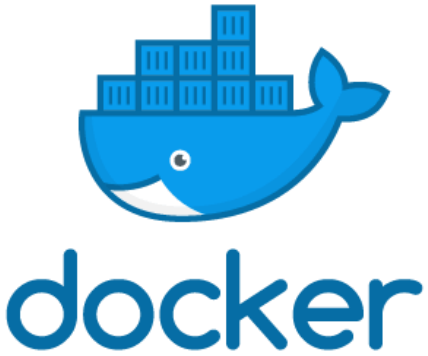
DOCKER NOVEMBER MEETUP

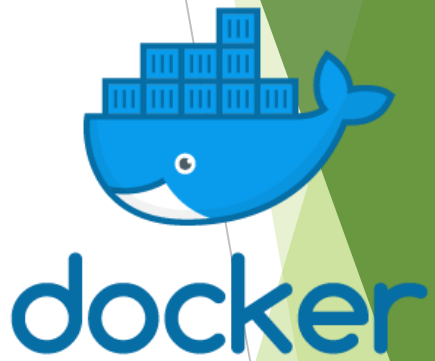
Docker Basics & Dockerizing your Micro services

By
Dinesh S



What You Can Expect

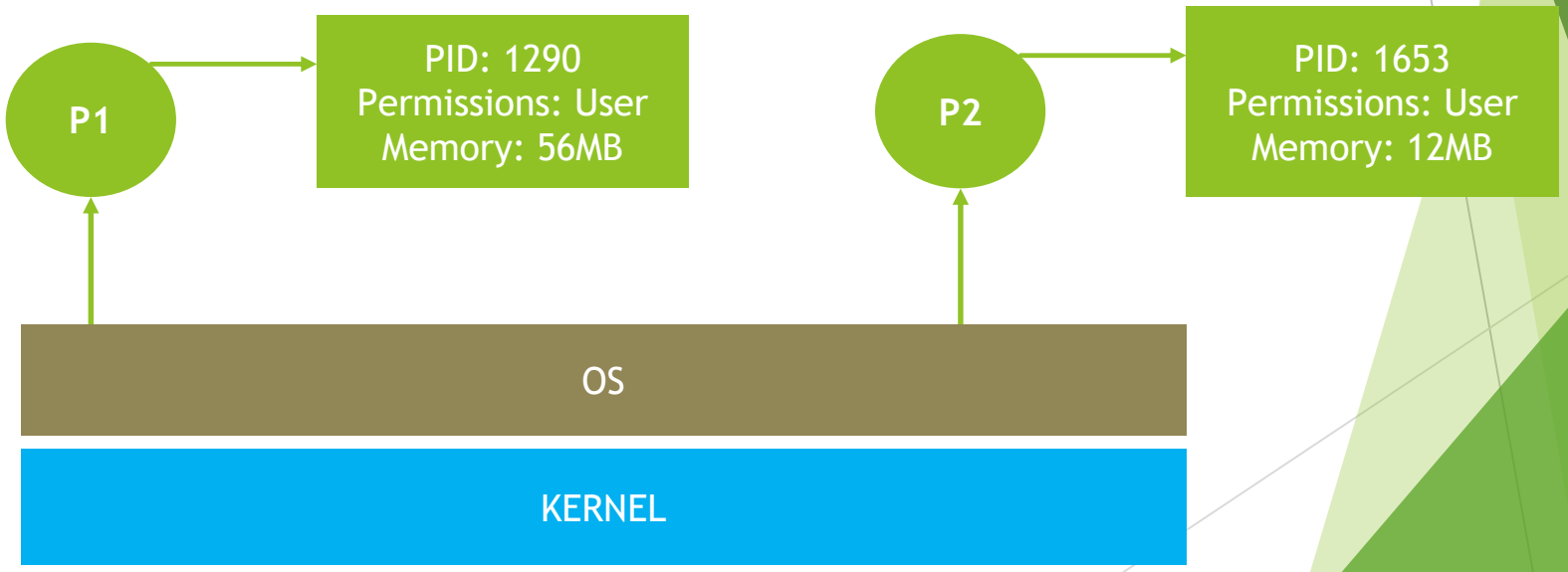
- How containers gets isolated
- What & How - Docker
- Brush up with Basic commands
- **Break**
- Microservices + Docker – a match made in Heaven
- Hands – on: Dockerize apps and test
- Networking
- Get Goodies and post phots



What a process can be

“A process is a running program that depends on the Host machine’s Memory, user & permissions and File system”

- A Process gets minimal isolation support from the Operating system.
- You cannot run two process with the same port in a host at the same time.
- A process gets the same privileges as the user who created the process.

