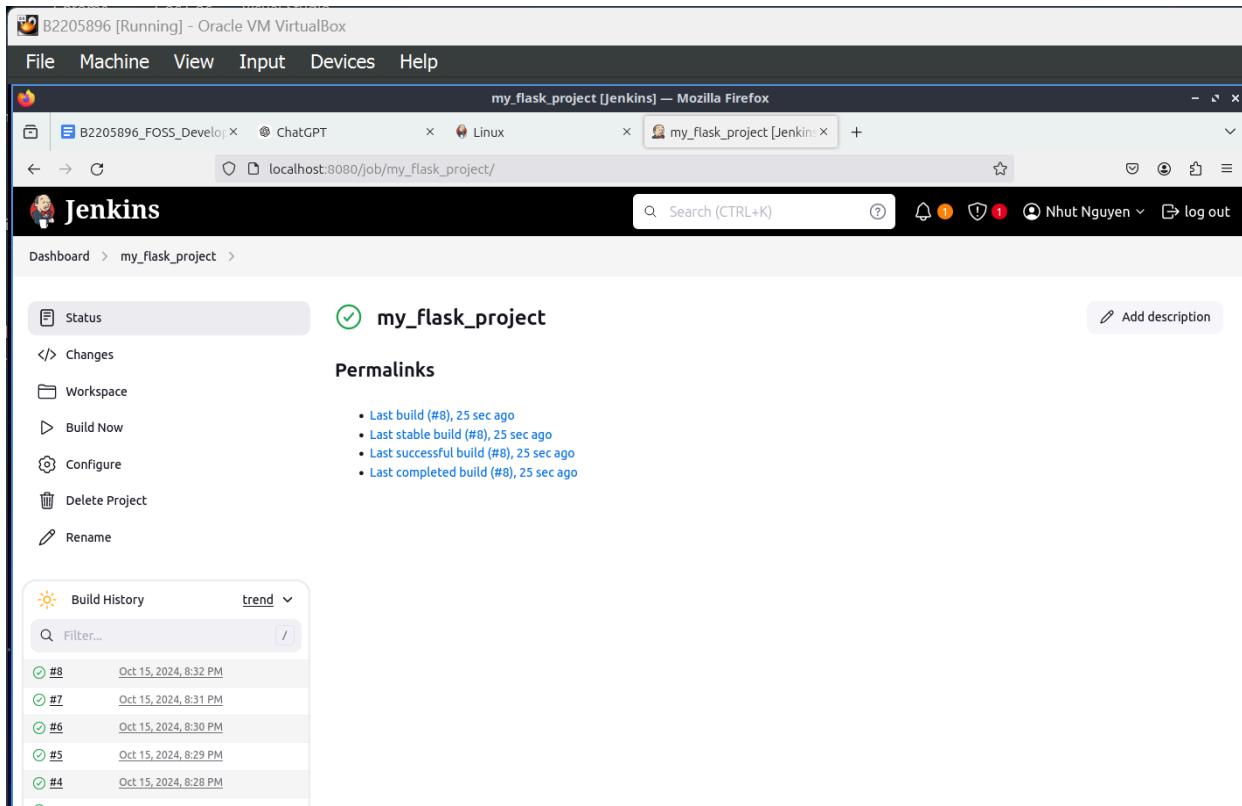
- On Jenkins dashboard, click "Create a new job", then choose "Freestyle project". Name your project as "my_flask_project"
- Under "Source Code Management" choose "Git", fill in your GitHub repository URL
- Under "Build Triggers" select "Build periodically", fill in "* * * * *" (build your project every minute)

```
* * * * *
```

- Under "Build" we will "Add build step", and select "Execute shell". Then fill in "docker build -t my-flask-image:latest ."

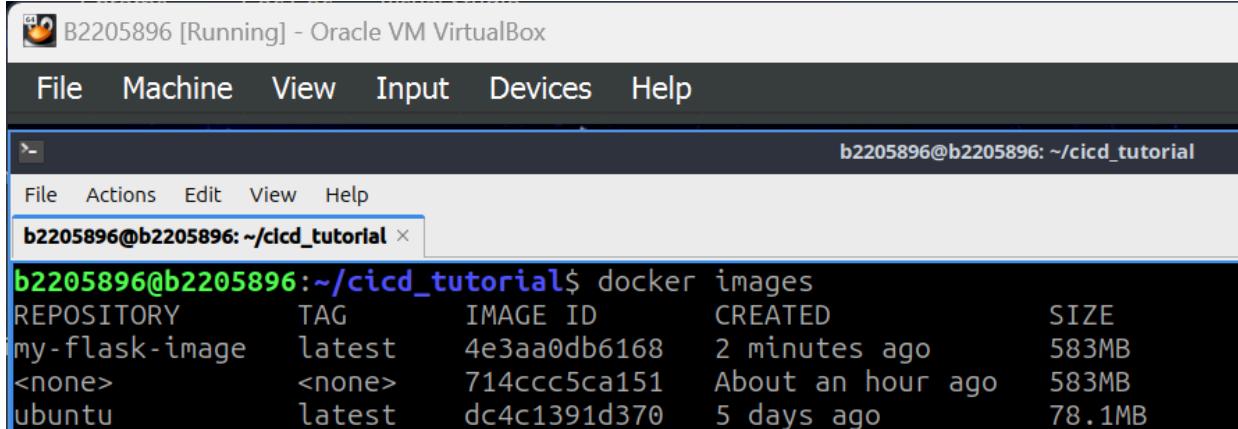
```
docker build -t my-flask-image:latest .
```

- Save your project. Then look at "Build history" to see that your project is built every minute.



