TESTING DISTRIBUTED MICRO-SERVICES

Carlos Sanchez

@csanchez csanchez.org

Watch online at carlossg.github.io/presentations

ABOUT ME

Senior Software Engineer @ CloudBees

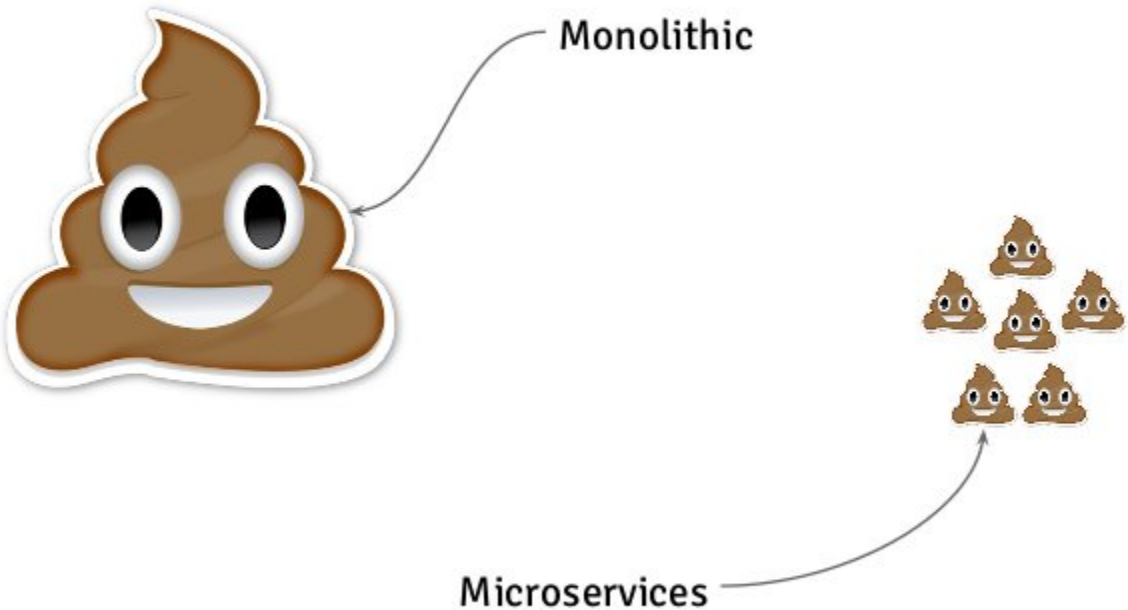
Author of Jenkins Kubernetes plugin

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

Google Cloud Platform "Expert"