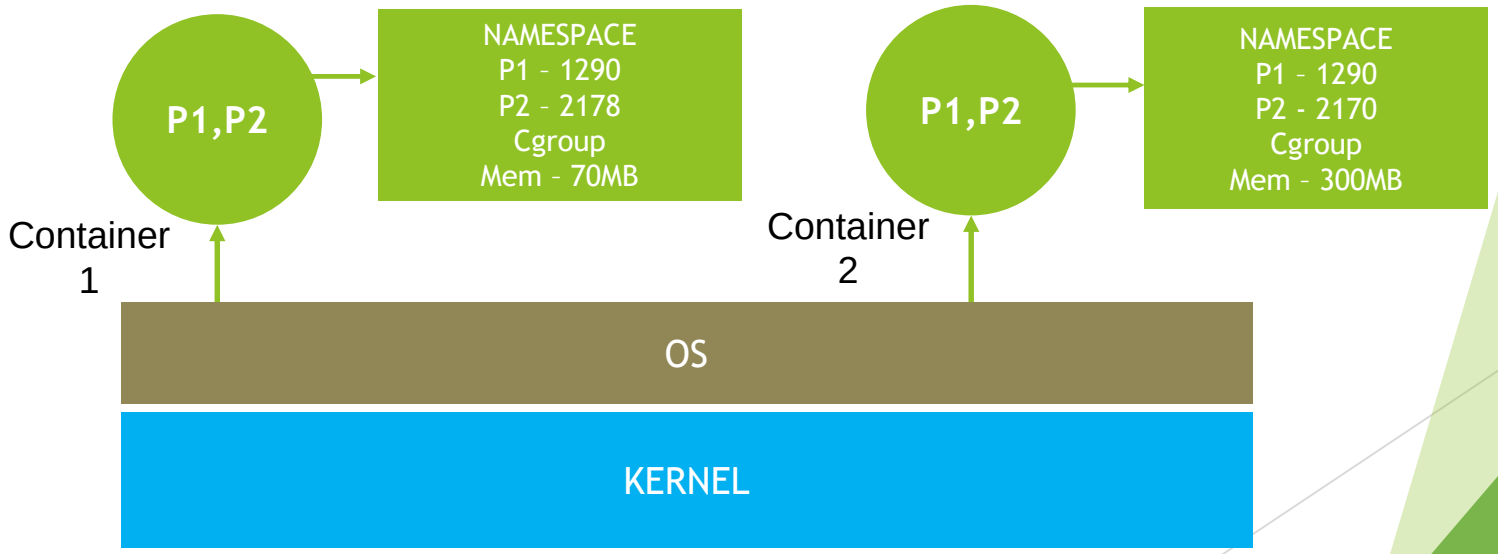


How containers gets isolated

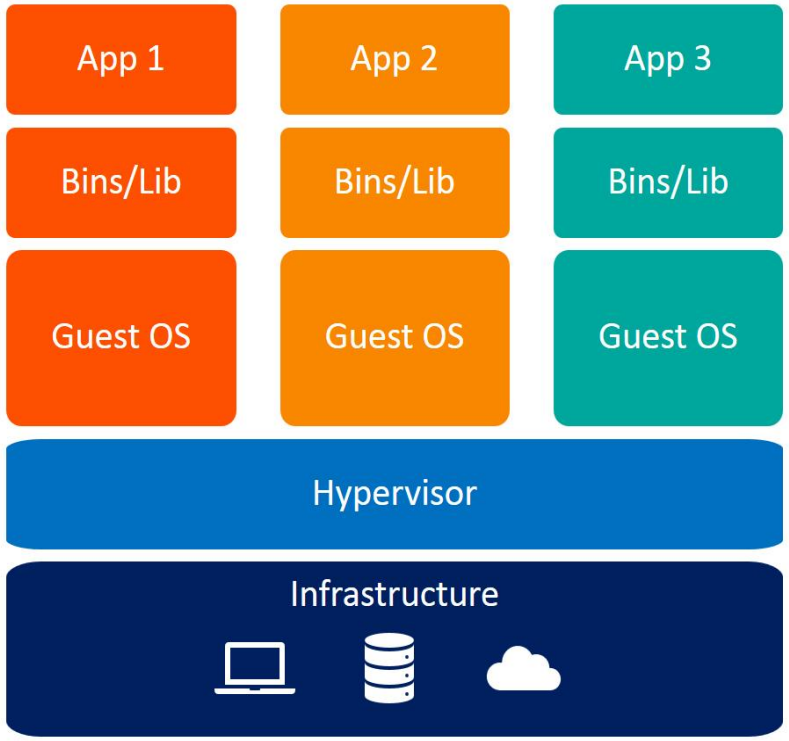
“ Containers are group of processes with some cool kernel features which makes the processes to pretend that they’re running on their own separate machine. “

- **Namespaces:** Isolates global resources for process – E.g.: A PID namespace isolates PID feature which can allocate same PID to different process in the same host.

