

Docking with Docker



A Gentle Introduction on Docker

Agenda

- Docker Overview
- Popular Vocabularies
- Some Geek Stuffs
- More Technical Stuffs with live demos
- Docker Compose with live demo
- QA Session



Arnab

Software Engineer

Cefalo Bangladesh Ltd.

Consultant for NHST Global Team

Email: arnab.shil@nhst.com

Website: ruddra.com

Shovon

Software Engineer

Cefalo Bangladesh Ltd.

Consultant for NHST Global Team

Email: arshovon@nhst.com

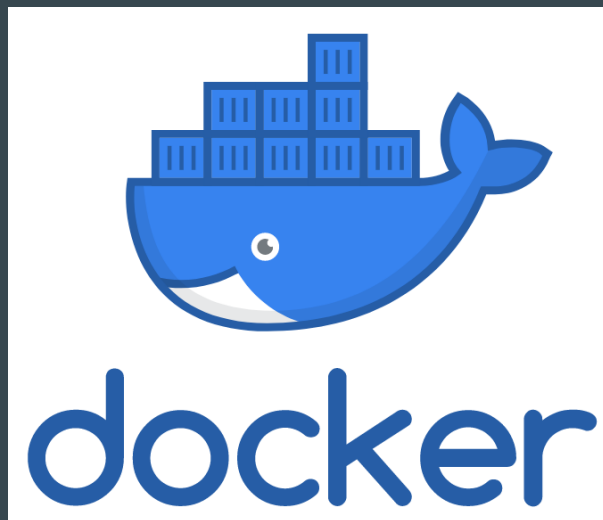
Website: [linkedin.com/in/arshovon](https://www.linkedin.com/in/arshovon)

Docker Overview

Why you need us to use Docker?

What is Docker?

- An open platform to build, ship, and run distributed applications, whether on laptops, data center VMs, or the cloud.
- Automates the deployment of applications inside containers.



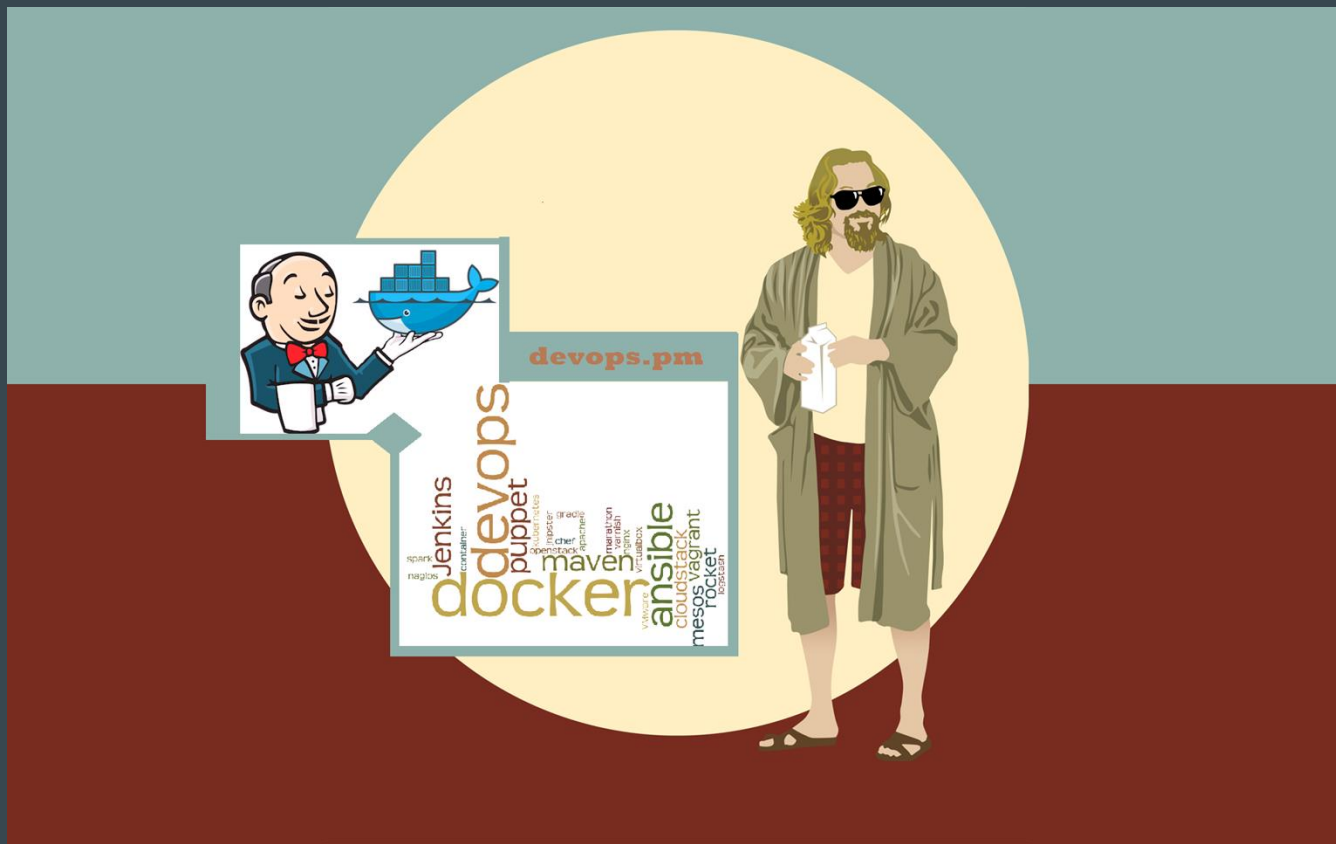
Development Setup != Production Setup (<2014 B.D.)



