Jenkins
User Conference

Scaling Jenkins with Docker and Kubernetes

Carlos Sanchez
@csanchez

PARENTAL ADVISORY

BUZZWORDS AHEAD

#jenkinsconf

Jenkins
User Conference

Containers & micro services

But it is not trivial

Docker

Linux containers

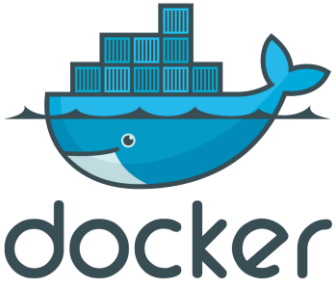
Union File System

File System

Users

Processes

Network



Kernel Sanders
@lstoll

The solution: Docker. The problem? You tell me.

OFFICIAL REPOSITORY

jenkins ★

Last pushed: 11 days ago

Repo Info Tags

Supported tags and respective `Dockerfile` links

- `latest`, `1.609.2` (*Dockerfile*)

For more information about this image and its history, please see the relevant manifest file (`library/jenkins`) in the `docker-library/official-images` GitHub repo.

Jenkins

The Jenkins Continuous Integration and Delivery server.

This is a fully functional Jenkins server, based on the Long Term Support release .

Jenkins

DOCKER PULL COMMAND

```
docker pull jenkins
```

DESCRIPTION

Official Jenkins Docker image

PUBLIC REPOSITORY

jenkinsci/jenkins ☆

Last pushed: 8 days ago

Repo Info Tags

Jenkins Continuous Integration and Delivery server.

This is a fully functional Jenkins server, based on the weekly releases .

Jenkins

Read documentation for usage

jenkinsci/jnlp-slave ☆

Last pushed: 6 days ago

Repo Info | Tags | Dockerfile | Build Details

Jenkins JNLP slave Docker image

A Jenkins slave using JNLP to establish connection.
See Jenkins Distributed builds for more info.

Usage :

```
docker run jenkinsci/jnlp-slave -url http://jenkins-server:port <secret> <slave
```

optional environment variables:

- *JENKINS_URL*: url for the Jenkins server, can be used as a replacement to -url option, or to set alternate jenkins URL
- *JENKINS_TUNNEL*: (HOST:PORT) connect to this slave host and port instead of Jenkins server, assuming this one do route TCP traffic to Jenkins master. Useful when when Jenkins runs behind a load balancer, reverse proxy, etc.

Jenkins
User Conference

Kubernetes

How would you design your infrastructure if you couldn't login? Ever.

Kelsey Hightower
CoreOS

Kubernetes

Container cluster orchestration

Docker containers across multiple hosts

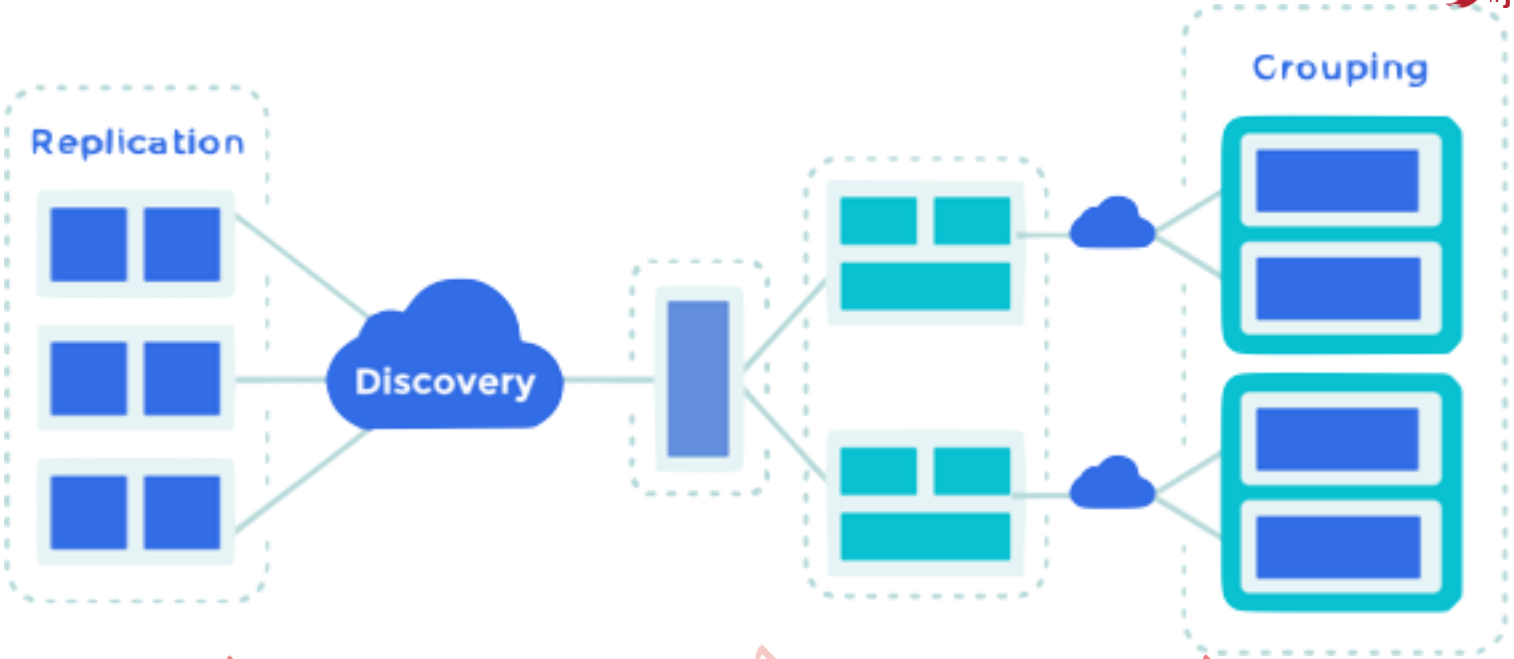
(nodes or minions)

