



Scaling Docker with Kubernetes

Carlos Sanchez
@csanchez

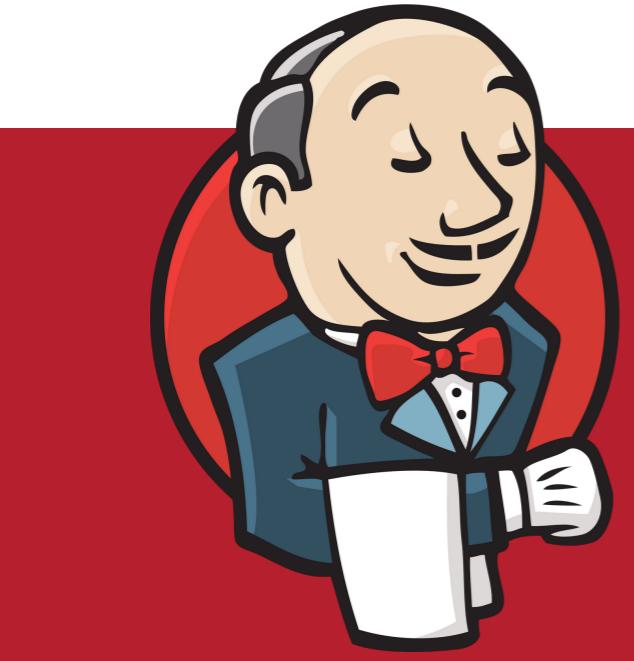
About

Senior Software Engineer @ CloudBees

Author of Jenkins Kubernetes plugin

Long time OSS contributor at Apache
Maven, Eclipse, Puppet,...





Containers & micro services

Docker

Linux containers

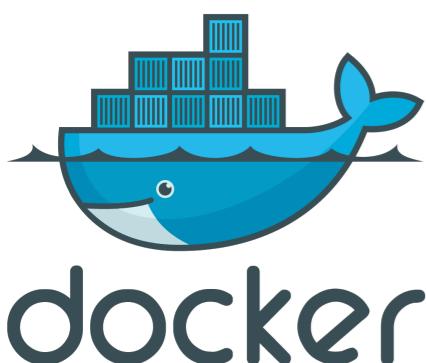
Union File
System

File System

Users

Processes

Network





But it is not trivial



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](#)

PUBLIC | AUTOMATED BUILD

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 Jenkins runs behind a load balancer, reverse proxy, etc.



Kubernetes

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

**Kelsey Hightower
CoreOS**

Kubernetes

Container cluster orchestration

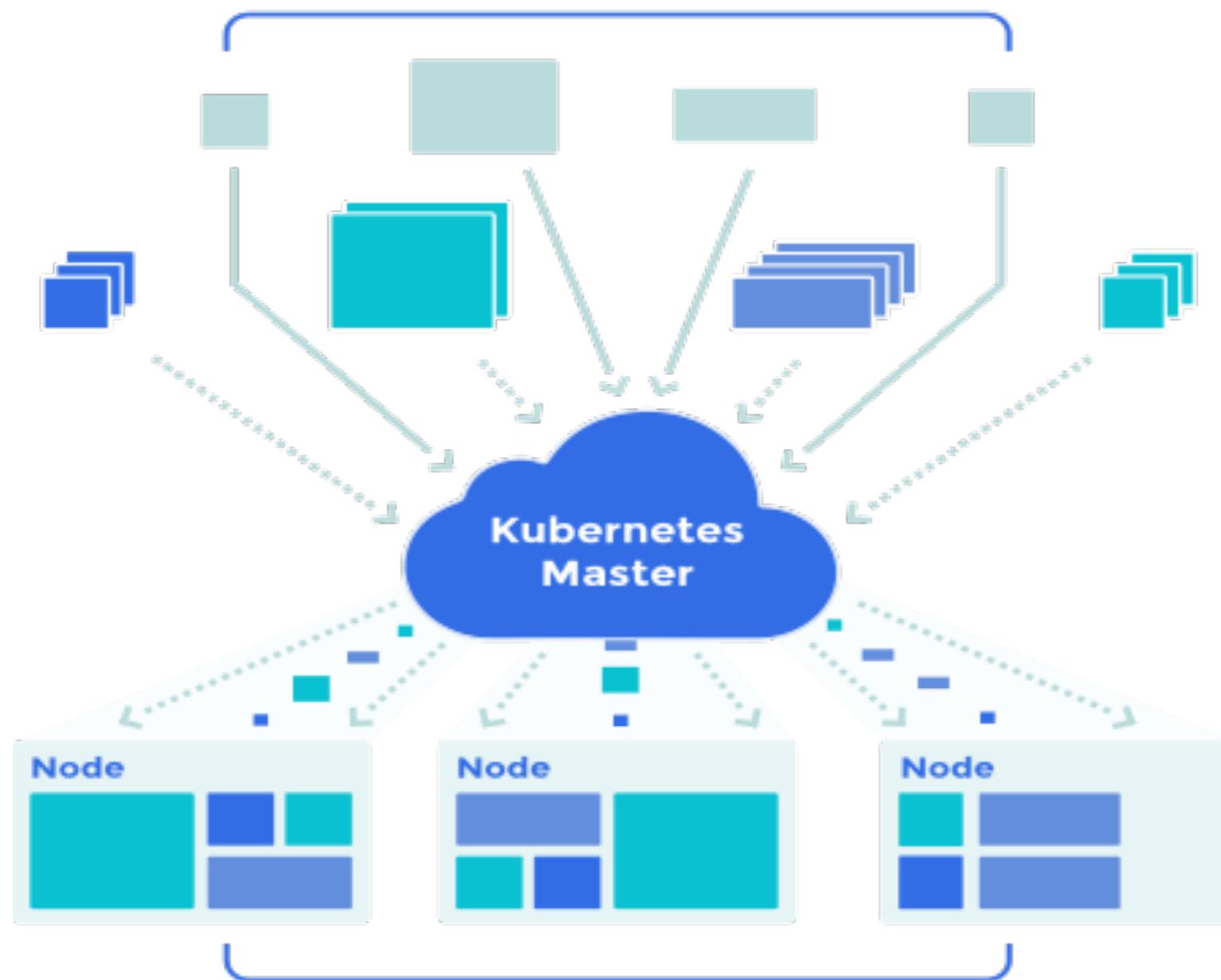
Docker containers across multiple hosts
(nodes or minions)

