USING KUBERNETES FOR

CONTINUOUS INTEGRATION

AND

CONTINUOUS DELIVERY

Carlos Sanchez

csanchez.org / @csanchez

ABOUT ME

Engineer @ CloudBees, Scaling Jenkins

Author of Jenkins Kubernetes plugin

Contributor to Jenkins and Maven official Docker images

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

Kernel Sanders
@lstoll

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

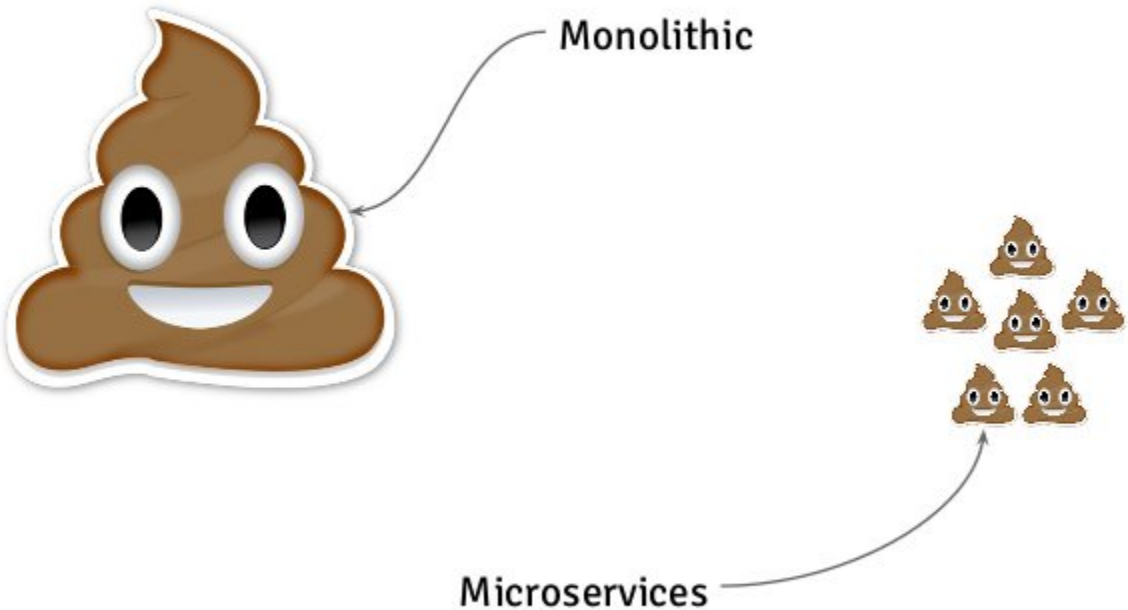
WHEN ONE MACHINE IS NO LONGER ENOUGH

- Running containers across multiple hosts
- Multiple environments: public cloud, private cloud, VMs or bare metal
- HA and fault tolerance

