



L02 – Docker

977-361 Software Architecture and Design

Mr. Wasin Thiengkunakrit

Expected Learning Outcome

- Understand Container Technology
- Understand the Docker technology
- Be able to understand the Docker Compose stack

Contents



- Introduction to Container
- Docker Installation
- Docker Compose



Introduction to Container

What is Container ?

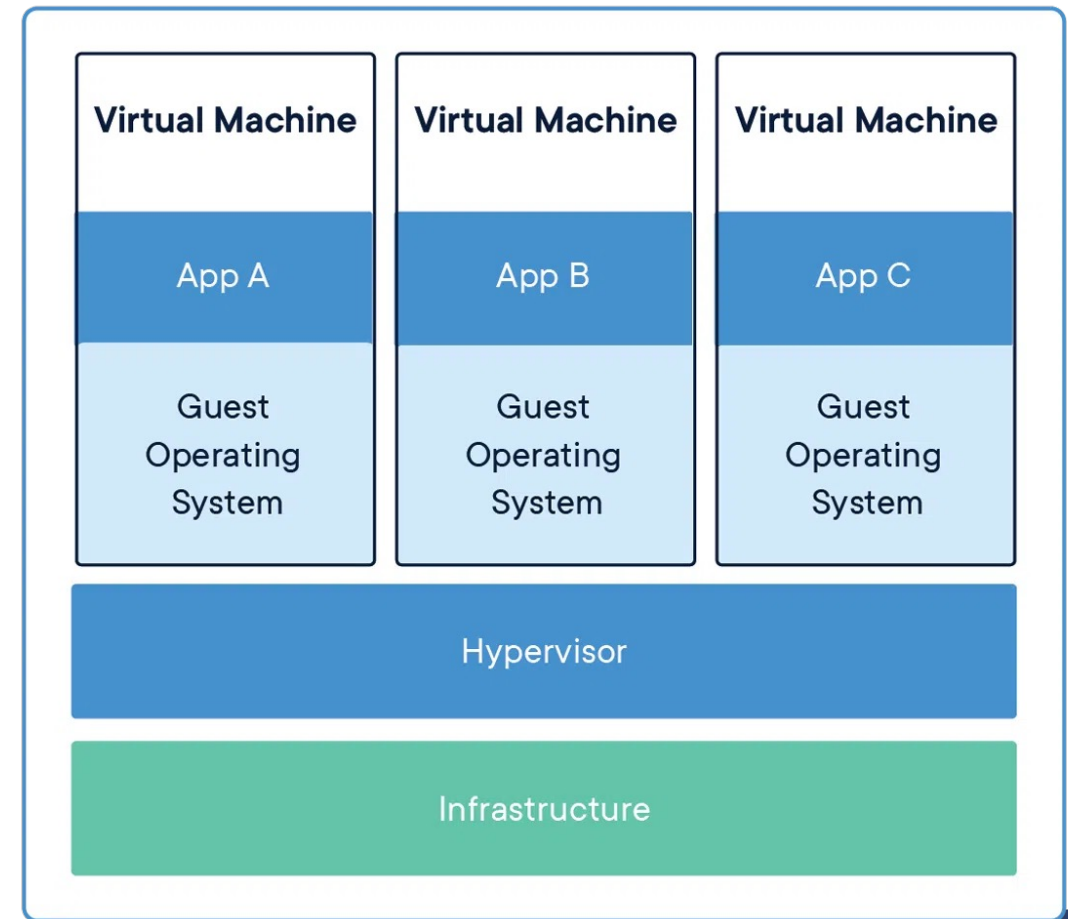
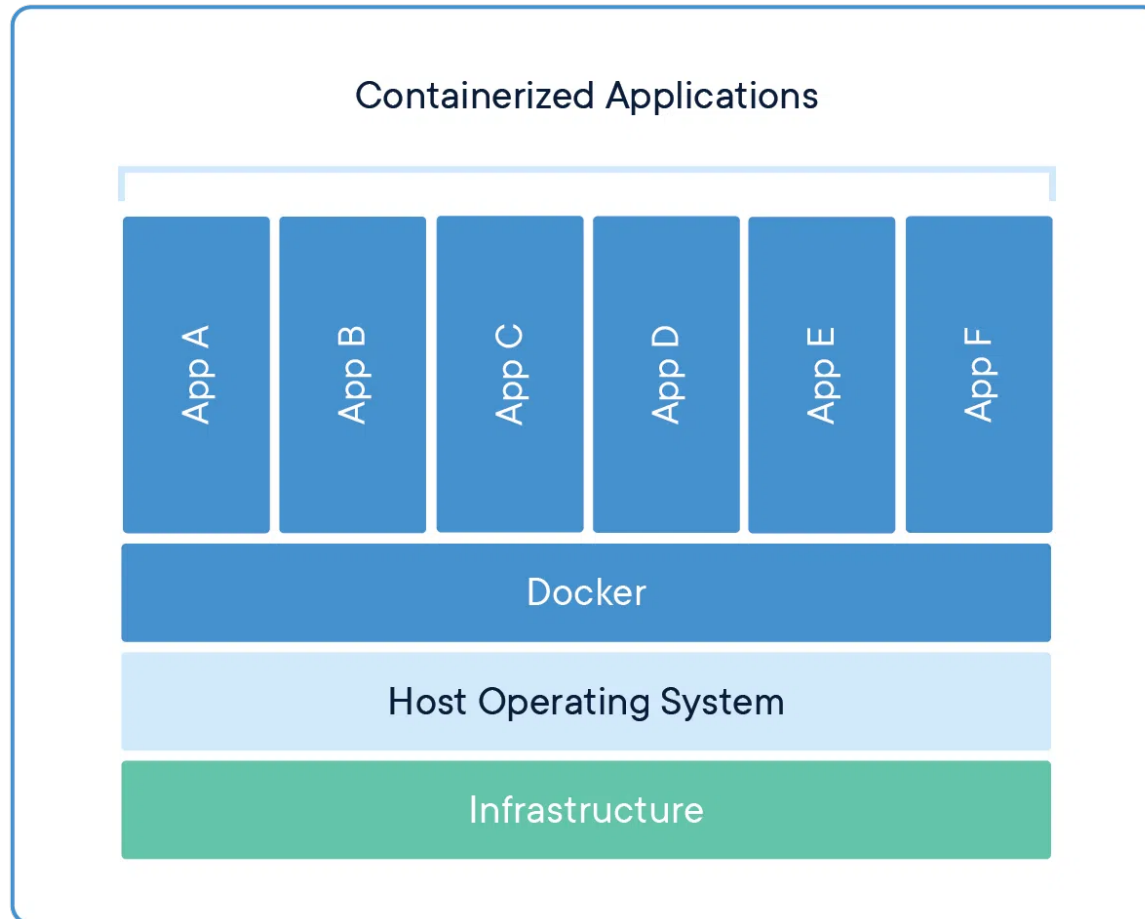
- Package software into standardized units for development, shipment, and deployment