Higher level API

Enforced state

Monitoring of endpoints

An ocean of
user containers



Scheduled and packed
dynamically onto nodes

Master

Kubernetes API Server

scheduling and synchronization

etcd

Kubernetes Controller Manager Server

implements replication algorithm watching
etcd

Node

Docker

Kubelet

ensures state of Pods

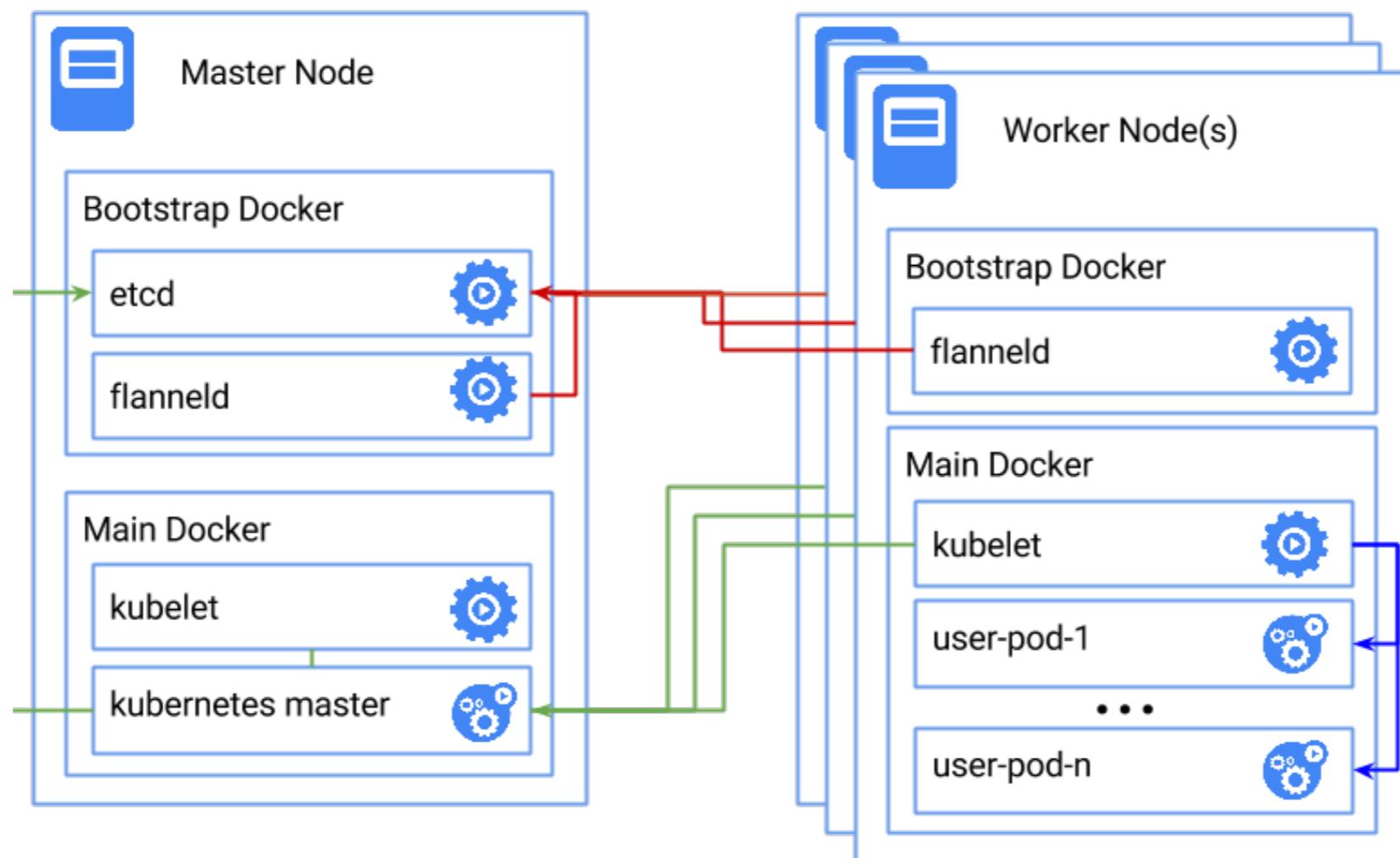
Kubernetes Proxy

simple network proxy

etcd

SkyDNS

ElasticSearch + Kibana



Providers

GKE

Azure

Vmware

Rackspace

oVirt

Vagrant

CloudStack

Ubuntu



oVirt



cloudstack



CloudBees

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

Node

```
---
```

```
kind: "Node"
apiVersion: "v1"
metadata:
  name: "gke-kubernetes-jenkins-e46fdcaa5-node-5gvr"
  selfLink: "/api/v1/nodes/gke-kubernetes-jenkins-e46fdcaa5-
node-5gvr"
  uid: "39f02f2c-73e8-11e5-a004-42010af00137"
  resourceVersion: "361180"
  creationTimestamp: "2015-10-16T09:28:14Z"
  labels:
    kubernetes.io/hostname: "gke-kubernetes-jenkins-e46fdcaa5-
node-5gvr"
spec:
  podCIDR: "10.172.0.0/24"
  externalID: "13482309745852170188"
  providerID: "gce://prefab-backbone-109611/us-central1-f/gke-
kubernetes-jenkins-e46fdcaa5-node-5gvr"
status:
  capacity:
    cpu: "1"
    memory: "3800808Ki"
    pods: "40"
  conditions:
  -
    type: "Ready"
    status: "True"
    lastHeartbeatTime: "2015-10-19T08:08:46Z"
    lastTransitionTime: "2015-10-16T09:28:51Z"
    reason: "kubelet is posting ready status"
  addresses:
  -
    type: "InternalIP"
    address: "10.240.0.4"
  -
    type: "ExternalIP"
    address: "104.154.35.119"
nodeInfo: ...
```