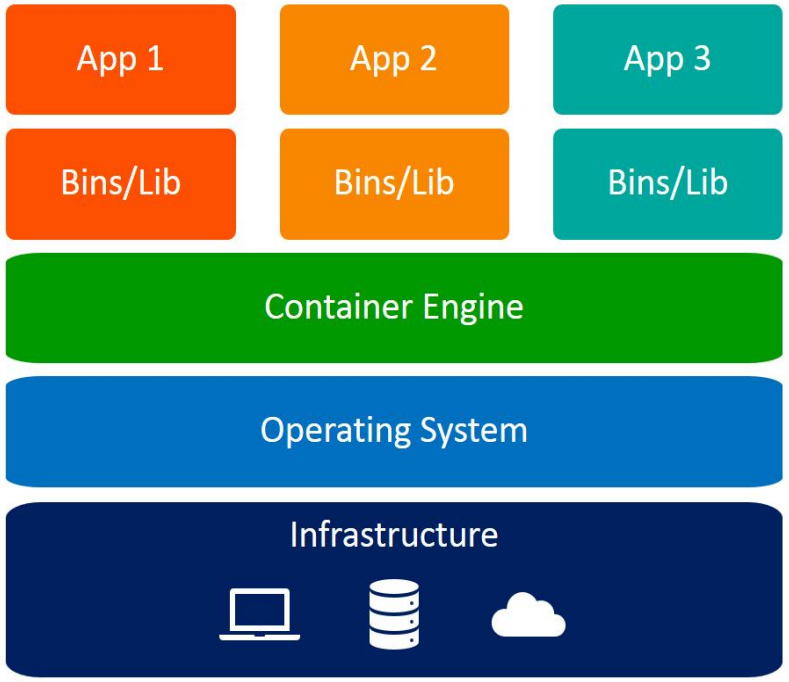
- **Cgroups:** Monitors & Limits Memory, CPU, Disk to your container process.



How containers make a difference



Virtual Machines



Containers

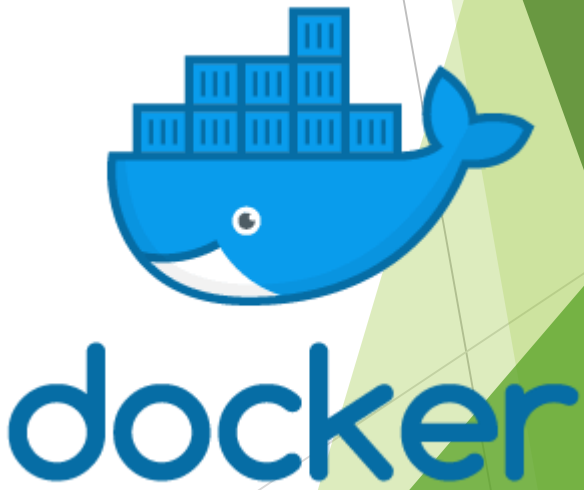
What is Docker

“Docker is a tool that automates the deployment of your application in Lightweight containers so that the application can work in different environments”

- Package your apps with required libraries, Configurations and dependencies as images.
- Makes use of Namespaces , Cgroups & Union File system to bring isolation of your apps.

