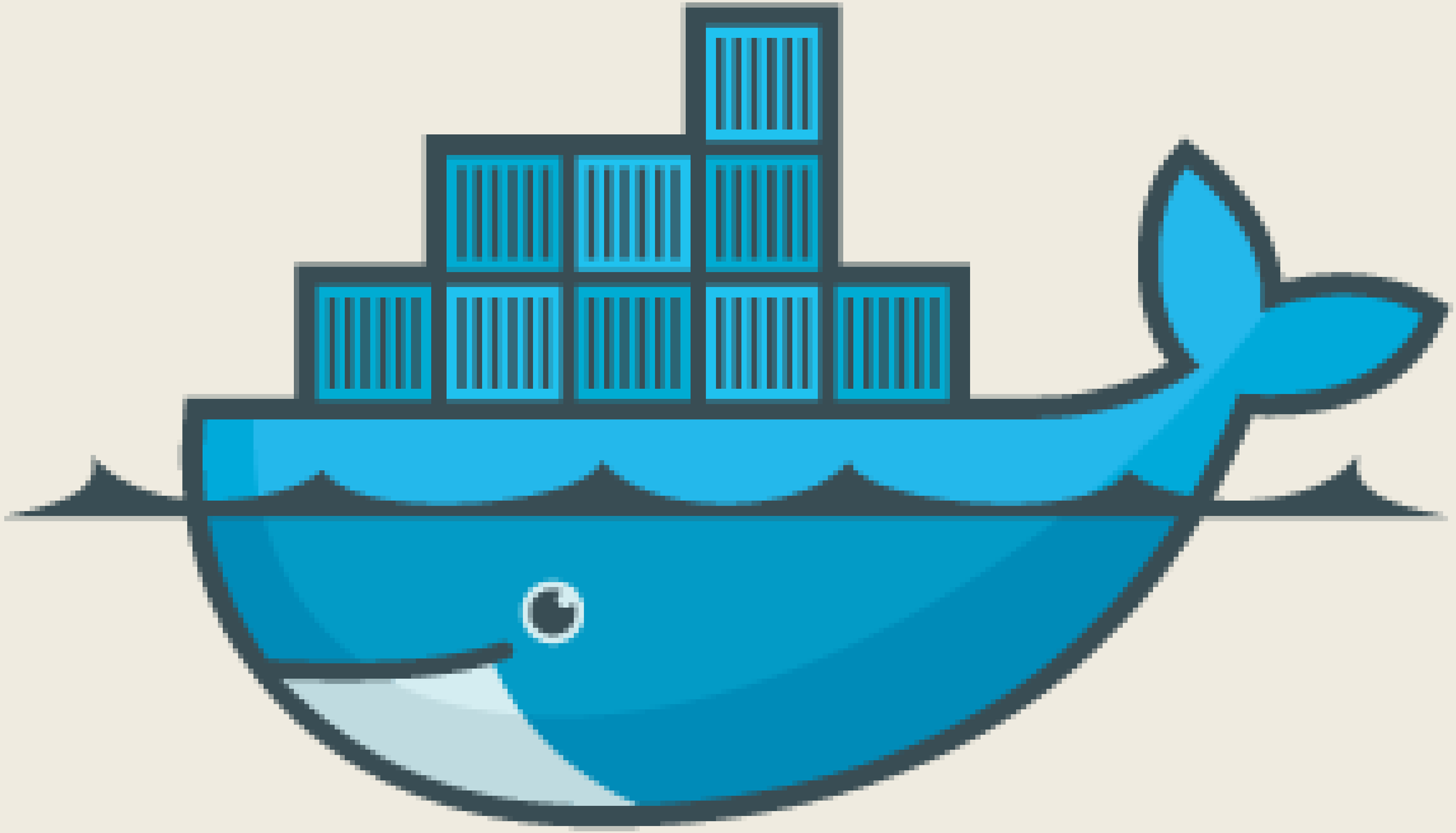
