

# Introduction to Docker

Mohd Faiz Hasim

faiz.hasim@servicerocket.com • faizhasim@gmail.com

---

Some Info

- Also available as [PDF](#), [EPUB](#) and [MOBI](#) formats.
- Hosted at [Github](#).
- Mistakes? Improvements? Make me a pull request.

## What is Docker?

---

Pretty much a **lightweight Virtual Machine**

- Own process space
  - Own network interface
  - Can run stuff as root
  - Implemented using [LXD](#) - new “hypervisor”
- 
- 
-

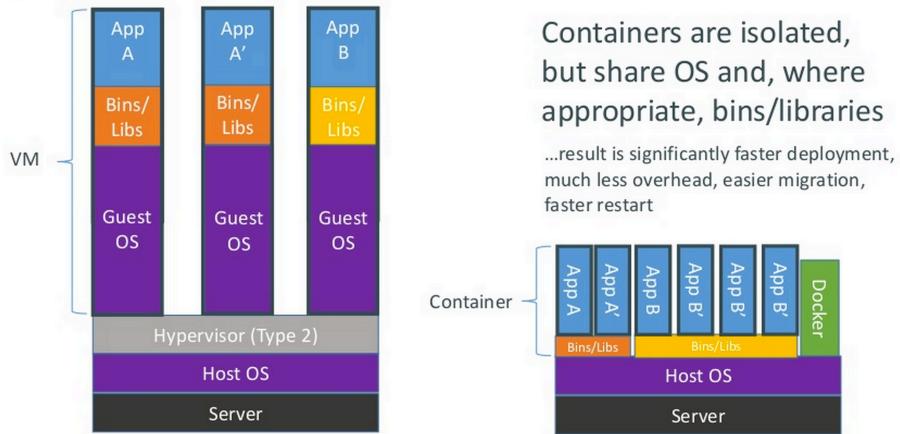


Figure 1: VM vs Container - From Docker

## What are the basics of the Docker system?

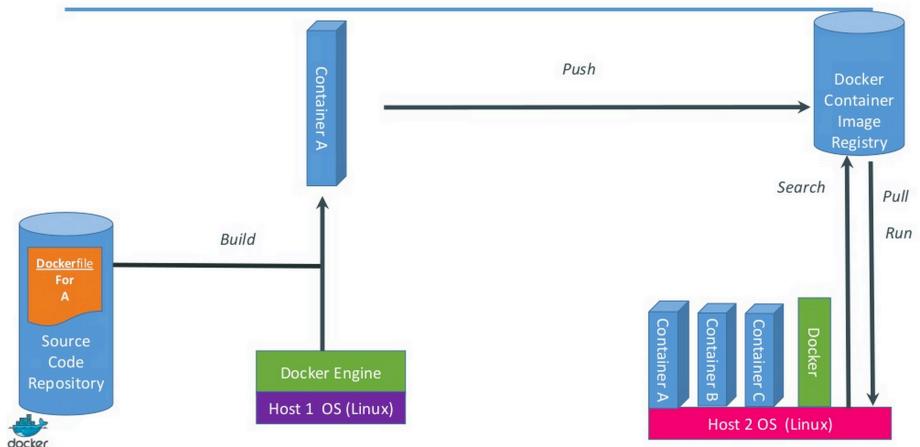


Figure 2: Docker System - From Docker

## Changes and Updates

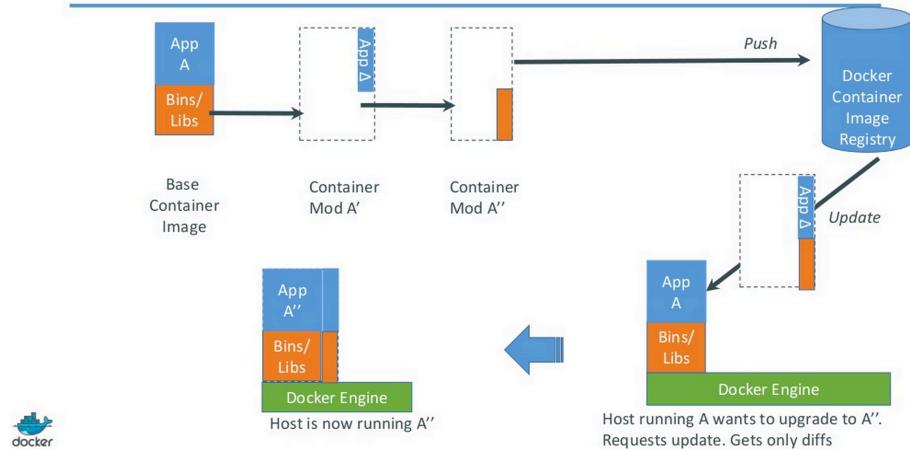


Figure 3: Changes and Update - From Docker

## What the big fuss?

### Shipment

- Works on my machine
- Works on your machine
- Works on staging
- Works on production

### How would you treat code?

- Push your source code on Github/Stash/Bitbucket?
- Peer review on your code?
- Share your source code?

### Docker works like Git

- Push your `Dockerfile` to Dockerhub (or your private Docker registry)
- Peer review your Docker architecture.
- Share your Dockerfile via Dockerhub/private Docker registry or build yourself from SCM

## Run everywhere

- Ubuntu/CentOS/Debian/BusyBox
- Physical or Virtual, cloud or on-premise

## Run anything

- Anything that you can run on Linux
- In other words, if you can install stuff on your Linux box, you can install on Docker

## Pulling busybox and play with it

---

```
docker search busybox
```

```
https://registry.hub.docker.com/search?q=busybox
```

---

```
docker pull busybox
```

```
docker pull node:latest
```

---

```
docker images
```

---

```
docker run busybox ps aux
```

```
docker run -i -t busybox /bin/sh
```

## Build Hello World in NodeJS

---

Running node:

```
docker run -it --rm --name hello-world-node node:latest node
```

---

```
mkdir helloworld
```

```
touch helloworld/index.js
```

```
var express = require('express');
var app = express();

app.get('/', function (req, res) {
  res.send('Hello World!')
});

var server = app.listen(3000, function () {
  var host = server.address().address;
  var port = server.address().port;
  console.log('Example app listening at http://%s:%s', host, port);
});
```

---

```
touch helloworld/Dockerfile
```

```
FROM      node:latest
MAINTAINER Mohd Faiz Hasim
```

```
ENV APP_PATH /user/src/myapp
```

```
EXPOSE 3000
```

```
ADD index.js $APP_PATH/index.js
RUN npm install --save express
```

```
WORKDIR $APP_PATH
```

```
CMD node index.js
```

---

```
docker build -t faizhasim/helloworld helloworld/
```

---

```
docker run -p=80:3000 -ti --rm --name hello-world-node faizhasim/helloworld
```

## Orchestration with [Fig](#)

---

```
touch fig.yml
```

---

```
helloworld:  
  build: helloworld/.  
  ports:  
    - "80:3000"
```

---

```
fig build
```

---

```
fig up
```

## Where should I go next?

For motivation...

Checkout [How we orchestrates 8 Docker containers to support our Learndot build pipeline?](#)

---

## Going forward

- Try to link your hello world container to read value from redis container.
  - Checkout examples from [Fig](#) website.
  - Ask Shuaib to demo to you.
- 

## Recommended reads

- The Docker Book by James Turnbull
  - Docker documentation website
- 

## Some interesting facts

- This slides is build using docker!
  - Fig will be part of official Docker orchestration tool
  - You can deploy Docker using Amazon Beanstalk
- 

Docker is not magic. You create magic around Docker.