Pod

Group of colocated containers

Same network namespace/IP

Environment variables

Shared volumes

- host mounted

- empty volumes

- GCE data disks

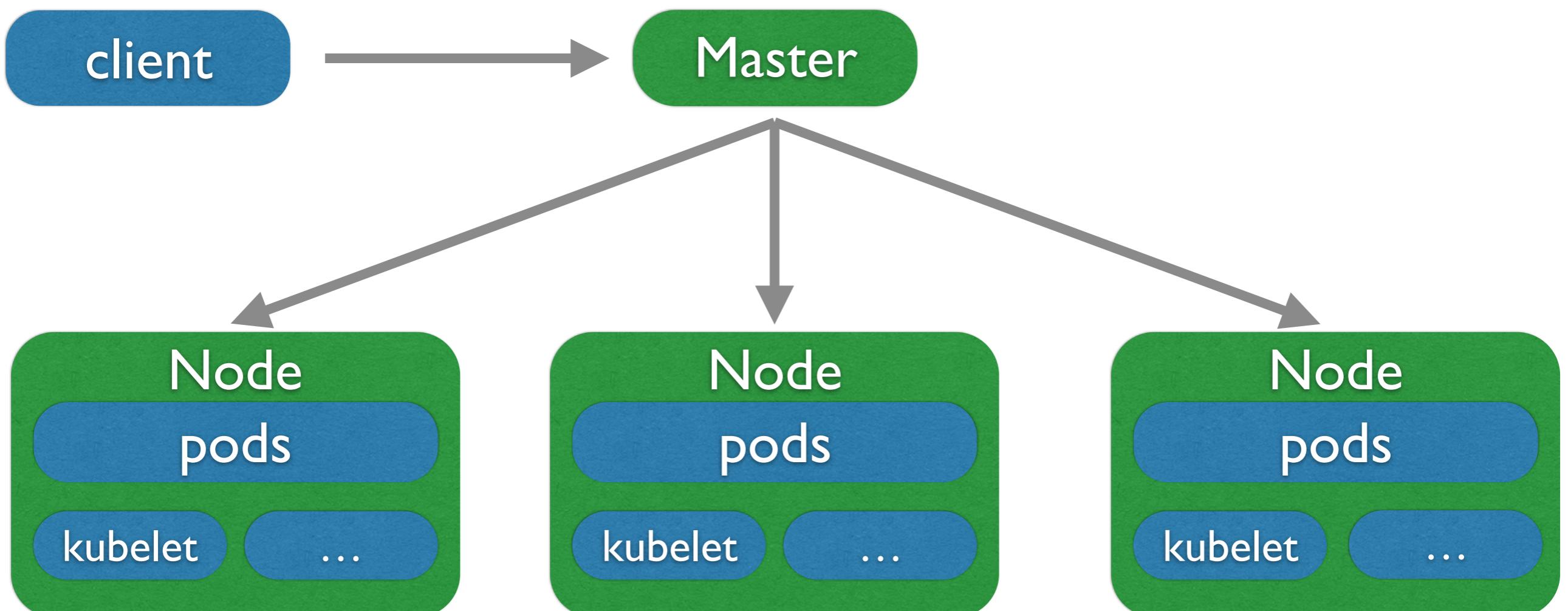
- AWS EBS volumes

- nfs

- glusterfs

- secrets

Pods



Pod

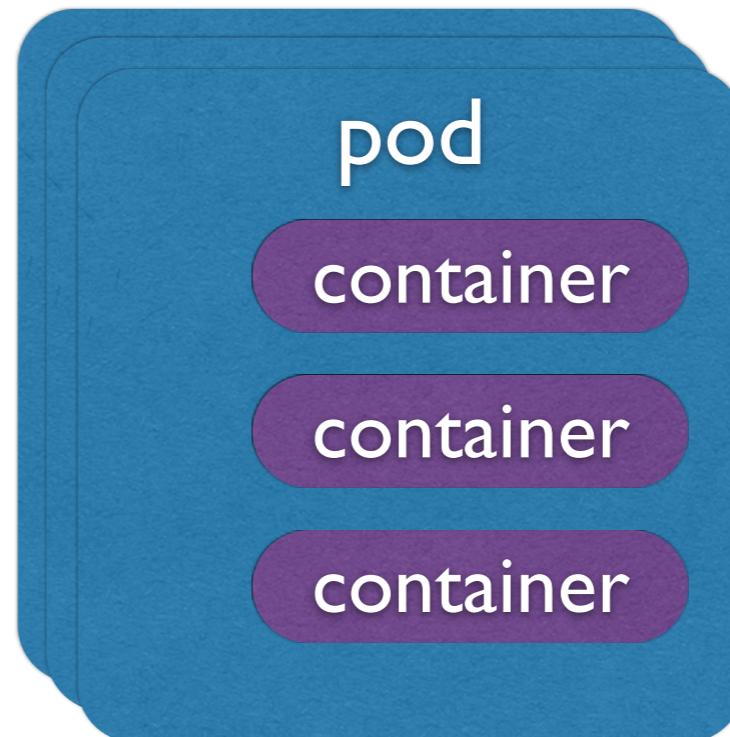
```
kind: "Pod"
apiVersion: "v1"
metadata:
  name: "jenkins"
  labels:
    name: "jenkins"
spec:
  containers:
    -
      name: "jenkins"
      image: "csanchez/jenkins-swarm:1.625.1-for-volumes"
      ports:
        - containerPort: 8080
        - containerPort: 50000
      volumeMounts:
        - name: "jenkins-data"
          mountPath: "/var/jenkins_home"
  volumes:
    - name: "jenkins-data"
      hostPath:
        path: "/home/docker/jenkins"
```

Replication controller

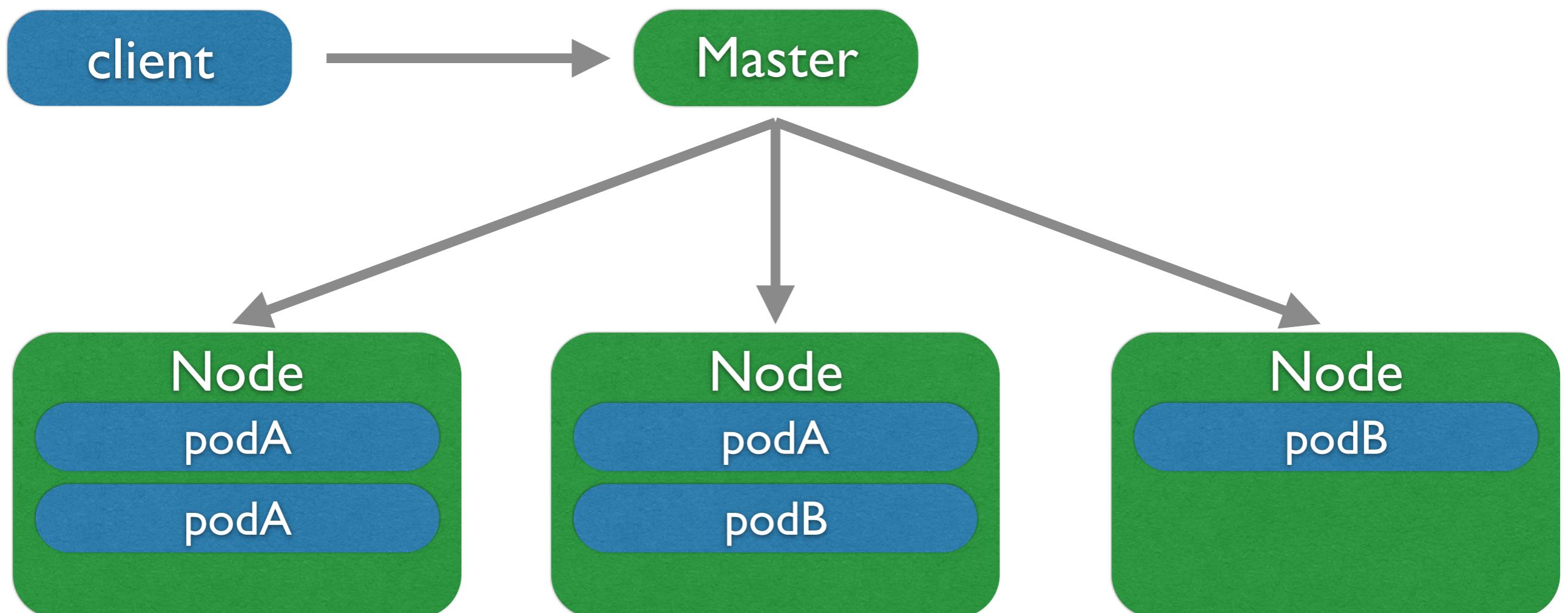
Ensure a number of pods are running

Pod templates

Rolling update



Replication controllers



CloudBees®



@DEVOPS_BORAT

DevOps Borat

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

Replication controller

```
apiVersion: "v1"
kind: "ReplicationController"
metadata:
  name: "jenkins"
  labels:
    name: "jenkins"
spec:
  replicas: 1
  template:
    metadata:
      name: "jenkins"
      labels:
        name: "jenkins"
    spec:
      containers:
        -
          name: "jenkins"
          image: "csanchez/jenkins-swarm:1.625.1-for-volumes"
          ports:
            - containerPort: 8080
            - containerPort: 50000
          volumeMounts:
            - name: "jenkins-data"
              mountPath: "/var/jenkins_home"
      volumes:
        - name: "jenkins-data"
          hostPath:
            path: "/var/jenkins"
```

