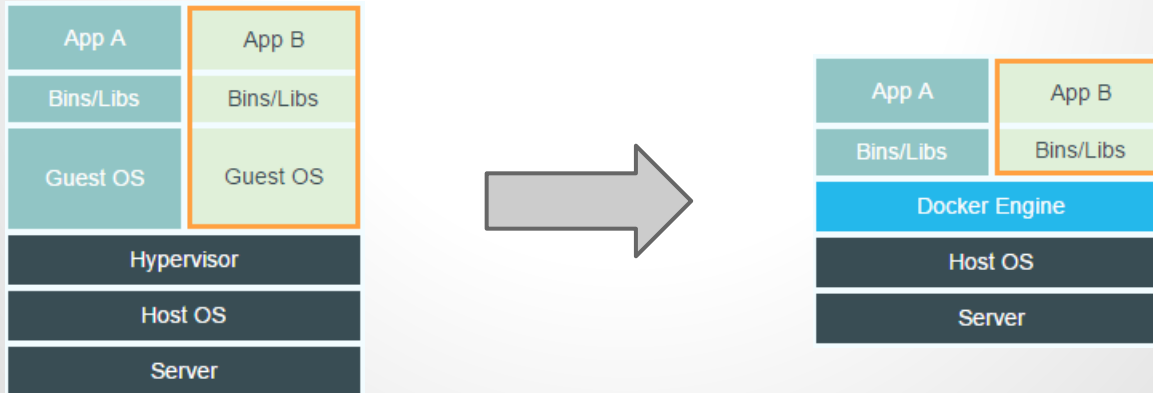


Docker, Kubernetes & Jube

Build, containerize & orchestrate

Docker

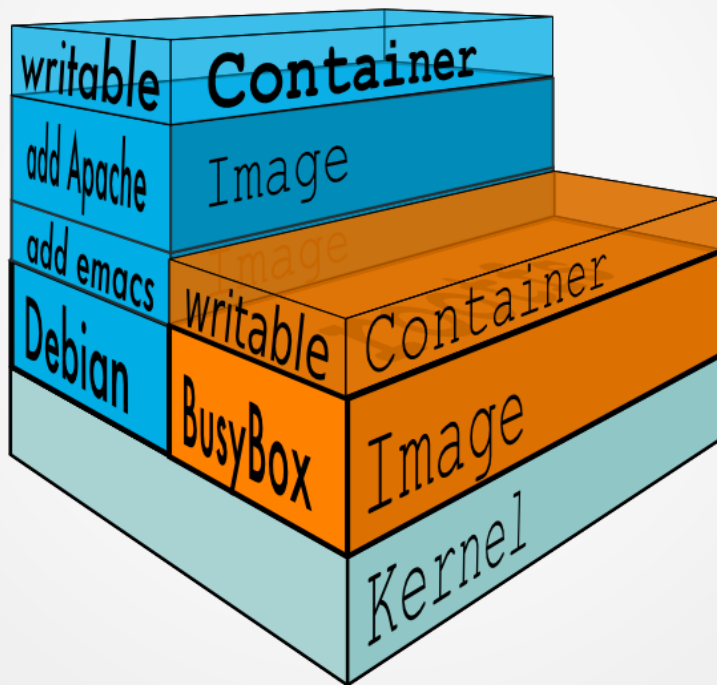
- Platform for building, distributing and running applications
- Lightweight container “virtualization”
- 10s of VMs => 100s, 1000s of containers per host



Docker concepts

- Images
 - A read-only package of an app & environment
 - Consists of Layers (some layers shared by images)
- Registries
 - Pull images from & push images to registry
 - Private & public registries (Docker Hub)
- Containers
 - A running image (“Process in a box”)
 - Read-Write layer on top of base image

Docker layers



Creating Docker images

- Use existing image (e.g. fedora, jboss/wildfly)
- Run a command (anything, even a shell)
 - Docker creates a new container from image
 - Allocates a read-write layer on top of image
 - Executes the command
- Run additional commands
- Look up last container id
- Commit the container as new image