Docker Engine helps in

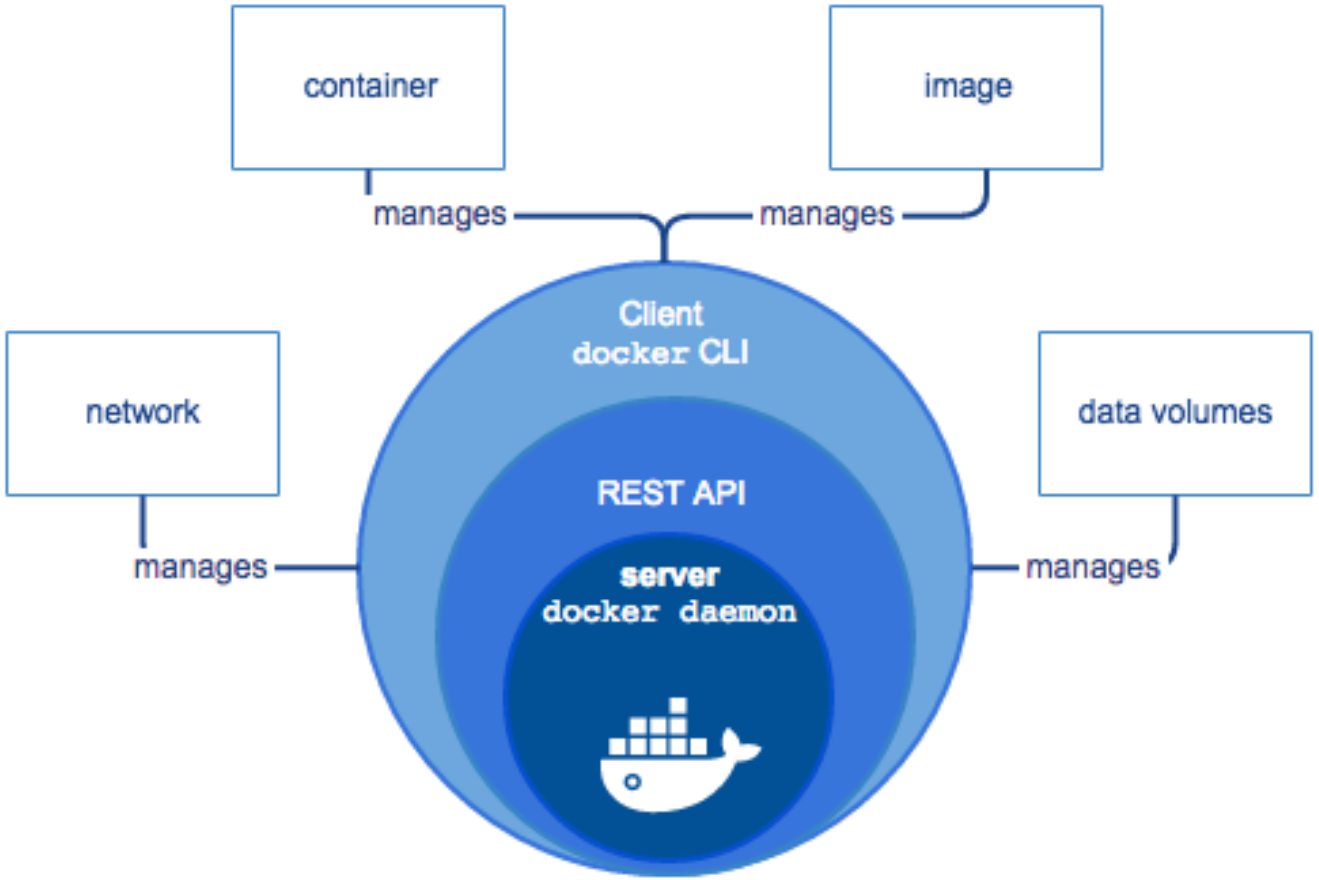
- Running Multiple Containers on the same hardware
- Maintaining each container in an isolated environment
- Easy and fast ways to build, configure & deploy.