Higher level API

Enforced state

Monitoring of endpoints


An ocean of
user containers
Kubernetes
Master
Node
Node
Node
Scheduled and packed
dynamically onto nodes


Replication
Discovery
Crouping

Master

Kubernetes API Server

- scheduling and synchronization

etcd

Kubernetes Controller Manager Server

- implements replication algorithm watching etcd

Node

Docker

Kubelet

- ensures state of Pods

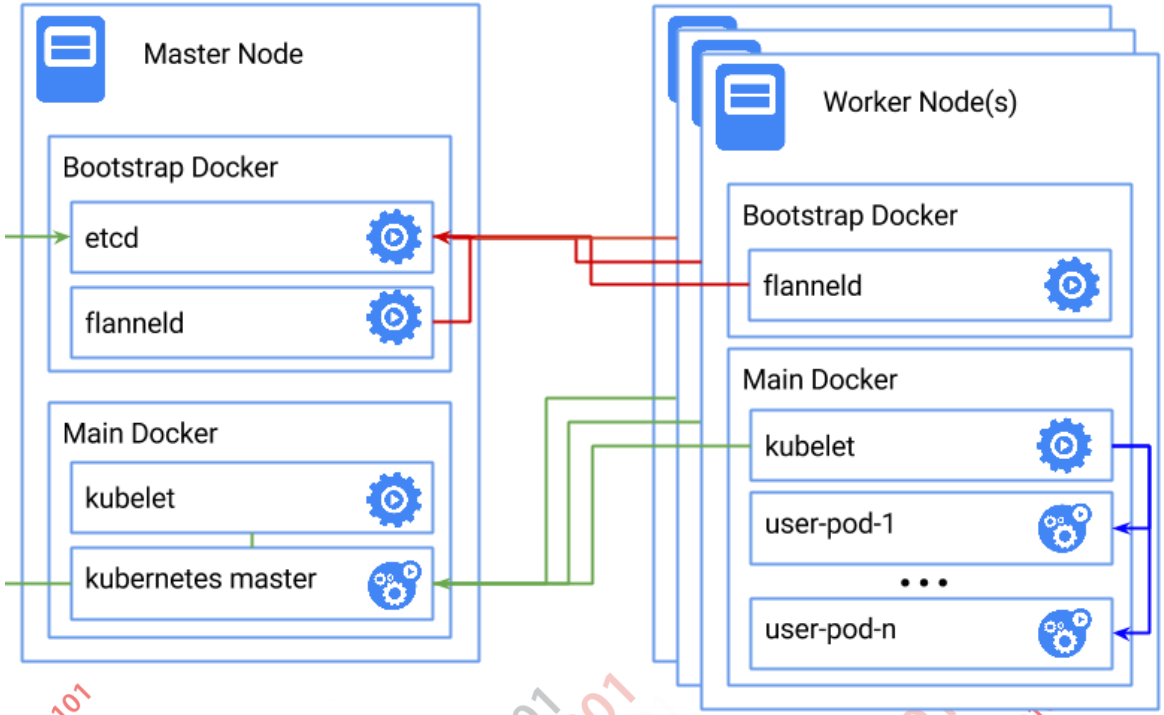
Kubernetes Proxy

- simple network proxy

etcd

SkyDNS

ElasticSearch + Kibana


Master Node
Bootstrap Docker
etcd
flanneld
Main Docker
kubelet
kubernetes master
Worker Node(s)
Bootstrap Docker
flanneld
Main Docker
kubelet
user-pod-1
...
user-pod-n

Providers

GKE

Azure

Vmware

Rackspace

oVirt

Vagrant

CloudStack

Ubuntu

oVirt

cloudstack

Cluster


```
export KUBERNETES_PROVIDER=gce
export KUBERNETES_NUM_MINIONS=2
cluster/kube-up.sh
```

Google Container Engine

Create a new container cluster

A container cluster is a managed group of uniform VM instances for hosting one or more containers. When you create a container, you must attach it to a container cluster.
Learn more

Name

cluster-1

Description (Optional)

Zone

us-central1-a

Machine type

n1-standard-1 (1 vCPU, 3.75 GB memory)

Cluster Size
Not including the Container Engine master which will be deployed in its own VM.

