
## CI AND CD AT SCALE

# SCALING JENKINS WITH DOCKER AND APACHE MESOS

Carlos Sanchez

csanchez.org / @csanchez

See online at http://carlossg.github.io/presentations

# ABOUT ME

Senior Software Engineer @ CloudBees

Contributor to the Jenkins Mesos plugin and the Java Marathon client

Author of Jenkins Kubernetes plugin

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

# OUR USE CASE

Scaling Jenkins

Your mileage may vary

# SCALING JENKINS

Two options:

- More build agents per master
- More masters

# SCALING JENKINS: MORE BUILD AGENTS

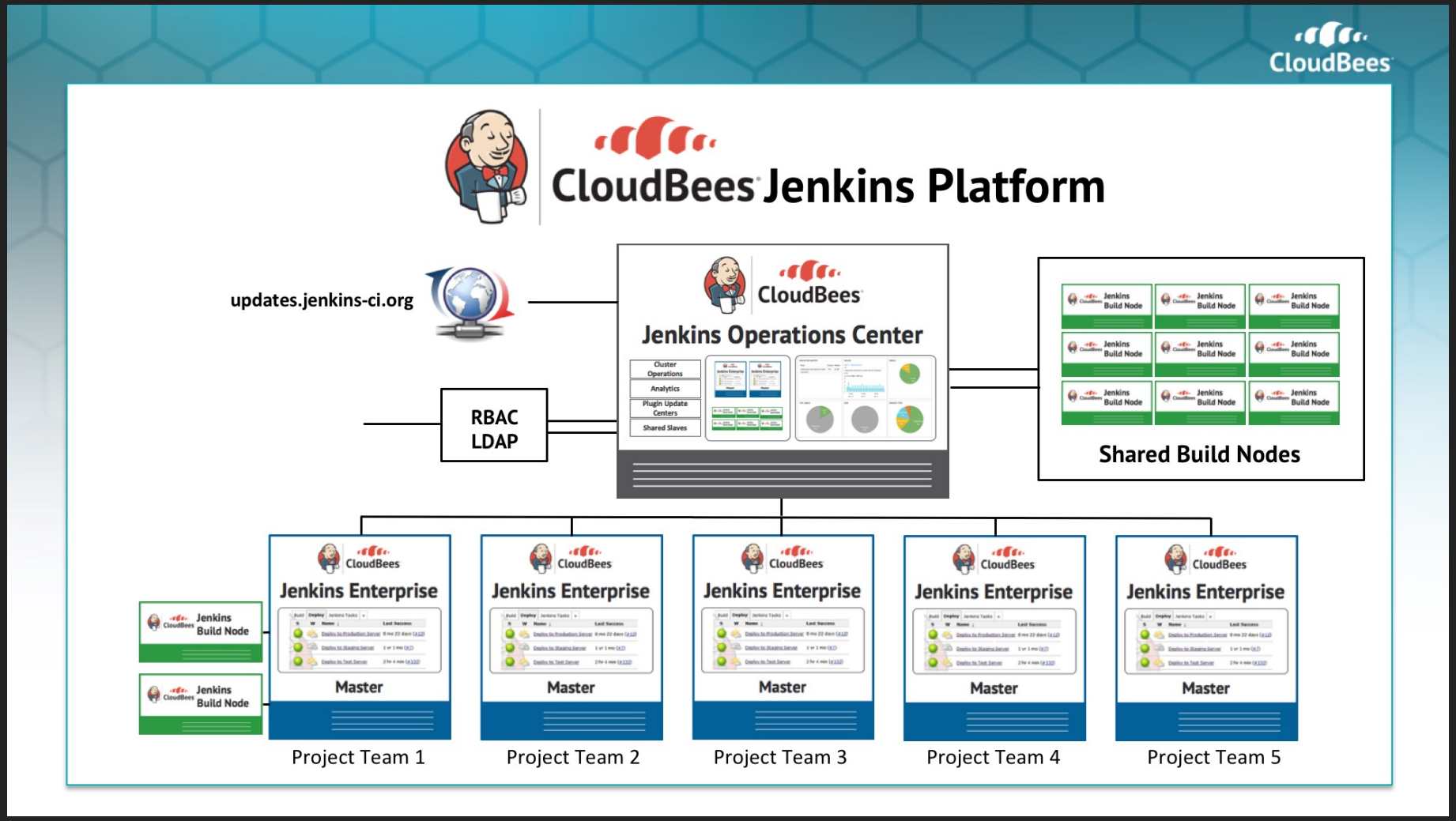
- Pros
  - Multiple plugins to add more agents, even dynamically
- Cons
  - The master is still a SPOF
  - Handling multiple configurations, plugin versions,...
  - There is a limit on how many build agents can be attached