War between Developer and Operator



Here Comes The Super Chief: Mr. Docker :)



Why Docker?

- Same setup everywhere
- Rapid deployment
- Easy maintainability
- Faster configurations



Build, Ship, Run, Any App Anywhere

From Dev



To Ops



Any App



Any OS



Anywhere



Physical



Virtual



Cloud

Docker Trends

65%
of orgs have
challenges
maintaining
legacy apps

59%
of orgs have
challenges from
inertia of legacy apps
and infrastructure

39%
of orgs are modernizing
legacy apps

44%
of orgs are adopting
microservices

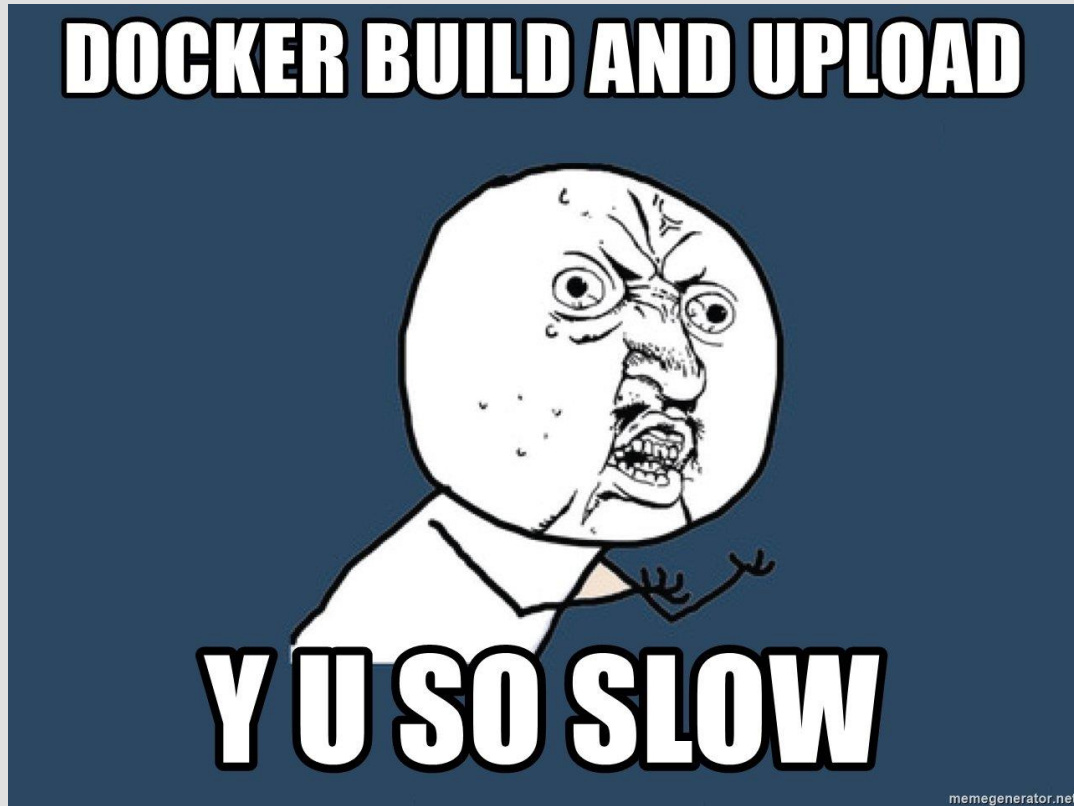


78%
are using, or planning to
use, Docker to build new
microservices applications.



71%
are using, or planning to
use, Docker to containerize
a legacy app.

Docker Disadvantages



- Slow build process
- Connect to existing services
- Resource consumption

Docker Vocabularies

Docker Container

A standardized unit of software
Provides OS level virtualization

Docker Hub / Store

Share images using Docker Hub repositories
Buy certified images from Docker store**

Docker Image

Includes code, libraries & environment
Container is a runtime instance of an image