DOCKER DOCKER DOCKER

Monolithic vs Microservices



OUR USE CASE

Scaling Jenkins

Your mileage may vary

Cloudbees
Jenkins Platform

Pipelines | Administration

Tiger

Cluster Summary | Virtual Machines | Masters

2000 Masters

2000 healthy | 0 warn | 0 critical

317 Virtual Machines

317 healthy | 0 warn | 0 critical

7 ElasticSearch

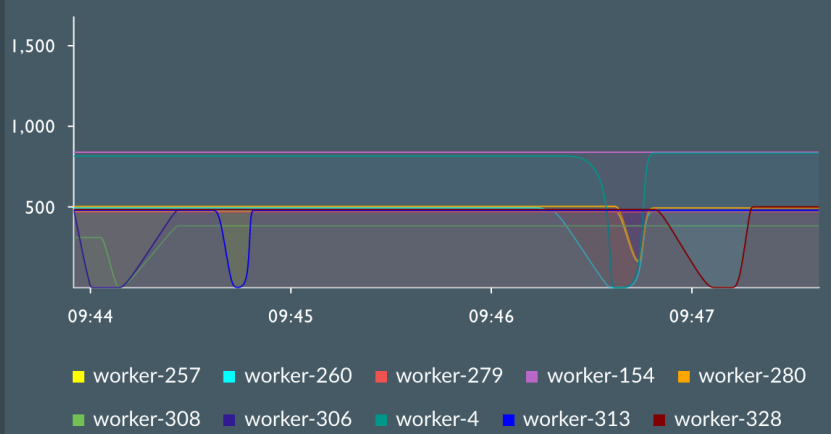
7 healthy | 0 warn | 0 critical

35% Core utilization

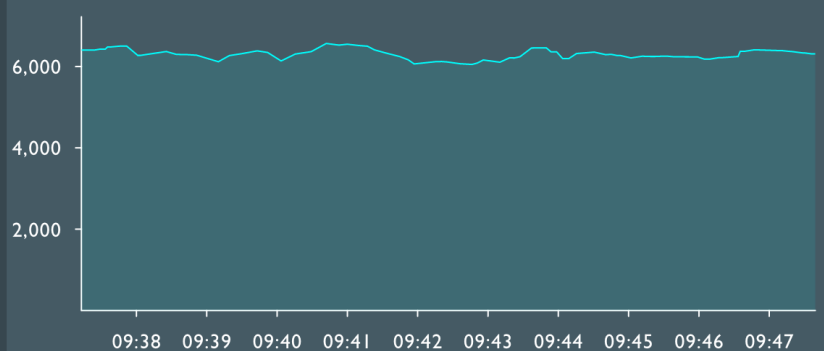
Used cores: 1300.5

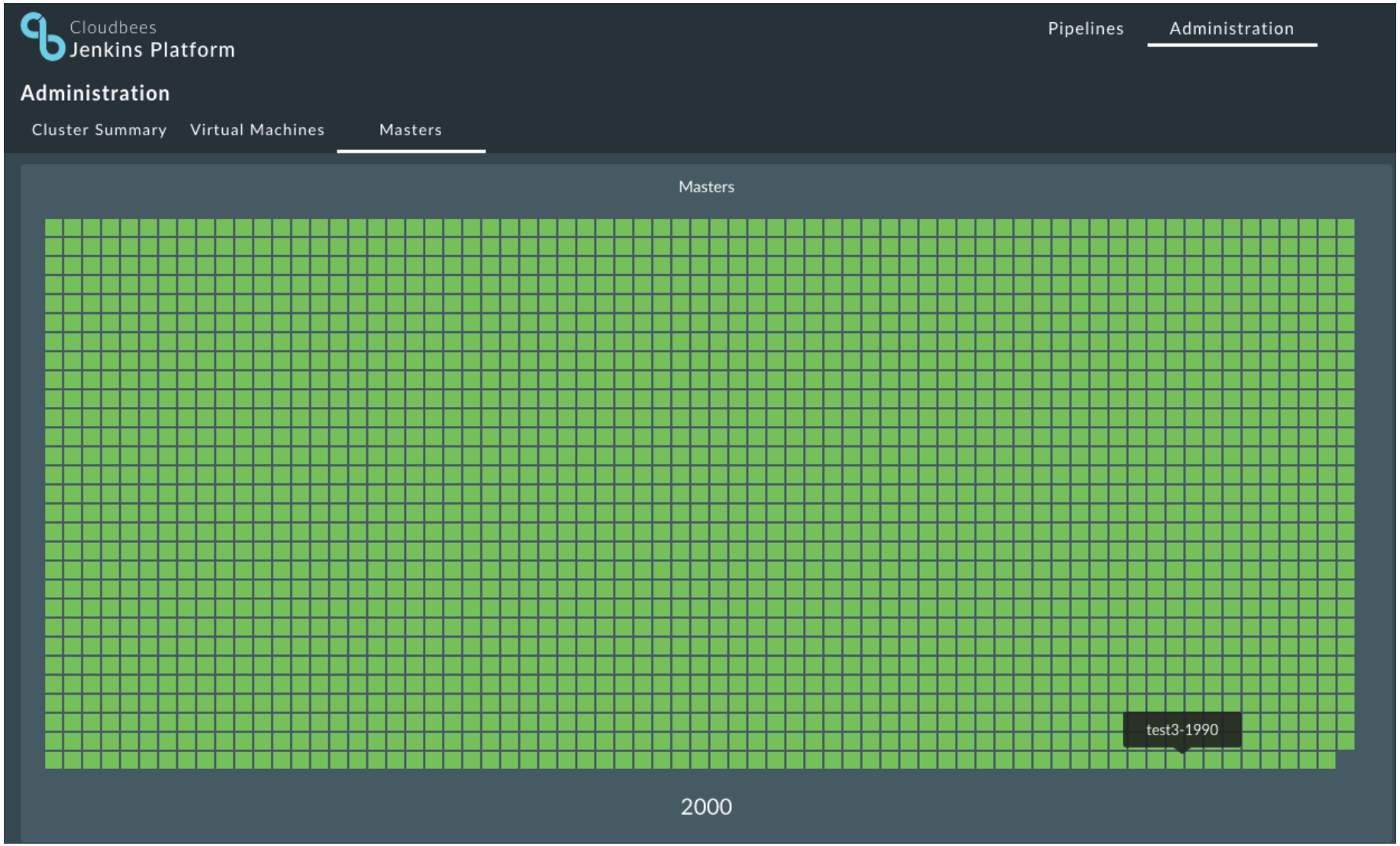
Total cores: 3748

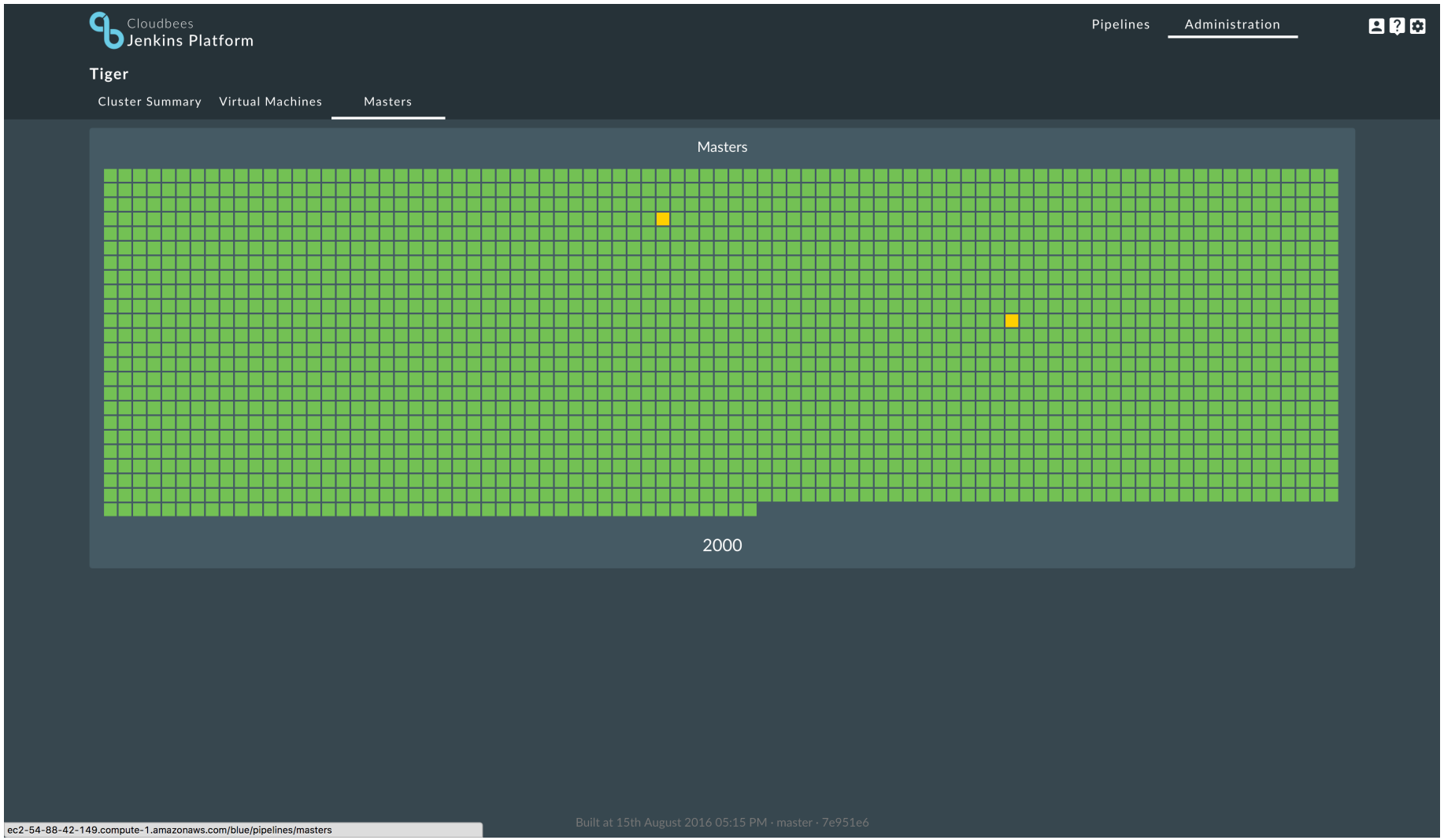
Workload



Executors




Cloudbees
Jenkins Platform
Pipelines
Administration
Administration
Cluster Summary
Virtual Machines
Masters
Masters
test3-1990
2000


Cloudbees
Jenkins Platform
Pipelines
Administration
Tiger
Cluster Summary
Virtual Machines
Masters
Masters
2000
ec2-54-88-42-149.compute-1.amazonaws.com/blue/pipelines/masters
Built at 15th August 2016 05:15 PM · master · 7e951e6

A 2000 JENKINS MASTERS CLUSTER