# SCALING JENKINS: MORE MASTERS

- Pros
  - Different sub-organizations can self service and operate independently
- Cons
  - Single Sign-On
  - Centralized configuration and operation

# CLOUDBEES JENKINS ENTERPRISE EDITION

## CloudBees Jenkins Operations Center



# CLOUDBEES JENKINS PLATFORM - PRIVATE SAAS EDITION

The best of both worlds

CloudBees Jenkins Operations Center with multiple masters

Dynamic build agent creation in each master

ElasticSearch for Jenkins metrics and Logstash

# BUT IT IS NOT TRIVIAL

# ARCHITECTURE

Docker Docker Docker

**Kernel Sanders**
@lstoll

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

Isolated Jenkins masters

Isolated build agents and jobs

Memory and CPU limits

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

*Kelsey Hightower*

# EMBRACE FAILURE!

# CLUSTER SCHEDULING

- Running in public cloud, private cloud, VMs or bare metal
  - Starting with AWS and OpenStack
- HA and fault tolerant
- With Docker support of course

# MESOSPHERE MARATHON

# TERRAFORM

# 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.


- Preinstall packages: Mesos, Marathon, Docker
- Cached docker images
- Other drivers: XFS, NFS,...
- Enhanced networking driver (AWS)

# STORAGE

Handling distributed storage

Servers can start in any host of the cluster

And they can move when they are restarted

Jenkins masters need persistent storage, agents (*typically*) don't

Supporting EBS (AWS) and external NFS

# SIDEKICK CONTAINER

A privileged container that manages mounting for other containers

Can execute commands in the host and other containers

# SIDEKICK CONTAINER *CASTLE*

Running in Marathon in each host

```
"constraints": [
  [
    "hostname",
    "UNIQUE"
  ]
]
```

A lot of magic happening with `nsenter`

both in host and other containers

- Jenkins master container requests data on startup using *entrypoint*
  - REST call to Castle
- Castle checks authentication
- Creates necessary storage in the backend
  - EBS volumes from snapshots
  - Directories in NFS backend

- Mounts storage in requesting container
  - EBS is mounted to host, then bind mounted into container
  - NFS is mounted directly in container
- Listens to Docker event stream for killed containers

# CASTLE: BACKUPS AND CLEANUP

Periodically takes S3 snapshots from EBS volumes in AWS

Cleanups happening at different stages and periodically

**EMBRACE FAILURE!**

# PERMISSIONS

Containers should not run as root

Container user id != host user id

i.e. `jenkins` user in container is always 1000 but matches `ubuntu` user in host

# CAVEATS

Only a limited number of EBS volumes can be mounted

Docs say `/dev/sd[f-p]`, but `/dev/sd[q-z]` seem to work too

Sometimes the device gets corrupt and no more EBS volumes can be mounted there

NFS users must be centralized and match in cluster and NFS server

# MEMORY

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?