1

Total Cores	1 vCPU
Total Memory	3.75 GB

The Container Engine master will be using an additional VM with 1 vCPU and 3.75 GB memory.

Network

Google Container Engine


```
gcloud beta container
  --project my-project
  clusters create cluster-1
  --machine-type g1-small
  --num-nodes 2
```

Tectonic by CoreOS

Jenkins User Conference

#jenkinsconf

TECTONIC Services Replication Controllers Pods Machines More My Account

+ Create a New Replication Controller

Replication Controllers

Replication Controller List

Filter replication controllers by name...

CONTROLLER ID	CONTROLLER LABELS	STATUS	POD SELECTOR	DESIRED COUNT
RC elasticsearch	app = search, type = storage, version = 1.5.0	5 of 5 pods	app = search, type = storage	5
kibana	app = search, type = ui, version = 4.0.2	1 of 1 pods	app = search, type = ui	1
logstash	app = search, type = collector, version = 1.4.2	1 of 1 pods	app = storage, type = collector	1

POD ID	POD LABELS	STATUS	CONTAINERS	MACHINE
P elasticsearch-0133o	app = search, type = storage, version = 1.5.0	Running	1	10.0.0.207
P elasticsearch-fplln	app = search, type = storage, version = 1.5.0	Running	1	10.0.0.210
P elasticsearch-gziey	app = search, type = storage, version = 1.5.0	Running	1	10.0.0.209
P elasticsearch-ls6k1	app = search, type = storage, version = 1.5.0	Running	1	10.0.0.208
P elasticsearch-oh43e	app = search, type = storage, version = 1.5.0	Running	1	10.0.0.206

Documentation

TECTONIC by CoreOS

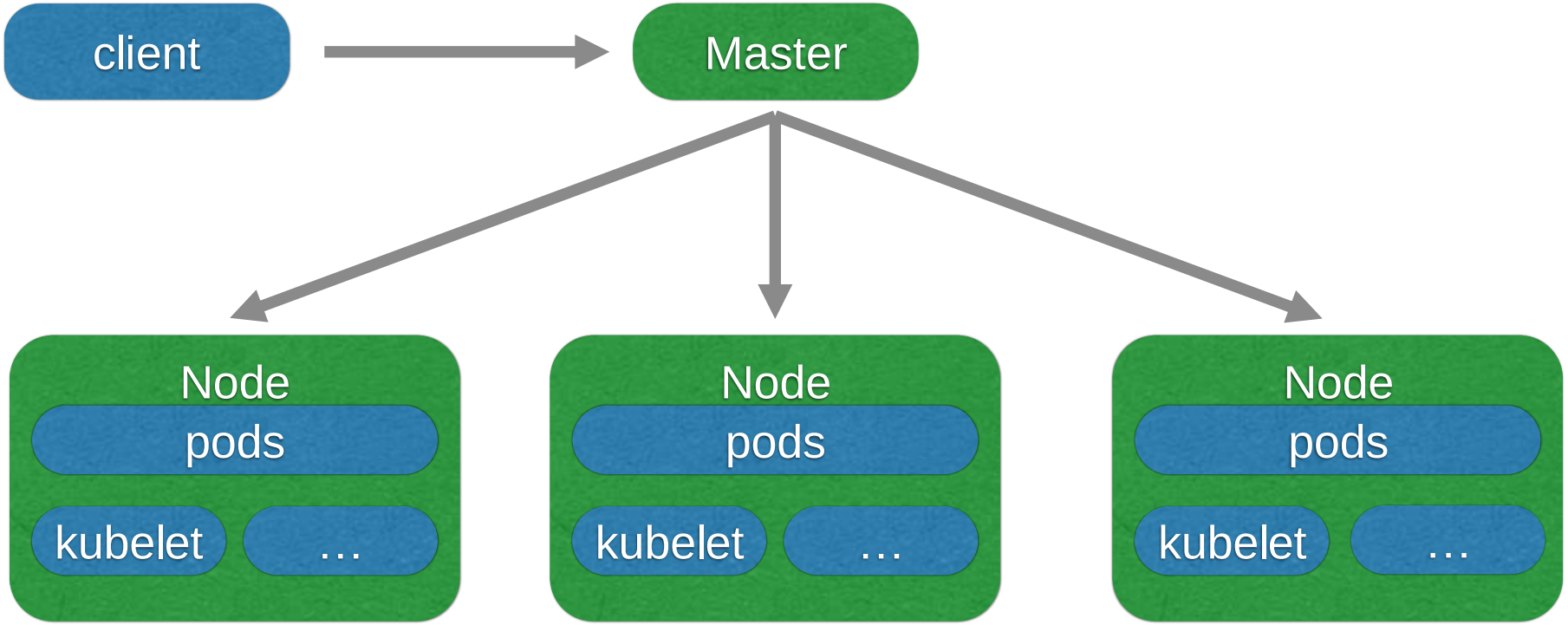
```
---
kind: "Node"
apiVersion: "v1"
metadata:
  name: "127.0.0.1"
  selfLink: "/api/v1/nodes/127.0.0.1"
  uid: "8c3192d2-48ef-11e5-8d27-bae1092286ff"
  resourceVersion: "55118"
  creationTimestamp: "2015-08-22T17:02:19Z"
  labels:
    kubernetes.io/hostname: "127.0.0.1"
spec:
  externalID: "127.0.0.1"
status:
  capacity:
    cpu: "0"
    memory: "0"
    pods: "40"
  conditions:
    -
      type: "Ready"
      status: "True"
      lastHeartbeatTime: "2015-08-26T19:38:20Z"
      lastTransitionTime: "2015-08-26T12:18:48Z"
      reason: "kubelet is posting ready status"
  addresses:
    -
      type: "LegacyHostIP"
      address: "127.0.0.1"
  nodeInfo:
    ...
```