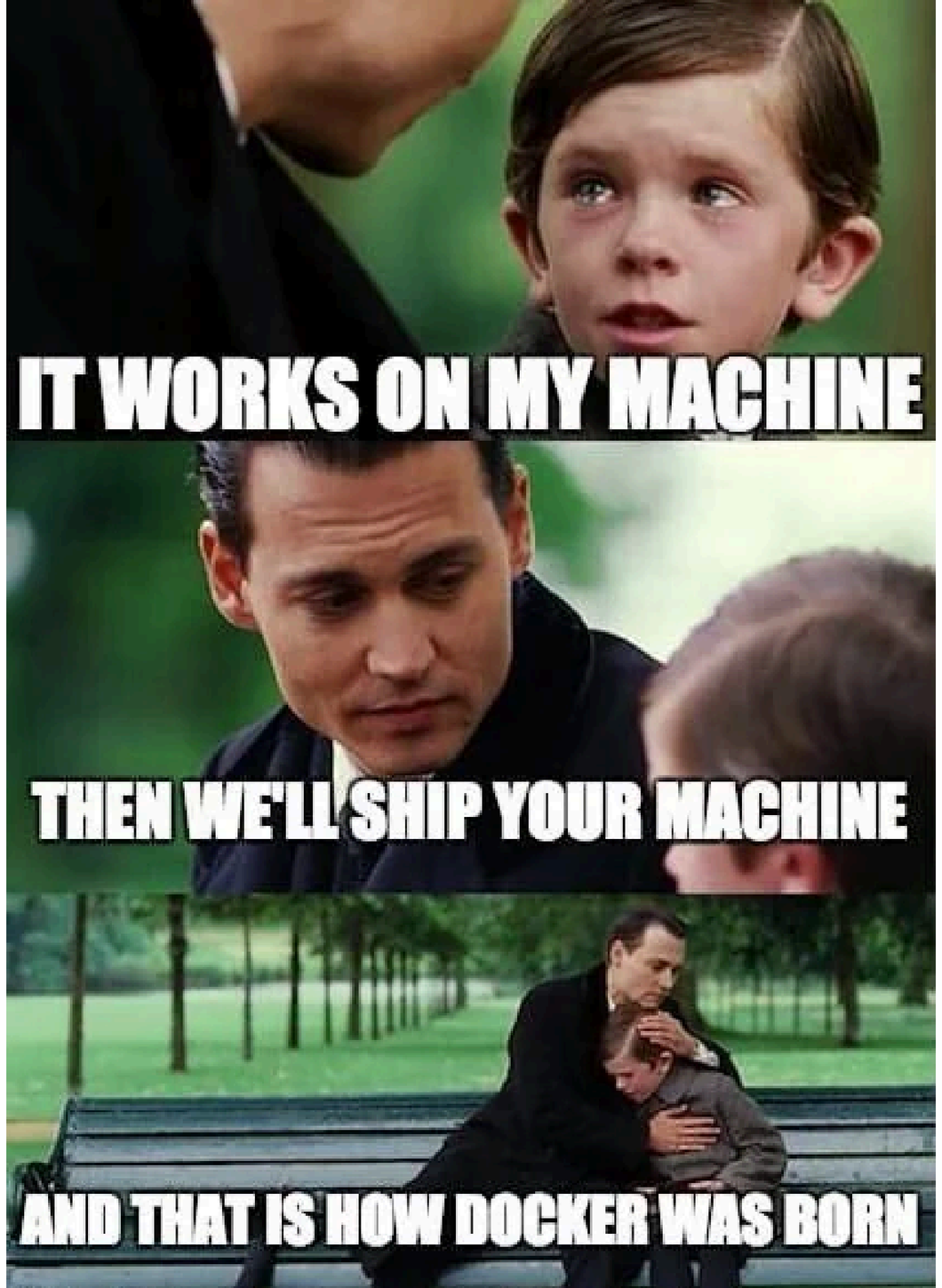
DevOps Connect