Services

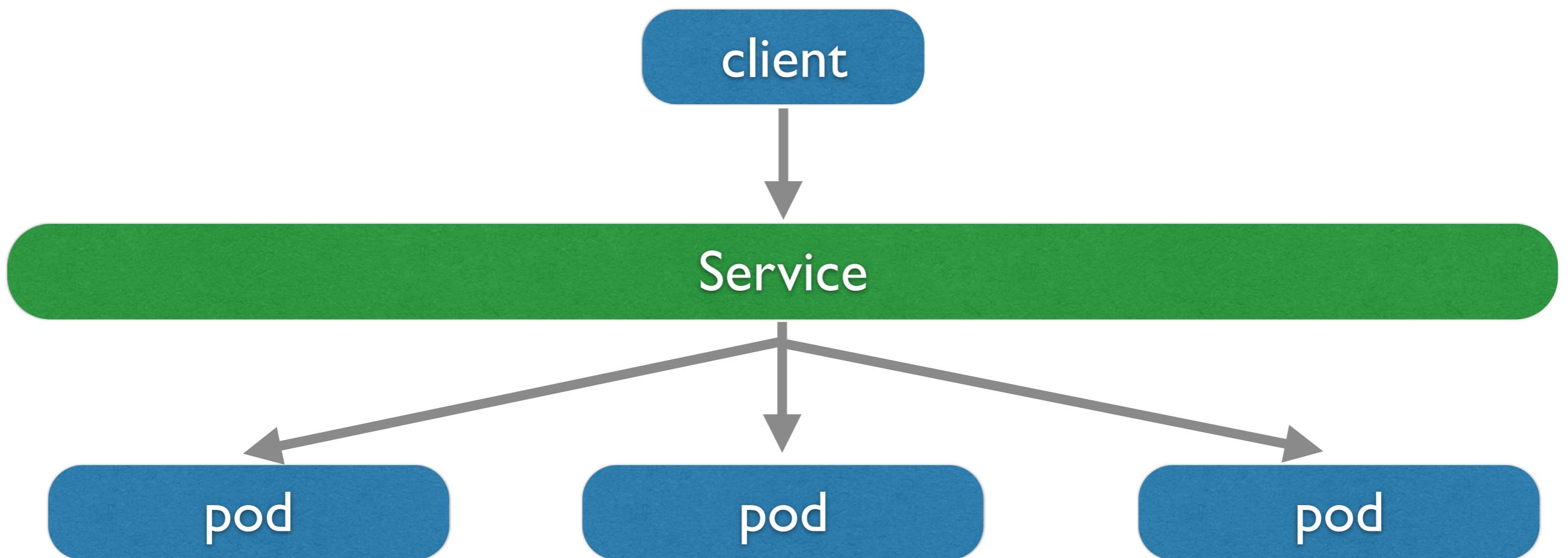
Pod discovery

IP per service

Route to pods selected with labels

Can create a load balancer in GCE and AWS

Services



```
apiVersion: "v1"
kind: "Service"
metadata:
  name: "jenkins"
spec:
  type: "LoadBalancer"
  selector:
    name: "jenkins"
  ports:
    -
      name: "http"
      port: 80
      targetPort: 8080
      protocol: "TCP"
    -
      name: "slave"
      port: 50000
      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

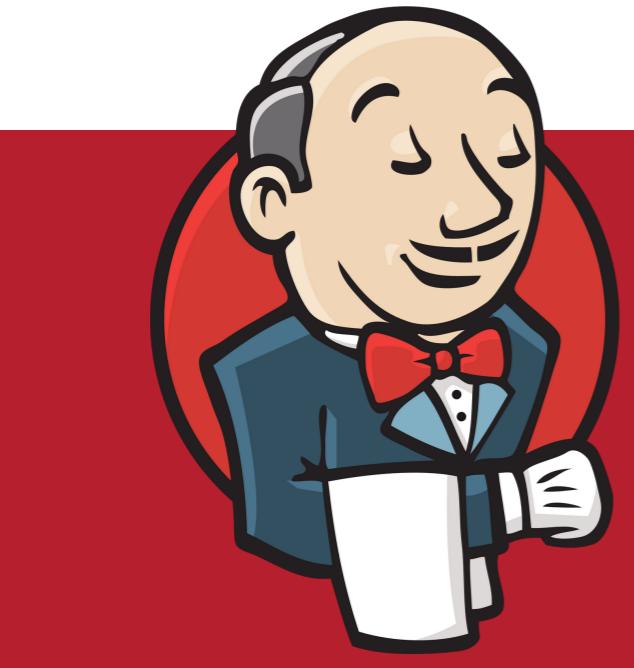
Only supported by GCE

CoreOS flannel

Creates an overlay network in other providers

Kubernetes cluster with docker-compose