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

Kelsey Hightower

Monolithic vs Microservices




kubernetes

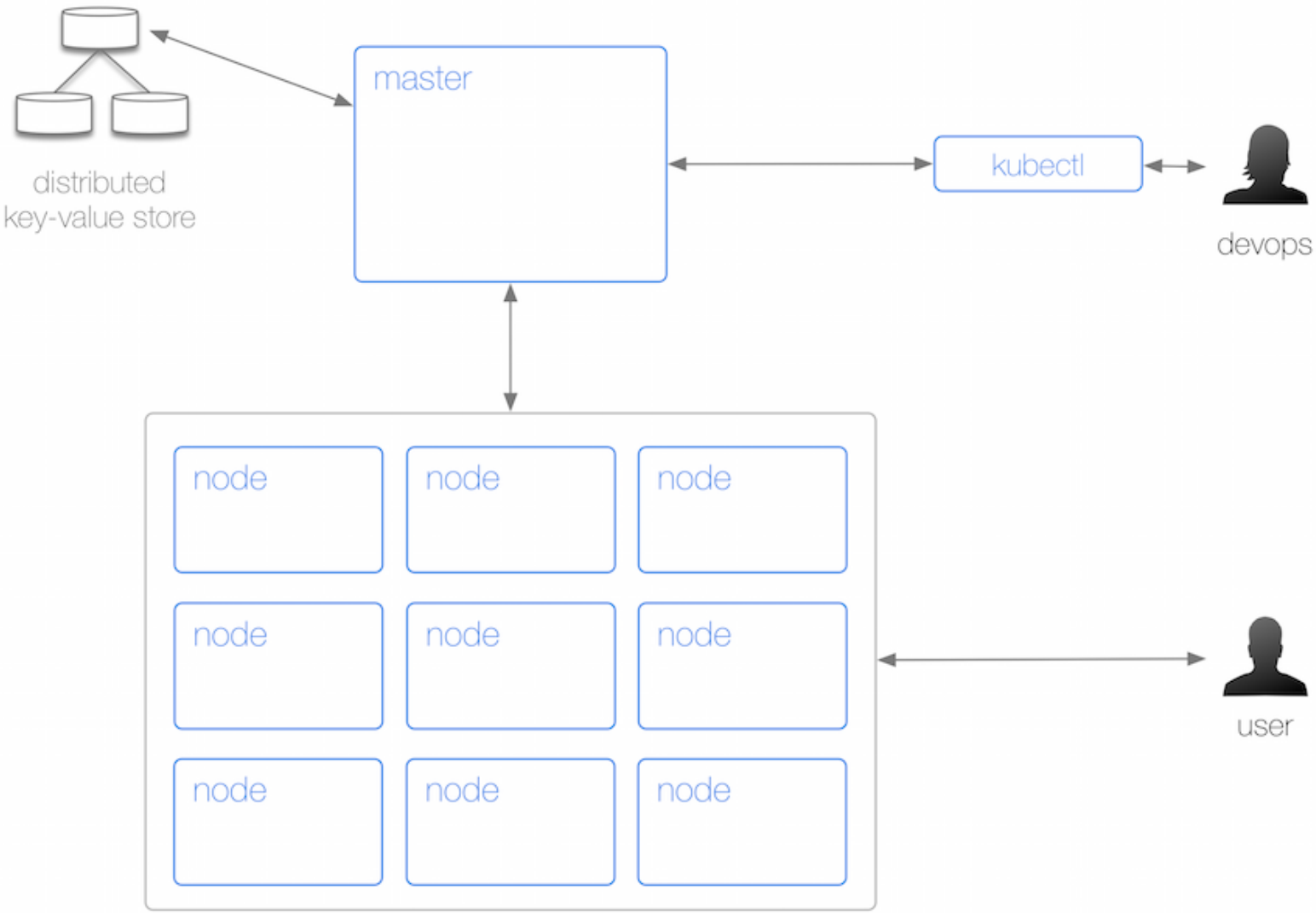
KUBERNETES

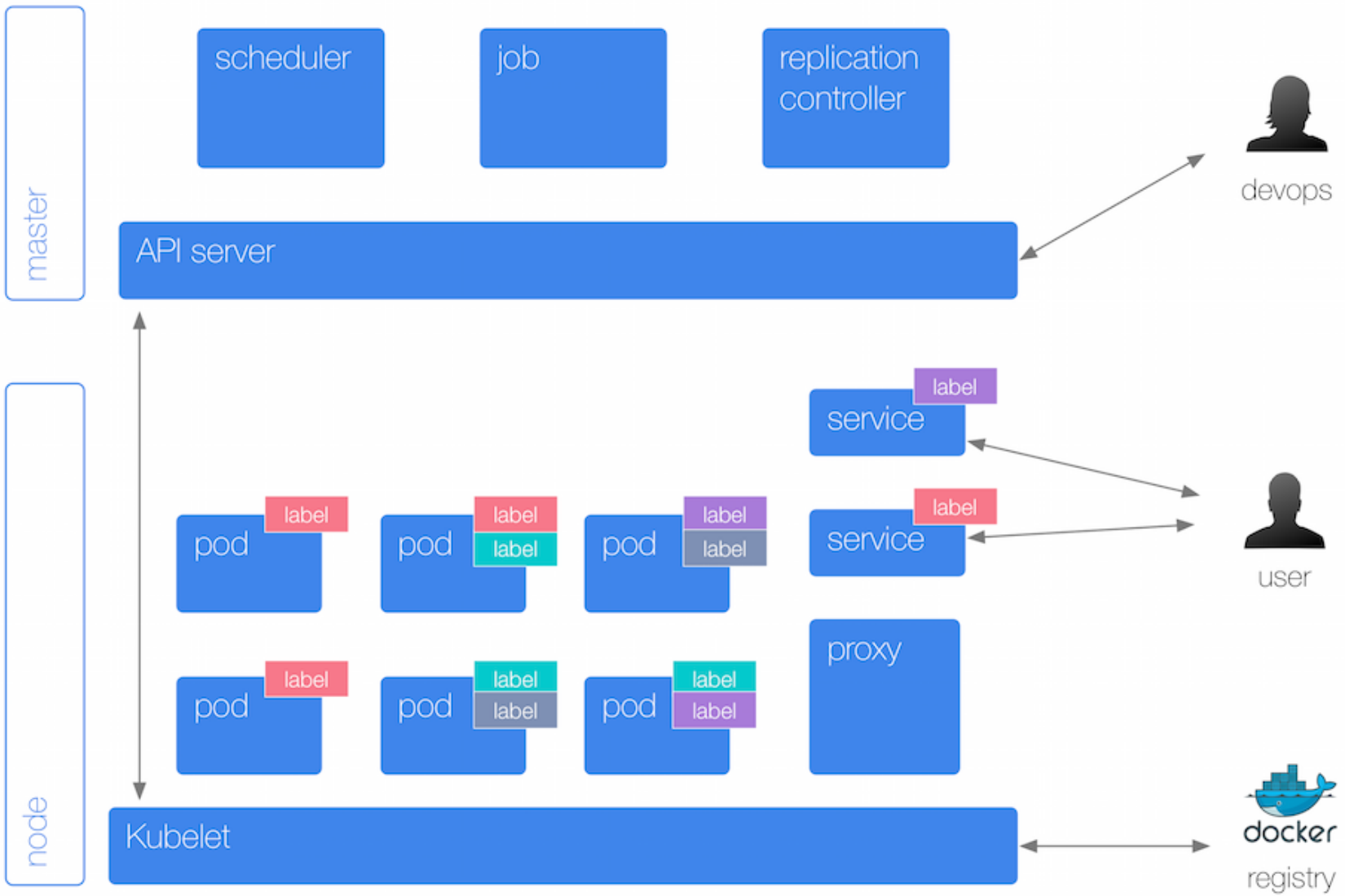
- Based on Google Borg
- Run in local machine, virtual, cloud
- Google provides Google Container Engine (GKE)
- Other services run by stackpoint.io, CoreOS Tectonic, Azure,...
- Minikube for local testing