Recent Updates

- Latest version: 19.03
- Recent update

Components of Docker Engine



Docker image

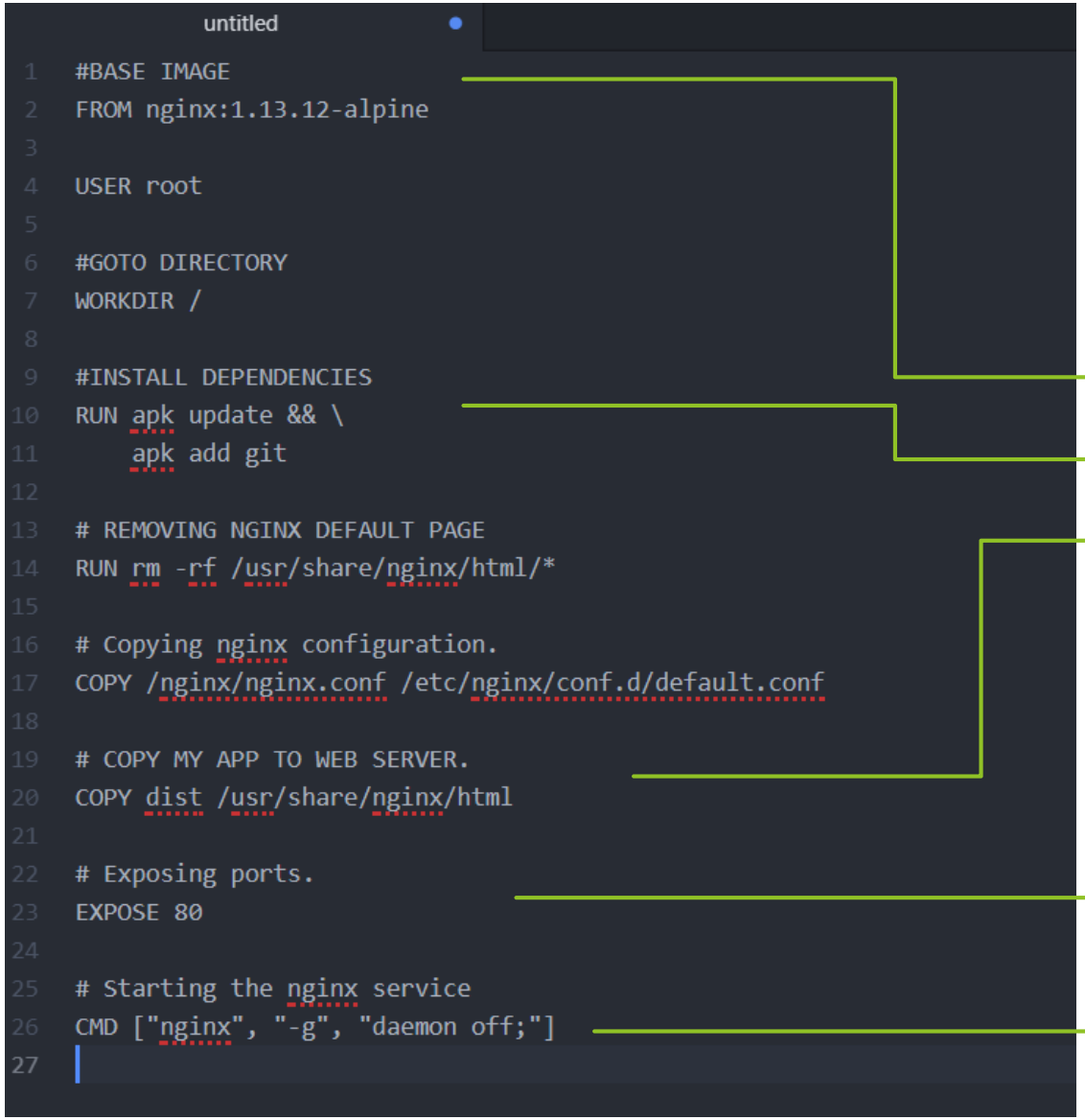
- A Docker image is a package with the set of instructions and dependencies that are needed for your app to run as docker container.
- A Docker image includes system libraries, tools, and other files and dependencies for the executable code.

How to create a docker image:

- Write a Dockerfile with a set of instructions for your application to run
- Run a Docker build

➢ docker build -t <image_name>:<Tag> <dockerfile location>

A Docker file



1) Choose your base image
2) Install your dependencies
3) Copy your app
4) Expose your app connection port
5) Start your service

Docker Networks

→ By Default docker creates its own network called **docker0.**
→ Containers in the same Docker network can talk to each other by their names.

Docker by Default provides 3 network drivers -

- **Host Network** – Uses the network of the Host machine.

- **Bridge Network** – Creates an isolated network where the only a container inside the Created bridge network can talk with one another.

- **None** – Disables the network on the container.

Others:

- **Overlay network** – Distributed network between multiple docker daemons running on different machines.

- **Macvlan** – Assigns mac address to each and every containers making it look like a physical machine.

Volumes

- Volumes help the containers to maintain persistent data.

- This is because by default docker containers are stateless.

Two types of Volumes

Host volumes also know as Bind mounts
- Uses the host machines storage location and directory structure

Docker volumes
- Managed volumes by docker daemon.
- Acts like a separate volume space

Docker Volumes

- Docker volumes are isolated storage volume created inside the docker layer.

- While bind mounts are dependent on the directory structure of the host machine,
- Volumes are completely managed by docker.