```
# 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
```



Related projects

Docker Machine

Provision Docker engines

VirtualBox, replaces boot2docker !

Amazon EC2

Microsoft Azure

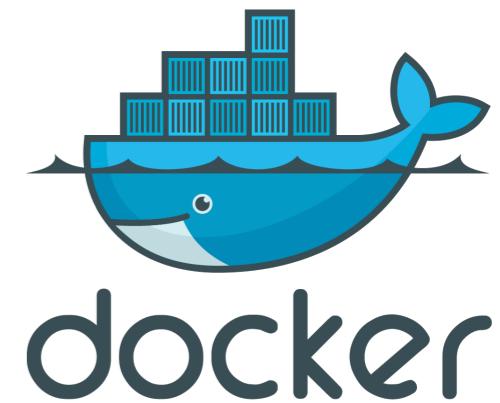
Google Compute Engine

OpenStack

Rackspace

VMware

....



Docker Swarm

Clustering for Docker containers

Using the same API

Integrates with Mesos / Mesosphere

And planned

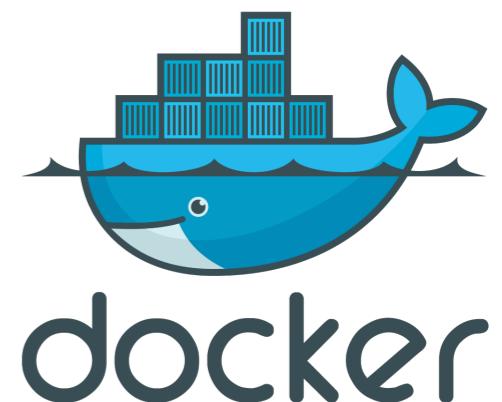
Amazon EC2 Container Service (ECS)

Google Kubernetes

IBM Bluemix Container Service

Joyent Smart Data Center

Microsoft Azure



Docker Compose

Orchestration of multi-container apps

Based on Fig

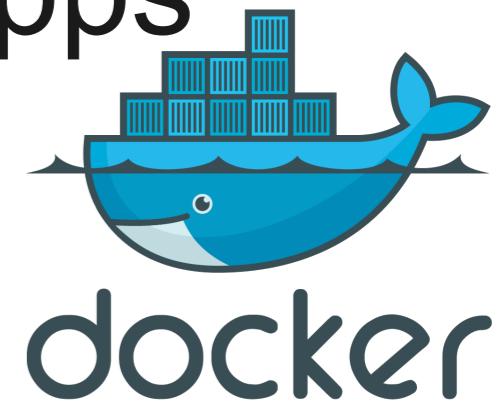
Defined by:

containers

configuration

links

volumes



Apache Mesos

A distributed systems kernel



Docker Containerizer