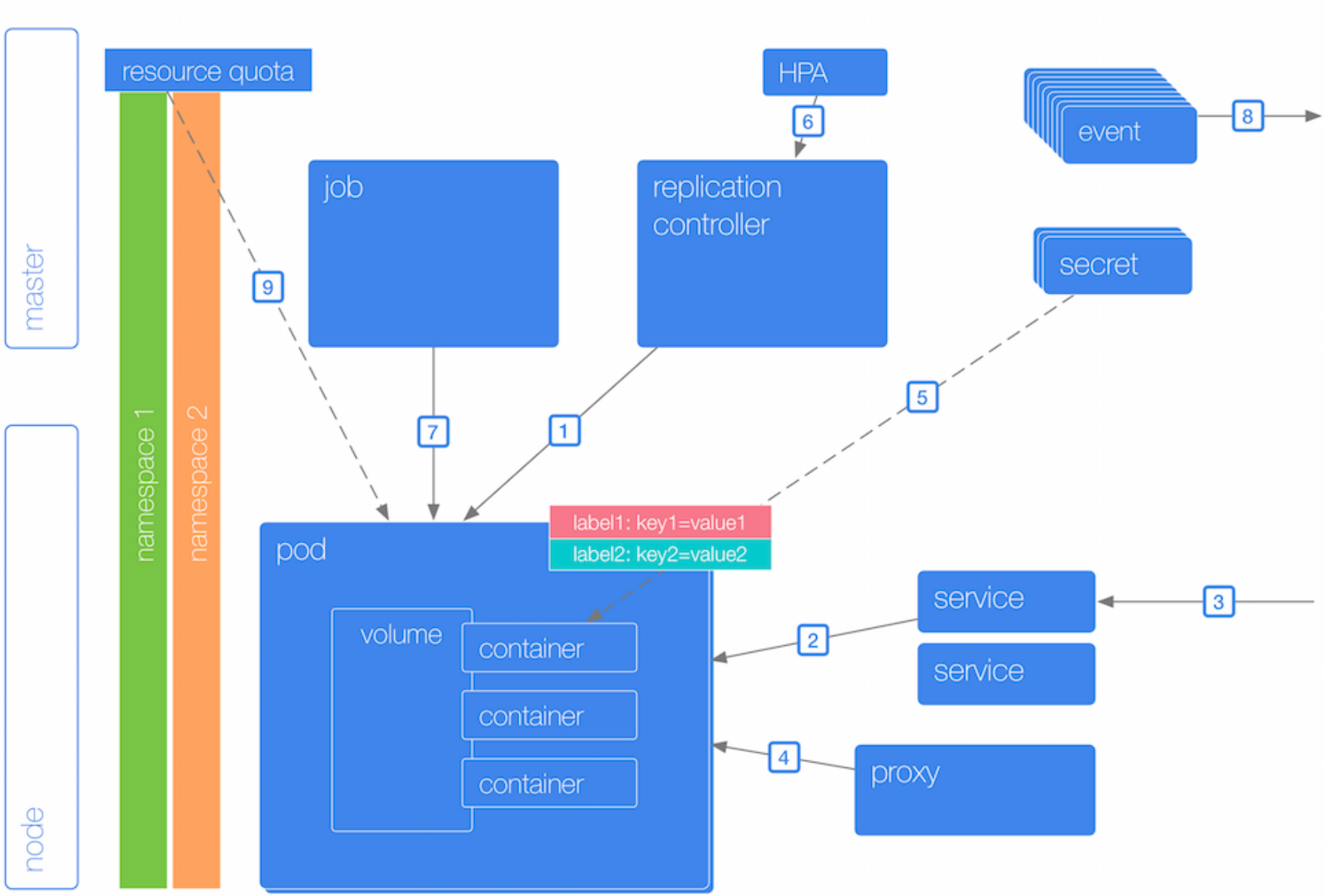
KUBERNETES

Free goodies:

- Declarative Syntax
- Pods (groups of colocated containers)
- Persistent Storage
- Networking Isolation


distributed
key-value store
master
kubectl
devops
node
node
node
node
node
node
node
node
node
user


master
scheduler
job
replication controller
devops
API server
node
label
service
pod
label
pod
label
label
pod
label
label
service
label
user
pod
label
pod
label
label
pod
label
label
proxy
Kubelet
docker
registry


master
node
resource quota
namespace 1
namespace 2
HPA
6
job
replication
controller
event
8
secret
9
7
1
5
label1: key1=value1
label2: key2=value2
pod
volume
container
container
container
service
3
2
service
4
proxy

@DEVOPS_BORAT

DevOps Borat

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

If you haven't automatically destroyed something by mistake, you are not automating enough

&

We can run both Jenkins **masters and agents** in Kubernetes

INFINITE SCALE!

