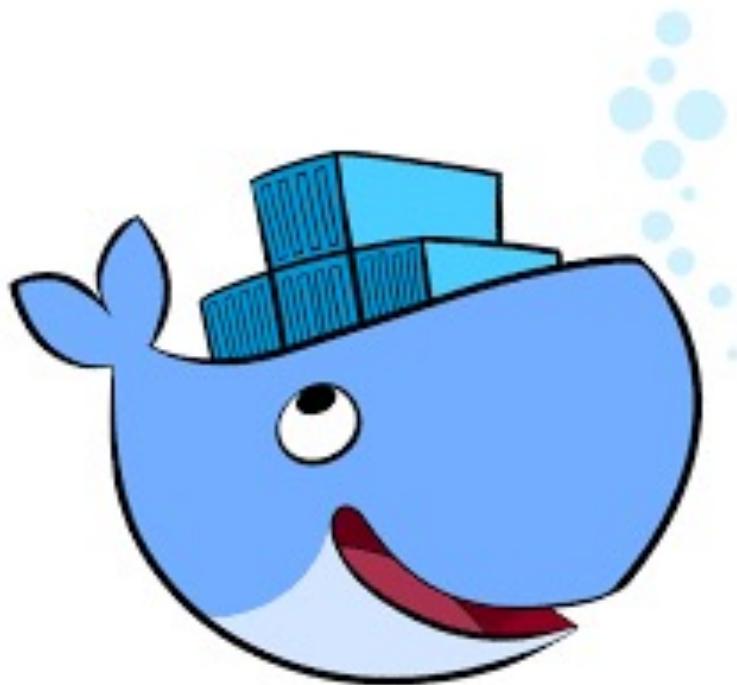


# Docker Containers

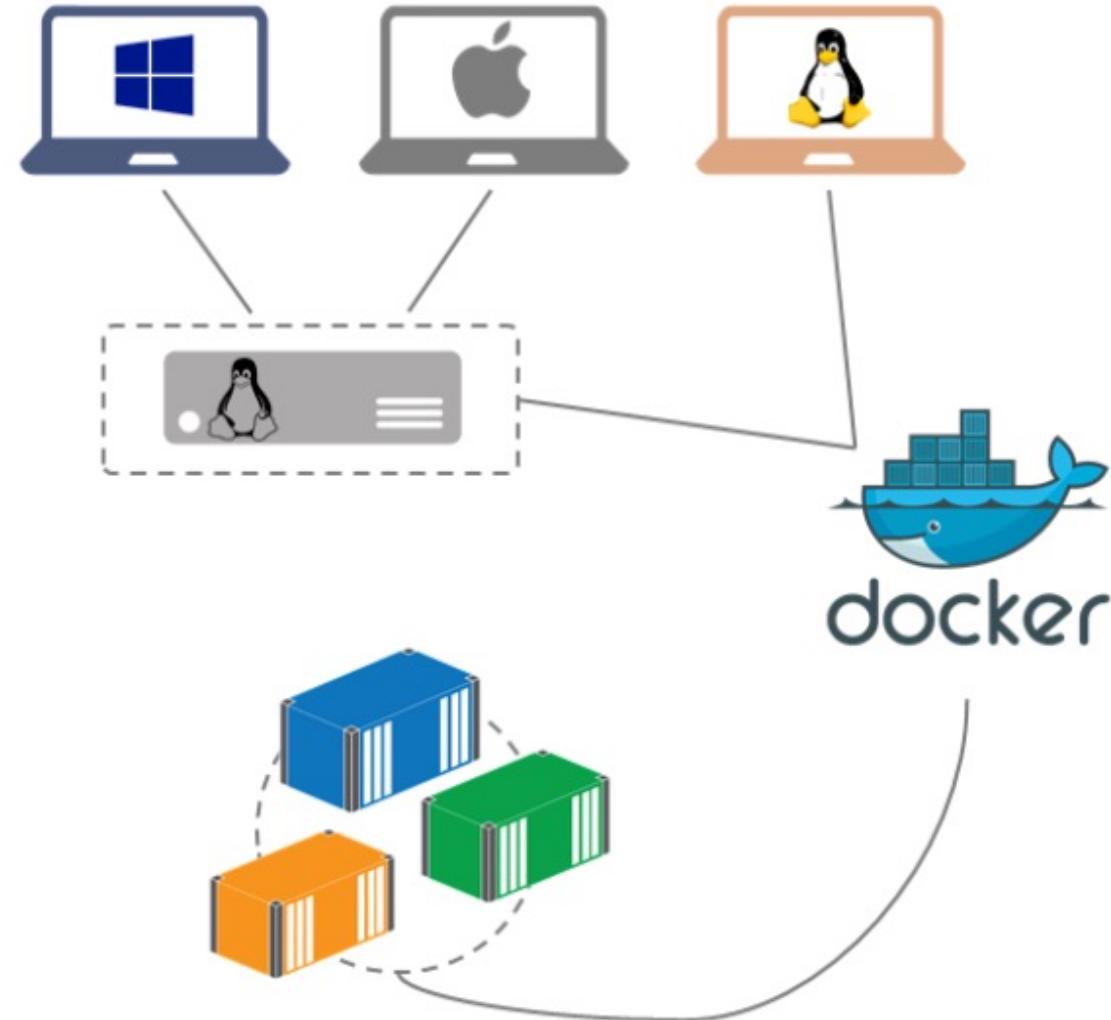
# What is Docker ?