- 3 Mesos masters (m3.xlarge: 4 vCPU, 15GB, 2x40 SSD)
- 317 Mesos slaves (c3.2xlarge, m3.xlarge, m4.4xlarge)
- 7 Mesos slaves dedicated to ElasticSearch: (c3.8xlarge: 32 vCPU, 60GB)

12.5 TB - 3748 CPU

Running 2000 masters and ~8000 concurrent jobs

ARCHITECTURE

Kernel Sanders
@lstoll

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

Isolated Jenkins masters

Isolated build agents and jobs

Memory and CPU limits

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

CLUSTER SCHEDULING

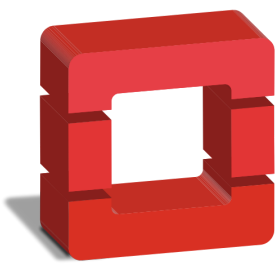
Distribute tasks across a cluster of hosts

HA and fault tolerant

With Docker support of course

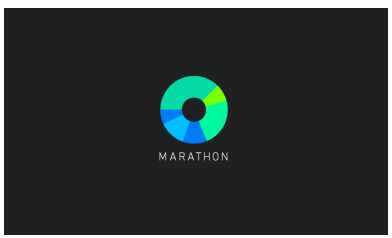
INFRASTRUCTURE

Running in public cloud, private cloud, VMs or bare metal



openstack
CLOUD SOFTWARE

APACHE MESOS & MESOSPHERE MARATHON



A distributed systems kernel

ALTERNATIVES

Docker Swarm / Kubernetes

"UNIT" TESTING DOCKER IMAGES

THE DOCKERFILE

A lot like a shell script

`RUN` comands

`COPY` files

...

DOCKERFILE