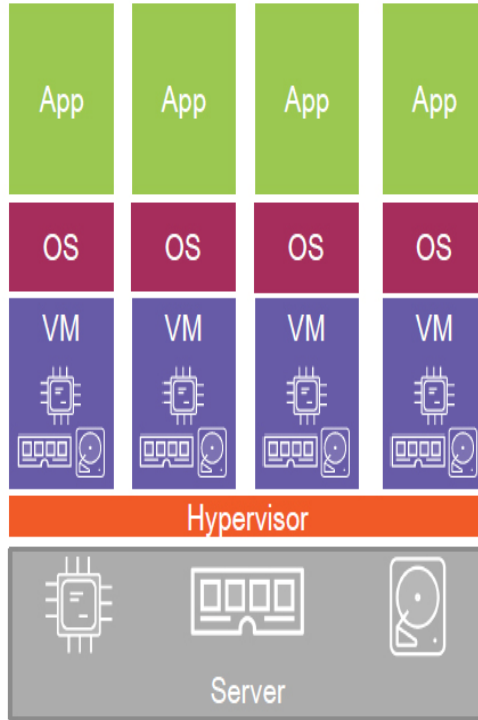
Docker Volume

Define data volumes for a service
Persist even when containers are terminated

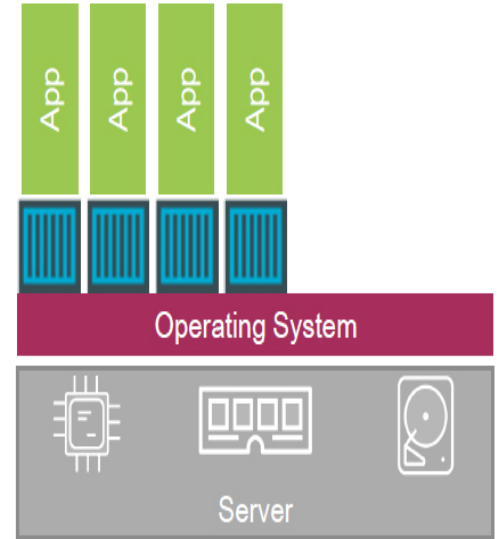
Geek Stuffs

Details on Docker's Internal things

Difference between Virtualization and Containerization

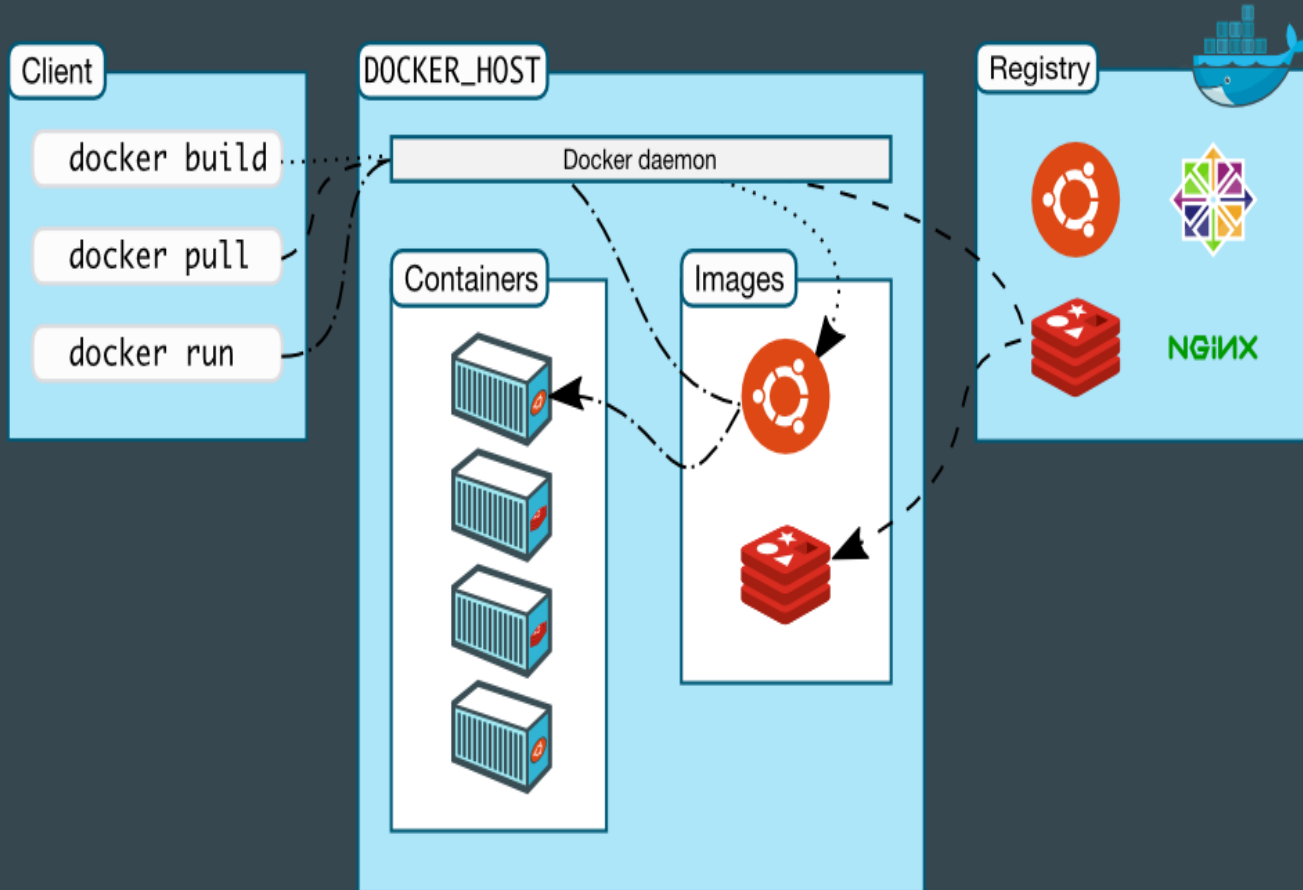


Hypervisor Architecture