---



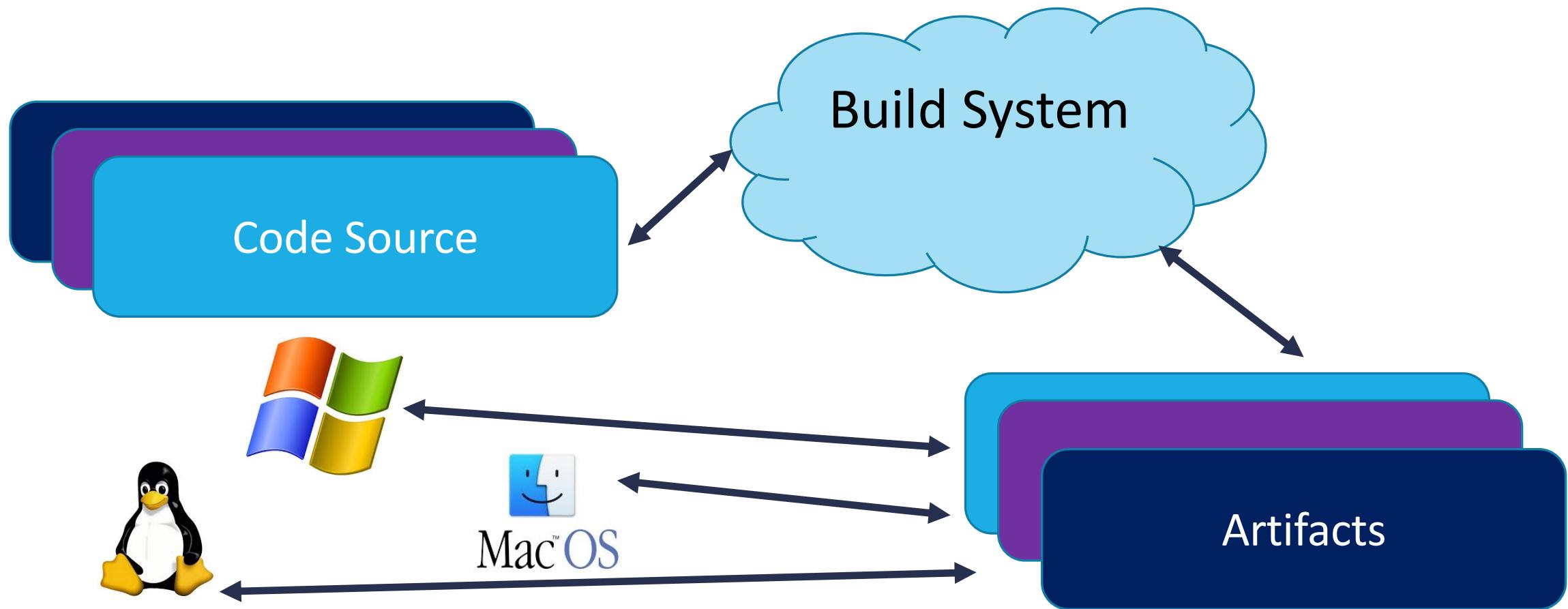
- Lightweight, open, secure platform
- Simplify building, shipping, running apps
- Shipping container system for code
- Runs natively on Linux or Windows Server
- Runs on Windows or Mac Development machines (with a virtual machine)
- Relies on "images" and "containers"