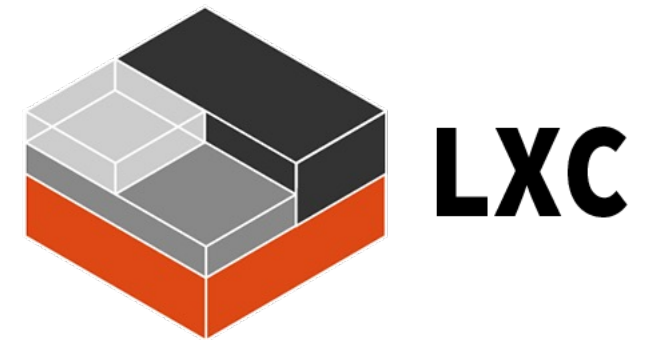
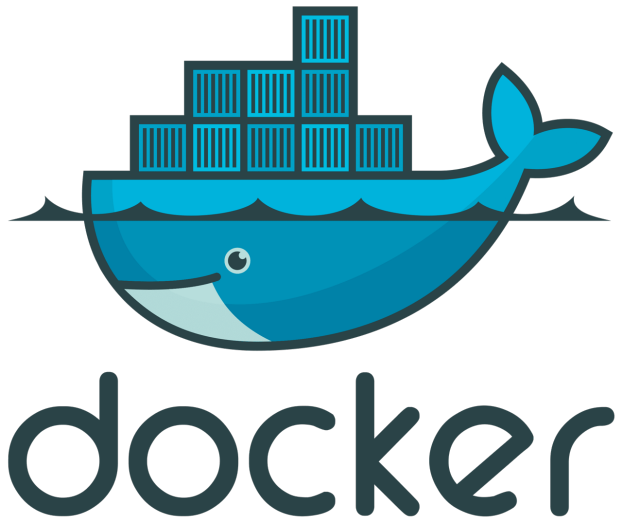
Container vs Virtual Machine (1)