Node

Pod

- Group of colocated containers
- Same network namespace/IP
- Environment variables
- Shared volumes
  - host mounted
  - empty volumes
  - GCE data disks
  - AWS EBS volumes

Pods


client
Master
Node
pods
kubelet
…
Node
pods
kubelet
…
Node
pods
kubelet
…

```yaml
---
kind: "Pod"
apiVersion: "v1"
metadata:
  name: "jenkins"
  labels:
    name: "jenkins"
spec:
  containers:
    -
      name: "jenkins"
      image: "csanchez/jenkins-swarm:1.609.2"
      ports:
        -
          containerPort: 8080
          hostPort: 8090
        -
          containerPort: 50000
          hostPort: 50000
      volumeMounts:
        -
          name: "jenkins-data"
          mountPath: "/var/jenkins_home"
  volumes:
    -
      name: "jenkins-data"
      hostPath:
        path: "/home/docker/jenkins"
```

Pod

Replication controller

Ensure a number of pods are running

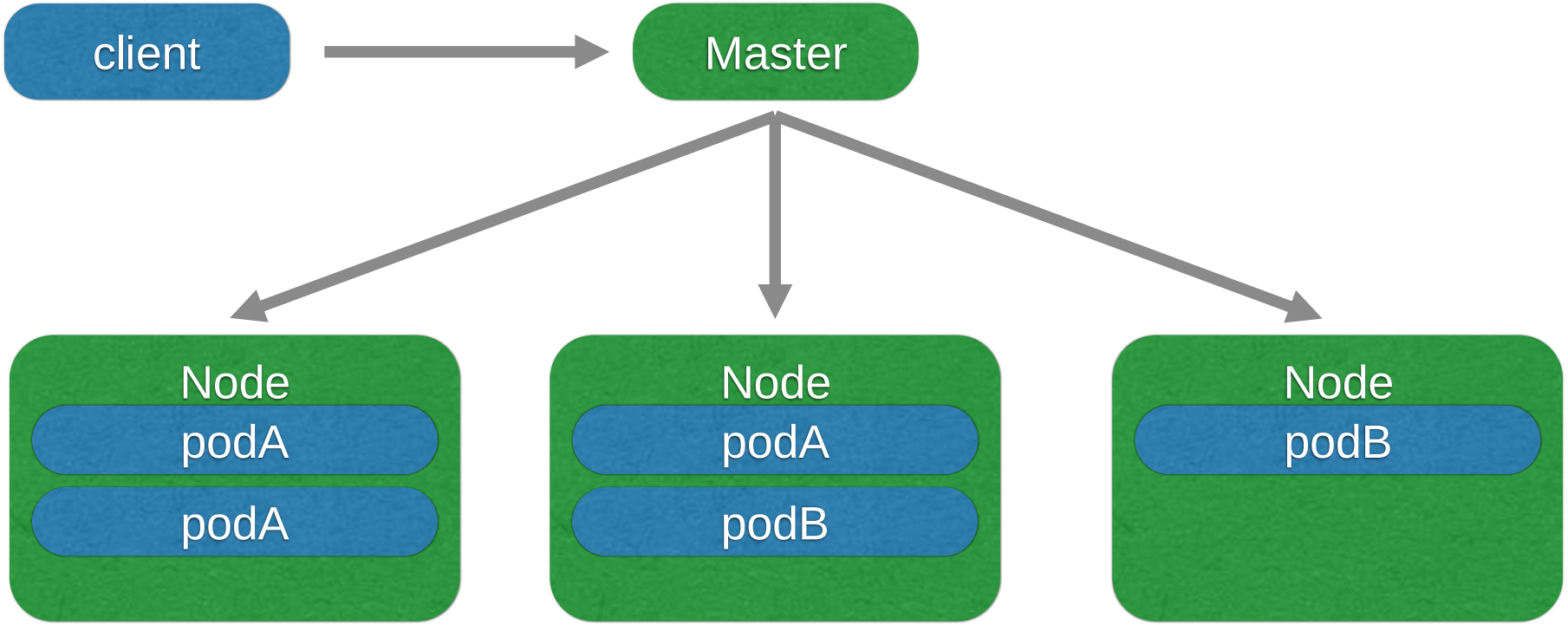
Pod templates

Rolling update

Replication controllers

Jenkins User Conference

#jenkinsconf



@DEVOPS_BORAT

DevOps Borat

To make error is human. To propagate error to all server in automatic way is #devops.

Replication controller


```yaml
---
apiVersion: "v1"
kind: "ReplicationController"
metadata:
  name: "jenkins-slave"
  labels:
    name: "jenkins-slave"
spec:
  replicas: 1
  template:
    metadata:
      name: "jenkins-slave"
      labels:
        name: "jenkins-slave"
    spec:
      containers:
        -
          name: "jenkins-slave"
          image: "csanchez/jenkins-swarm-slave:2.0"
          env:
          command:
```

Replication controller


```
command:
  - "/usr/local/bin/jenkins-slave.sh"
  - "-master"
  - "http://$(JENKINS_SERVICE_HOST):$(JENKINS_SERVICE_PORT)"
  - "-tunnel"
  - "$(JENKINS_SLAVE_SERVICE_HOST):$(JENKINS_SLAVE_SERVICE_PORT)"
  - "-username"
  - "jenkins"
  - "-password"
  - "jenkins"
  - "-executors"
  - "1"
```

Services

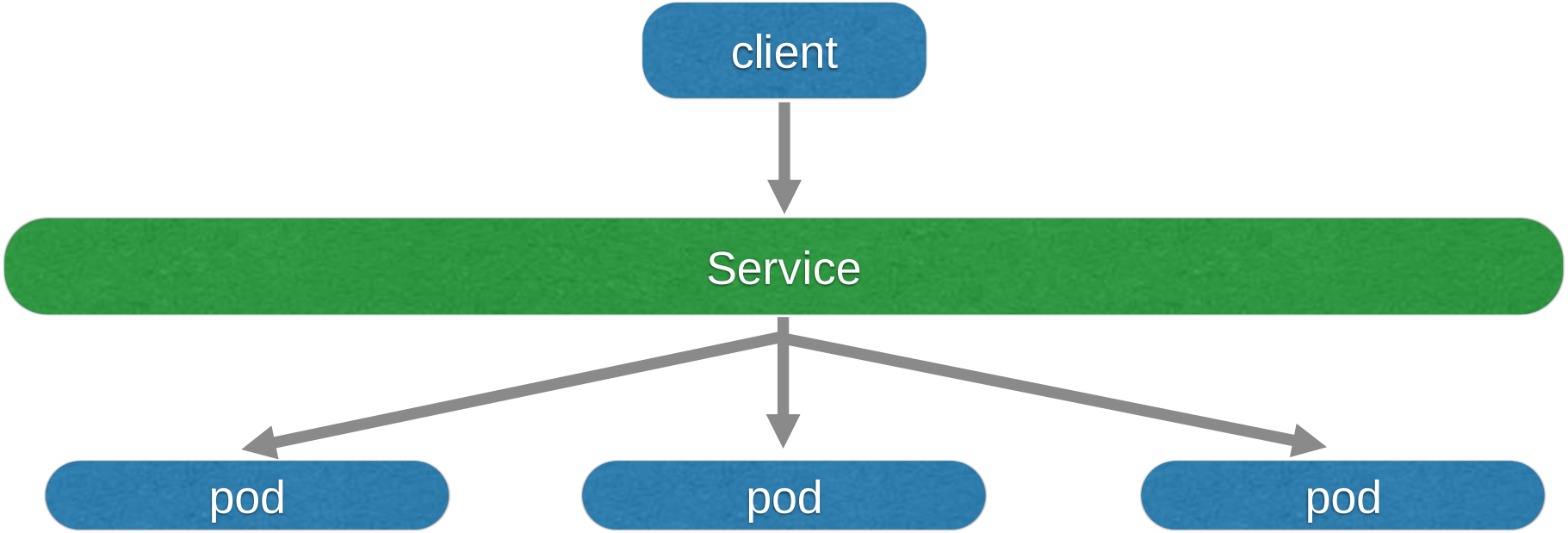
Pod discovery

IP per service

Route to pods selected with labels

Can create a load balancer in GCE and AWS

Services



```yaml
---
apiVersion: "v1"
kind: "Service"
metadata:
  name: "jenkins"
spec:
  type: "NodePort"
  selector:
    name: "jenkins"
  ports:
    -
      name: "http"
      port: 8090
      nodePort: 32080
      protocol: "TCP"
```

```yaml
---
apiVersion: "v1"
kind: "Service"
metadata:
  name: "jenkins-slave"
spec:
  type: "NodePort"
  selector:
    name: "jenkins"
  ports:
    -
      name: "http"
      port: 50000
      nodePort: 32050
      protocol: "TCP"
```

