Building Clustered Applications with Kubernetes and Docker

Steve Watt

redhat.
Why Kubernetes?
Kubernetes Architectural Overview

Kubernetes Cluster

Kubernetes Master Server(s)
- etcd
- API Server
- Scheduler
- Controller Manager

Linux Server(s)

Kubernetes Node
- Docker
- Kubelet
- Kubernetes Proxy

Linux Server

Kubernetes Proxy

Kubernetes Node
- Docker
- Kubelet
- Kubernetes Proxy

Linux Server

Kubernetes Node
- Docker
- Kubelet

Linux Server
Installing Kubernetes

- Hosted Services: Google Compute Engine
- Support for a wide variety of Infrastructure (Azure, Rackspace, vSphere, AWS)
- Support for several OS’ (RHEL, CentOS, Fedora, Debian, Ubuntu, Atomic, CoreOS)
- Local but automated (Vagrant/Ansible) * Magic *
- Local but manual (Fedora) * What I use *
What do you need to know to get started?

Pods
- Pods describe a tightly coupled group of containers that typically need to be deployed on the same server because they share resources.
- Pods are described in Pod Files which also articulate what shared resources the containers in the Pod require. An example of a Shared Resource is a Kubernetes Volume.
- Pods are deployed on Kubernetes Nodes.

Labels
- Pods describe a tightly coupled group of containers that typically need to be deployed on the same server because they share resources.

Services
- Pods and containers will die and get scheduled on different machines. Services provide you with the ability to have a consistent, known endpoint for your pod that stays the same, regardless of where your pod is moved to.

Replication Controllers (RCs)
- Replication Controllers are used to scale out Pods.
- RCs produce identical instances of pods that correspond to a replica count that has been set.
- Services will route requests between multiple instances of the same Pod.

Volumes
- Allow you to mount Local and Network Storage Systems to the containers with your Pods.
Awesome! Now, let's build some applications.
Example 1 – Build a Scalable Web Application with Local Storage
Example 2 – Then use Labels to Hot “Upgrade” to Shared Storage
Example 3 – Now let’s add a Database

Kubernetes Cluster

Kubernetes Node 1
- NGINX Pod
- NGINX Container

Kubernetes Node 2
- MySQL Pod
- MySQL Container

ClusterFS

Ceph
- RBD
- Network Disk

Browser
Request

NGINX Service
NGINX RC

@wattsteve
Example 4 – Deploy Apache Spark to Analyze your data

Kubernetes Cluster

Kubernetes Node 1
Spark Worker Pod
Spark Worker Container

Kubernetes Node 2
Spark Master Pod
Spark Master Container
Spark Worker Pod
Spark Worker Container

Kubernetes Master
Spark Master Service
Spark Worker RC

Terminal Or Zeppelin

GlusterFS
Example 5 – Deploying Storage Platforms in Kubernetes

Kubernetes Cluster

Browser Request

Kubernetes Master

NGINX Service

NGINX RC

Kubernetes Node 1

GlusterFS Pod

GlusterFS Container

Kubernetes Node 2

GlusterFS Pod

GlusterFS Container

Kubernetes Node 3

NGINX Pod

NGINX Container

Mount

GlusterFS

NGINX Service

NGINX RC

GlusterFS Pod

GlusterFS Container

NGINX Pod

NGINX Container
Thanks for Listening!

Interested in following up?

Steve Watt

swatt@redhat.com

Examples:

https://github.com/wattsteve/containercon