- Container and VM has similar resource isolation and allocation benefits, but function differently
 - Containers virtualize the operating system instead of hardware.
 - Containers are more portable and efficient.
- Multiple Containers run on same machine and shared OS kernel
 - Container take less space than VM

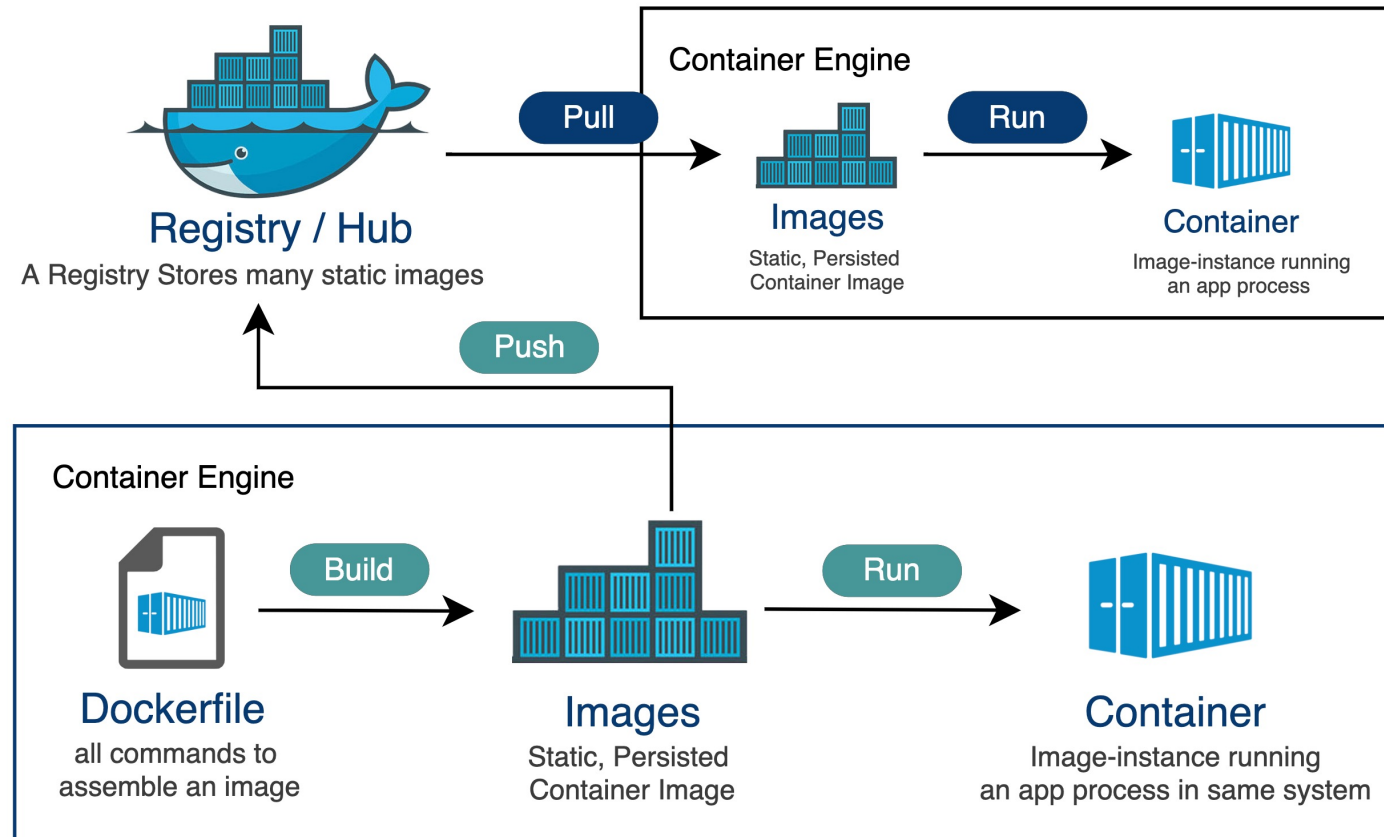
Container vs Virtual Machine (2)