Benefits

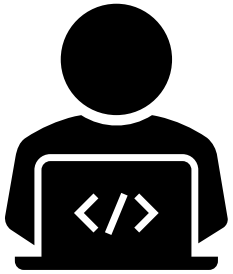
- Works on both Linux & windows
- Volumes can be shared by different containers
- Volumes can use external storage locations

Attaching an NFS to docker container

```
docker volume create --driver local   \
--opt type=nfs  \
--opt o=addr=192.168.1.1,rw \
--opt device=:/path/to/dir \
 mynfsvolume
```

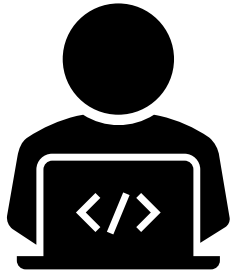
Where does Docker really help

It works in my Environment



DEVELOPER

We followed the deployment document



IT OPS

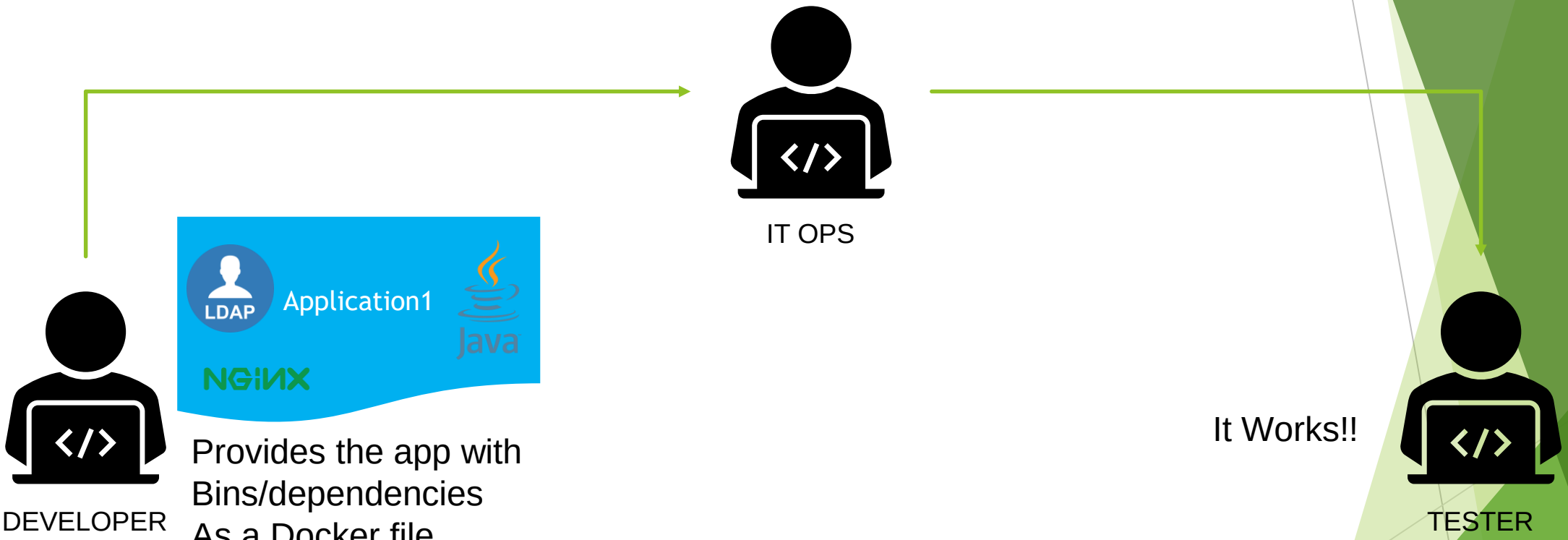
It does not work in my environment

QA

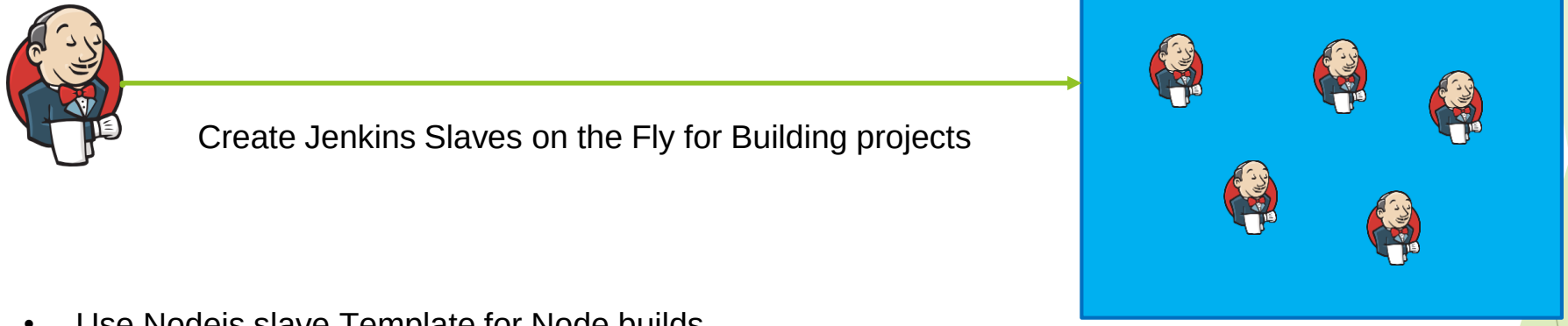
Wait, I just found the problem,
Dev has openjdk-10 and QA has openjdk-8

