cloudbees

FROM MONOLITH TO DOCKER DISTRIBUTED APPLICATIONS

Carlos Sanchez

@csanchez

Watch online at carlossg.github.io/presentations

ABOUT ME
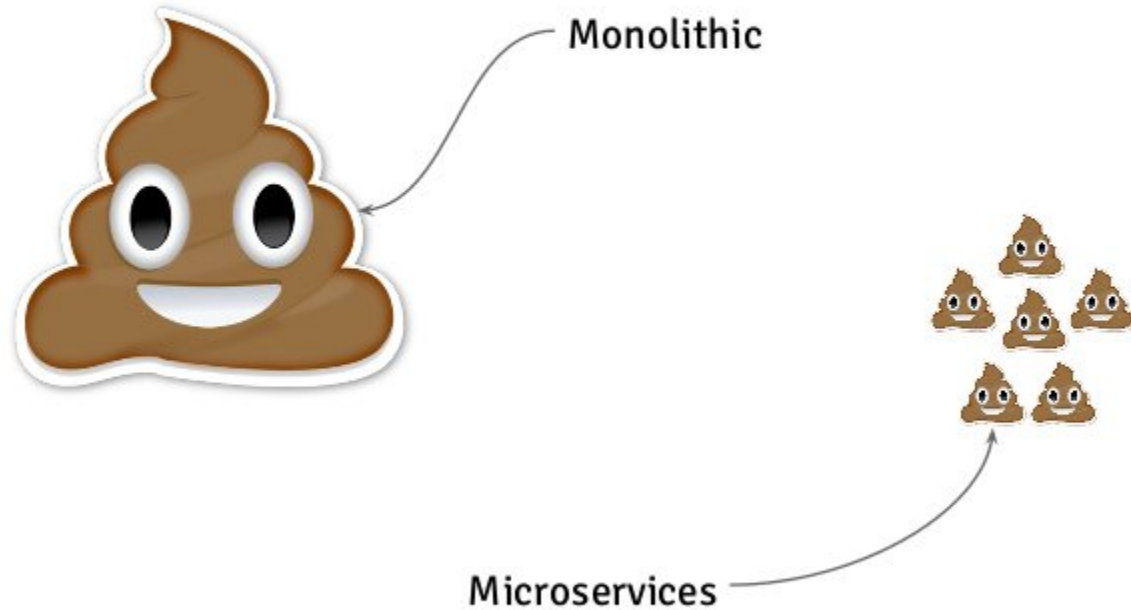
Senior Software Engineer @ CloudBees

Author of Jenkins Kubernetes plugin

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

DOCKER DOCKER DOCKER

Monolithic vs Microservices

Monolithic

Microservices

@alvaro_sanchez

odobo

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
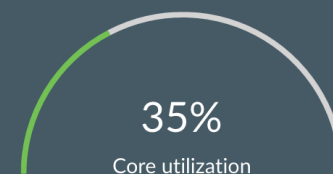
Used cores: 1300.5

Total cores: 3748

Workload

1,500

1,000

500

09:44   09:45   09:46   09:47
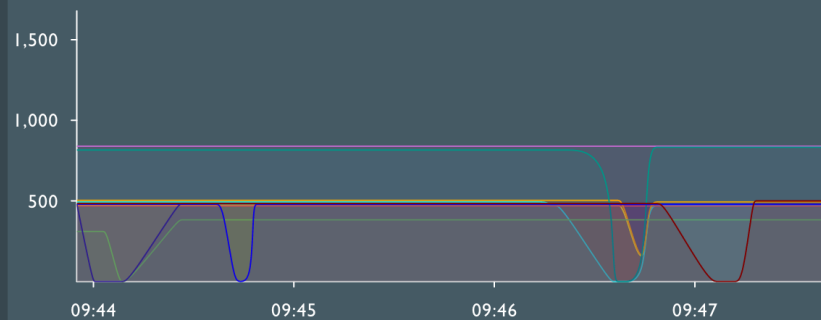
■ worker-257 ■ worker-260 ■ worker-279 ■ worker-154 ■ worker-280
■ worker-308 ■ worker-306 ■ worker-4 ■ worker-313 ■ worker-328

Executors

6,000

4,000

2,000

09:38 09:39 09:40 09:41 09:42 09:43 09:44 09:45 09:46 09:47
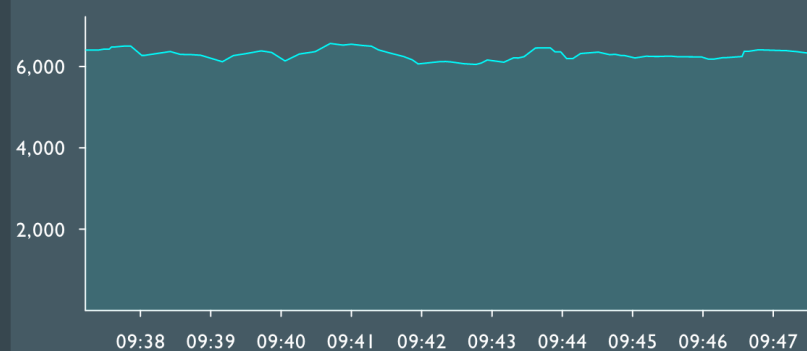
Administration