Container Providers



Docker Build & Deployment



Docker Installation

- `sudo apt-get update`
- `sudo apt install docker.io`
- `sudo systemctl start docker`
- `sudo systemctl enable docker`
- `sudo usermod -aG docker $USER`

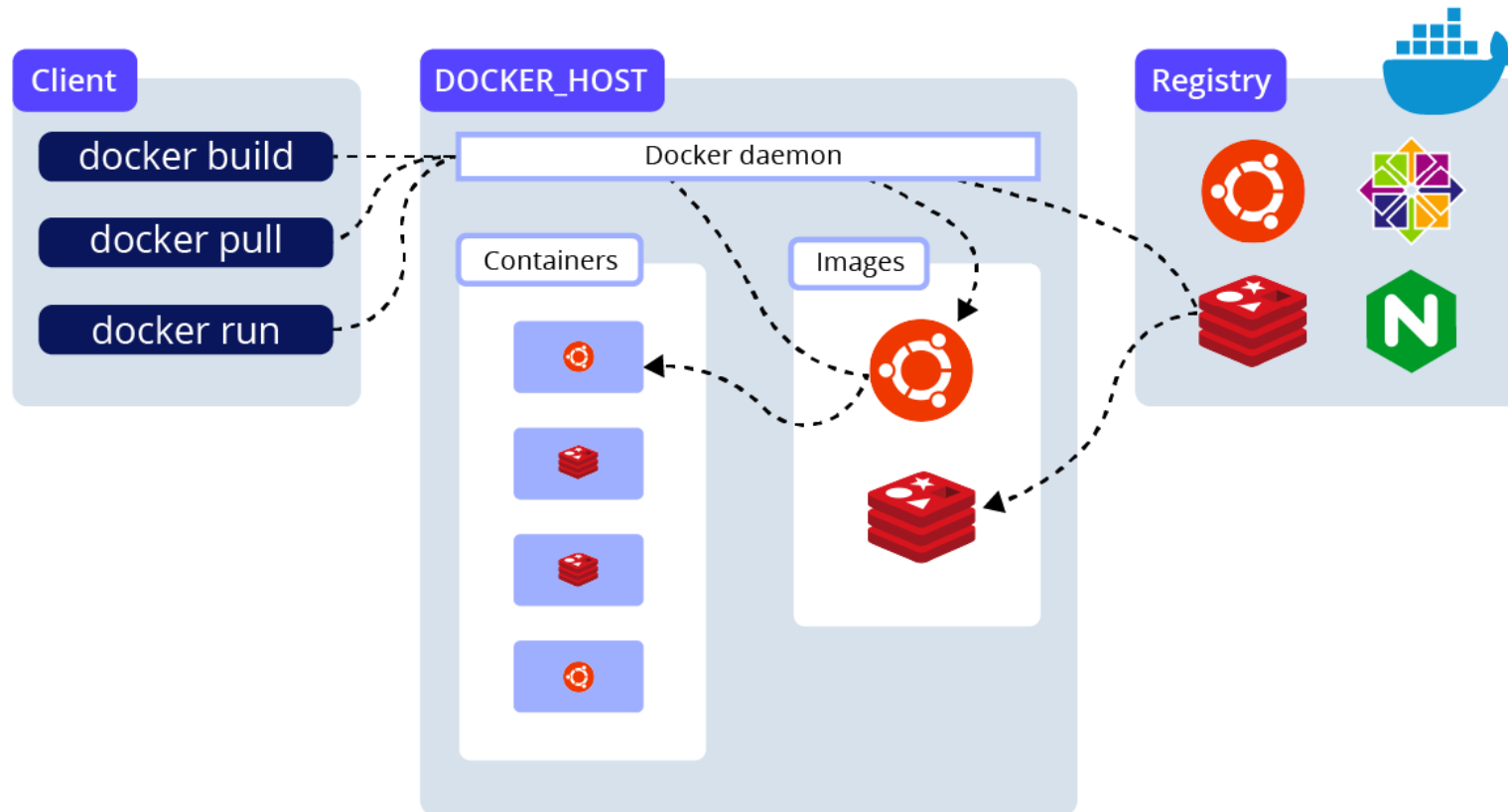
Docker Hello World

- docker --version
 - docker run hello-world
 - docker image ls
- Docker execute commands
 - docker pull
 - docker run
 - docker ps
 - docker exec
 - docker stop
 - docker kill



Docker Architecture Components

Docker client, Docker host, and Docker registry make up the Docker architecture and provide a flexible and efficient way to build, deploy, and run applications in any environment.



<https://www.educative.io/blog/docker-compose-tutorial>

Docker Share Resource

- We'll use Nginx as webserver and share **index.html** from host machine to container
 - Create an index.html file at /www/index.html
 - Run the command

```
<html>
  <body>
    <h1> Hello World from Nginx!!</h1>
  </body>
</html>
```

```
docker run -v /www:/usr/share/nginx/html:ro -p 8080:80 nginx
```

- Use web browser to **http://<server_ip>:8080**
- Try **editing** the website

Docker Share Resource (2)

- Let's run nginx in background mode

```
docker run -v /www:/usr/share/nginx/html:ro -p 8080:80 -d nginx
```

- Show running container

```
docker ps
```

- Stop running container

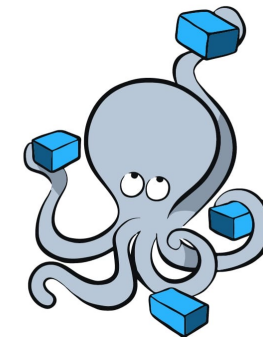
```
docker stop <container name>
```



Docker Compose

Docker Compose

- Docker Compose is a tool that was developed to help **define** and share **multi-container applications**.
- With Compose, we can create a **YAML** file to define the services and with a single command, can spin everything up or tear it all down.