Services

Networking

all containers can communicate with all other containers without NAT

all nodes can communicate with all containers (and vice-versa) without NAT

the IP that a container sees itself as is the same IP that others see it as

Containers in a Pod can talk using localhost

Networking

Every machine in the cluster is assigned a full subnet

ie. node A 10.0.1.0/24 and node B 10.0.2.0/24

Simpler port mapping

Only supported by GCE

CoreOS flannel

- Creates an overlay network in other providers

#jenkinsconf

Jenkins
User Conference

Related projects

Docker Machine

Provision Docker engines

- VirtualBox, replaces boot2docker !
- Amazon EC2
- Microsoft Azure
- Google Compute Engine
- OpenStack
- Rackspace
- VMware
- …

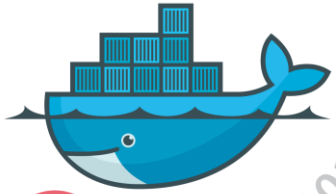
Docker Compose

Orchestration of multi-container apps

Based on Fig

Defined by:

- containers
- configuration
- links
- volumes



docker

#jenkinsconf

Jenkins
User Conference

Kubernetes and Jenkins

Kubernetes cluster with docker-compose

#jenkinsconf

```
# Docker Compose definition for a one node Kubernetes cluster
# Based on Docker Cookbook example
# https://github.com/how2dock/docbook/ch05/docker
etcd:
  image: kubernetes/etcd:2.0.5.1
  net: "host"
  command: /usr/local/bin/etcd --addr=127.0.0.1:4001 --bind-addr=0.0.0.0:4001 --data-dir=/var/etcd/data
master:
  image: gcr.io/google_containers/hyperkube:v1.0.1
  net: "host"
  volumes:
    - /var/run/docker.sock:/var/run/docker.sock
  command: /hyperkube kubelet --api_servers=http://localhost:8080 --v=2 --address=0.0.0.0 --enable_server --hostname_override=127.0.0.1 --
config=/etc/kubernetes/manifests
proxy:
  image: gcr.io/google_containers/hyperkube:v1.0.1
  net: "host"
  privileged: true
  command: /hyperkube proxy --master=http://127.0.0.1:8080 --v=2
```

https://github.com/carlossg/kubernetes-jenkins

Jenkins master pod


```
kind: "Pod"
 apiVersion: "v1"
 metadata:
   name: "jenkins"
   labels:
     name: "jenkins"
 spec:
   containers:
     -
       name: "jenkins"
       image: "csanchez/jenkins-swarm:1.609.2"
       ports:
         -
           containerPort: 8080
           hostPort: 8090
         -
           containerPort: 50000
           hostPort: 50000
```

https://github.com/carlossg/kubernetes-jenkins

storage options


```
      volumeMounts:
        -
          name: "jenkins-data"
          mountPath: "/var/jenkins_home"
    volumes:
     -
      name: "jenkins-data"
      hostPath:
        path: "/home/docker/jenkins"
#       gcePersistentDisk:
#         pdName: my-data-disk
#         fsType: ext4
#       awsElasticBlockStore:
#         volumeID: aws://<availability-zone>/<volume-id>
#         fsType: ext4
```

https://github.com/carlossg/kubernetes-jenkins

Jenkins master services


```yaml
---
apiVersion: “v1"
kind: "Service"
metadata:
  name: "jenkins"
spec:
  type: "NodePort"
  selector:
    name: "jenkins"
  ports:
    -
      name: "http"
      port: 8090
      nodePort: 32080
      protocol: “TCP"
```

```yaml
---
apiVersion: "v1"
kind: "Service"
metadata:
  name: "jenkins"
spec:
  type: "NodePort"
  selector:
    name: "jenkins"
  ports:
    -
      name: "http"
      port: 8090
      nodePort: 32080
      protocol: "TCP"
```

https://github.com/carlossg/kubernetes-jenkins

Jenkins slaves replication pool


```yaml
---
apiVersion: "v1"
kind: "ReplicationController"
metadata:
  name: "jenkins-slave"
  labels:
    name: "jenkins-slave"
spec:
  replicas: 1
  template:
    metadata:
      name: "jenkins-slave"
      labels:
        name: "jenkins-slave"
    spec:
      containers:
```