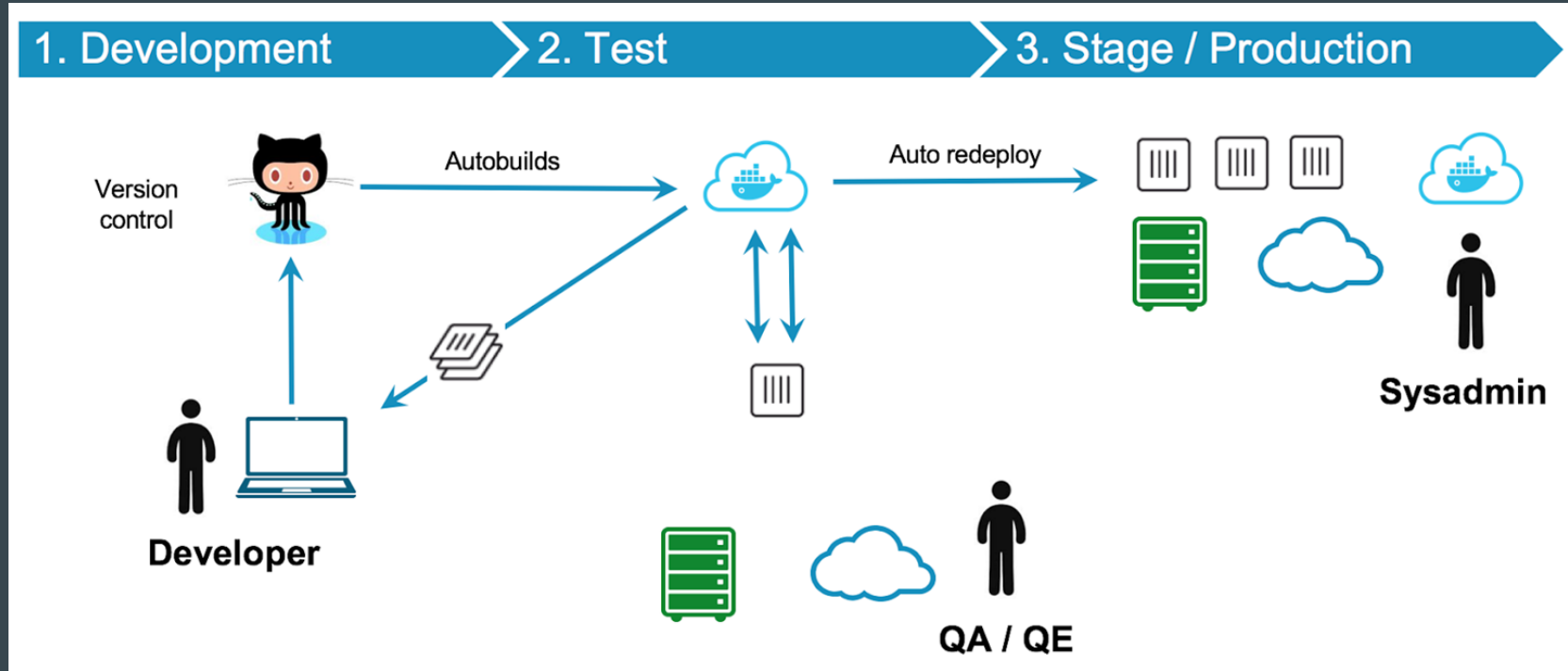


Container Architecture

Docker Architecture



Continuous Integration (CI) & Continuous Deployment (CD) Workflow



More Technical Stuff

Run docker in your machine

Docker Install

MacOs: <https://docs.docker.com/docker-for-mac/install/#download-docker-for-mac>

Windows:

<https://docs.docker.com/docker-for-windows/install/>

Ubuntu:

<https://docs.docker.com/install/linux/docker-ce/ubuntu/#install-using-the-repository>

Docker Pull

Docker Pull means to pull image from dockerhub registry(hub.docker.com).

Command:

```
docker pull <image-name>:<tag>
```

Example:

```
docker pull mysql:latest
```

Docker Build

What is Docker Build?

Building Image from a Dockerfile.

