Docker

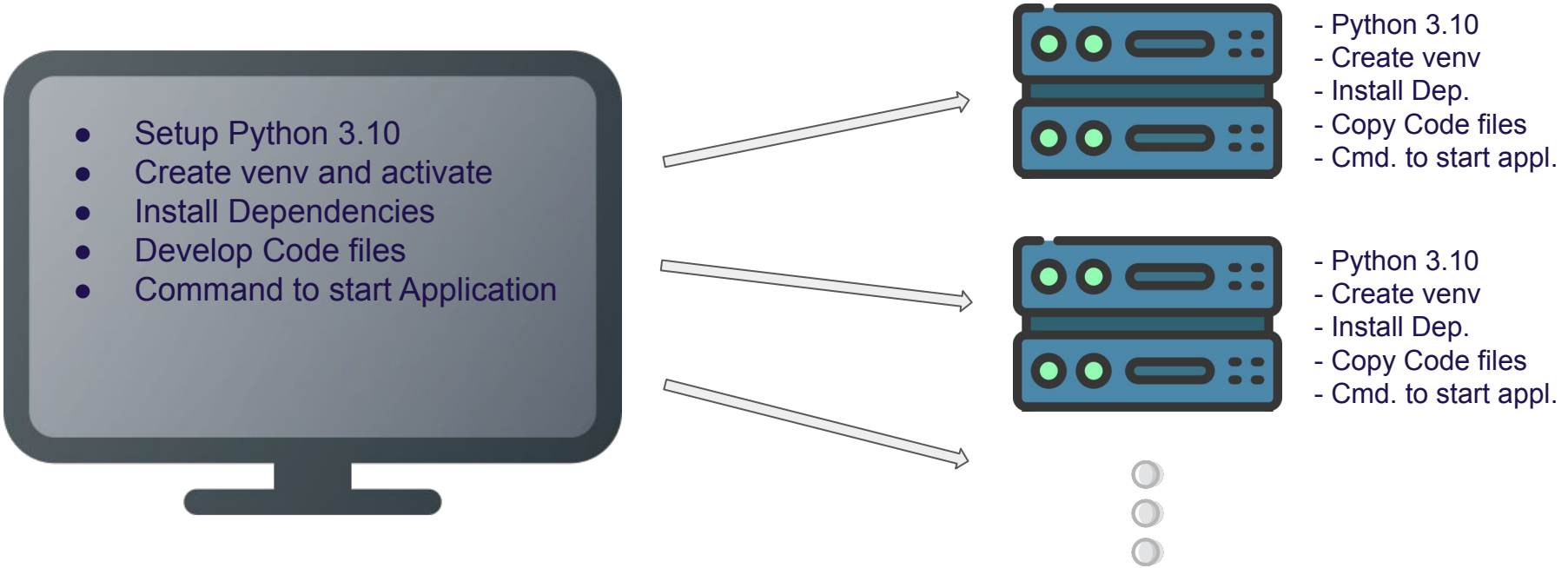
for Containerizing Application

Objectives

1. How to Run your Application on a Server? (Challenges in it)
2. What is meant by “Containerize your Application”
3. How to Containerize Application using “Docker”
4. How to create a “Docker Image”?
5. Basic Docker Commands
6. Demo
    a. Docker Basics
    b. Dockerize and push FastAPI Application

How to Run your Application on a Server?

- First, Build your application on your system, then take it to server



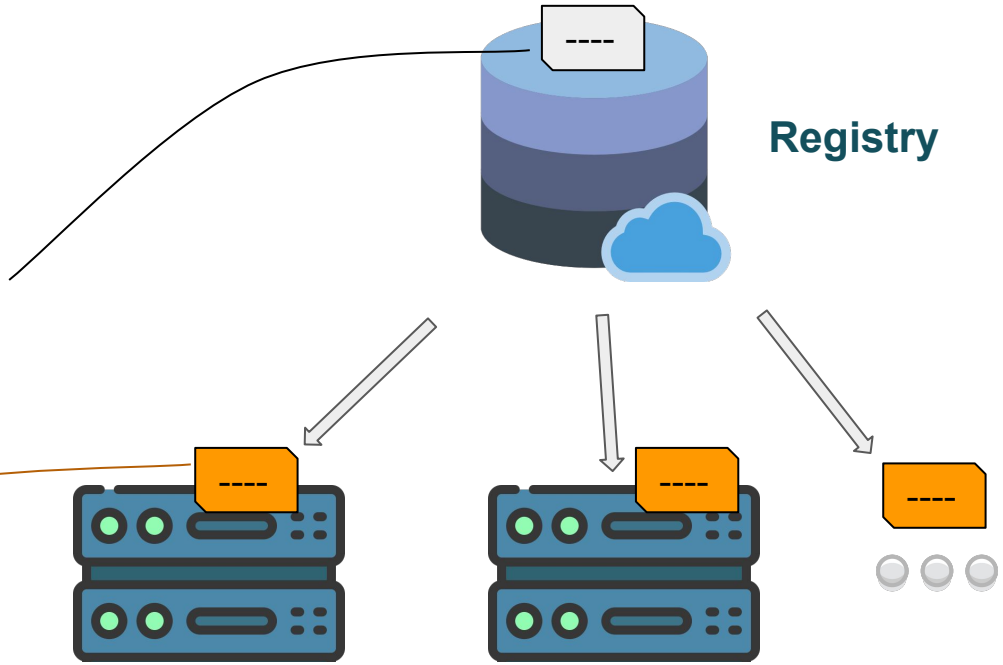
“ Process becomes Troublesome ”