Dockerfile

- Build an image automatically
- Specifies base image and instructions:
 - FROM <existing image>
 - ADD <local file> <path inside image>
 - RUN <cmd>
 - EXPOSE <port>
 - ENV <name> <value>
 - CMD <cmd>

Dockerfile - example

```
# Use latest jboss/base-jdk:7 image as the base
FROM jboss/base-jdk:7

# Set the WILDFLY_VERSION env variable
ENV WILDFLY_VERSION 8.2.0.Final

# Add the WildFly distribution to /opt
RUN cd $HOME && curl http://download.jboss.
org/wildfly/$WILDFLY_VERSION/wildfly-$WILDFLY_VERSION.tar.gz | tar zx && mv
$HOME/wildfly-$WILDFLY_VERSION $HOME/wildfly

# Set the JBOSS_HOME env variable
ENV JBOSS_HOME /opt/jboss/wildfly

# Expose the ports we're interested in
EXPOSE 8080 9990

# Set the default command to run on boot
CMD ["/opt/jboss/wildfly/bin/standalone.sh", "-b", "0.0.0.0"]
```

Kubernetes

- Orchestration system for Docker containers
- Provides basic mechanisms for:
 - deployment
 - maintenance
 - scaling
- Auto-restarting, re-scheduling & replicating containers

Kubernetes architecture

- Master node
 - etcd (distributed key value store)
 - Kubernetes API Server (REST) (+ Scheduler)
 - Kubernetes Controller Manager Server
- Minions
 - Docker
 - Kubelet
 - Kubernetes Proxy

Kubernetes concepts

- Pods
- Volumes
- Labels
- Replication controllers
- Services

Pods

apiVersion: v1beta1

id: www

desiredState:

manifest:

version: v1beta1

id: X

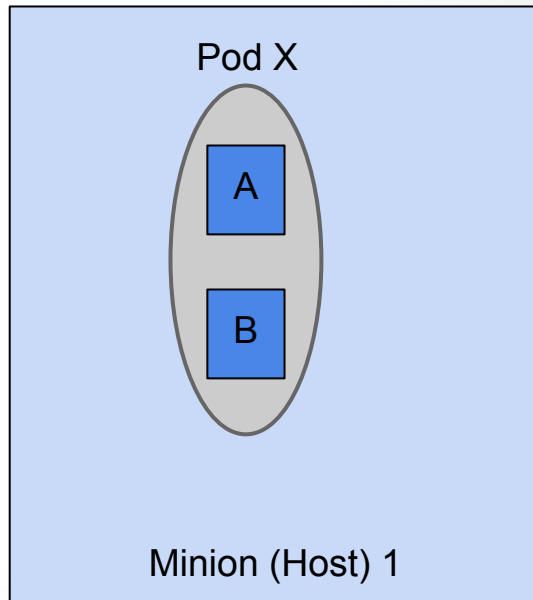
containers:

- name: nginx

image: dockerfile/nginx

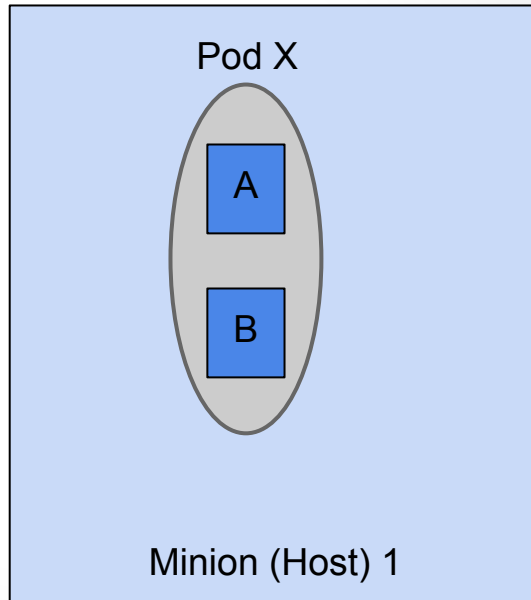
- name: mydb

image: foo/mycooldb



Pods (continued)