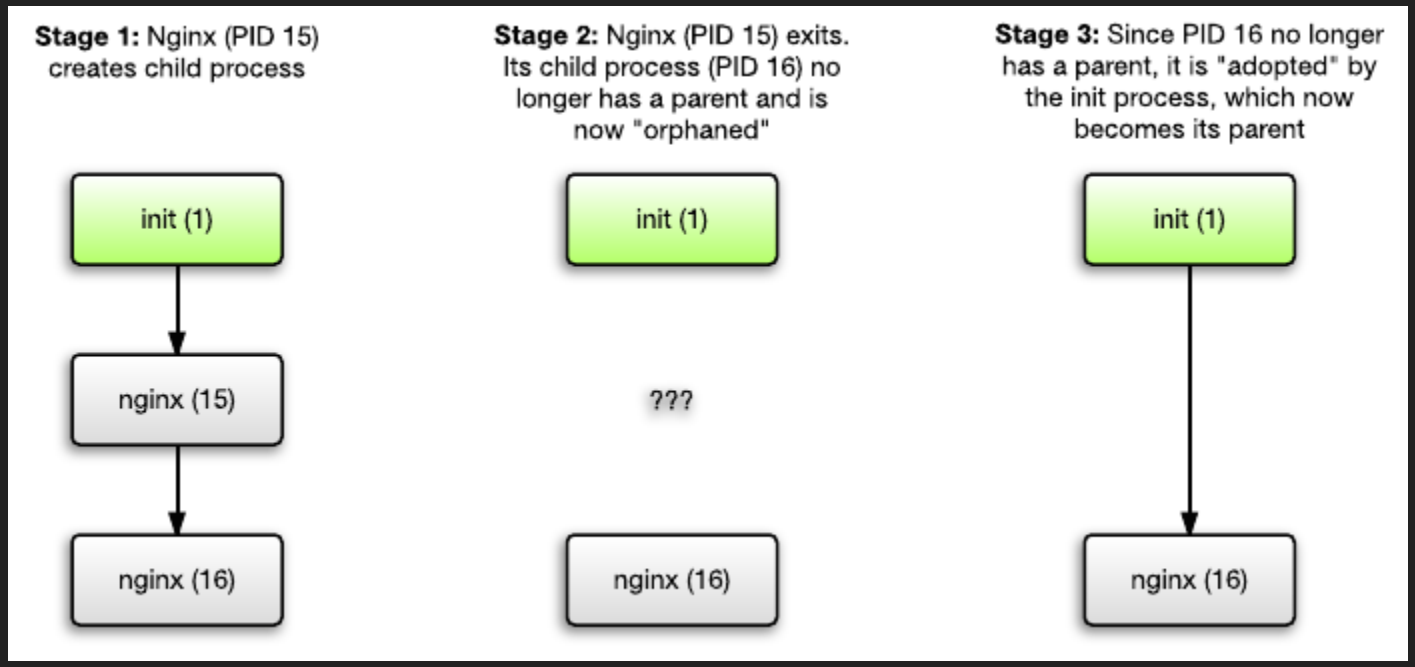

iCACHONDEO.COM

# WHAT ABOUT THE JVM?

# WHAT ABOUT THE CHILD PROCESSES?

# OTHER CONSIDERATIONS

# ZOMBIE REAPING PROBLEM




Zombie processes are processes that have terminated but have not (yet) been waited for by their parent processes.

The init process -- PID 1 -- task is to "adopt" orphaned child processes

source

# THIS IS A PROBLEM IN DOCKER

Jenkins build agent run multiple processes

But Jenkins masters too, and they are long running

# TINI

Systemd or SysV init is too heavyweight for containers

*All Tini does is spawn a single child (Tini is meant to be run in a container), and wait for it to exit all the while reaping zombies and performing signal forwarding.*

# PROCESS REAPING

Docker 1.9 gave us trouble at scale, rolled back to 1.8

Lots of *defunct* processes

# NETWORKING

Jenkins masters open several ports

- HTTP
- JNLP Build agent
- SSH server (Jenkins CLI type operations)

# NETWORKING: HTTP

We use a simple `nginx` reverse proxy for

- Mesos
- Marathon
- ElasticSearch
- CJOC
- Jenkins masters

Gets destination host and port from Marathon

# NETWORKING: HTTP

Doing both

- domain based routing `master1.pse.example.com`
- path based routing `pse.example.com/master1`
  - because not everybody can touch the DNS or get a wildcard SSL certificate

# NETWORKING: JNLP

Build agents started dynamically in Mesos cluster can connect to masters internally

Build agents manually started outside cluster get host and port destination from HTTP, then connect directly

# NETWORKING: SSH

SSH Gateway Service

Tunnel SSH requests to the correct host

Simple configuration needed in client

```
Host=*.ci.cloudbees.com
ProxyCommand=ssh -q -p 22 ssh.ci.cloudbees.com tunnel %h
```

allows to run

```
ssh master1.ci.cloudbees.com
```

# SCALING

New and interesting problems

# A 300 JENKINS MASTERS CLUSTER

- 3 Mesos masters (m3.xlarge: 4 vCPU, 15GB, 2x40 SSD)
- 80 Mesos slaves (m3.xlarge)
- 7 Mesos slaves dedicated to ElasticSearch: (r3.2xlarge: 8 vCPU, 61GB, 1x160 SSD)