Jenkins Kubernetes Plugin

- Dynamic Jenkins agents, running as Pods
- Multi-container support
  - One Jenkins agent image, others custom
- Pipeline support for both agent Pod definition and execution
- Persistent workspace

ON DEMAND JENKINS AGENTS

```
podTemplate(label: 'mypod') {
  node('mypod') {
    sh 'Hello world!'
  }
}
```

GROUPING CONTAINERS (PODS)

```
podTemplate(label: 'maven', containers: [
  containerTemplate(name: 'maven', image: 'maven:3.3.9-jdk-8-alpine',
    ttyEnabled: true, command: 'cat') ]) {

  node('maven') {
    stage('Get a Maven project') {
      git 'https://github.com/jenkinsci/kubernetes-plugin.git'
      container('maven') {
        stage('Build a Maven project') {
          sh 'mvn -B clean package'
        }
      }
    }
  }
}
```

USING DECLARATIVE PIPELINE TOO

```
pipeline {
  agent {
    kubernetes {
      label 'mypod'
      containerTemplate {
        name 'maven'
        image 'maven:3.3.9-jdk-8-alpine'
        ttyEnabled true
        command 'cat'
      }
    }
  }
  stages {
    stage('Run maven') {
      steps {
        container('maven') {
          sh 'mvn -version'
        }
      }
    }
  }
}
```

PODS: MULTI-LANGUAGE PIPELINE

```
podTemplate(label: 'maven-golang', containers: [
  containerTemplate(name: 'maven', image: 'maven:3.3.9-jdk-8-alpine',
    ttyEnabled: true, command: 'cat'),
  containerTemplate(name: 'golang', image: 'golang:1.8.0',
    ttyEnabled: true, command: 'cat')]) {

  node('maven-golang') {
    stage('Build a Maven project') {
      git 'https://github.com/jenkinsci/kubernetes-plugin.git'
      container('maven') {
        sh 'mvn -B clean package'
      }
    }

    stage('Build a Golang project') {
      git url: 'https://github.com/hashicorp/terraform.git'
      container('golang') {
        sh """
        mkdir -p /go/src/github.com/hashicorp
        ln -s `pwd` /go/src/github.com/hashicorp/terraform
        cd /go/src/github.com/hashicorp/terraform && make core-dev
        """
      }
    }
```

PODS: SELENIUM

Example:

- Jenkins agent
- Maven build
- Selenium Hub with
  - Firefox
  - Chrome

5 containers

```
podTemplate(label: 'maven-selenium', containers: [
  containerTemplate(name:'maven-firefox',image:'maven:3.3.9-jdk-8-alp
    ttyEnabled: true, command: 'cat'),
  containerTemplate(name:'maven-chrome',image:'maven:3.3.9-jdk-8-alpi
    ttyEnabled: true, command: 'cat'),
  containerTemplate(name: 'selenium-hub', image: 'selenium/hub:3.4.0'
  // because containers run in the same network space, we need to
  // make sure there are no port conflicts
  // we also need to adapt the selenium images because they were
  // designed to work with the --link option
  containerTemplate(name: 'selenium-chrome',
    image: 'selenium/node-chrome:3.4.0', envVars: [
    containerEnvVar(key: 'HUB_PORT_4444_TCP_ADDR', value: 'localhost'
    containerEnvVar(key: 'HUB_PORT_4444_TCP_PORT', value: '4444'),
    containerEnvVar(key: 'DISPLAY', value: ':99.0'),
    containerEnvVar(key: 'SE_OPTS', value: '-port 5556'),
  ]),
  containerTemplate(name: 'selenium-firefox',
    image: 'selenium/node-firefox:3.4.0', envVars: [
    containerEnvVar(key: 'HUB_PORT_4444_TCP_ADDR', value: 'localhost'
    containerEnvVar(key: 'HUB_PORT_4444_TCP_PORT', value: '4444'),
    containerEnvVar(key: 'DISPLAY', value: ':98.0'),
    containerEnvVar(key: 'SE_OPTS', value: '-port 5557'),
  ])
```

```
node('maven-selenium') {
  stage('Checkout') {
    git 'https://github.com/carlossg/selenium-example.git'
    parallel (
      firefox: {
        container('maven-firefox') {
          stage('Test firefox') {
            sh """
            mvn -B clean test -Dselenium.browser=firefox \
              -Dsurefire.rerunFailingTestsCount=5 -Dsleep=0
            """
          }
        }
      },
      chrome: {
        container('maven-chrome') {
          stage('Test chrome') {
            sh """
            mvn -B clean test -Dselenium.browser=chrome \
              -Dsurefire.rerunFailingTestsCount=5 -Dsleep=0
            """
          }
        }
      }
```