- Resource sharing & communication
- Not fully isolated
- Scheduled to a node
- Containers are auto-restarted
- If a node dies, its pods are deleted (not rescheduled)



Volumes

```
desiredState: manifest:
```

```
containers:
```

```
- name: A
```

```
image: foo/A
```

```
volumeMounts:
```

```
- name: vol1
```

```
mountPath: /data/vol1
```

```
volumes:
```

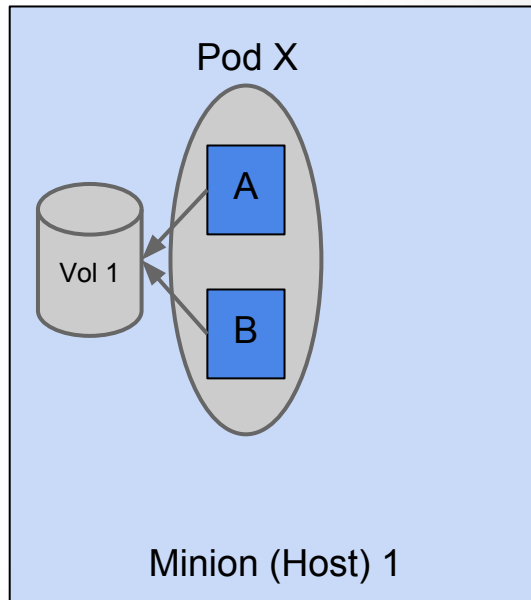
```
- name: vol1
```

```
source:
```

```
emptyDir: {}
```

Or:

```
hostDir: /opt/data/vol1
```



Labels

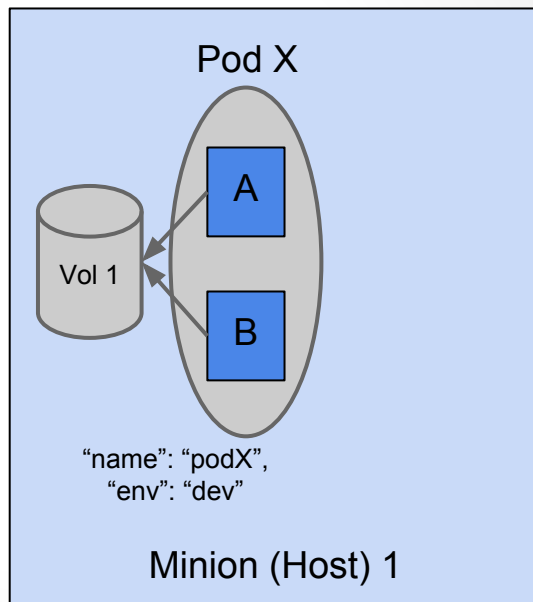
- key-value pairs
- for categorizing things