in association with

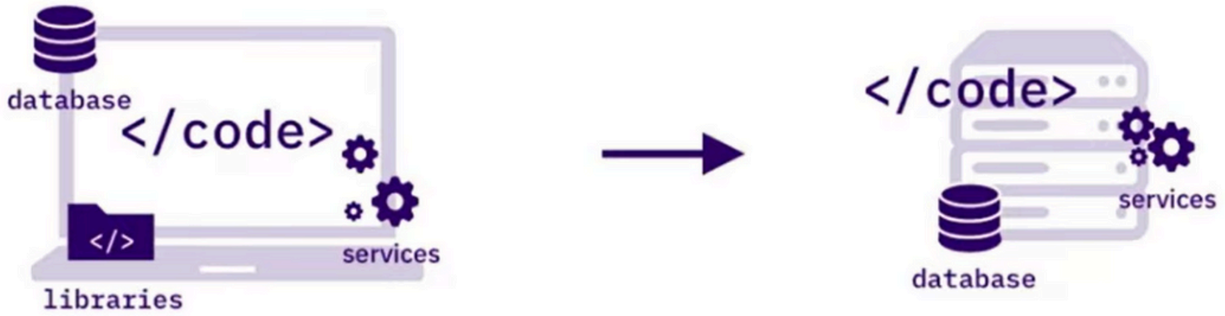
विद्वान् सर्वत्र पूज्यते।
Since 1909

PVG's COET & M
Department of Computer Engineering




IT WORKS ON MY MACHINE
THEN WE'LL SHIP YOUR MACHINE
AND THAT IS HOW DOCKER WAS BORN

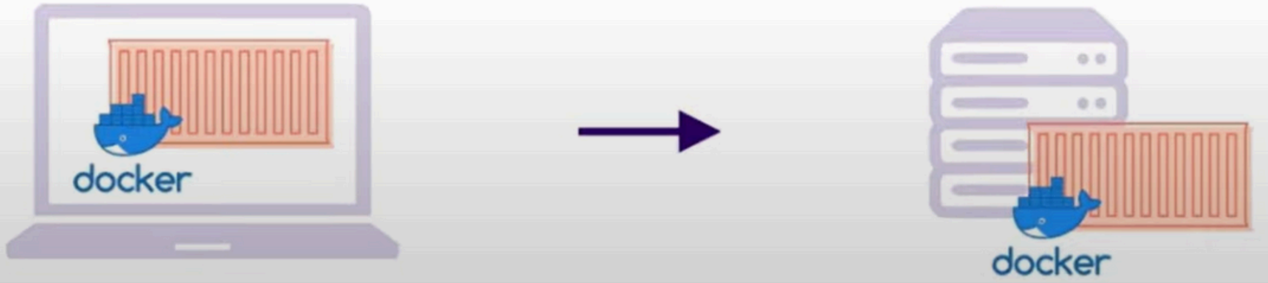