# What is Docker



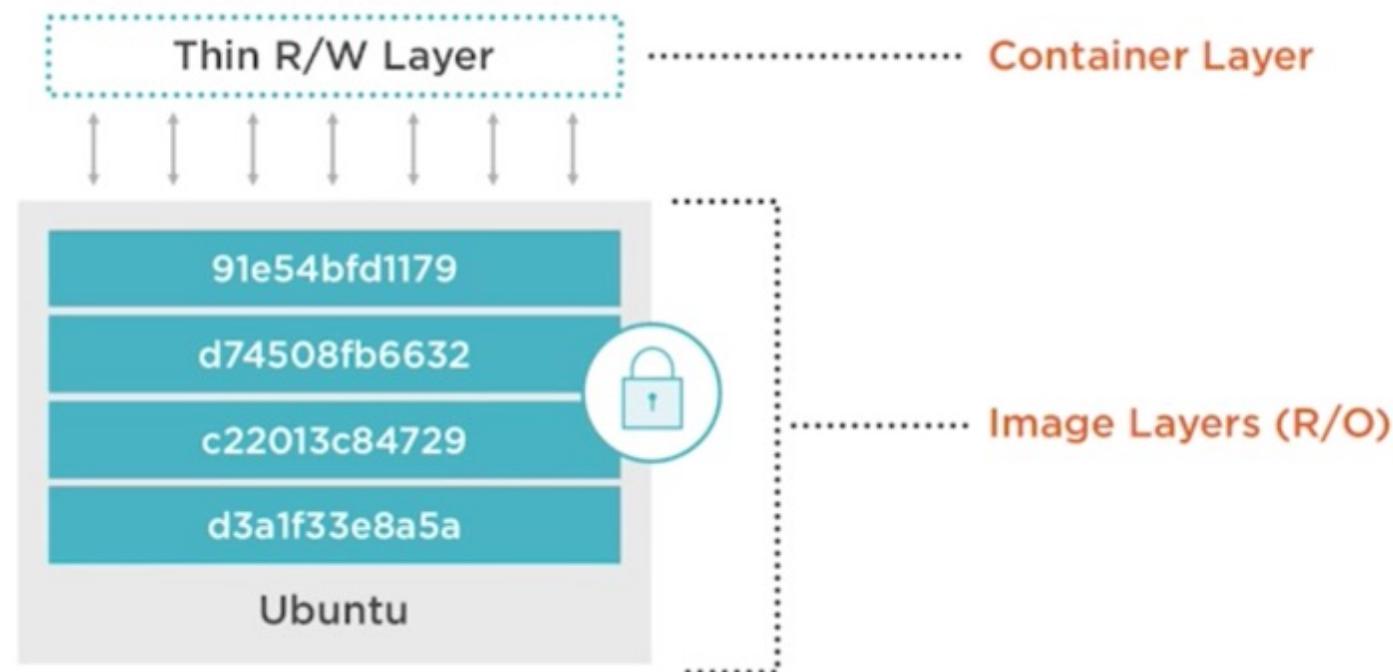
# Build Processes without Docker

---



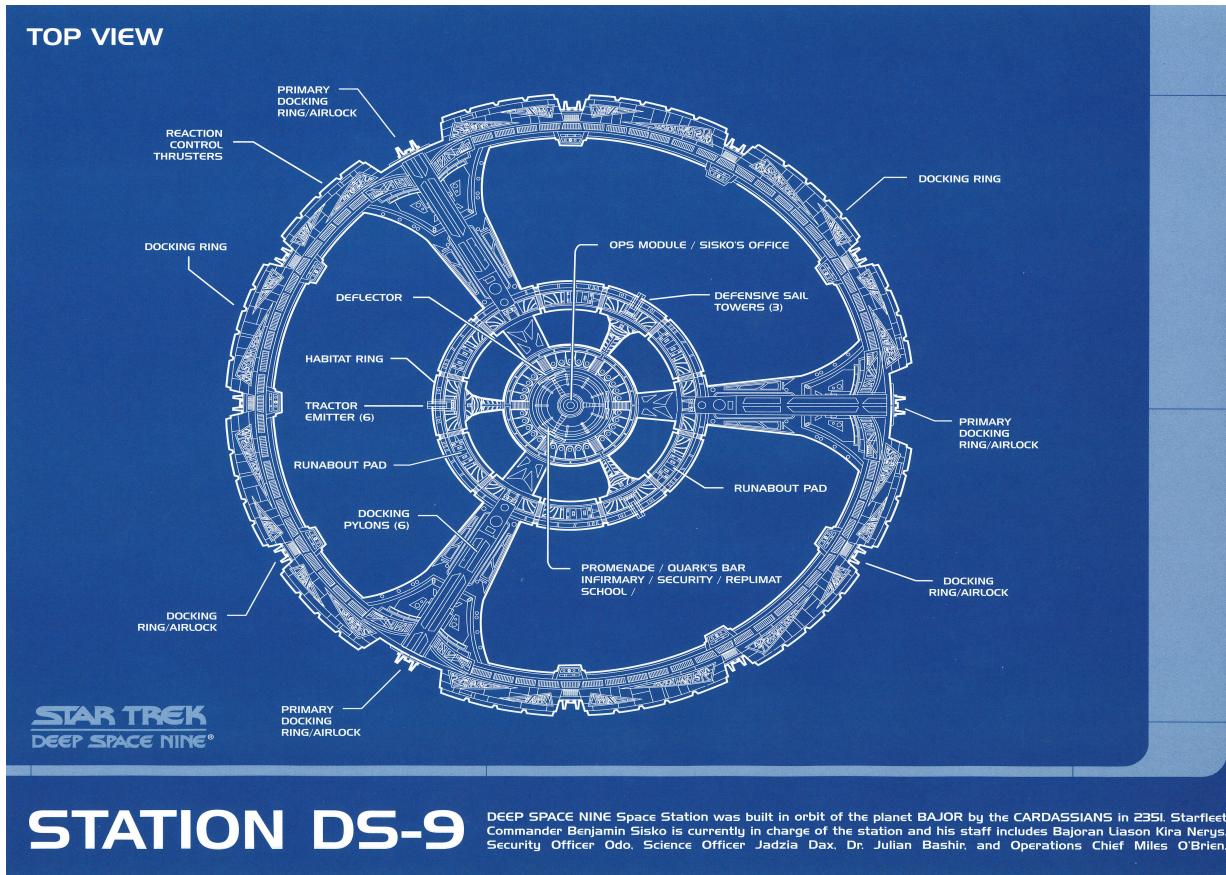
# Docker Images and File Layers

---

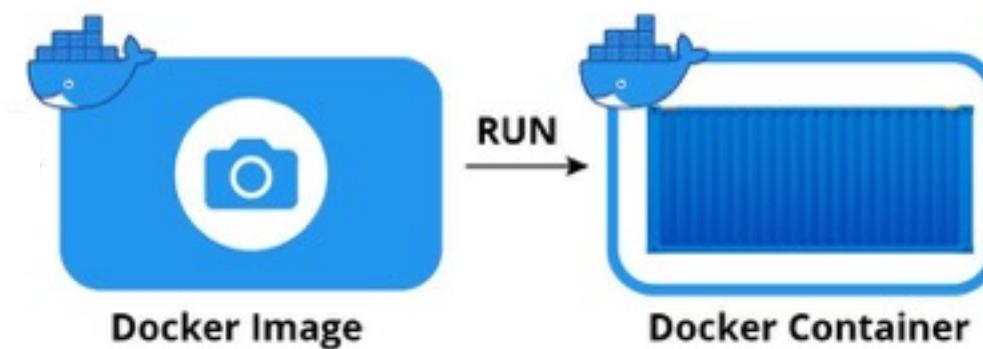


# Docker Images blue-print

---



# What is Docker Image and Container



# Docker Demo

## nginx



# NGINX

```
$ docker container run nginx:alpine
Unable to find image 'nginx:alpine' locally
alpine: Pulling from library/nginx
019300c8a437: Pull complete
2425a41f485c: Pull complete
26e59859b15d: Pull complete
a69539b662c9: Pull complete
Digest: sha256:6cf0606c8010ed70f6a6614f8c6dfedb5e2d207b5dd4b0fab846bbc26f263e
Status: Downloaded newer image for nginx:alpine
```