`"env": "dev", "env": "prod", "env": "qa"`

`"rel": "stable", "rel": "canary"`

`"partition": "custA", "partition": "custB"`

- label selectors

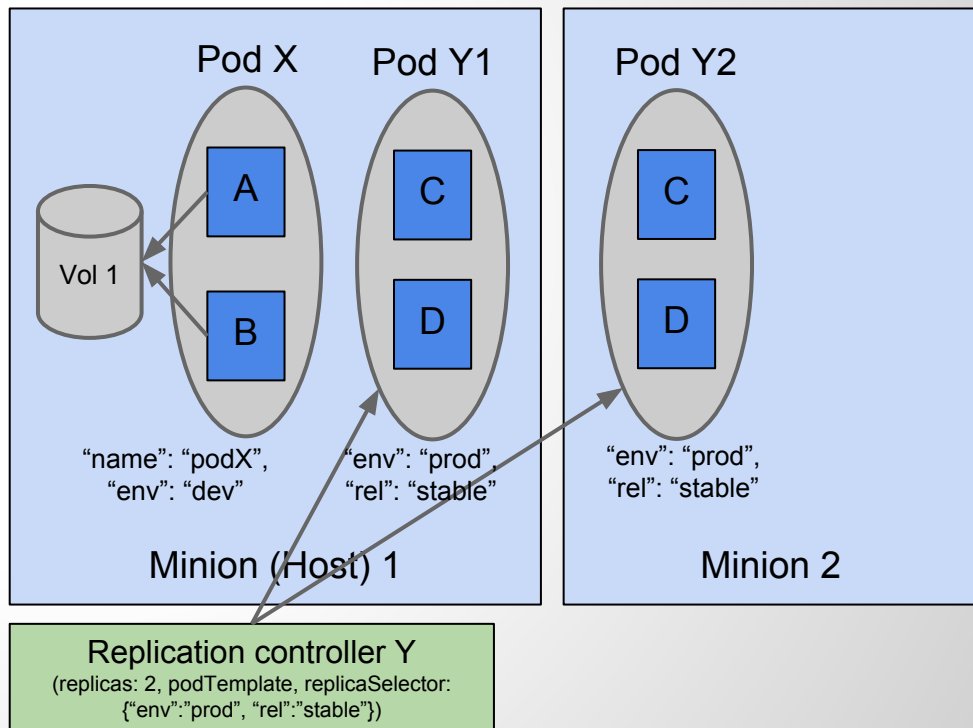


Replication controllers

```

id: replicationControllerY
kind: ReplicationController
desiredState:
  replicas: 2
  replicaSelector:
    env: prod
    rel: stable
podTemplate:
  desiredState:
    manifest:
      ...
  labels:
    - env: prod
    - rel: stable

