Docker: Starting Small

For Docker Ottawa Meetup
by: Patrick Santos

Overview

- Docker at small scale: My use case

- Why get started with Docker
  - (if you already have VMs)
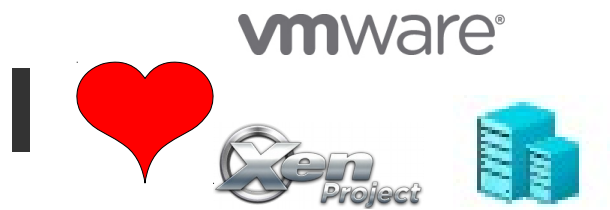
- How to start using Docker

- Demos: Basics

Experienced users can
start sleeping here

Docker at Small Scale (+why)

- Heterogeneous servers from:
  - Research projects / demos
  - TCO calculation vs cloud
- Docker helps
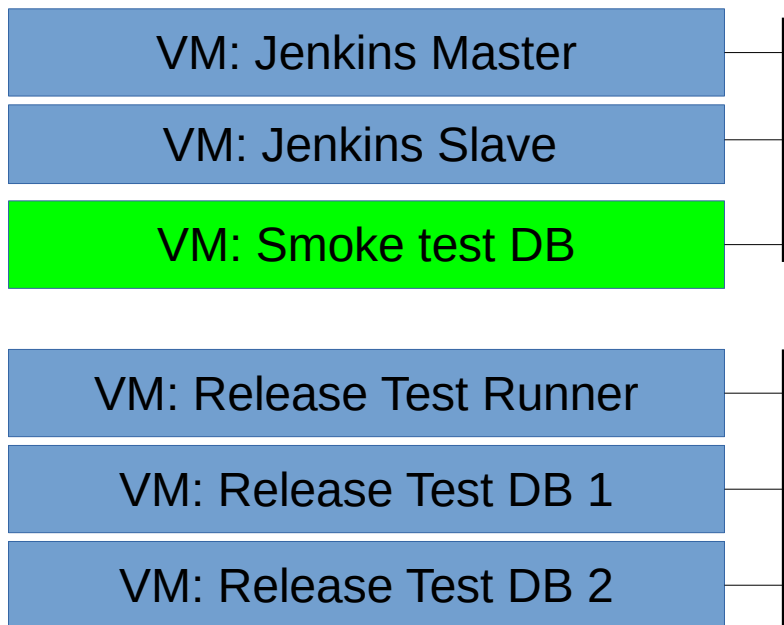 - Better scaling
  - Setup / teardown

I ♥ vmware Xen Project : **Why Docker?**

- Docker works with your VMs (+ other tooling)

- Speed to spin up / tear down

  - Seconds versus minutes

- Resource consolidation

  - ↓30-35% storage, ↓7% RAM, ↓25% compute [1]

How do I start?

1) Install Docker on your machine: https://store.docker.com/search?type=edition&offering=community

2) Play around / learn: https://github.com/docker/labs

3) Find something to "Dockerize"

First Docker ~~Victim~~ Candidate

<table>
<tr><td>VM: Jenkins Master</td></tr>
<tr><td>VM: Jenkins Slave</td></tr>
<tr><td>VM: Smoke test DB</td></tr>
</table>

<table>
<tr><td>VM: Release Test Runner</td></tr>
<tr><td>VM: Release Test DB 1</td></tr>
<tr><td>VM: Release Test DB 2</td></tr>
</table>

Test setup before Docker

- Not in production!

- Docker image available

- Ephemeral

First Experiments

- Run a Docker container

  - docker run --rm -it ubuntu:16.04 /bin/bash

- Check the space savings versus VM

  - docker ps

  - docker inspect

- Try pipes with it!

  - docker run ubuntu:16.04 echo hello | tee .docker.out

[2]

Replacing a Database VM - 1

- Find suitable container

  - PostgreSQL:
    https://store.docker.com/images/postgres

- Note – official images:

  - Community containers have a username, e.g.
    docker pull **circleci**/postgres

Replacing a Database VM - 2

- Look in hub.docker.com or store.docker.com page for instructions

<div style="border:1px solid #ccc">

start a postgres instance

```
$ docker run --name some-postgres -e POSTGRES_PASSWORD=mysecretpassword -d postgres
```

This image includes `EXPOSE 5432` (the postgres port), so standard container linking will make it automatically available

to the linked containers. The default `postgres` user and database are created in the entrypoint with `initdb`.

</div>

Connecting to the Database

- With another Docker container

  - []

- Native process from the same machine

  - Watch port or firewall rules

- Expose port from another machine

  - []

Next Steps

- Docker volumes to persist data

  - https://docs.docker.com/engine/admin/volumes/volumes

- Learn more:

  - https://github.com/docker/labs

  - Creating your own Dockerfile

  - Docker Swarm mode

Suppl: Docker vs VMs

Caveat: Checking containers

- For security conscious: Auditing still applies!

- Example insecure Dockerfile:

```
# Install Kafka, Zookeeper and other needed things
RUN apt-get update && \
    apt-get install -y zookeeper wget supervisor dnsutils && \
    rm -rf /var/lib/apt/lists/* && \
    apt-get clean && \
    wget -q http://apache.mirrors.spacedump.net/kafka/ (...) && \
    tar xfz /tmp/kafka_"$SCALA_VERSION"-"$KAFKA_VERSION".tgz -C /opt && \
    rm /tmp/kafka_"$SCALA_VERSION"-"$KAFKA_VERSION".tgz
```

References

[1] Mike Coleman. 2017. *Docker?!? But I'm a SYSADMIN!* Retrieved Aug. 11, 2017, from https://youtu.be/M7ZBF-JJWVU

[2] Docker. 2017. *Running your first container*. Retrieved Aug. 16, 2017, from
https://github.com/docker/labs/blob/master/beginner/chapters/alpine.md

[3] Docker. 2017. postgres. Retrieved Aug. 16, 2017, from
https://store.docker.com/images/postgres