before



local status: OK

server status: ERROR
E: library XYZ missing

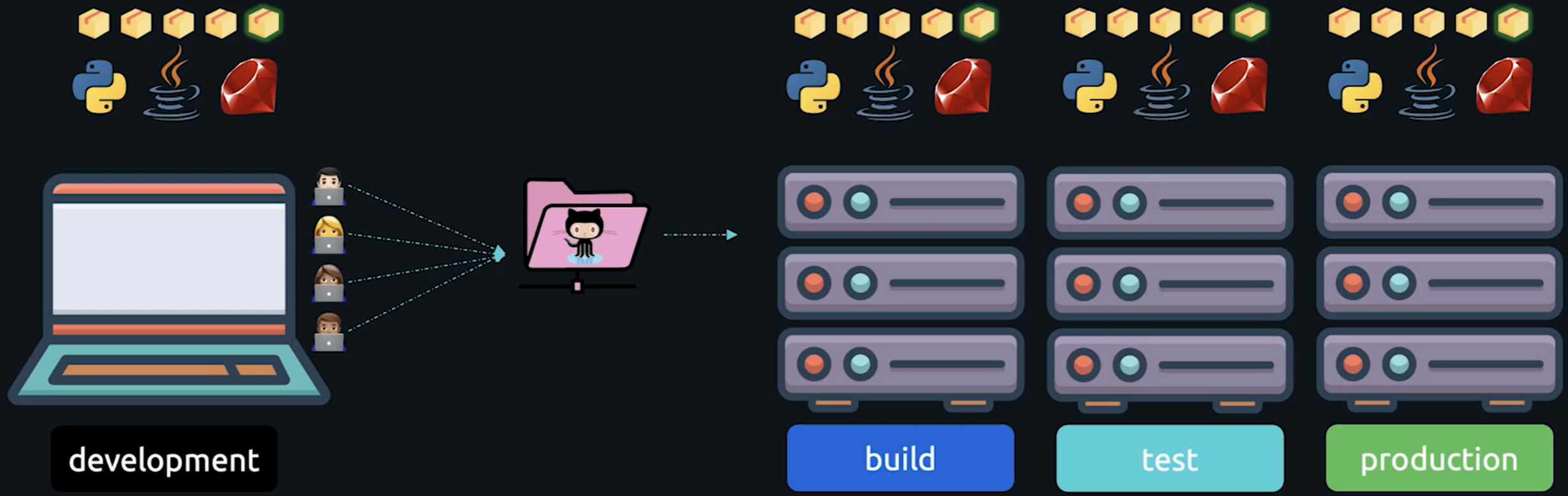
after



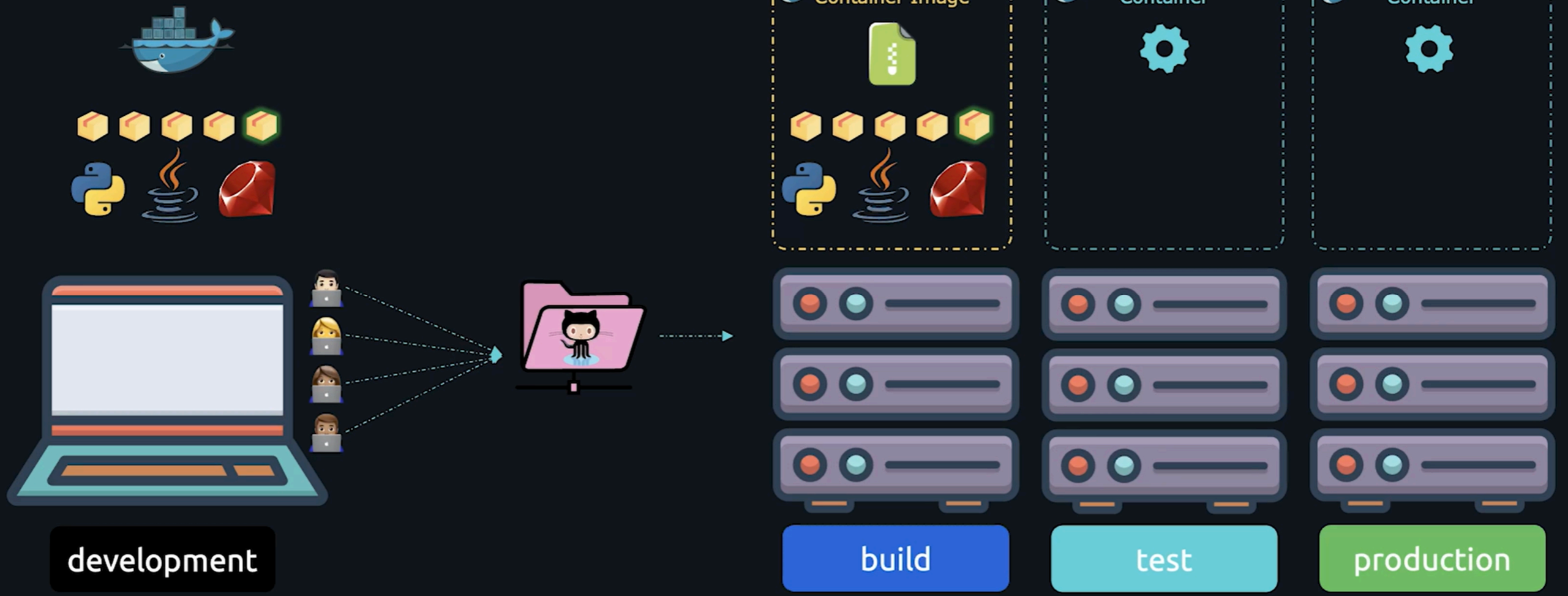
local status: OK

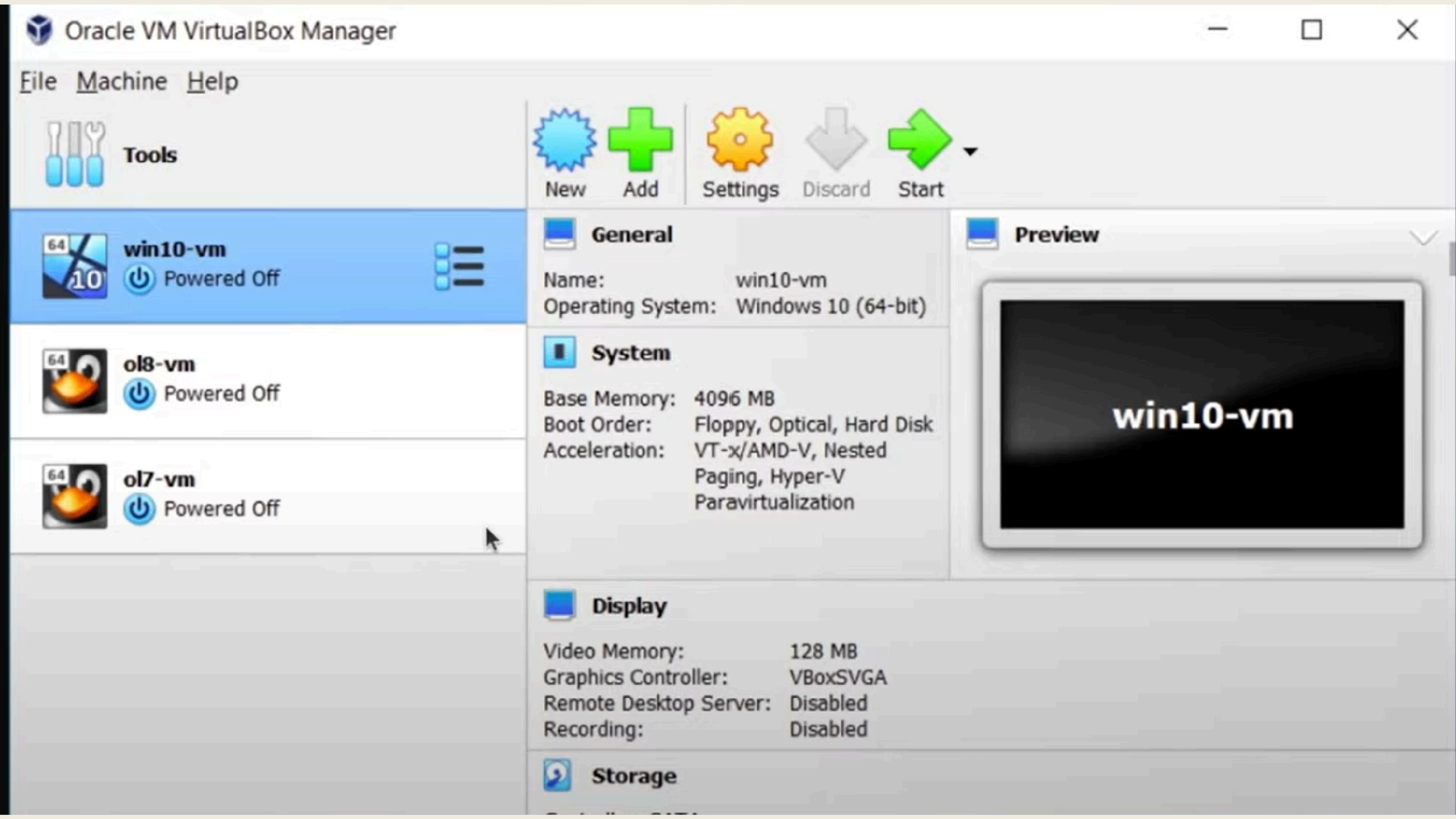
server status: OK

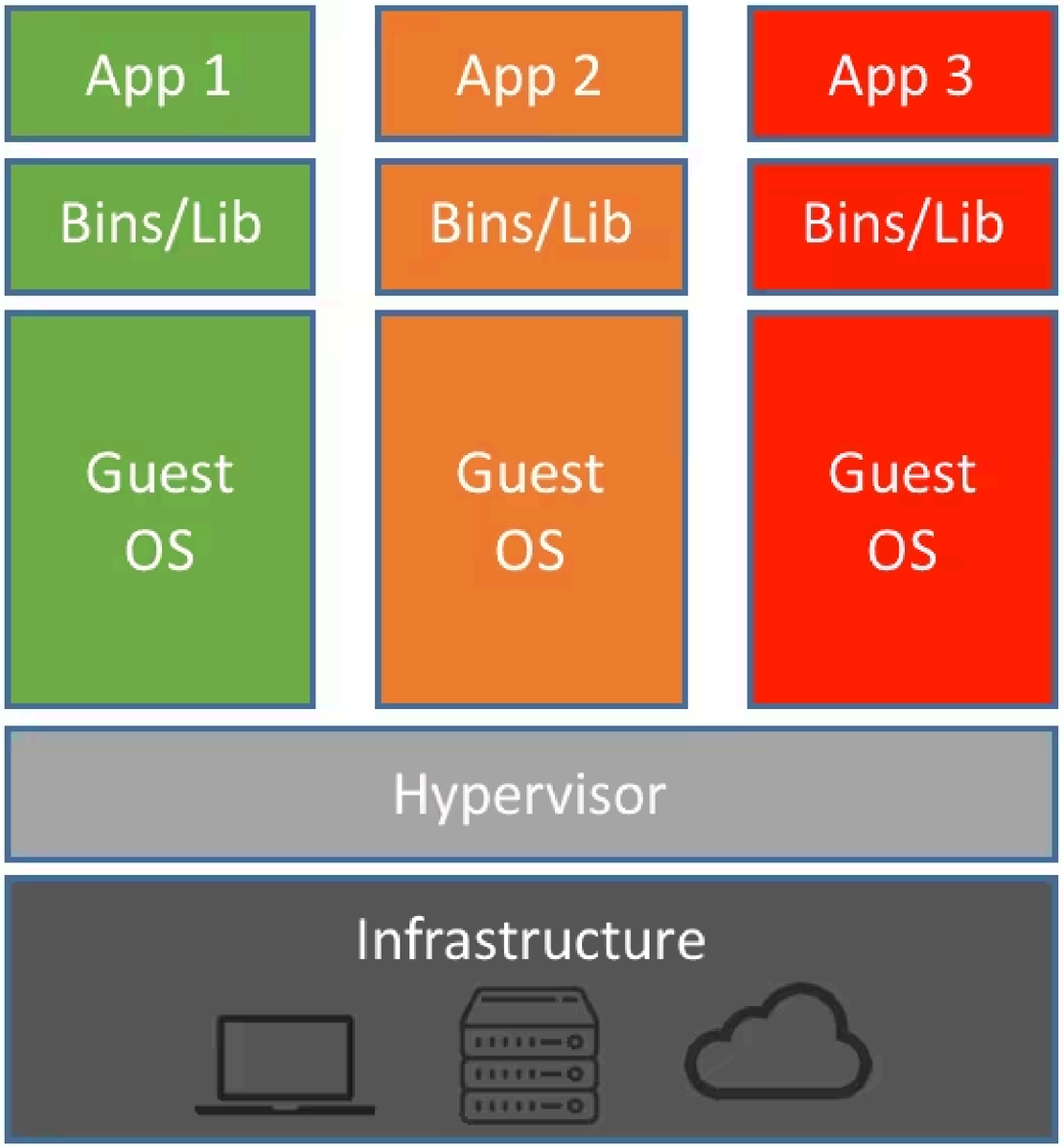