```

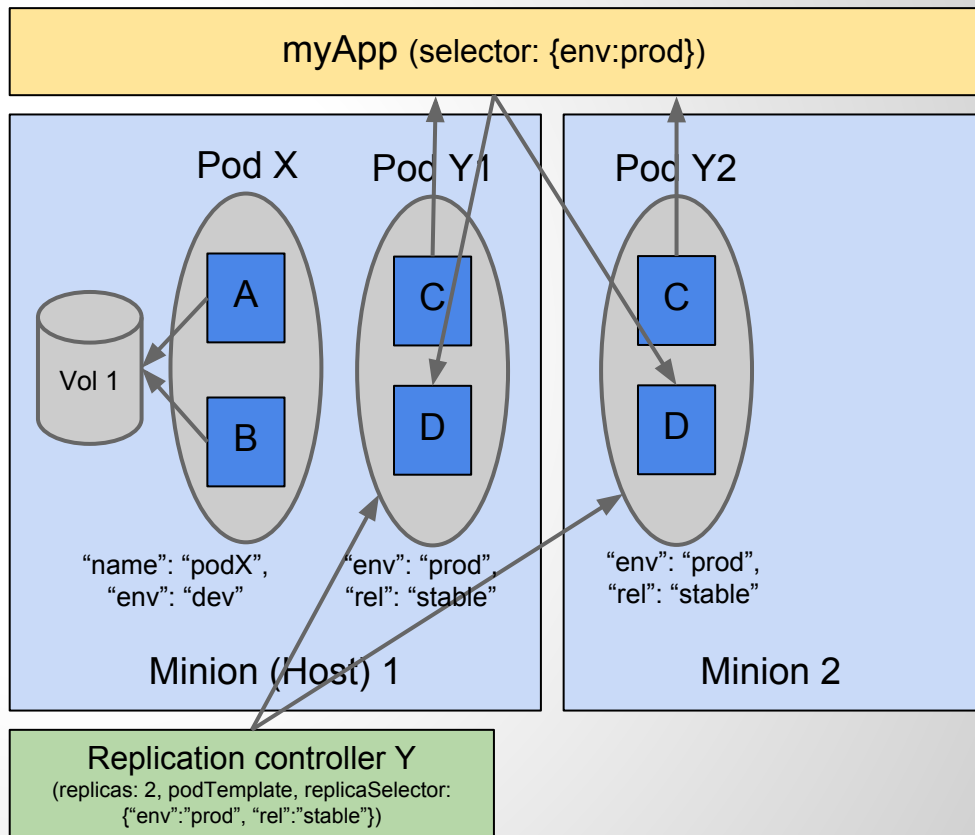


Services

```

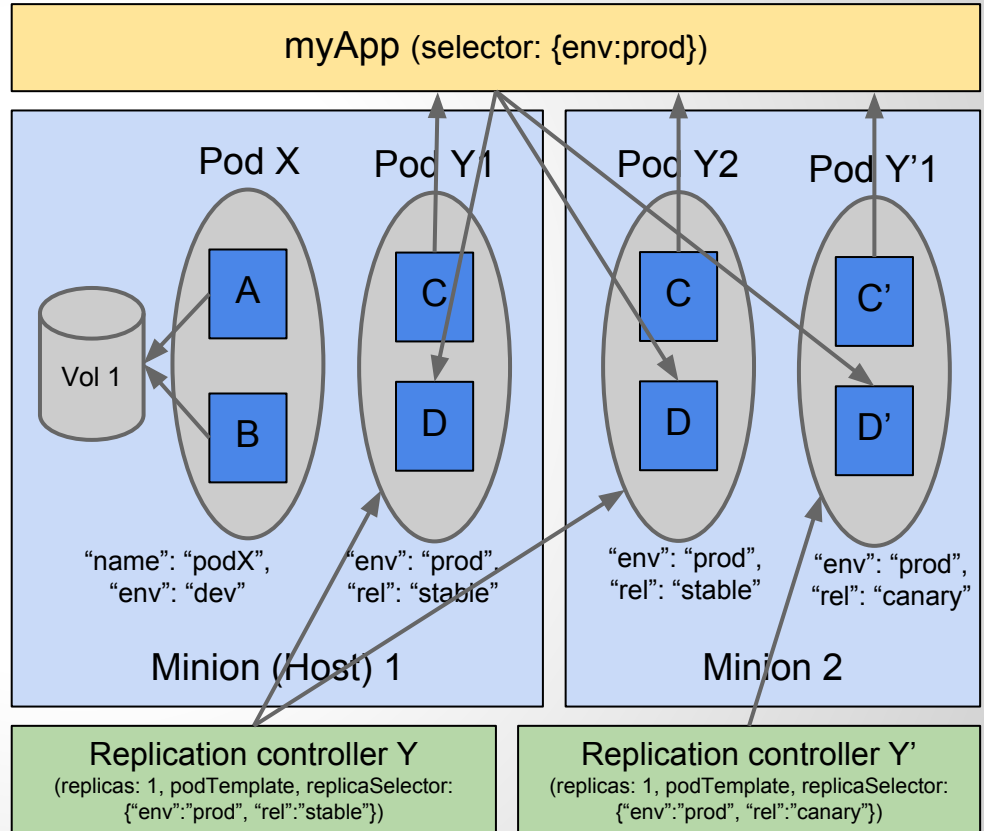
id: myApp
kind: Service
apiVersion: v1beta1
port: 1234
selector:
  env: prod
containerPort: 2345
  