STORAGE

Persistent volumes

- GCE disks
- GlusterFS
- NFS
- EBS
- etc

USING PERSISTENT VOLUMES

```
apiVersion: "v1"
kind: "PersistentVolumeClaim"
metadata:
  name: "maven-repo"
  namespace: "kubernetes-plugin"
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
```

```
podTemplate(label: 'maven', containers: [
  containerTemplate(name: 'maven', image: 'maven:3.3.9-jdk-8-alpine',
    ttyEnabled: true, command: 'cat')
  ], volumes: [
  persistentVolumeClaim(mountPath: '/root/.m2/repository',
    claimName: 'maven-repo', readOnly: false)
  ]) {

  node('maven') {
    stage('Build a Maven project') {
      git 'https://github.com/jenkinsci/kubernetes-plugin.git'
      container('maven') {
          sh 'mvn -B clean package'
      }
    }
  }
}
```

MEMORY LIMITS

Scheduler needs to account for container memory requirements and host available memory

Prevent containers for using more memory than allowed

Memory constraints translate to Docker --memory

https://kubernetes.io/docs/concepts/configuration/manage-compute-resources-container/#how-pods-with-resource-limits-are-run

WHAT DO YOU THINK HAPPENS WHEN?

Your container goes over memory quota?


¡CACHONDEO.COM

NEW JVM SUPPORT FOR CONTAINERS

JDK 8u131+ and JDK 9

```
$ docker run -m 1GB openjdk:8u131 java \
  -XX:+UnlockExperimentalVMOptions \
  -XX:+UseCGroupMemoryLimitForHeap \
  -XshowSettings:vm -version
VM settings:
    Max. Heap Size (Estimated): 228.00M
    Ergonomics Machine Class: server
    Using VM: OpenJDK 64-Bit Server VM
```

Running a JVM in a Container Without Getting Killed

https://blog.csanchez.org/2017/05/31/running-a-jvm-in-a-container-without-getting-killed

NEW JVM SUPPORT FOR CONTAINERS

```
$ docker run -m 1GB openjdk:8u131 java \
  -XX:+UnlockExperimentalVMOptions \
  -XX:+UseCGroupMemoryLimitForHeap \
  -XX:MaxRAMFraction=1 -XshowSettings:vm -version
VM settings:
    Max. Heap Size (Estimated): 910.50M
    Ergonomics Machine Class: server
    Using VM: OpenJDK 64-Bit Server VM
```

Running a JVM in a Container Without Getting Killed

https://blog.csanchez.org/2017/05/31/running-a-jvm-in-a-container-without-getting-killed

CPU LIMITS

Scheduler needs to account for container CPU requirements and host available CPUs