# Dockerfile

---

```
FROM mono:latest  
  
MAINTAINER Riccardo Terrell  
  
ENV CODEDIR /var/code  
RUN mkdir -p $CODEDIR  
copy . $CODEDIR  
WORKDIR $CODEDIR  
  
RUN chmod +x $CODEDIR  
RUN xbuild ./ActorRemote.sln /property:Configuration=Release  
  
EXPOSE 9234  
  
ENTRYPOINT ["mono", "./AkkaExamples/bin/Release/AkkaExamples.exe"]
```



Dockerfile



Docker Image

# DockerFile

```
# Use an official Python runtime as a parent image
FROM python:2.7-slim

# Set the working directory to /app
WORKDIR /app

# Copy the current directory contents into the container at /app
COPY . /app

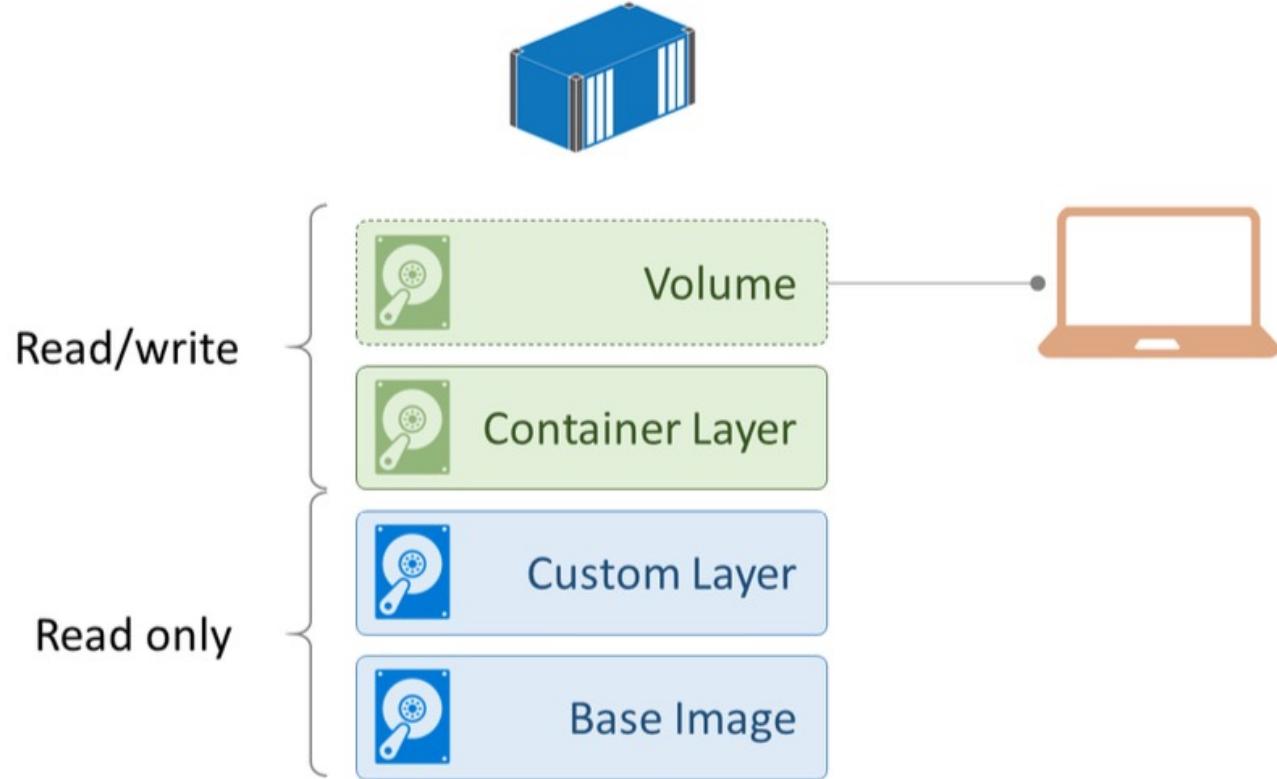
# Install any needed packages specified in requirements.txt
RUN pip install --trusted-host pypi.python.org -r requirements.txt

# Make port 80 available to the world outside this container
EXPOSE 80

# Define environment variable
ENV NAME World

# Run app.py when the container launches
CMD ["python", "app.py"]
```