docker
Compose

Install Docker Compose

- Search for our package

```
apt-cache search docker | grep compose
```

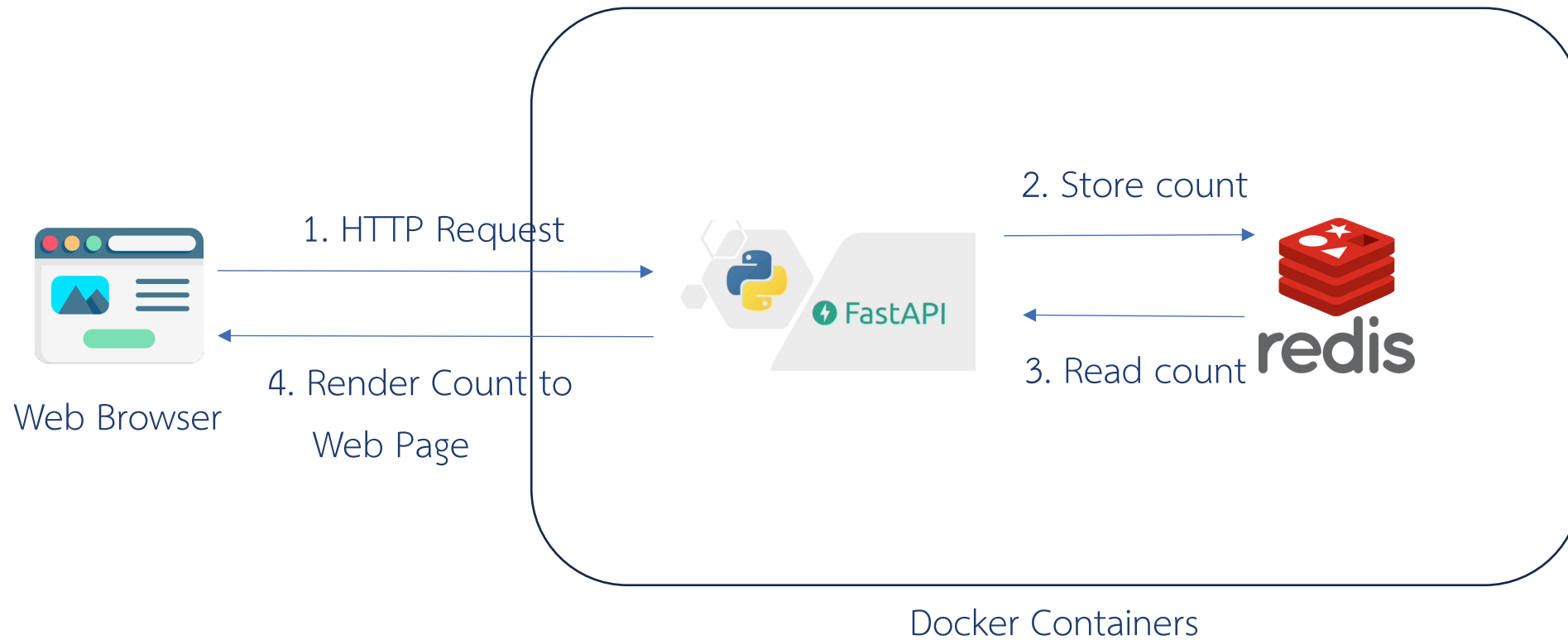
- Let's install

```
sudo apt-get install docker-compose-v2
```

- Verify installation

```
docker compose --version
```

Our compose a scenario



Prepare project files

- Create a folder **L02**
- Copy these files into the project folder
 - main.py – Your source code
 - requirement.txt – Dependency file
 - Dockerfile – docker builder script (We'll learn this later)

Create a docker-compose file

- Create a file called “**docker-compose.yml**”

```
services:
  fast-api:
    build: .
    ports:
      - "9000:8000"
  redis:
    image: "redis:alpine"
```

Run our project

- Run with command

```
docker compose up
```

- Browse to http://<server_ip>:9000
- Use ctrl-c to stop and exit the docker-compose

```
docker compose up -d
```