**Total: 1.5TB 376 CPUs**

Running 300 masters and ~3 concurrent jobs per master

Masters: 2GB 0.1 CPU / Build agents: 512MB 0.1 CPU

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | master-0286 | | 3 | 3 | 1 | 1.642.2.1 | |
| | | master-0287 | | 3 | 2 | 1 | 1.642.2.1 | |
| | | master-0288 | | 3 | 0 | 1 | 1.642.2.1 | |
| | | master-0289 | | 3 | 0 | 3 | 1.642.2.1 | |
| | | master-0290 | | 3 | 0 | 2 | 1.642.2.1 | |
| | | master-0291 | | 3 | 0 | 1 | 1.642.2.1 | |
| | | master-0292 | | 3 | 3 | 1 | 1.642.2.1 | |
| | | master-0293 | | 3 | 3 | 1 | 1.642.2.1 | |
| | | master-0294 | | 3 | 0 | 2 | 1.642.2.1 | |
| | | master-0295 | | 3 | 2 | 1 | 1.642.2.1 | |
| | | master-0296 | | 3 | 1 | 2 | 1.642.2.1 | |
| | | master-0297 | | 3 | 0 | 2 | 1.642.2.1 | |
| | | master-0298 | | 3 | 0 | 2 | 1.642.2.1 | |
| | | master-0299 | | 3 | 0 | 1 | 1.642.2.1 | |
| | | master-0300 | | 3 | 0 | 2 | 1.642.2.1 | |

Mesos Frameworks Slaves Offers

## Slaves

| | |
|---|---|
| Activated | 87 |
| Deactivated | 0 |

## Tasks

| |
|---|
| Staged |
| Started |
| Finished |
| Killed |
| Failed |
| Lost |

## Resources

| | CPUs | Mem |
|---|---|---|
| Total | 376 | 1507.9 GB |
| Used | 192.500 | 1457.7 GB |
| Offered | 0 | 0 B |
| Idle | 183.500 | 50.2 GB |

| | | | | | |
|---|---|---|---|---|---|
| 8fb234eb6d5e47049a1d07f0e297cd97-mesos | 8fb234eb6d5e47049a1d07f0e297cd97-mesos | | minutes ago | 36.compute-1.amazonaws.com | |
| mesos-jenkins-be97d6997b6e473d8acabea8ef8587f8-mesos | task mesos-jenkins-be97d6997b6e473d8acabea8ef8587f8-mesos | RUNNING | 24 minutes ago | ec2-54-164-181-123.compute-1.amazonaws.com | Sandbox |
| mesos-jenkins-0727b10e0bdd4711b34470bef33e2ff9-mesos | task mesos-jenkins-0727b10e0bdd4711b34470bef33e2ff9-mesos | RUNNING | 25 minutes ago | ec2-54-85-24-59.compute-1.amazonaws.com | Sandbox |
| mesos-jenkins-c0e330cab95b410b929a1e01cb93e108-mesos | task mesos-jenkins-c0e330cab95b410b929a1e01cb93e108-mesos | RUNNING | 25 minutes ago | ec2-54-165-41-44.compute-1.amazonaws.com | Sandbox |
| mesos-jenkins-7786f1fa4ea24d2a904c35095dcdd157-mesos | task mesos-jenkins-7786f1fa4ea24d2a904c35095dcdd157-mesos | RUNNING | 25 minutes ago | ec2-54-175-146-38.compute-1.amazonaws.com | Sandbox |
| mesos-jenkins-df03482cbf8644998b6712489c73268e-mesos | task mesos-jenkins-df03482cbf8644998b6712489c73268e-mesos | RUNNING | 25 minutes ago | ec2-54-175-113-162.compute-1.amazonaws.com | Sandbox |
| mesos-jenkins-cbf3857cfd8045698bc3e56b7af8c6e8-mesos | task mesos-jenkins-cbf3857cfd8045698bc3e56b7af8c6e8-mesos | RUNNING | 26 minutes ago | ec2-54-164-243-131.compute-1.amazonaws.com | Sandbox |
| mesos-jenkins-d3a4b6a5d72f497ca03b2c8d657f59e0-mesos | task mesos-jenkins-d3a4b6a5d72f497ca03b2c8d657f59e0-mesos | RUNNING | 28 minutes ago | ec2-54-83-61-112.compute-1.amazonaws.com | Sandbox |