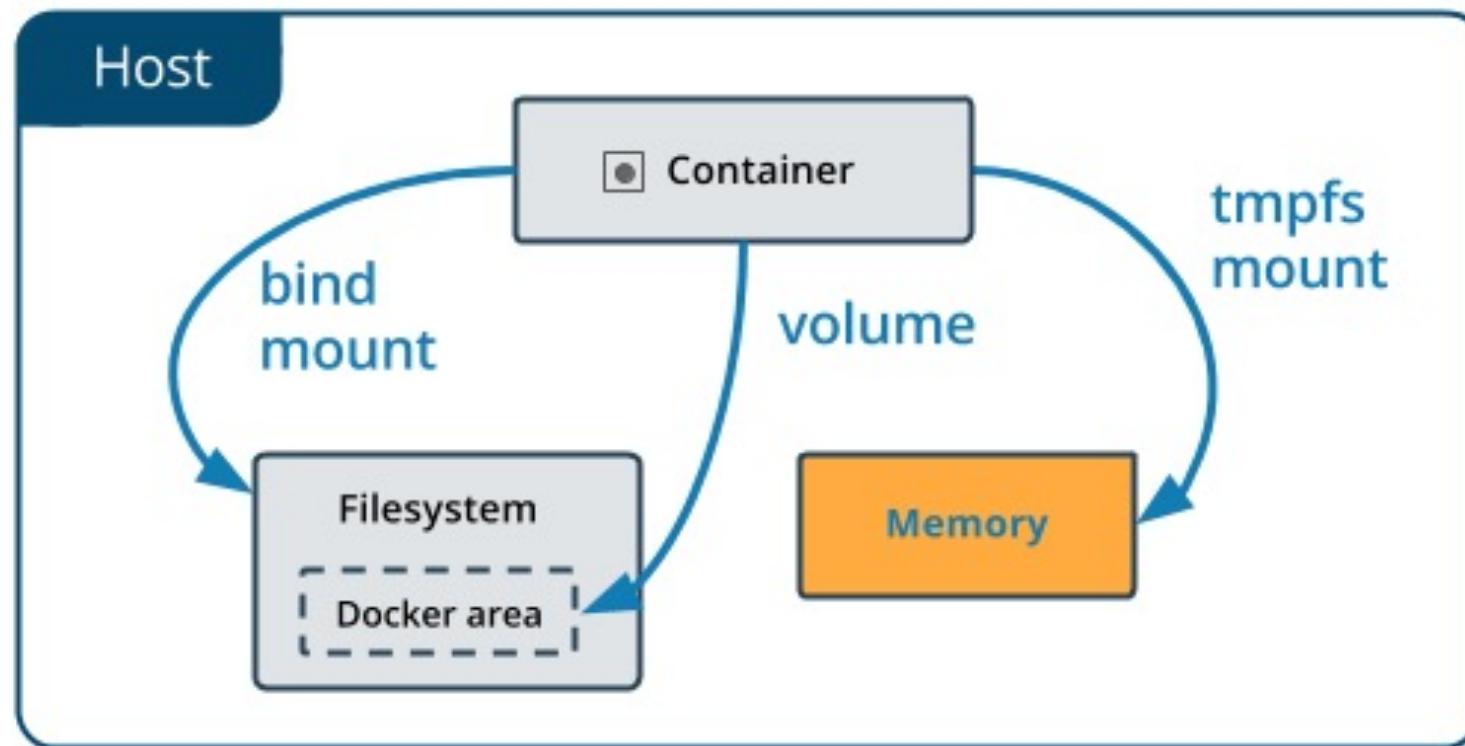
# Docker Storage



```
docker container run -it --rm -v ~/v1:/v1 ubuntu-with-volume
```

