Docker...

...and DevOps - 20190718

Introductions and About me

- Rajshri Mohan
- Proud batch of 2010-2014 BE CSE
- I work at IQZ Systems
  - I build stuff
  - I fix things
  - I lead a team
- What this talk is going to be about
  - DevOps
  - Docker
  - Industry and Docker
  - A demo

Before we do this...

- You can call me “Rajshri”. Or “dude”. Or “bro”.
  - Or anything, but “sir”.
- Feel free to stop me anytime if you do not understand something I mention.
- Be vocal. I like engaging in conversations.
  - And I hate sermons.

What made this possible?

DevOps?

“DevOps is a set of practices that automates the processes between software development and IT teams, in order that they can build, test, and release software faster and more reliably.” - Atlassian

“DevOps is a set of software development practices that combine software development (Dev) and information technology operations (Ops) to shorten the systems development life cycle while delivering features, fixes, and updates frequently in close alignment with business objectives.” - Wikipedia

DevOps?

- Traditional method of software deployment has shortcomings
- Continuous Integration & Continuous Deployment
- Infrastructure as Code

Containerization

- Transportation of goods
  - Old school methods
  - Shipping containers
- Container Engine
  - Is Docker the only container engine?
    - Docker and rkt
- Why isolate?

Containerization

- VM
  - Bulky and bloated
  - High resource utilization
  - Architectural differences
  - Hypervisor
- Containers
  - Lightweight
  - Resource sharing due to minimal footprint
  - Architecture agnostic
  - Runtime/Engine

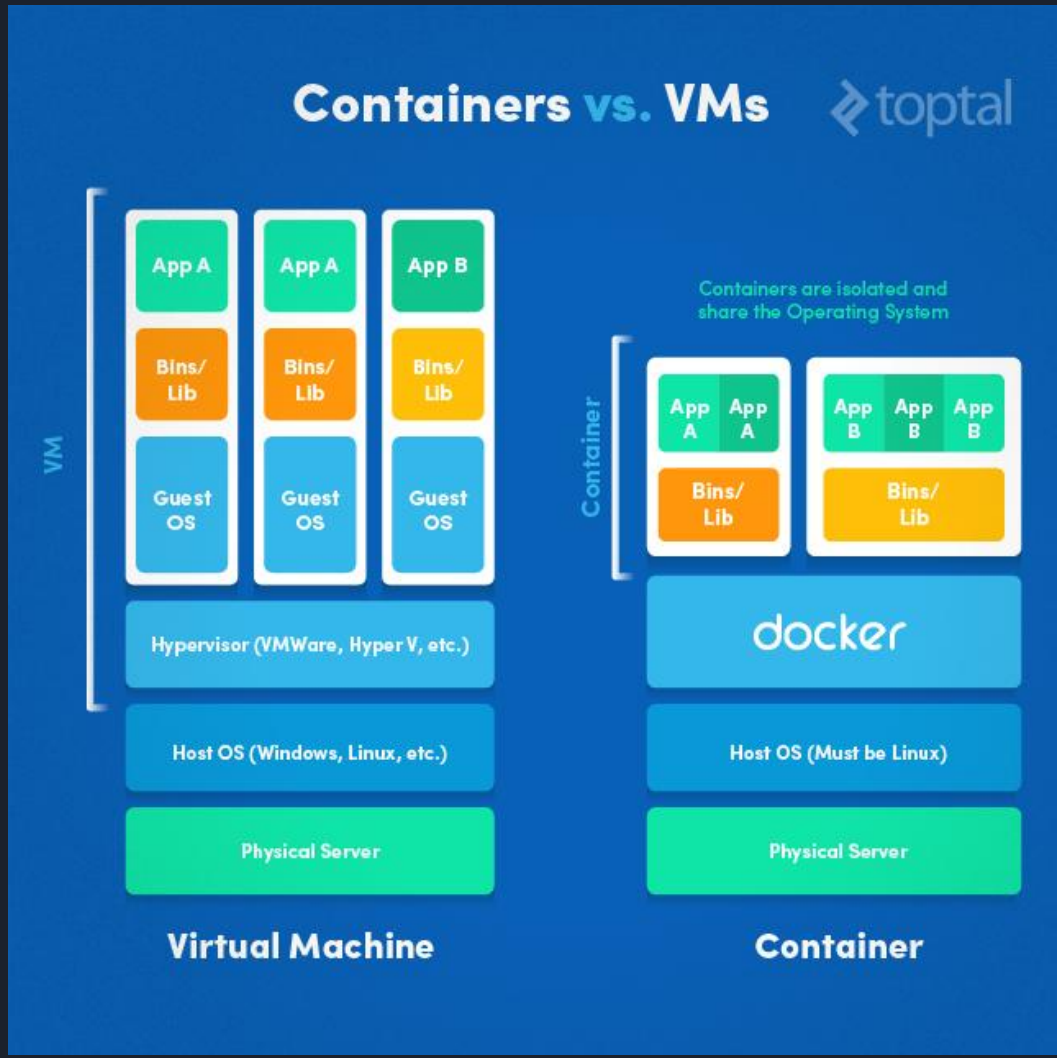

Image shamelessly stolen from TopTal :P

Docker and DevOps

- *“Build once; Deploy anywhere.”*
- Brings infrastructure management to the developer.
- Reduces unnecessary man power. Or in another perspective, makes man power highly efficient.

Containerization in Docker

- The Dockerfile
- Filesystem isolation
  - Volumes
- Network isolation
- Caching and Layers

The Dockerfile

- Defines how a container should be built.
- Caching and Layers
- A more complex example (external)

```
# Set the base image to node
FROM node:10.16.0

# Add metadata
LABEL maintainer.name="Rajshri Mohan K S" \

maintainer.email="rajshri.m@iqzsystems.com"

# Change shell from sh to bash
SHELL ["/bin/bash", "-c"]

# Set environment variables
ENV WORK_DIR /app/

# Get app port
ARG PORT

# Set Working Directory
WORKDIR $WORK_DIR

# Set up app
COPY package*.json $WORK_DIR
RUN npm install
COPY . $WORK_DIR
RUN npm run build

# Expose port
EXPOSE $PORT

CMD ["node","./dist/main.js"]
```

“But it works in my computer”

- The problem
- How Docker solves this
- Platform agnostic deployments
  - Windows
  - Linux
  - Mac, etc.
- Advantages for developers, testers, and business.

Demo

Adoption

- Cloud providers
  - AWS
  - Azure
  - GCP
- Deployment Tools
  - Jenkins
  - GitLab
  - And more…
- Kubernetes

Advantages

- How Docker enables CI/CD

Questions?

Thanks and stuff...

- Contact
- IQZ is hiring ;)

```
return 0;
```