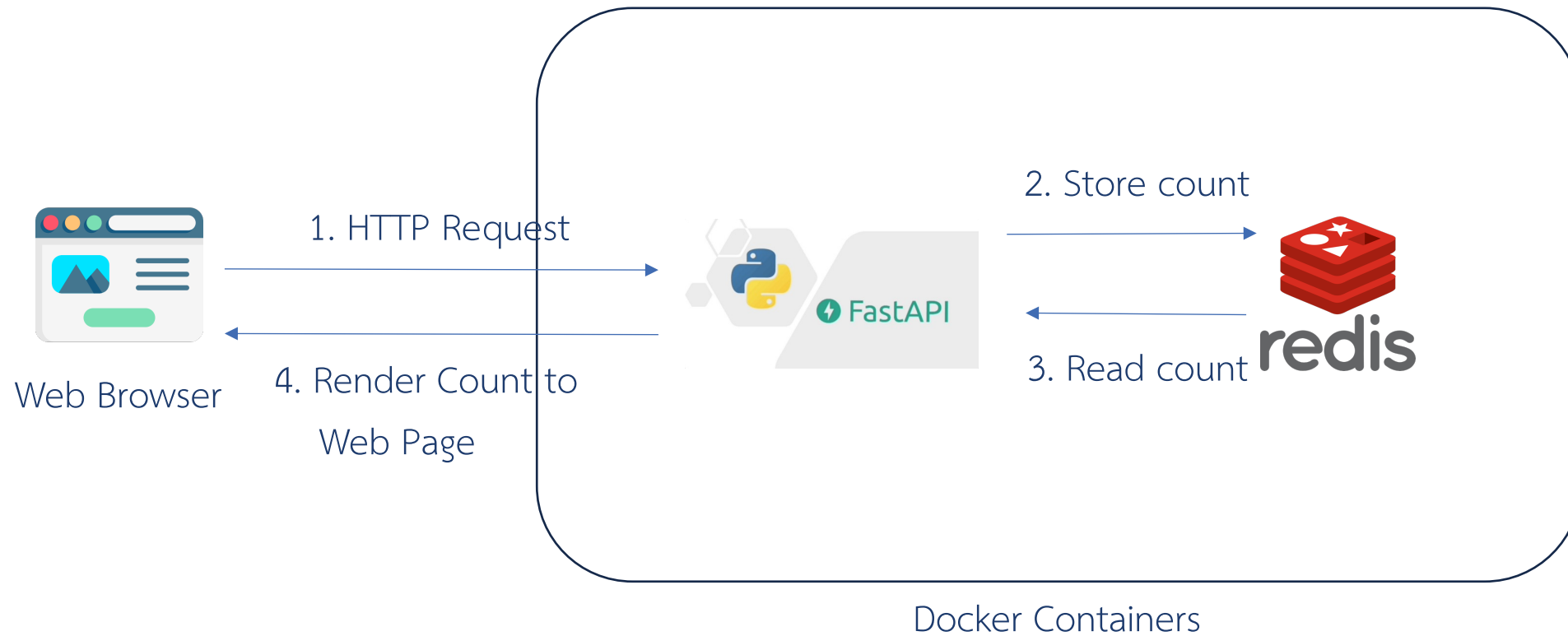
```
docker compose stop
```

```
docker compose down
```