Deployment made easy

Build & Deploys docker image to QA



Efficient use of Infrastructure



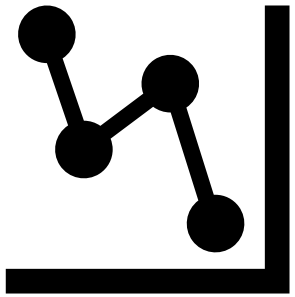
Create Jenkins Slaves on the Fly for Building projects

Single Build machine for Jenkins Environment

- Use Nodejs slave Template for Node builds
- Use OpenJDK slave template for Spring builds
- Use custom templates

Other Benefits

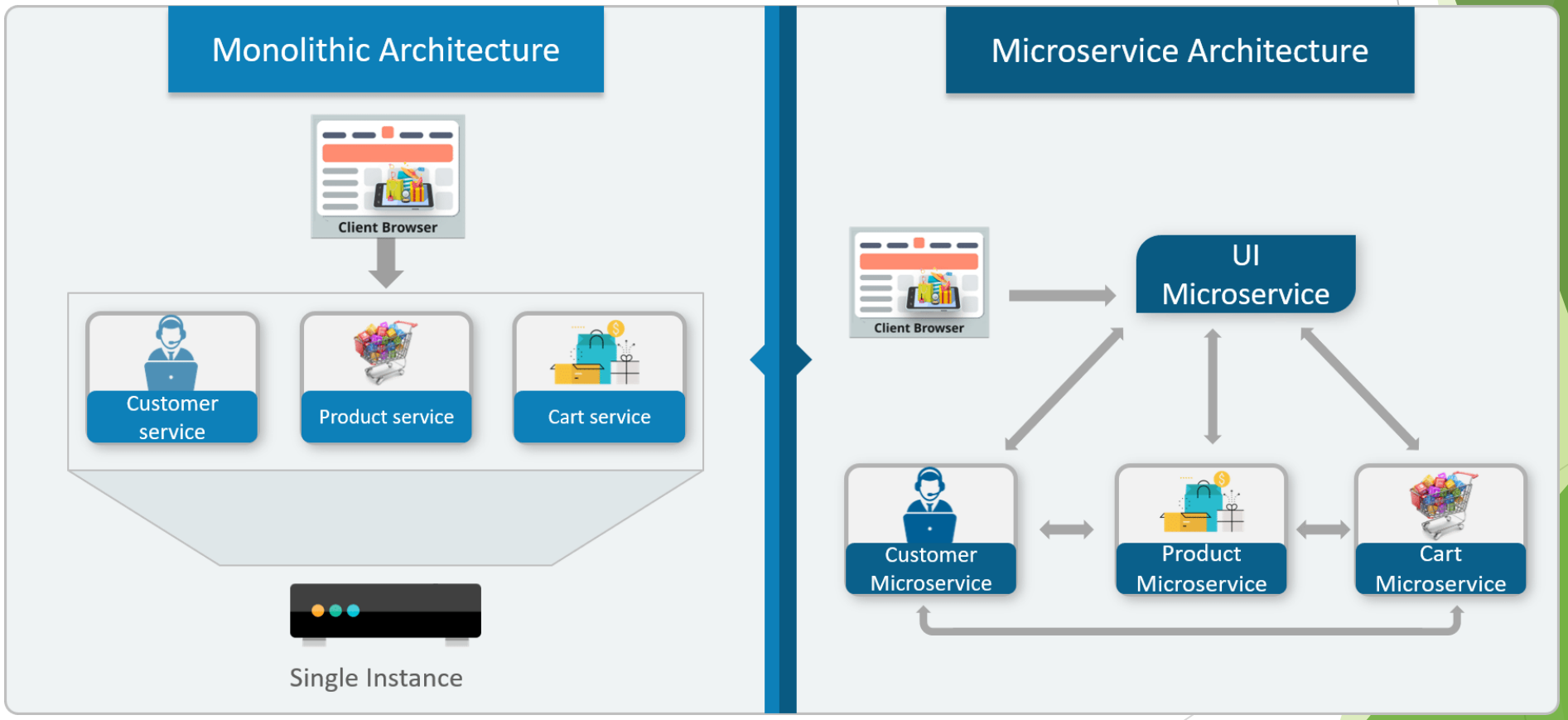
- Faster Build cycles and deployments which **reduces Mean time to Release.**
- Faster Rollbacks in case of issues which **reduces Mean time to Recover**.



