docker fundamentals

Intro

Intro to docker

Docker platform

Engine

Hub

Machine

Swarm

Compose

Kitematic

Problems addressed by VM

Allow better use of hardware

No need to reserve one machine per service

Quickly setup an environment

Scalability

Scalability

New vm's can be spun up quickly on demand

Now, firms are provisioning vm's in the cloud.

Firms only have to pay for whay they use, don't evven have to buy a server

Getting Started

Using Docker

Working Examples

Containers

OS-level vs Hypervisor Virtualization

OS level virtualization

Kernel-managed isolated user space

Multiple containers can be running at a time

Kernel manages the containers as opposed to a separate hypervisor-type thing

Hypervisor virtualization

VMWare

Microsoft

Duplication of an entire OS running in a virtual resource space

From the POV of a user, each container feels like a separate server

For many OS, kernel also provides resource mechanisms such that one container's activities don't impact other containers

Unix containers

Can be considered an extension of the OS chroot mechanismroo

Containers versus Hypervisor virtualization

Speed

Containers are much quicker launching

Containers load information into memory

Containers are much more resilient

Soft memory

Running an application out of memory as opposed to being swapped to / from disk

Enables quick relaunches / crash mgmt

Security

Running apps in memory limits ability of enemies to hack your apps

Docker

Standardization of containers

Container API's allow all processes related to container to work better

Provides standardized container API for hosting apps

Technology

Written in Google GO

cgroup

namespace

AUFS union filesystem

lxc underneath

Host OS's

Ubuntu - older versions require you to update the kernel

Docker isn't yet prod ready (< 1.0)

Installation

Ubuntu

AWS

Vagrant

Vagrant to AWS

Docker Fundamentals

Starting docker

As daemon

Interactivey

Docker must first be started before any containers can be launched

Docker binds to unix socket (not a TCP port)

Docker group

Can use group to allow read/write to unix socket by docker group

Allows clients in the docker group to avoid having to use sudo when executing commands against the docker container

Once docker is running, you can launch containers

Running apps and services

Docker repository

Hosted collection of tagged images

In total, makes up a container's filesystem.

Docker commands

docker run

launch a container

docker version

display verion info

docker ps

list containers running

docker search

search for images

docker pull

download os image

Terms

central index

EC2

Vagrant

Dockerized app

App built so that it can be added to standardized docker container

AWS

Azure

Rackspace

Vagrant

Chef

Puppet

OpenStack

Soft memory

Hypervisor

Virtualization mechanism

Virtual Machine Monitor (VMM)

Runs on a host machine along with one or more VM's

Presents the guest OS's with a virtual operating platform and manages execution

of the guest OS's

Virtual Machine (VM)

Called a guest machine

OpenVZ

Linux VServer

Virtualization

FreeBSD Jails

Container

Refers to an OS-level virtualization

Zone

Separate servers within an OS

Partition

Jail

Another name for a container

Continuous delivery

apt-get

Hypervisor

Each VM requires it's own OS, cpu, memory, etc

More VM's => more resources needed

Container-based virtualization

OS creates separate container's, each having its own set of resources

Hardware emulation

Virtualization

Allows running multiple contexts on a single machine.

Concepts

Container

Guest instance created by the kernel

Docker-hub

Public docker repository

Docker image

Docker container