CPU requests translates into Docker `--cpu-shares`

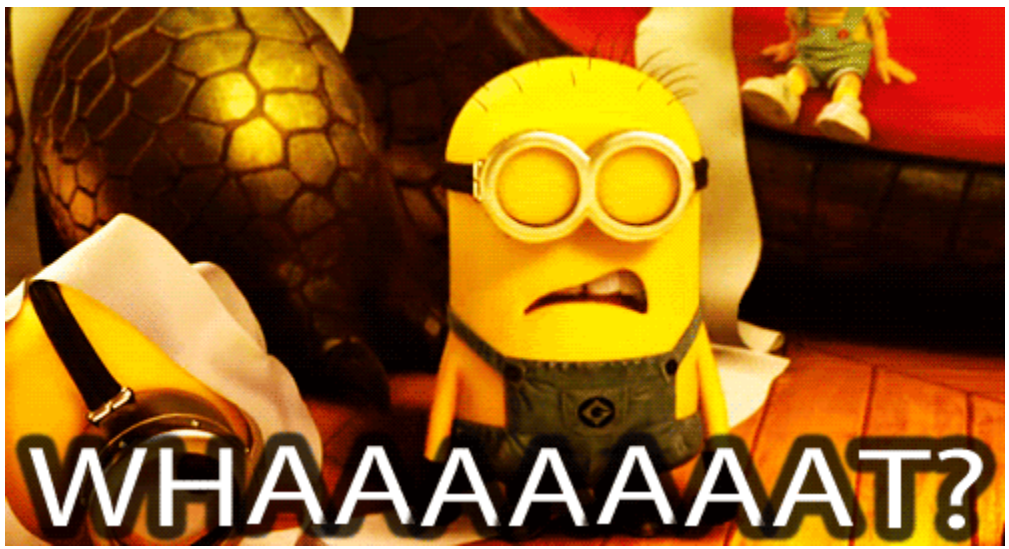
CPU limits translates into Docker `--cpu-quota`

https://kubernetes.io/docs/concepts/configuration/manage-compute-resources-container/#how-pods-with-resource-limits-are-run

WHAT DO YOU THINK HAPPENS WHEN?

Your container tries to access more than one CPU

Your container goes over CPU limits



Totally different from memory

RESOURCE REQUESTS AND LIMITS

```
podTemplate(label: 'mypod', containers: [
    containerTemplate(
        name: 'maven', image: 'maven', ttyEnabled: true,
        resourceRequestCpu: '50m',
        resourceLimitCpu: '100m',
        resourceRequestMemory: '100Mi',
        resourceLimitMemory: '200Mi')]) {
...
}
```

DEPLOYING TO KUBERNETES

DEPLOYING TO KUBERNETES

```
podTemplate(label: 'deployer', serviceAccount: 'deployer', containers
    containerTemplate(name: 'kubectl', image: 'lachlanevenson/k8s-kub
      command: 'cat', ttyEnabled: true)
]){
  node('deployer') {
    container('kubectl') {
      sh "kubectl apply -f my-kubernetes.yaml"
    }
  }
}
```

DEPLOYING TO KUBERNETES

kubernetes-pipeline-plugin

```
podTemplate(label: 'deploy', serviceAccount: 'deployer') {

  stage('deployment') {
    node('deploy') {
      checkout scm
      kubernetesApply(environment: 'hello-world',
        file: readFile('kubernetes-hello-world-service.yaml'))
      kubernetesApply(environment: 'hello-world',
        file: readFile('kubernetes-hello-world-v1.yaml'))
    }}

  stage('upgrade') {
    timeout(time:1, unit:'DAYS') {
      input id: 'approve', message:'Approve upgrade?'
    }
    node('deploy') {
      checkout scm
      kubernetesApply(environment: 'hello-world',
        file: readFile('kubernetes-hello-world-v2.yaml'))
    }}
}
```

Or Azure kubernetes-cd-plugin

```
kubernetesDeploy(
  credentialsType: 'KubeConfig',
  kubeConfig: [path: '$HOME/.kube/config'],

  configs: '*.yaml',
  enableConfigSubstitution: false,
)
```

СПАСИБО

csanchez.org


csanchez


carlossg