```

- **Env vars:**
MYAPP_SERVICE_HOST,
MYAPP_SERVICE_PORT
- **future: DNS**



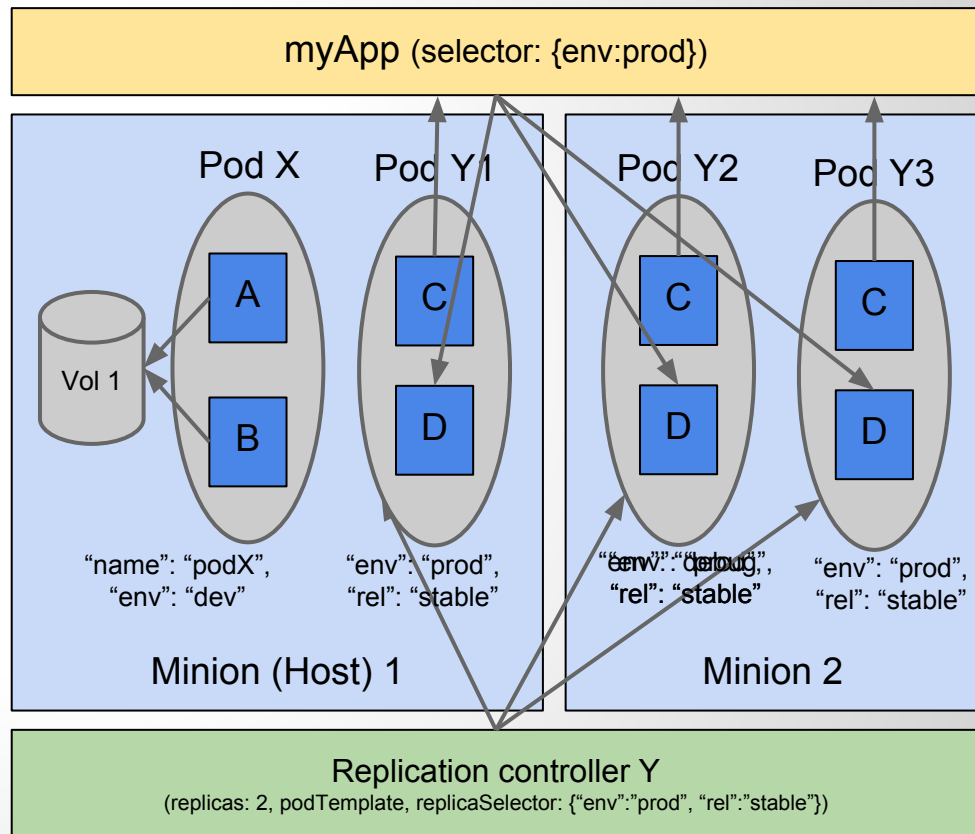
Canary releases

- New replication controller (rel:canary)
- stable controller: replicas--
- canary controller: replicas++



Debug pods

- labels of a running pod can be changed
- way of adding or removing pods from services or replication controllers



OpenShift v3

- Platform-As-A-Service
- Kubernetes extensions
 - Application Templates
 - Single JSON file for configuring Kube resources
 - Parameterizable (see [example](#))
 - Builds
 - Hosts source code in git repos
 - Performs builds and hosts private docker images
 - Kick off new builds on git-push

OpenShift v3 - Fabric8

- Fabric8 v2
 - Aggregate logging
 - LogSpout, LogStash, Fluentd, Elastic, Kibana
 - Aggregate metrics
 - kAdvisor, InfluxDB, Grafana
 - Auto-scaling
 - Should be in K8s ...
 - AScaler

OpenShift v3 - EAP

- EAP
 - KubePing
 - CE Arquillian

What is Jube?

- Issues ...
 - I'm not using Linux
 - env switching pita
 - I want to debug my app
 - local vs. remote
 - Everything is a Docker image
 - I just want to use my (Java) app

What is Jube?

- Solution?
 - Java based Kubernetes mock
 - It's all about Kubernetes REST API
 - And similar Kubernetes-like behavior
 - Replication, master election, ...
- Where is Docker?
 - Nope, no Docker here
 - Zip images
 - Lifecycle scripts

Demo (ping us after presentation ;-)

- Jube
- Fabric8
- Hawtio
- WildFly
- Cluster
- Hello servlet

Resources

- <https://github.com/jubeio/jube>
- <http://fabric8.io/>
- <https://github.com/hawtio/hawtio>
- <http://www.wildfly.org>

Resources

- <https://www.docker.com/>
- <https://www.docker.com/tryit/>
- <https://registry.hub.docker.com/>

- <http://kubernetes.io/>
- <https://github.com/GoogleCloudPlatform/kubernetes>
- <https://godoc.org/github.com/GoogleCloudPlatform/kubernetes/pkg/api>

- <https://github.com/openshift/origin>
- <https://blog.openshift.com/openshift-v3-deep-dive-docker-kubernetes/>