https://github.com/carlossg/kubernetes-jenkins

Jenkins slaves replication pool


```
-
  name: "jenkins-slave"
  image: "csanchez/jenkins-swarm-slave:2.0"
  command:
    - "/usr/local/bin/jenkins-slave.sh"
    - "-master"
    - "http://$(JENKINS_SERVICE_HOST):$(JENKINS_SERVICE_PORT)"
    - "-tunnel"
    - "$(JENKINS_SLAVE_SERVICE_HOST):$(JENKINS_SLAVE_SERVICE_PORT)"
    - "-username"
    - "jenkins"
    - "-password"
    - "jenkins"
    - "-executors"
    - "1"
```

https://github.com/carlossg/kubernetes-jenkins

Jenkins cluster in Kubernetes


```
kubectl get nodes
kubectl create --validate -f pod.yml
kubectl get pods
kubectl create --validate -f service-http.yml
kubectl create --validate -f service-slave.yml
kubectl get services
kubectl create --validate -f replication.yml
kubectl get pods
kubectl scale replicationcontrollers --replicas=20 jenkins-slave
```

https://github.com/carlossg/kubernetes-jenkins

#jenkinsconf

Jenkins
User Conference

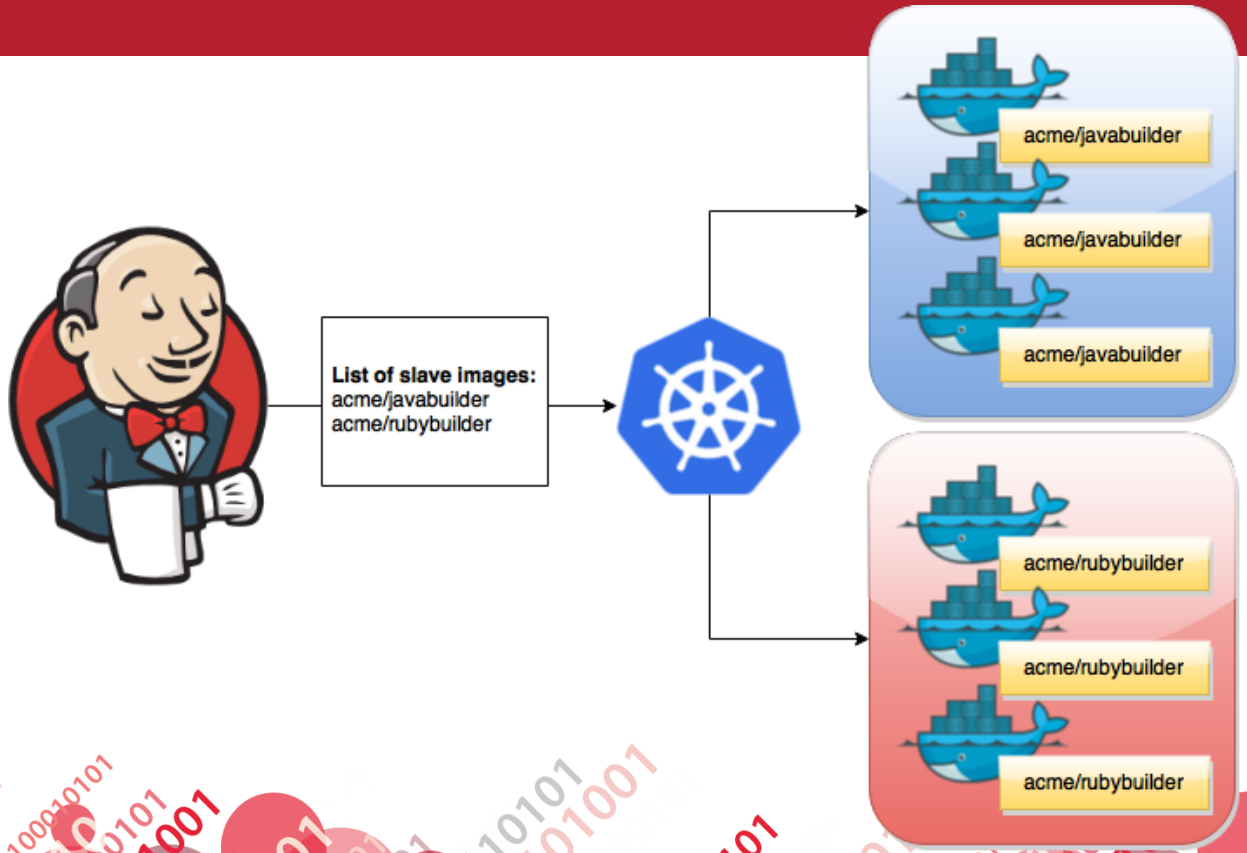
Kubernetes Jenkins plugin

Kubernetes Jenkins plugin

As a plugin

on demand slaves

https://github.com/jenkinsci/kubernetes-plugin


List of slave images:
acme/javabuilder
acme/rubybuilder
acme/javabuilder
acme/javabuilder
acme/javabuilder
acme/rubybuilder
acme/rubybuilder
acme/rubybuilder