Marathon & Chronos



MESOS

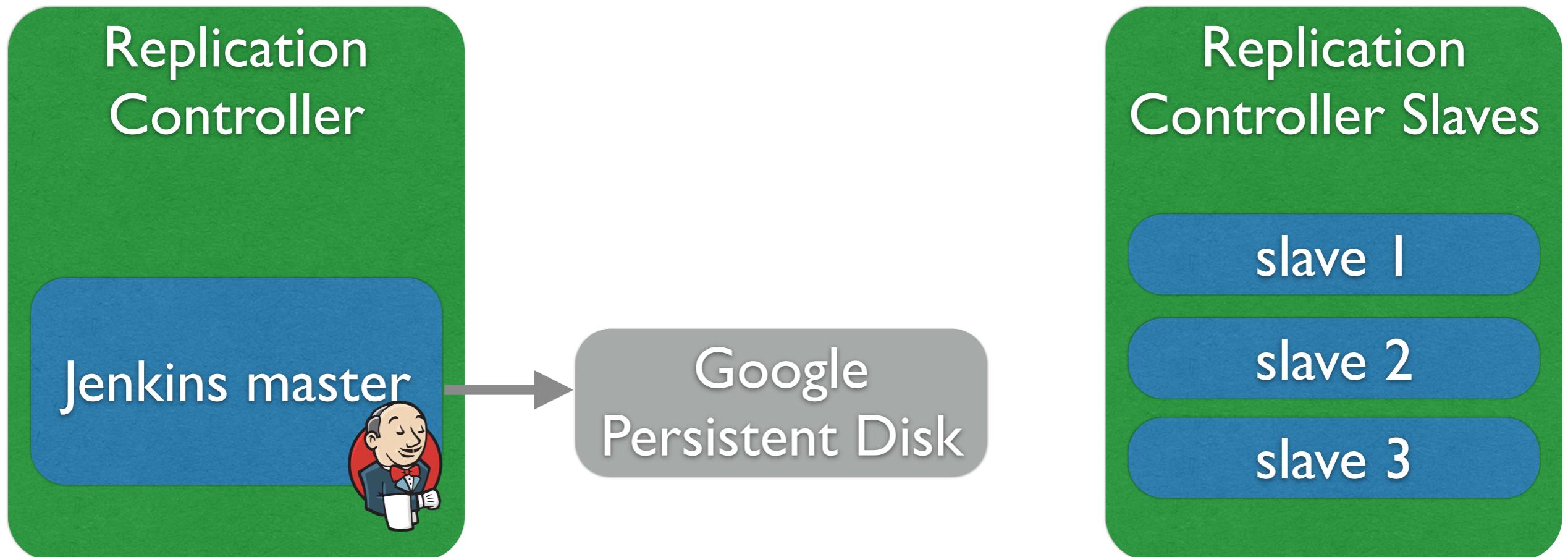


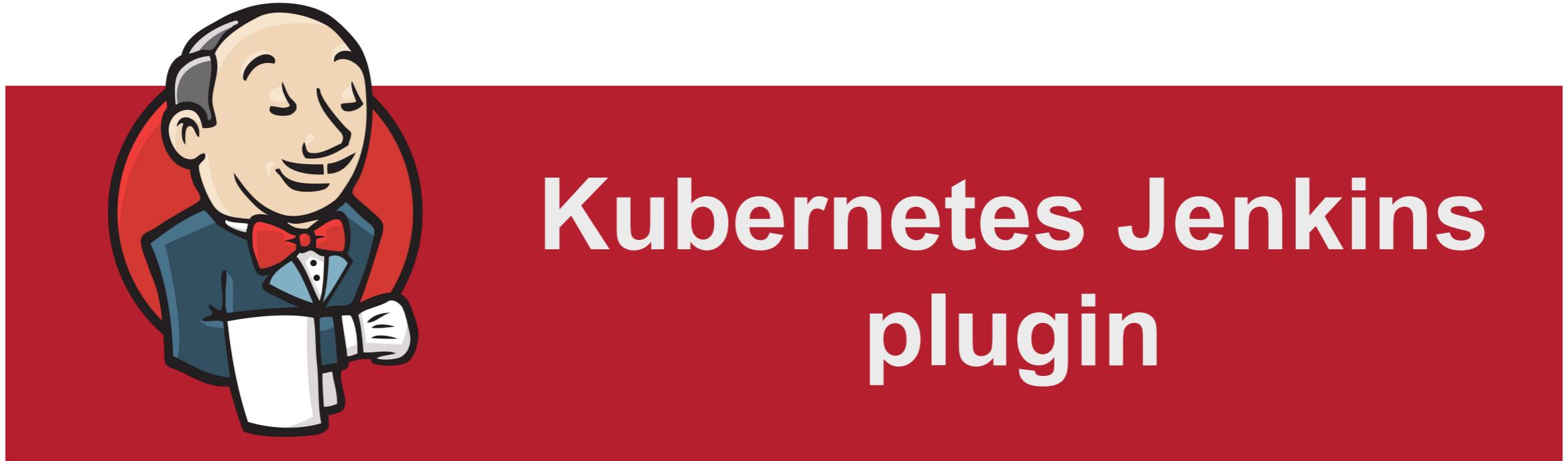
CloudBees



Kubernetes and Jenkins

Initial solution





Kubernetes Jenkins plugin

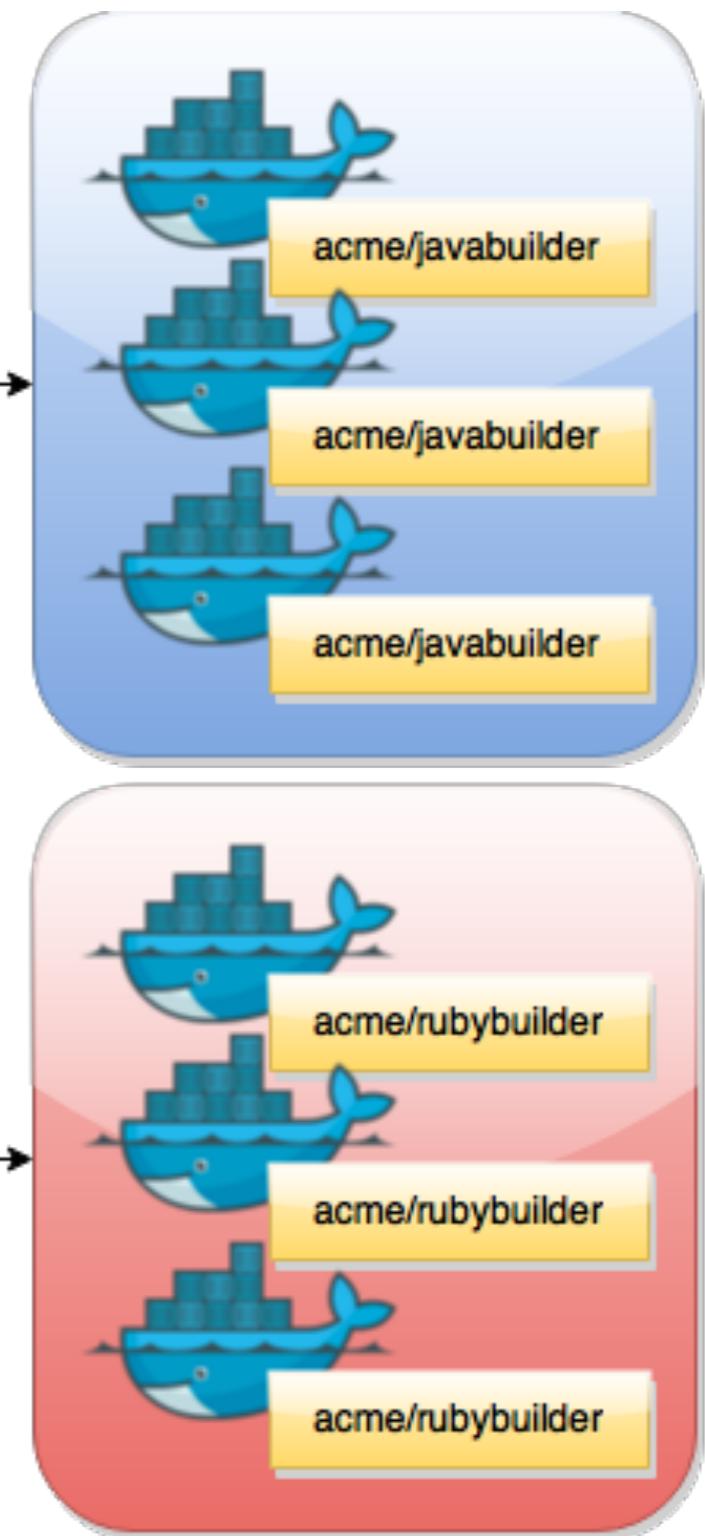
Kubernetes Jenkins plugin

As a plugin
on demand slaves

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



List of slave images:
acme/javabuilder
acme/rubybuilder



Kubernetes Jenkins plugin

pods, not replication controllers

Jenkins Cloud API

Fabric8 Java API

Workflow support

Kubernetes

Name: My Kubernetes cluster

Kubernetes URL: https://kubernetes.default.svc.cluster.local

Kubernetes server certificate key:

Disable https certificate check:

Kubernetes Namespace: kubernetes-plugin

Credentials: 598980c4-2090-4ce2-bc18-5f57432669c9 (My Kubernetes cluster)