```
FROM openjdk:8-jdk

RUN apt-get update && apt-get install -y git curl && rm -rf /var/lib/

ARG JENKINS_VERSION
ENV JENKINS_VERSION ${JENKINS_VERSION:-2.19.3}
ARG JENKINS_SHA=e976706363940 92af40cc626f8e07b092105c07b

ARG JENKINS_URL=https://repo.jenkins-ci.org/public/org/jenkins-ci/mai

RUN curl -fsSL ${JENKINS_URL} -o /usr/share/jenkins/jenkins.war \
  && echo "${JENKINS_SHA}  /usr/share/jenkins/jenkins.war" | sha1sum

COPY jenkins-support /usr/local/bin/jenkins-support
COPY jenkins.sh /usr/local/bin/jenkins.sh
ENTRYPOINT ["/usr/local/bin/jenkins.sh"]
```

Mocking and stubbing are your friends

BUILDING WITH JENKINS DOCKER PIPELINE

```
def maven = docker.image('maven:3.3.9-jdk-8');

stage 'Mirror'
maven.pull()
docker.withRegistry('https://secure-registry/',
    'docker-registry-login') {

  stage 'Build'
  maven.inside {
    sh "mvn -B clean package"
  }

  stage 'Bake Docker image'
  def pcImg = docker.build(
    "examplecorp/spring-petclinic:${env.BUILD_TAG}", 'app')

  pcImg.push();
}
```