Build From Remote Source(Git)

Command:

```
docker build <git-repository-link>#<branch_name> -t="image-name"
```

Example:

```
docker build https://github.com/ruddra/docker-example.git#build-from-remote  
-t='docker-remote'
```

Custom Dockerfile

```
#Dockerfile Example
```

```
from python:latest
```

```
add . /code
```

```
WORKDIR /code
```

```
RUN pip install -r requirements.pip
```

```
EXPOSE 5000
```

```
CMD python flask_app.py
```

Build Within Local Machine

Command:

```
docker build <path-to-docker-file> -t="<image-name>"
```


Docker Run

Command:

```
docker run <image-name> <docker-port>:<local-machine-port>
```

Docker Set Environment Variable

Command:

```
docker run -e <ENVIRONMENT-VARIABLE-1>=<VALUE-1> <ENVIRONMENT-VARIABLE-2>=<VALUE-2> <image-name>
```

Example:

```
docker run -e MYSQL_ROOT_PASSWORD=root -e MYSQL_USER=root -e MYSQL_PASSWORD=root -e MYSQL_DATABASE=mydb mysql
```

Docker Hub AKA Docker Cloud

Login Command:

```
docker login
```

Tag your Image for Docker Hub:

```
docker tag my_image $DOCKER_ID_USER/my_image
```

Push your Image to Docker Hub:

```
docker push $DOCKER_ID_USER/my_image
```

Push your Image for Docker Hub:

```
docker pull $DOCKER_ID_USER/my_image
```

Docker Useful Commands

Remove Container:

```
docker rm <container-id>
```

Show running container:

```
docker ps
```

Show all containers:

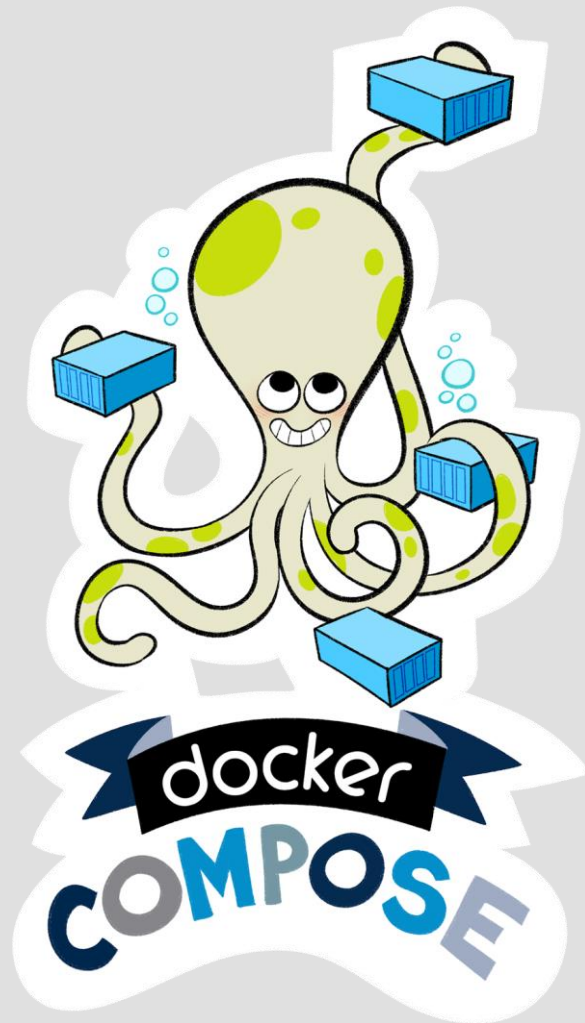
```
docker ps -a
```

Docker shell:

```
docker exec -it <container-id-OR-name> bash
```

Docker Compose

- Tool for defining and running multi-container Docker applications
- Use a YAML file to configure application's services
- Call all the services using single command
- Same everywhere:
 - Production
 - Staging
 - Development
 - Testing
 - CI



Docker Compose Useful Commands

Build Containers:

```
docker-compose build
```

Create and start containers:

```
docker-compose up
```

Stop and remove containers, networks, images, and volumes:

```
docker-compose down
```

Execute a command in a running container:

```
docker-compose exec
```

Docker Compose Useful Commands (Continue)

Start services:

```
docker-compose start
```

Stop services:

```
docker-compose stop
```

Restart services:

```
docker-compose restart
```

Remove stopped containers:

```
docker-compose rm
```

Questions Time



Thank you

Please don't ask more
questions :)