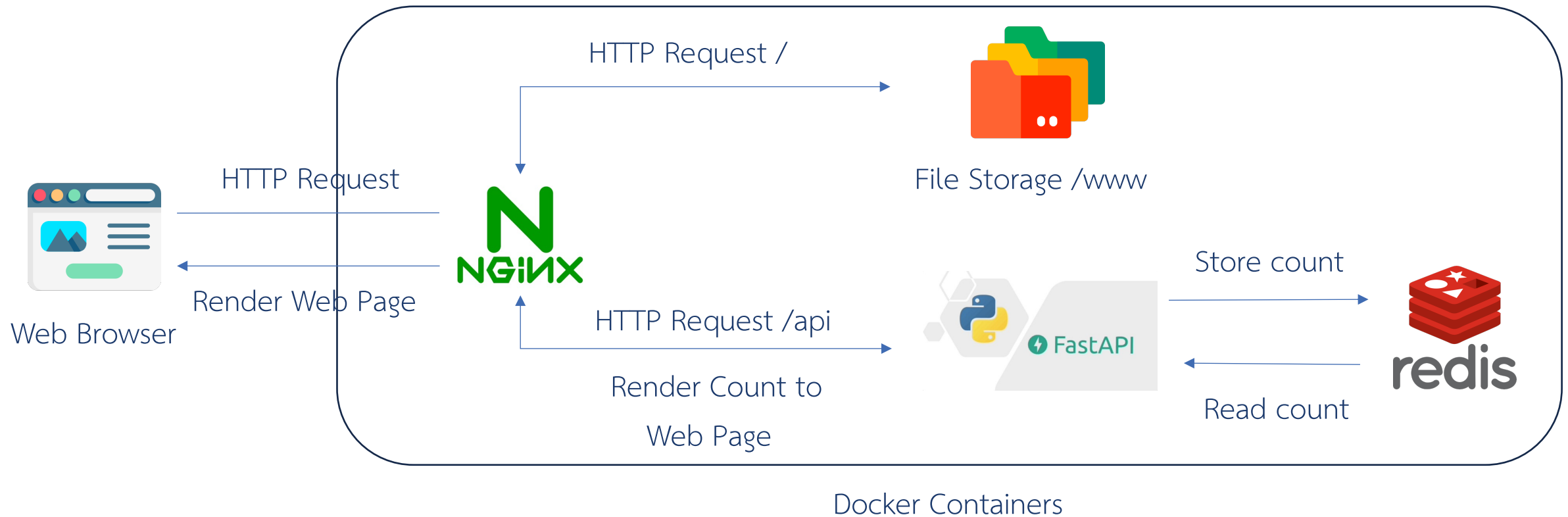


Docker Compose + Nginx

Our architecture



Our architecture + Nginx



Let's look at default Nginx config

- Start nginx docker

```
docker run -v /www:/usr/share/nginx/html:ro -p 8080:80 -d nginx
```

- Go into container

```
docker exec -it <container_id> /bin/bash
```

- i – interactive mode, t – pseudo TTY

- Print out configuration

```
cat /etc/nginx/conf.d/default.conf
```

Copy out the configuration

- Copy the configuration file from docker

```
docker cp <container_id>:/etc/nginx/conf.d/default.conf .
```

- Rename the file

```
$mv default.conf nginx_default.conf
```

Create Nginx routing



- Create a file in our project folder called “`nginx_default.conf`”

```
server {  
    listen      80;  
    listen  [::]:80;  
    server_name localhost;  
  
    location / {  
        root    /usr/share/nginx/html;  
        index  index.html index.htm;  
    }  
  
    location /api {  
        proxy_pass http://fast-api:8000/;  
        proxy_set_header Host $host;  
        proxy_set_header X-Real-IP $remote_addr;  
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
        proxy_set_header X-Forwarded-Proto $scheme;  
    }  
}
```

Edit docker-compose.yml (1)

```
services:
  fast-api:
    build: .
    ports:
      - "9000:8000"
  redis:
    image: "redis:alpine"
```

Edit docker-compose.yml (2)

```
services:
  nginx:
    image: "nginx:stable-bullseye"
    ports:
      - "8000:80"
    volumes:
      - /www:/usr/share/nginx/html
  fast-api:
    build: .
    expose:
      - 8000
  redis:
    image: "redis:alpine"
```

Try to access - http://<server_ip>/ and http://<server_ip>/api

Edit docker-compose.yml (3)



```
services:
  nginx:
    image: "nginx:stable-bullseye"
    ports:
      - "8000:80"
    volumes:
      - /www:/usr/share/nginx/html
      - ./nginx_default.conf:/etc/nginx/conf.d/default.conf
  fast-api:
    build: .
    expose:
      - 8000
  redis:
    image: "redis:alpine"
```

Access our container

- View Nginx Log

```
docker compose logs -f --tail 3 nginx
```

- Access Nginx shell console

```
docker compose exec nginx /bin/bash
```

- Stop and remove container

```
docker compose stop <container_name>
```

Environment Variables (1)

- docker-compose.yml can define custom variable for more flexible deployment
- Create “.env” file in our project folder

```
# Nginx configuration  
WWW_PORT=8090  
WWW_PATH=/www
```


Environment Variables (2)

- Edit `docker-compose.yml`

```
nginx:
  image: "nginx:stable-bullseye"
  ports:
    - "${WWW_PORT}:80"
  volumes:
    - ${WWW_PATH}:/usr/share/nginx/html
    - ./nginx_default.conf:/etc/nginx/conf.d/default.conf
```



Docker is stateless

- Try copy file with “docker cp” command and restart the docker container