TESTING WITH JENKINS DOCKER PIPELINE

Build + test + promotion

Promotion = different Docker registries for different environments

- dev
- staging
- production

TESTING WITH JENKINS DOCKER PIPELINE

```
pcImg = docker.image("examplecorp/spring-petclinic:dev")

stage 'Test'
docker.withRegistry('https://dev.docker.example.com/',
    'docker-registry-login') {

    pcImg.withRun { petclinic ->
        sh "test -f /var/some_file"
    }
}

stage 'Promote'
docker.withRegistry('https://staging.docker.example.com/',
    'docker-registry-login') {

    pcImg.push()
}
```

DOCKER WORKFLOW PLUGIN DEMO

https://github.com/jenkinsci/docker-workflow-plugin/tree/master/demo

USING BATS

Testing using shell scripts!

```
bats tests/install-plugins.bats
```

Examples from
https://github.com/jenkinsci/docker/tree/master/tests

BATS

```
#!/usr/bin/env bats

SUT_IMAGE=bats-jenkins

load 'test_helper/bats-support/load'
load 'test_helper/bats-assert/load'
load test_helpers

@test "plugins are installed with plugins.sh" {
  run docker build -t $SUT_IMAGE-plugins $BATS_TEST_DIRNAME/plugins
  assert_success
  run bash -c "docker run --rm $SUT_IMAGE-plugins ls -1 \
    /var/jenkins_home/plugins"
  assert_success
  assert_line 'maven-plugin.jpi'
  assert_line 'maven-plugin.jpi.pinned'
  assert_line 'ant.jpi'
  assert_line 'ant.jpi.pinned'
}
```

BATS

```
@test "test jenkins arguments" {
  local version=$(grep 'ENV JENKINS_VERSION' Dockerfile | \
    sed -e 's/.*:-\(.*\)}/\1/')
  # need the last line of output
  assert "${version}" docker run --rm --name $SUT_CONTAINER \
    -P $SUT_IMAGE --help --version | tail -n 1
}
```

ISOLATION

There is no 100% isolation

MEMORY ISSUES WITH CONTAINERS

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

Prevent containers for using more memory than allowed

Memory constrains translate to Docker --memory

WHAT DO YOU THINK HAPPENS WHEN?

Your container goes over memory quota?


¡CACHONDEO.COM

WHAT ABOUT THE JVM?

WHAT ABOUT THE CHILD PROCESSES?

END TO END TESTING OF MULTIPLE CONTAINERS

JENKINS PIPELINE WITH MULTIPLE CONTAINERS

```
pcImg = docker.image("examplecorp/spring-petclinic:dev")

stage 'Test'
docker.withRegistry('https://dev.docker.example.com/',
    'docker-registry-login') {

    pcImg.withRun {petclinic ->
      testImg.inside("--link=${petclinic.id}:petclinic") {
        wrap([$class: 'Xvnc',
          takeScreenshot: true,
          useXauthority: true]) {
          sh "mvn -B clean test"
        }
      }
    }
}

stage 'Promote'
docker.withRegistry('https://staging.docker.example.com/',
    'docker-registry-login') {
    pcImg.push()
}
```

USING CONTAINER GROUPS

DOCKER COMPOSE

Runs multiple containers

Containers can access each other

DOCKER COMPOSE

```
version: '2'

services:

  example:
    image: acme/example:latest
    ports:
      - "5050:5050"

  maven:
    image: maven:3-jdk-8
    command: mvn test
```

JENKINS KUBERNETES PLUGIN

The Jenkins job can run in a Kubernetes Pod

(group of containers)

Containers in a Pod can access each other at `localhost`

KUBERNETES PLUGIN PIPELINES

Ex. run maven tests against webapp with Selenium, using pre-made Docker images

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

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

HOW ARE WE DOING IT

HOW ARE WE DOING IT

On each commit and PR

Provisioning of the infrastructure using Terraform

Installation of the cluster scheduler (Mesos & Marathon)