- Then see if your image is in Docker (**take a screenshot**)

```
$docker images
```



B2205896 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

b2205896@b2205896: ~/cicd_tutorial

```
b2205896@b2205896:~/cicd_tutorial$ docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
my-flask-image  latest   4e3aa0db6168  2 minutes ago  583MB
<none>          <none>  714ccc5ca151  About an hour ago  583MB
ubuntu          latest   dc4c1391d370  5 days ago   78.1MB
```

- Modify your Flask application:

```
$nano flask_docker.py
```

```
from flask import Flask
app = Flask(__name__)
@app.route('/')

def hello_world():
    return 'Hello FOSS, Hello CI/CD using Jenkins'
if __name__ == '__main__':
    app.run(debug=True,host='0.0.0.0')
```

- Commit and push your project files to GitHub

```
$git add .
$git commit -m "second commit"
$git push origin master
```

- Wait 1 minute, then run your image

B2205896 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

b2205896@b2205896: ~/cicd_tutorial

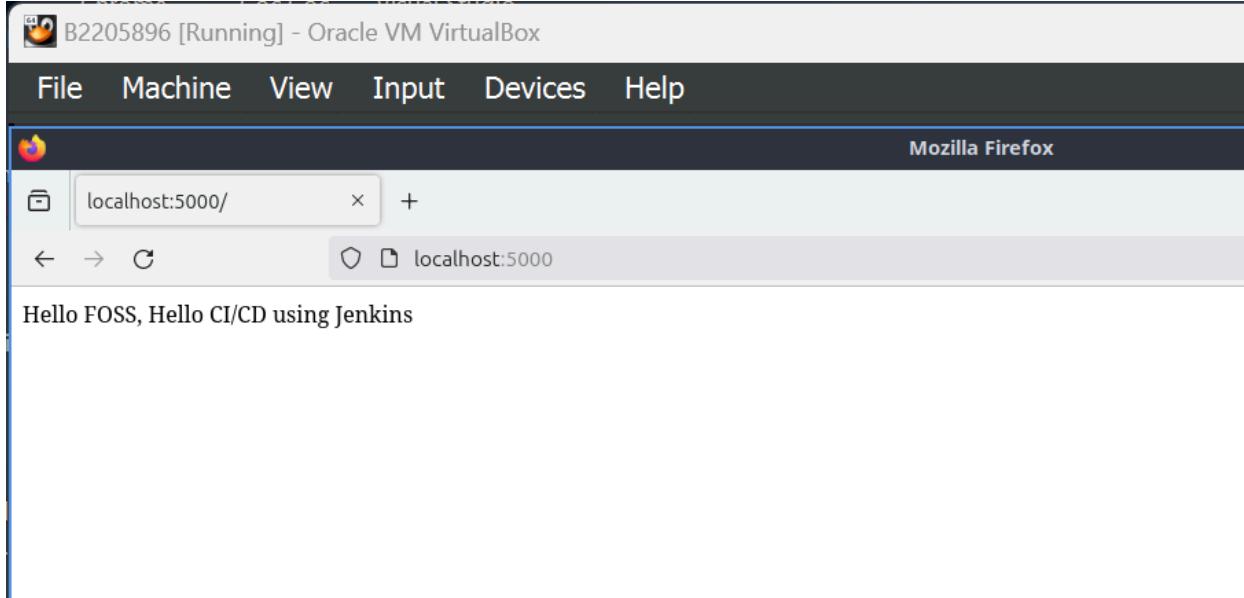
File Actions Edit View Help

b2205896@b2205896: ~/cicd_tutorial x

```
b2205896@b2205896:~/cicd_tutorial$ docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
my-flask-image  latest   d862a10f4a90  About a minute ago  583MB
<none>          <none>   4e3aa0db6168  8 minutes ago    583MB
<none>          <none>   714ccc5ca151  About an hour ago  583MB
ubuntu          latest   dc4c1391d370  5 days ago   78.1MB
```

```
$docker run -d -p 5000:5000 my-flask-image
$docker ps
```

- Access the application from a browser (<http://localhost:5000>) (take a screenshot)



- On your Jenkins project configure, under "Build Triggers", do not forget to deselect "Build periodically"

---END---