Lets start the hands on

What are Microservices

- Microservice is a Development methodology.
- In Microservice Architecture, each service is **self-contained** and implements a **single business capability.**

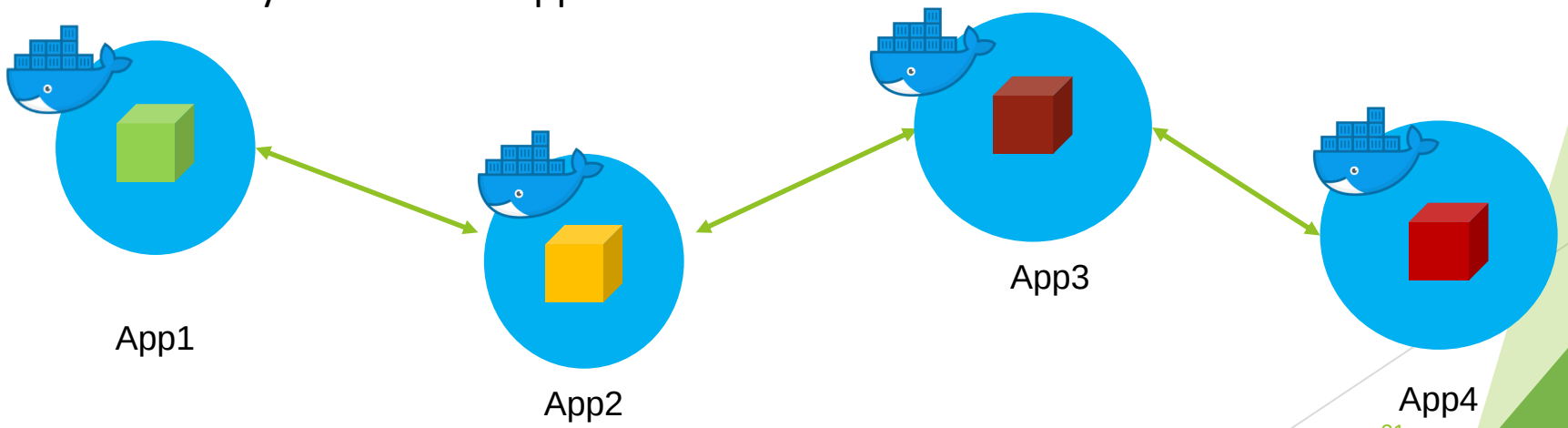


But the Problem is….

- What will happen if you create a dozen microservices for your app?.
- And what if you decide to build several microservices with different technology stacks?.
  E.g. Spring boot, Django, Angular app.
- Your team will soon be in trouble as Operations have to manage even more environments than they would with a traditional monolithic application.

How Containers can help

- Packaging each of your application as Docker containers would make things easier in terms of Deployment, Infrastructure.
- Each App microservice must have a separate Dockerfile with specific instructions for each image.
- Micro service development helps you to find loop-holes in your applications is easily and provides isolation of business functionalities.
- On the same hand Containerizing your micro service helps you in releasing your fixes and features of your Business apps with ease

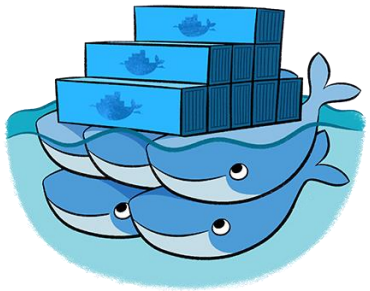


Container Management system

Amazon EKS

Google Kubernetes Engine

Azure Kubernetes Service (AKS)



Docker Swarm

Kubernetes

RED HAT® OPENSHIFT

Other Interesting tools

JENKINSX

SKAFFOLD

Lets start the hands on

THANK YOU