Containerize your Application

- Once the application is build on your system, create a **template** of it
- Use **an instance** of that **template** to run the app anywhere

- Python:3.10 Installed
- Dependencies Installed
- Code files present
- Commands configured to start the appl.

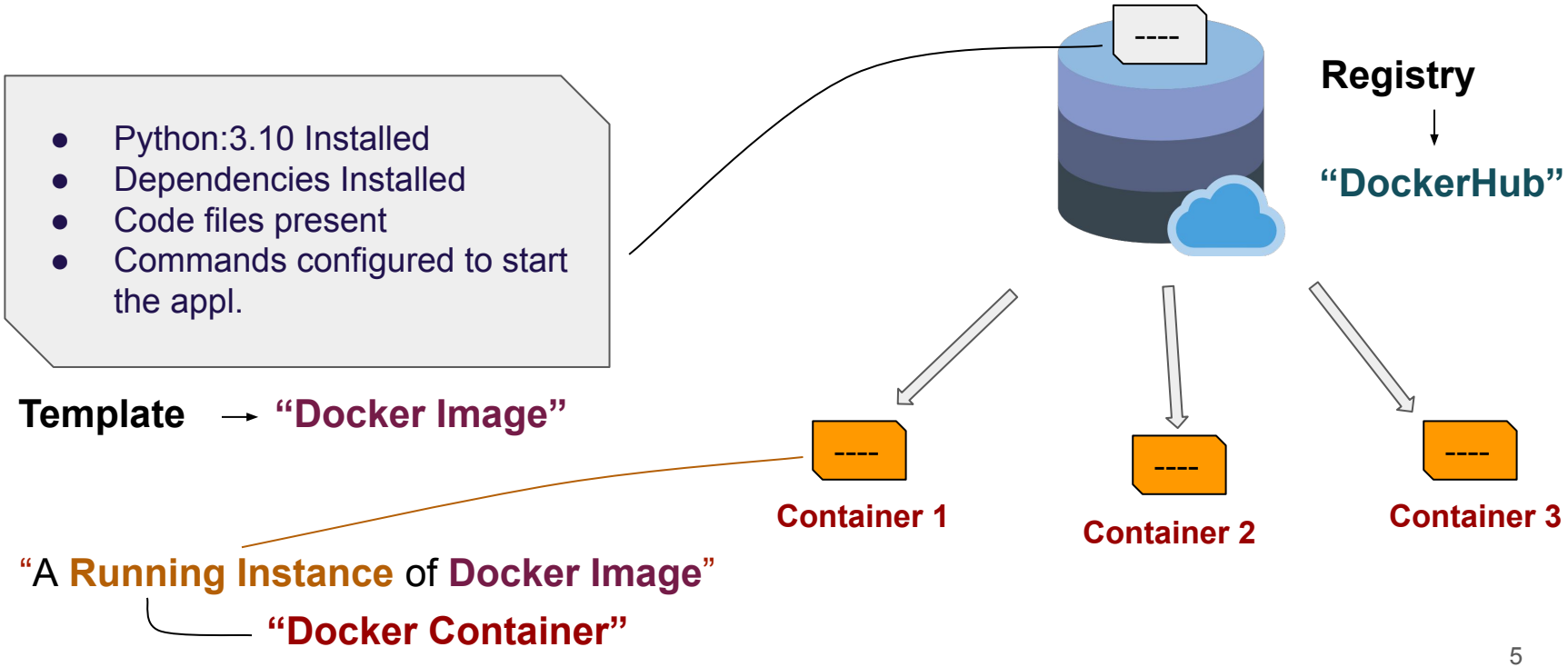
Template



"A **Running Instance** of **template**"