Jenkins URL: http://10.175.249.78

Jenkins tunnel:

Connection Timeout: 5

Read Timeout: 15

Container Cap: 10

Images

Kubernetes Pod Template

Name: jnlp slave

Labels:

Docker image: jenkinsci/jnlp-slave

Jenkins slave root directory: /home/jenkins

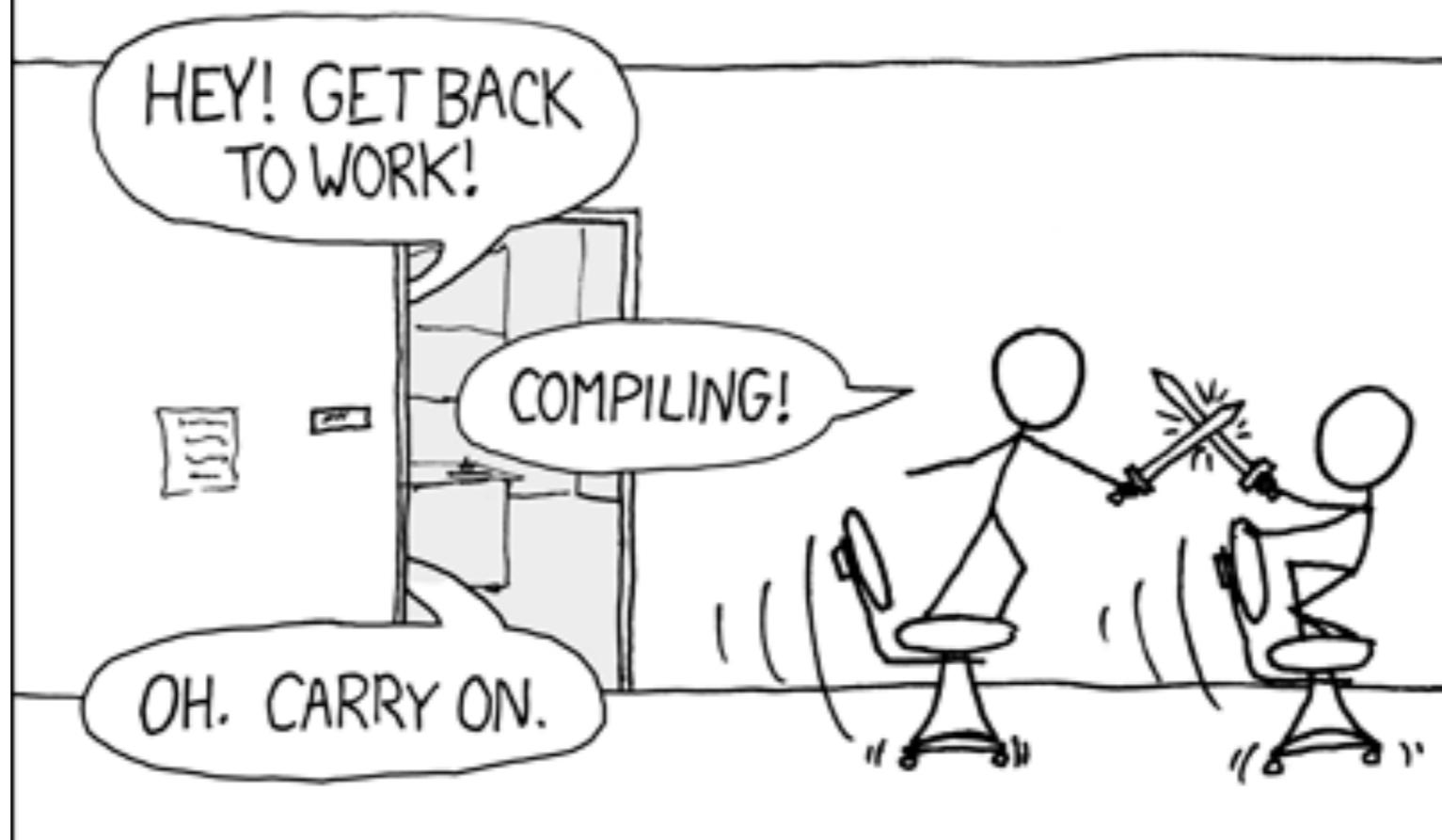
Command to run slave agent:

Arguments to pass to the command:

Max number of instances:



THE #1 PROGRAMMER EXCUSE
FOR LEGITIMATELY SLACKING OFF:
"MY CODE'S COMPILING."



Roadmap

Stable API

When Kubernetes Java lib is stable

Using new Jenkins Cloud/Containers APIs

Use jobs in Kubernetes 1.1

Example code and slides

Available at

<http://slideshare.csanchez.org>

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

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

<http://blog.csanchez.org>



Дякую