CI/CD Pipeline



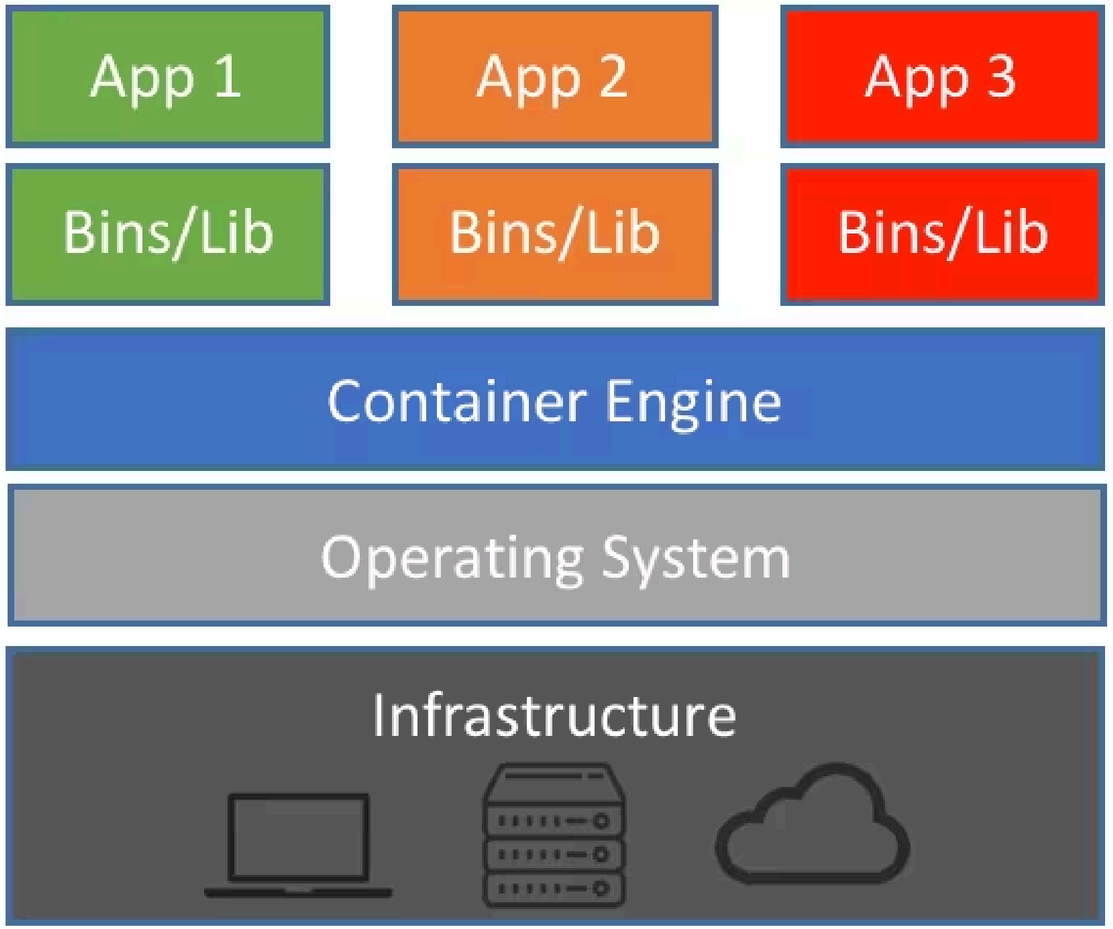
CI/CD Pipeline




Oracle VM VirtualBox Manager
File Machine Help
Tools
New
Add
Settings
Discard
Start
win10-vm
Powered Off
ol8-vm
Powered Off
ol7-vm
Powered Off
General
Name: win10-vm
Operating System: Windows 10 (64-bit)
System
Base Memory: 4096 MB
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, Hyper-V Paravirtualization
Preview
win10-vm
Display
Video Memory: 128 MB
Graphics Controller: VBoxSVGA
Remote Desktop Server: Disabled
Recording: Disabled
Storage