Kubernetes Jenkins plugin

pods, not replication controllers

Jenkins Cloud API

Fabric8 Java API

Workflow support

Jenkins
User Conference

#jenkinsconf

Kubernetes

Name

Kubernetes URL: http://localhost:8080

Kubernetes server certificate key

Credentials: - none - | Add

Test Connection

Kubernetes Namespace: default

Jenkins URL: http://192.168.1.104:10000/jenkins

Jenkins tunnel

Connection Timeout: 5

Read Timeout: 15

Container Cap: 10

Images

Kubernetes Pod Template

Name

Labels

Docker image: csanchez/jenkins-slave

Jenkins slave root directory: /home/jenkins

Command to run slave agent

Arguments to pass to the command

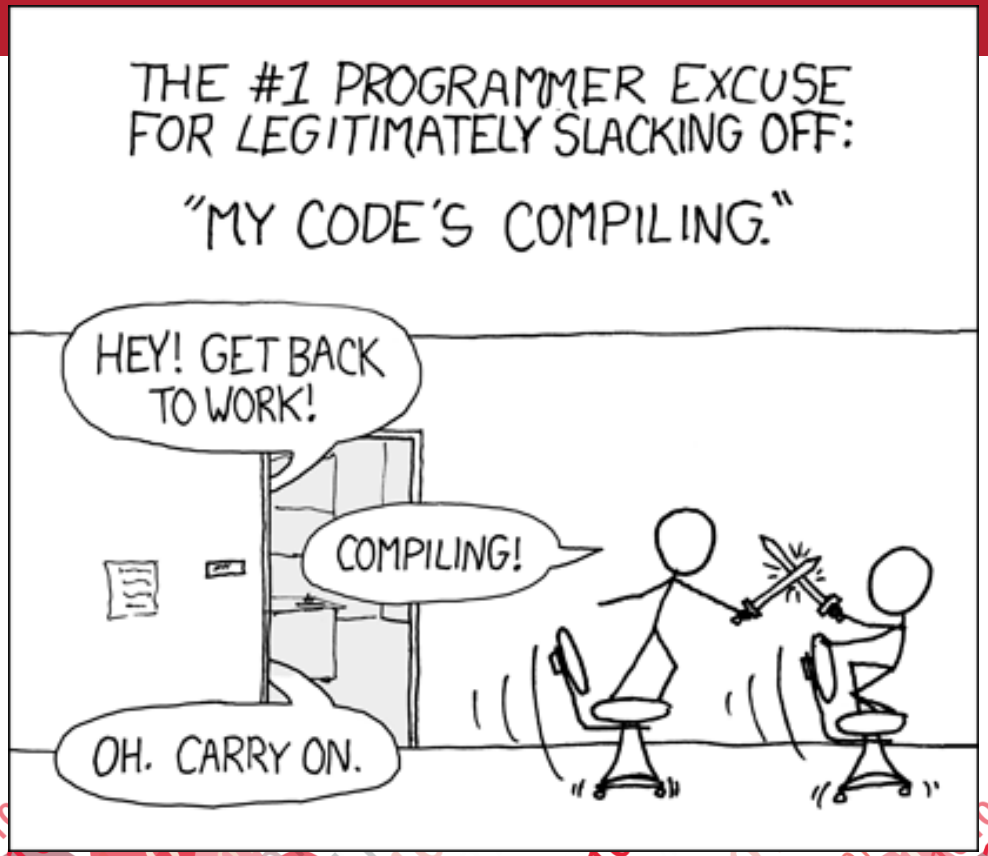
Max number of instances

Run in privileged mode

Delete Template

Add Pod Template

List of Images to be launched as slaves


THE #1 PROGRAMMER EXCUSE FOR LEGITIMATELY SLACKING OFF:
"MY CODE'S COMPILING."
HEY! GET BACK TO WORK!
COMPILING!
OH. CARRY ON.

Roadmap

Stable API

When Kubernetes Java lib is stable

Using new Jenkins Cloud/Containers APIs

Example code and slides

Available at

http://slideshare.csanchez.org

https://github.com/carlossg/kubernetes-jenkins

http://blog.csanchez.org

Jenkins
User Conference

Thanks!

Thanks to our Sponsors!

#jenkinsconf

CloudBees
The Enterprise Jenkins Company

perfecto mobile
Pivotal
XebiaLabs
Deliver Faster

SOASTA
Performance is Everything
vmware
JFrog

BlazeMeter
IBM
Sonatype
PERFORCE

BLACKDUCK
info stretch
ElasticBox
CPRIME

GerritForge
keynote
MOBILE LABS

sauce LABS
Xamarin

PLATFORM9
{} Codenvy
DevOps.com
DZone
SERENA