Continuously creating clusters from scratch

TERRAFORM

Infrastructure-As-Code

TERRAFORM

```
resource "aws_instance" "worker" {
    count = 1
    instance_type = "m3.large"
    ami = "ami-xxxxxx"
    key_name = "tiger-csanchez"
    security_groups = ["sg-61bc8c18"]
    subnet_id = "subnet-xxxxxx"
    associate_public_ip_address = true
    tags {
        Name = "tiger-csanchez-worker-1"
        "cloudbees:pse:cluster" = "tiger-csanchez"
        "cloudbees:pse:type" = "worker"
    }
    root_block_device {
        volume_size = 50
    }
}
```

TERRAFORM

- State is managed
- Runs are idempotent
  - `terraform apply`
- Sometimes it is too automatic
  - Changing image id will restart all instances

@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

HOW ARE WE DOING IT

In AWS and OpenStack

5 different combinations

More combinations on demand

HOW ARE WE DOING IT

After creation we launch acceptance tests

Some python scripts

Some Selenium tests

Clusters get destroyed at the end

MONITORING IS THE NEW TESTING

We gather from the cluster:

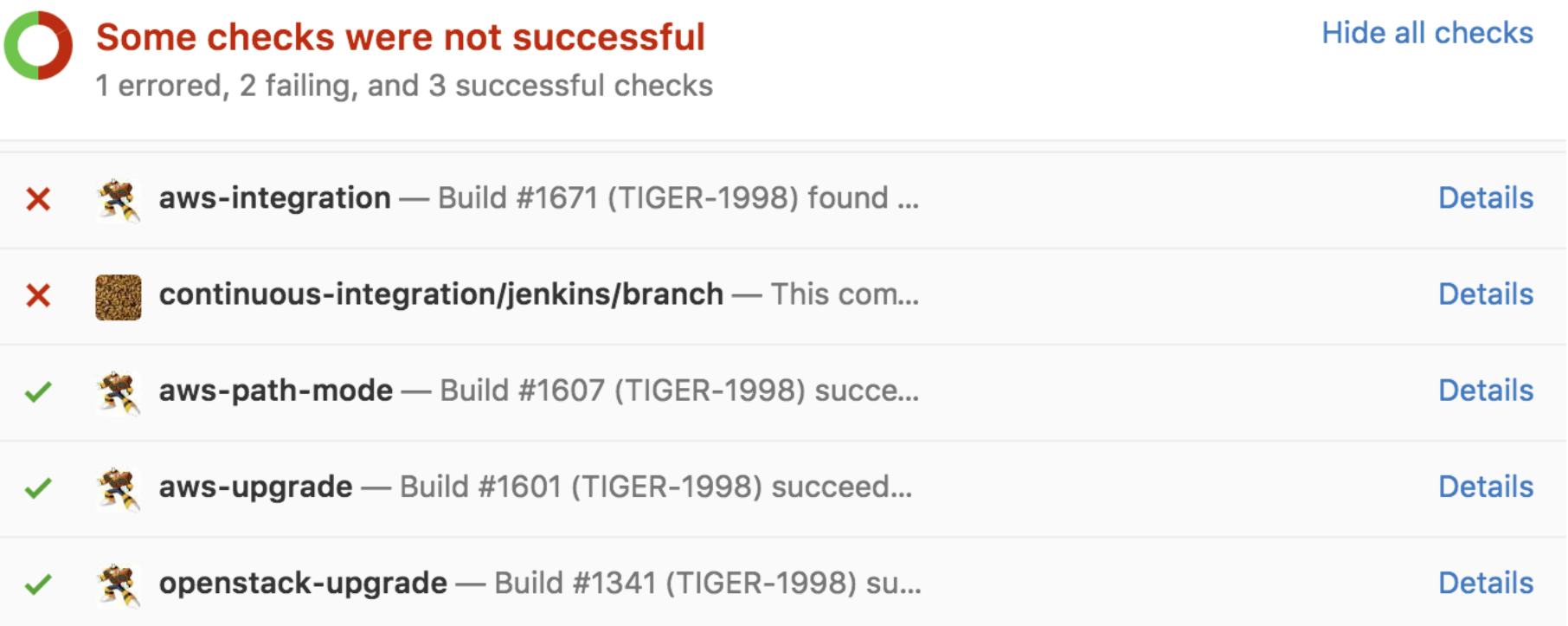
- logs
- configuration
- outputs

Attached to the build to diagnose errors in CI

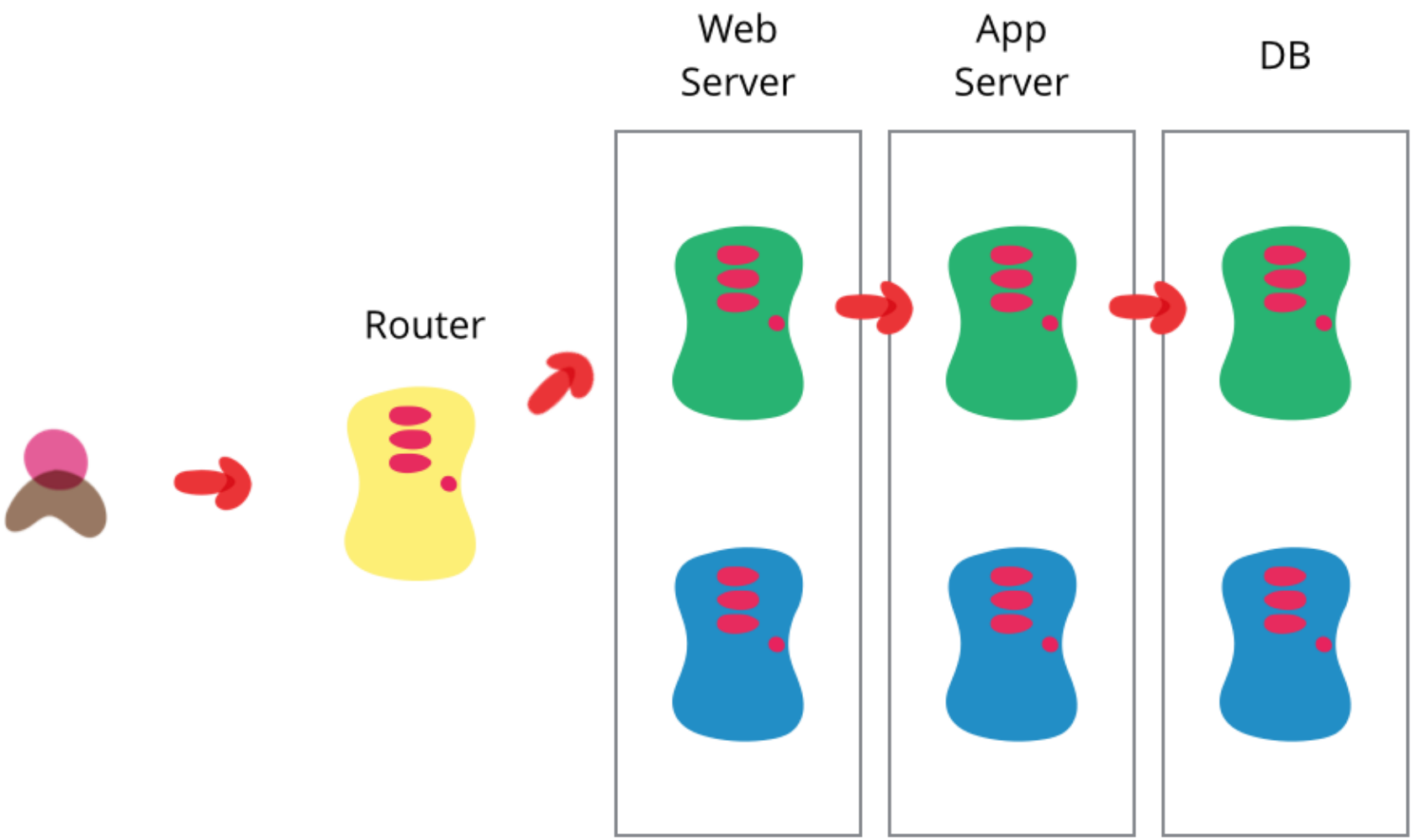
but also used by our customers to send us information