| | | | | | |
|---|---|---|---|---|---|
| /masters/master-0286 | 2048 | 0.2 | 1 / 1 | | Running |
| /masters/master-0287 | 2048 | 0.2 | 1 / 1 | | Running |
| /masters/master-0288 | 2048 | 0.2 | 1 / 1 | | Running |
| /masters/master-0289 | 2048 | 0.2 | 1 / 1 | | Running |
| /masters/master-0290 | 2048 | 0.2 | 1 / 1 | | Running |
| /masters/master-0291 | 2048 | 0.2 | 1 / 1 | | Running |
| /masters/master-0292 | 2048 | 0.2 | 1 / 1 | | Running |
| /masters/master-0293 | 2048 | 0.2 | 1 / 1 | | Running |
| /masters/master-0294 | 2048 | 0.2 | 1 / 1 | | Running |
| /masters/master-0295 | 2048 | 0.2 | 1 / 1 | | Running |
| /masters/master-0296 | 2048 | 0.2 | 1 / 1 | | Running |
| /masters/master-0297 | 2048 | 0.2 | 1 / 1 | | Running |
| /masters/master-0298 | 2048 | 0.2 | 1 / 1 | | Running |
| /masters/master-0299 | 2048 | 0.2 | 1 / 1 | | Running |
| /masters/master-0300 | 2048 | 0.2 | 1 / 1 | | Running |

cluster | nodes | rest | more | elasticsearch @ Xandu

7 nodes | 36 indices | 1,080 shards | 13,905,081 docs ↑ 1,892 | 29.62GB ↑ 52.31MB

filter nodes by name | ☑ master | ☑ data | ☑ client

| name | load average | | cpu % | | heap usage % | | disk usage % | | uptime |
|---|---|---|---|---|---|---|---|---|---|
| Aminedi<br>6d8918690840<br>10.16.177.217:31090<br>JVM: 1.8.0_66-internal ES: 1.7.3 | 0.7 | 3min: 0.6<br>5min: 0.6 | 3.0 | user: 3<br>sys: 0 | 50.0 | used: 13.67GB<br>max: 27.28GB | 8.0 | free: 135.34GB<br>total: 147.51GB | 12d. |
| Basilisk<br>0bc739541f3a<br>10.16.191.236:31090<br>JVM: 1.8.0_66-internal ES: 1.7.3 | 0.6 | 3min: 0.4<br>5min: 0.3 | 4.0 | user: 4<br>sys: 0 | 71.0 | used: 19.45GB<br>max: 27.28GB | 8.0 | free: 135.26GB<br>total: 147.51GB | 12d. |
| Daytripper<br>edf9aa370ce1<br>10.16.136.3:31090<br>JVM: 1.8.0_66-internal ES: 1.7.3 | 0.8 | 3min: 0.6<br>5min: 0.5 | 4.0 | user: 4<br>sys: 0 | 26.0 | used: 7.27GB<br>max: 27.28GB | 8.0 | free: 135.22GB<br>total: 147.51GB | 12d. |
| Kymaera<br>3cdd93388aa1<br>10.16.204.83:31090<br>JVM: 1.8.0_66-internal ES: 1.7.3 | 0.1 | 3min: 0.3<br>5min: 0.3 | 4.0 | user: 4<br>sys: 0 | 32.0 | used: 8.95GB<br>max: 27.28GB | 8.0 | free: 136.16GB<br>total: 147.51GB | 12d. |
| Man-Bull<br>52aa7d5d0a38<br>10.16.29.39:31090<br>JVM: 1.8.0_66-internal ES: 1.7.3 | 0.6 | 3min: 0.6<br>5min: 0.6 | 3.0 | user: 3<br>sys: 0 | 32.0 | used: 8.89GB<br>max: 27.28GB | 8.0 | free: 135.27GB<br>total: 147.51GB | 12d. |
| Order<br>6d98bb17dd97<br>10.16.61.74:31090<br>JVM: 1.8.0_66-internal ES: 1.7.3 | 0.3 | 3min: 0.4<br>5min: 0.4 | 4.0 | user: 4<br>sys: 0 | 34.0 | used: 9.36GB<br>max: 27.28GB | 9.0 | free: 134.91GB<br>total: 147.51GB | 12d. |
| Xandu<br>c43e9727582f<br>10.16.160.225:31090<br>JVM: 1.8.0_66-internal ES: 1.7.3 | 0.6 | 3min: 0.6<br>5min: 0.5 | 3.0 | user: 3<br>sys: 0 | 66.0 | used: 18.16GB<br>max: 27.28GB | 8.0 | free: 135.74GB<br>total: 147.51GB | 12d. |