Containerize Application - “Docker”

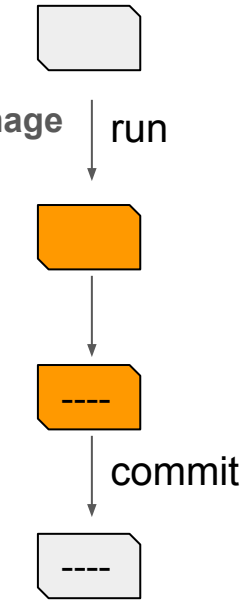
- Few terminologies to note:



How to create a “Docker Image”?

- Without **Dockerfile**

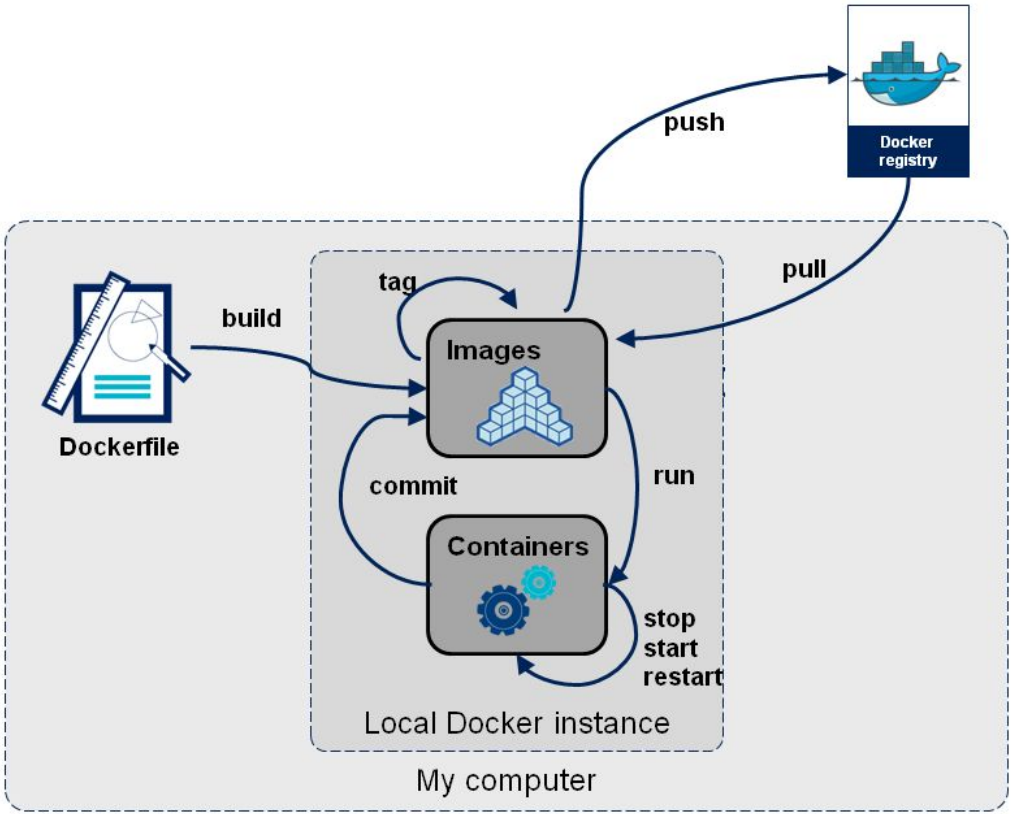
1. Run a Container using a **Base Image**
2. Inside Container
   a. **Manually** add your files
   b. Install dependencies
3. **Commit** the changes



- With **Dockerfile**

```
# pull python base image
FROM python:3.10
# copy application files
ADD . .
# specify working directory
WORKDIR /titanic_model_api
# update pip
RUN pip install --upgrade pip
# install dependencies
RUN pip install -r requirements.txt
# expose port for application
EXPOSE 8001
# start fastapi application
CMD ["python", "app/main.py"]
```

Basic Docker Commands




```
>> docker build . -t imgName:1
>> docker run --name=myApp imgName:1
        >> docker stop myApp
        >> docker start myApp
        >> docker restart myApp
        >> docker commit myApp newImg:1
>> docker tag newImg:1 username/newImg:2
>> docker push username/newImg:2
>> docker pull username/newImg:2
```

1. Docker Setup and Basics

Demo...

2. Dockerize and push FastAPI Application

Demo...

Thank you!