HOW ARE WE DOING IT

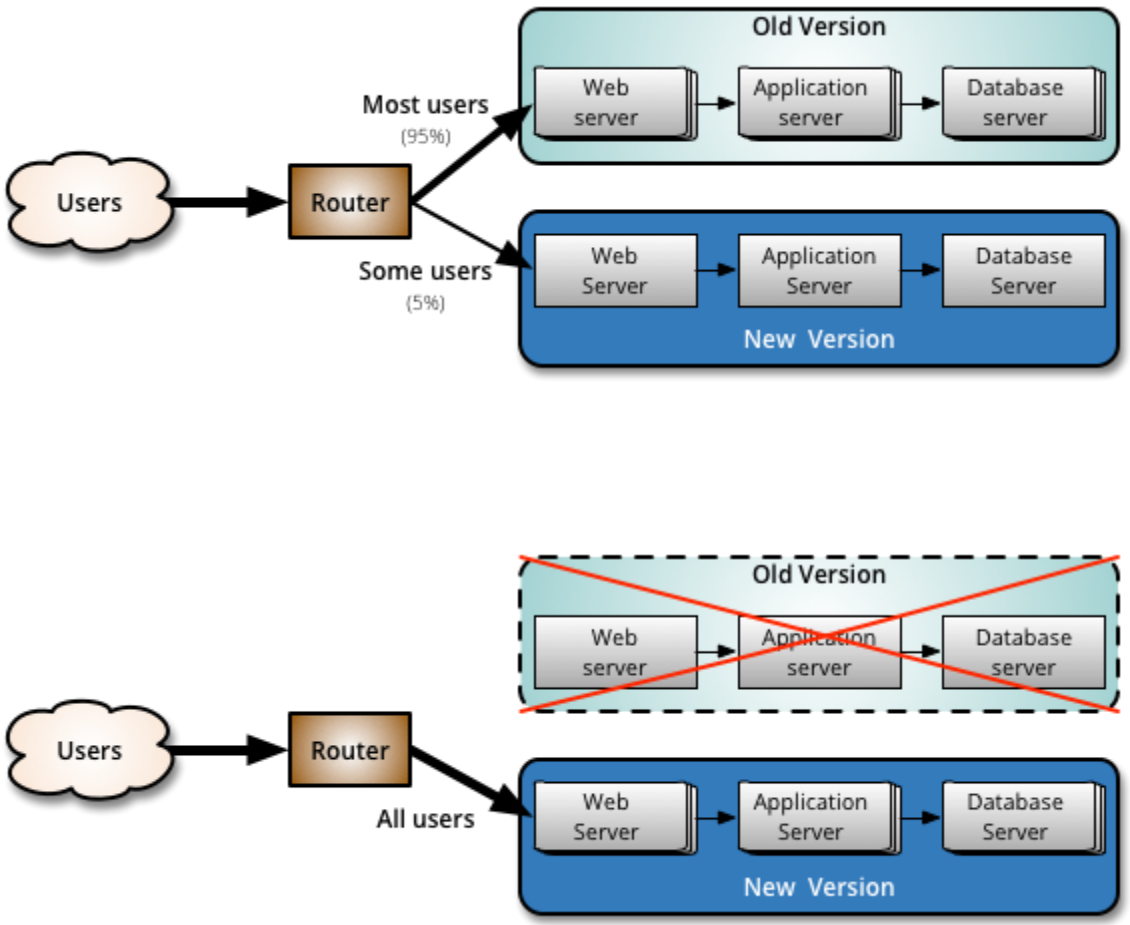
Feedback is published to Github PR



BLUE-GREEN UPGRADES



CANARY DEPLOYMENTS



SCALING

New and interesting problems

AWS

Resource limits: VPCs, S3 snapshots, some instance sizes

Rate limits: affect the whole account

Always use different accounts for testing/production and possibly different teams

Retrying is your friend, but with exponential backoff

EMBRACE FAILURE!

OPENSTACK

Custom flavors

Custom images

Different CLI commands

There are not two OpenStack installations that are the same

DANKE!

RATE THIS SESSION IN AGILETESTINGDAYS.COM

csanchez.org


csanchez


carlossg

cloudbees®