# Docker Sharing Volumes

---



# Docker Orchestrating

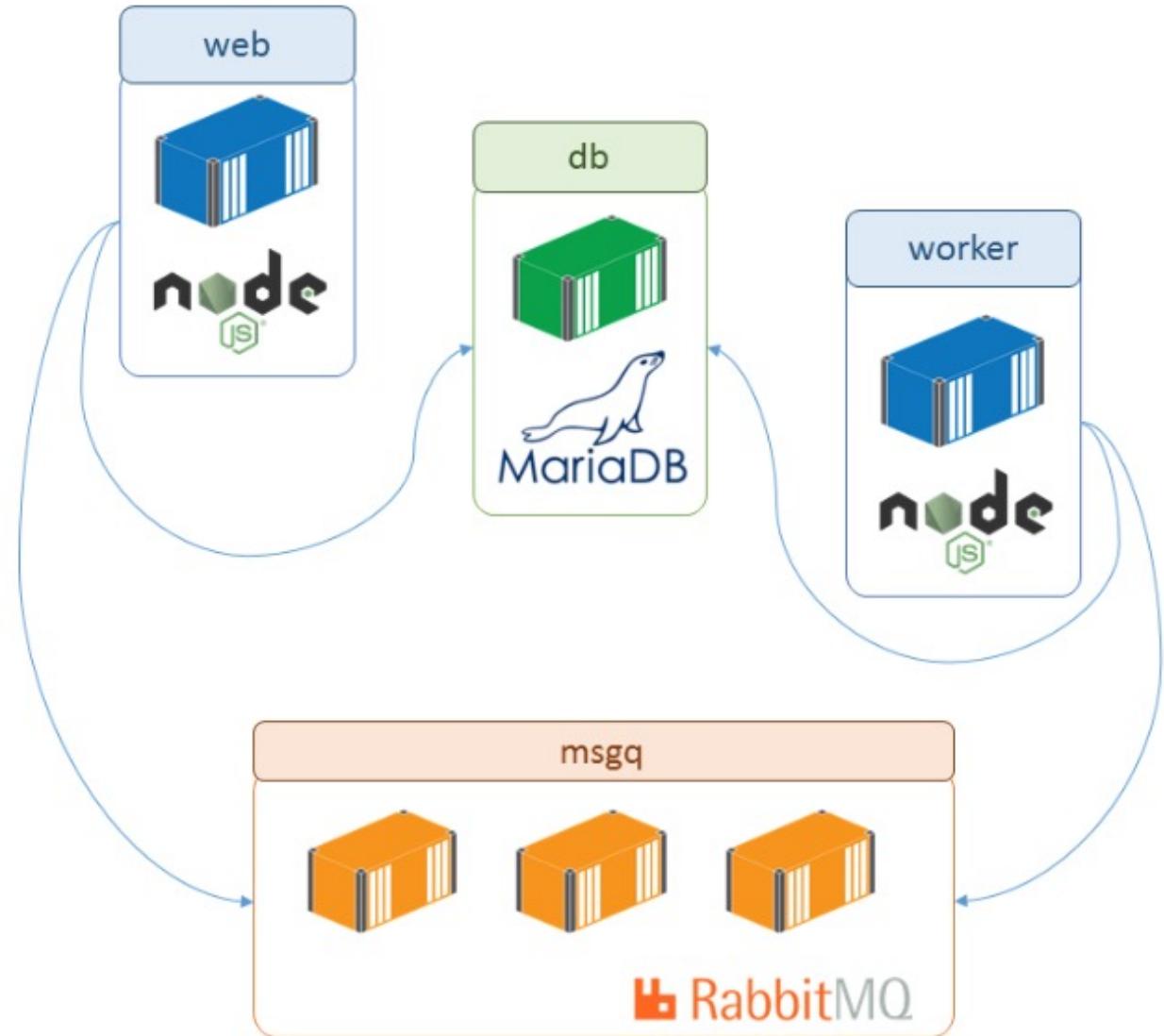
Networks



# Docker-Compose

```
version: "3.7"
services:
  webapp:
    build:
      context: ./dir
      dockerfile: Dockerfile-alternate
    args:
      buildno: 1
    volumes:
      - type: volume
        source: mydata
        target: /data
    volume:
      nocopy: true
```

# Docker-Compose



# Docker Commands

Few useful commands

List Docker CLI commands

```
docker  
docker container --help
```

Display Docker version and info

```
docker --version  
docker version  
docker info
```

Execute Docker image

```
docker run hello-world
```

List Docker images

```
docker image
```

List Docker containers (running, all, all in quiet mode)

```
docker container  
docker container --all  
docker container --aq
```

Create image using this directory's Dockerfile

```
docker build -t friendlyhello .
```

Run "friendlyhello" mapping port 4000 to 80

```
docker run -p 4000:80 friendlyhello docker run -d -p 4000:80 friendlyhello
```

List all images on this machine

```
docker image ls -a
```

Remove specified image from this machine

```
docker image rm <image id>
```

Remove all images from this machine

```
docker image rm $(docker image ls -a -q) docker login
```

# Docker Networks

---

```
docker network create my-network
```

```
docker container run -d --name webcontainer --network my-network nginx:alpine
```

D

E

M

O

f10

delete

return