Machine Virtualization

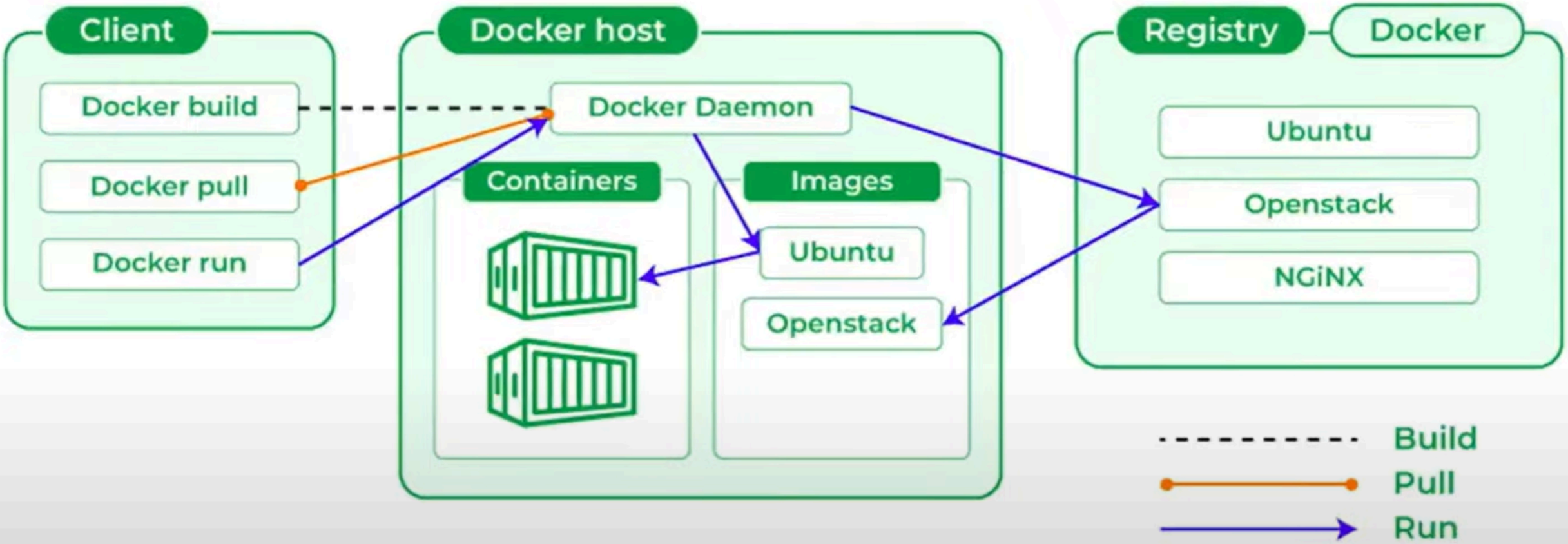


Containers

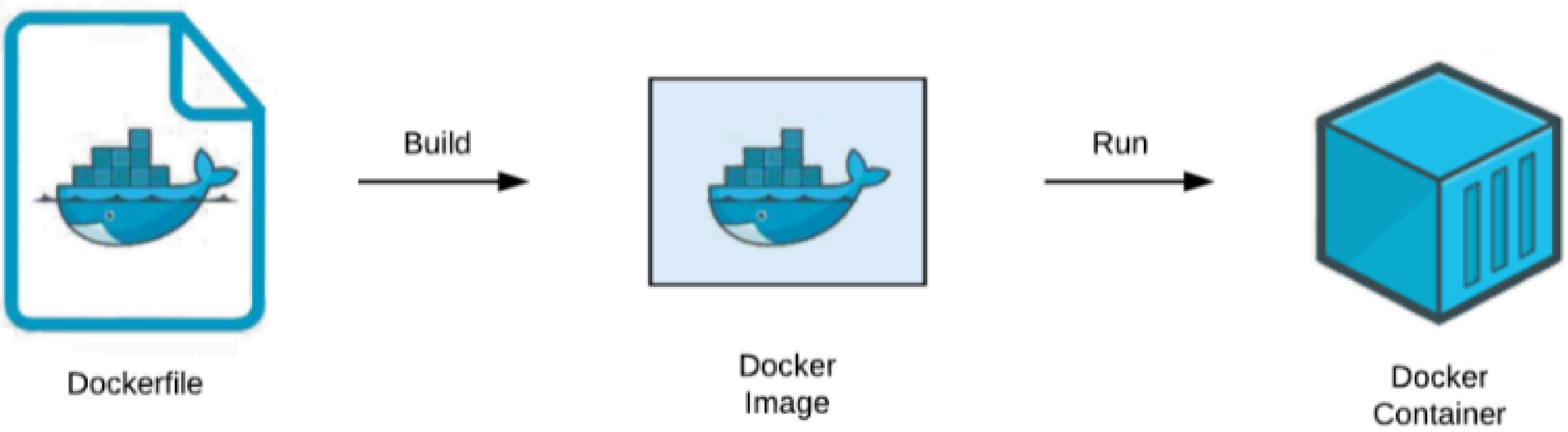
Virtual Machine	*Container*
Heavyweight	• Lightweight
Limited performance	• Native performance
Each VM runs in its own OS	• All containers share the host OS
Hardware-level virtualization	• OS virtualization
Startup time in minutes	• Startup time in milliseconds
Allocates required memory	• Requires less memory space
Fully isolated and hence more secure	• Process-level isolation, possibly less secure


HOMEZ
ONLINE

Docker Architecture



Docker uses a client-server architecture to manage and run containers:


Build
Run
Dockerfile
Docker
Image
Docker
Container

```
# Use an official Python base image
FROM python:3.12-slim

# Set the working directory inside the container
WORKDIR /app

# Copy only requirements first (better for caching)
COPY requirements.txt .

# Install dependencies
RUN pip install -r requirements.txt

# Copy the rest of the application code
COPY . .

# Expose the port your app listens on (optional but helpful)
EXPOSE 5000

# Command to run the application
CMD ["python", "app.py"]
```

BUILD A DOCKER IMAGE

docker build -t myapp:1.0 .

Explanation

docker build - Creates an image from a Dockerfile.

-t myapp:1.0 - Tags the image as myapp with version 1.0.

. - The build context (current directory containing the Dockerfile).

RUN A DOCKER CONTAINER

docker run -d -p 8080:80 --name myapp_container myapp:1.0

Explanation

-d - Detached mode (runs in background).

-p 8080:80 - Maps host port 8080 to container port 80.

--name myapp_container - Assigns a custom name.

myapp:1.0 - Image name and tag to run.

LIST DOCKER IMAGES

```
docker images
```

Explanation

Shows all images with repository, tag, ID, size.

LIST CONTAINERS

```
docker ps                    #Running containers only:

docker ps -a                 #All containers (including stopped):
```

Displays container ID, name, image used, status, ports, and more.

```
# Build image from Dockerfile in current directory
docker build -t mywebapp:latest .

# Run container in detached mode, expose port 5000 on host
docker run -d -p 5000:5000 --name web_container mywebapp:latest

# Verify
docker images       # List images

docker ps           # List running containers

docker ps -a        # List all containers
```

THANKS :)