cluster | nodes | rest | more | elasticsearch @ Kymaera

7 nodes | 36 indices | 1,080 shards | 13,903,189 docs ↑ 2,348 | 29.57GB ↑ 89.24MB

filter indices by name | ☑ closed (0) | ☐ special (0) | filter nodes by name | « 1-5 of 36 selected indices

| | builds-20160227<br>shards: 15 * 2 \| docs: 173,098 \| size: 111.81MB | builds-20160228<br>shards: 15 * 2 \| docs: 172,941 \| size: 111.58MB | builds-20160229<br>shards: 15 * 2 \| docs: 173,034 \| size: 111.83MB | builds-20160301<br>shards: 15 * 2 \| docs: 172,659 \| size: 111.45MB | builds-20160302<br>shards: 15 * 2 \| docs: 172,342 \| size: 111.75MB |
|---|---|---|---|---|---|
| Aminedi<br>6d8918690840<br>heap disk cpu load | 3 10 1 9 | 3 10 1 8 | 14 10 13 7 | 14 10 0 13 7 | 12 3 10 4 |
| Basilisk<br>0bc739541f3a<br>heap disk cpu load | 14 0 13 7 8 | 0 7 11 4 | 3 10 1 0 7 | 14 3 10 6 0 | 12 6 11 5 |
| Daytripper<br>edf9aa370ce1<br>heap disk cpu load | 12 6 11 4 | 12 14 1 5 8 | 12 3 9 5 | 12 2 5 8 | 1 0 7 8 |
| Kymaera<br>3cdd93388aa1<br>heap disk cpu load | 3 10 11 4 | 2 11 9 4 | 12 1 5 8 | 1 11 4 8 | 14 13 7 5 |
| Man-Bull<br>52aa7d5d0a38<br>heap disk cpu load | 2 0 9 5 | 6 2 13 9 | 6 13 11 4 | 3 6 13 9 | 14 6 2 1 9 |
| Order<br>6d98bb17dd97<br>heap disk cpu load | 14 2 7 8 | 14 6 0 13 7 | 2 11 4 8 | 1 7 11 4 | 2 0 13 9 8 |
| Xandu<br>c43e9727582f<br>heap disk cpu load | 12 6 1 13 5 | 12 3 10 5 | 14 6 2 0 9 | 12 2 9 5 | 3 10 11 4 |

# TERRAFORM AWS

- Instances
- Keypairs
- Security Groups
- S3 buckets
- ELB
- VPCs

# AWS

Resource limits: VPCs, S3 snapshots, some instance sizes

Rate limits: affect the whole account

Retrying is your friend, but with exponential backoff

# AWS

Running with a patched Terraform to overcome timeouts and AWS *eventual consistency*

```
<?xml version="1.0" encoding="UTF-8"?>
<DescribeVpcsResponse xmlns="http://ec2.amazonaws.com/doc/2015-10-01/
  <requestId>8f855bob-3421-4cff-8c36-4b517eb0456c</requestld>
  <vpcSet>
    <item>
      <vpcId>vpc-30136159</vpcId>
      <state>available</state>
      <cidrBlock>10.16.0.0/16</cidrBlock>
                                ...
</DescribeVpcsResponse>
2016/05/18 12:55:57 [DEBUG] [aws-sdk-go] DEBUG: Response ec2/Describe
--[ RESPONSE] ------------------------------------
HTTP/1.1 400 Bad Request
<Response><Errors><Error><Code>InvalidVpcID.NotFound</Code><Message>
The vpc ID 'vpc-30136159' does not
exist</Message></Error></Errors>
```

# TERRAFORM OPENSTACK

- Instances
- Keypairs
- Security Groups
- Load Balancer
- Networks

# OPENSTACK

Custom flavors

Custom images

Different CLI commands

There are not two OpenStack installations that are the same

# THE FUTURE

New framework using Netflix Fenzo

Runs under marathon, exposes REST API that masters call

- Affinity
- Reduce number of frameworks
- Faster to spawn new build agents because framework is not started
- Pipeline durable builds, can survive a restart of the master
- Dedicated workers for builds

# THANKS

csanchez.org

csanchez

carlossg