Cluster Summary   Virtual Machines   Masters

Masters

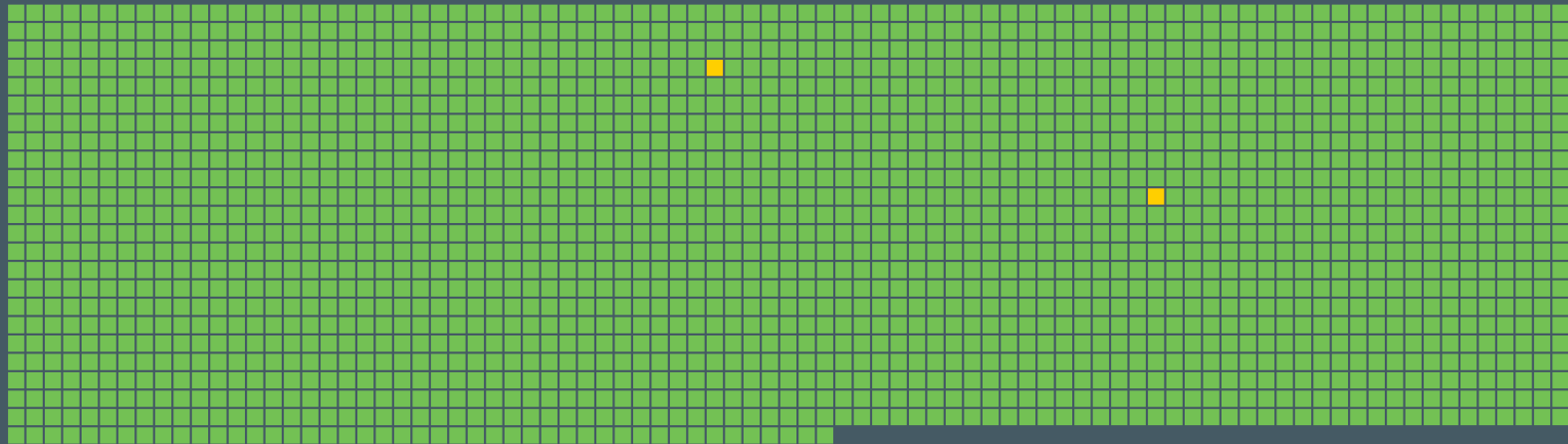test3-1990

2000

Cloudbees
Jenkins Platform
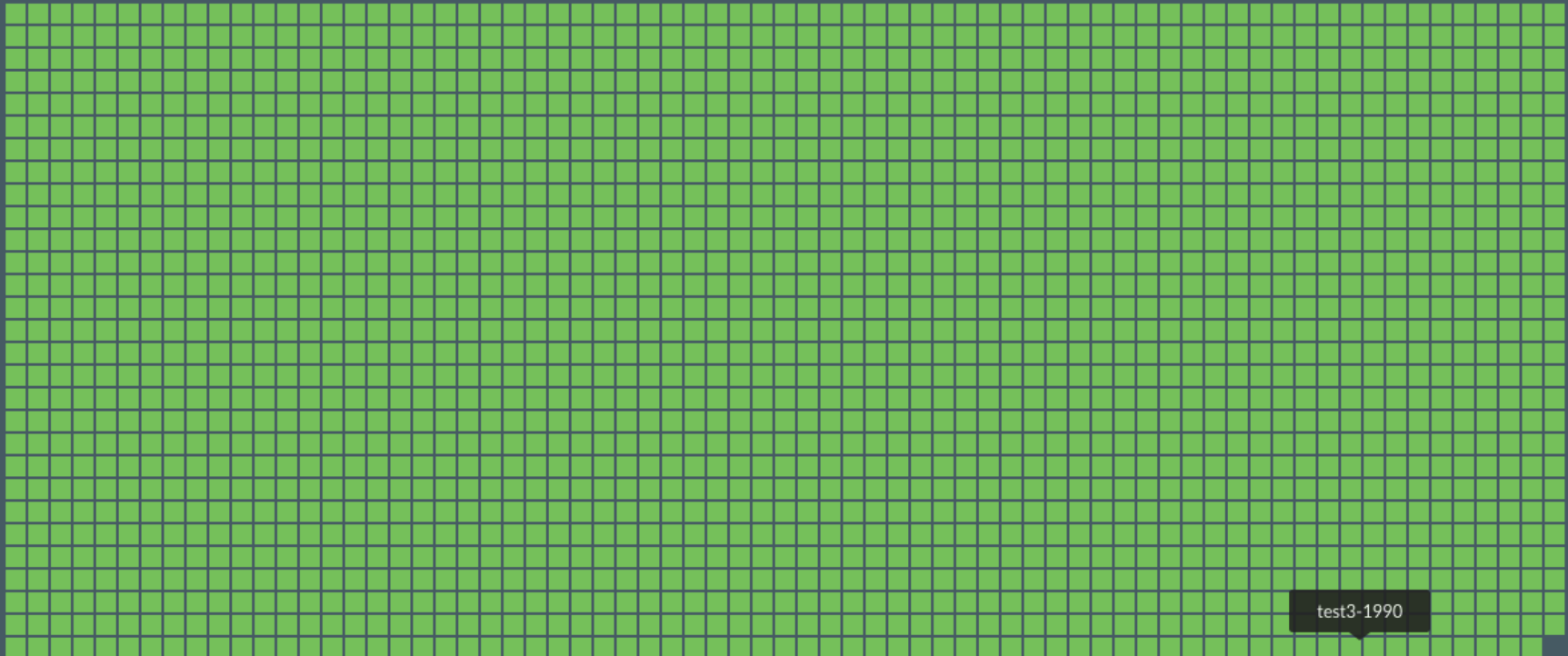
Tiger

Cluster Summary   Virtual Machines   Masters

Masters

2000

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)

2.2TB - 2758 cores

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

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 .

DOCKER PULL COMMAND

```
docker pull jenkins
```

DESCRIPTION

Official Jenkins Docker image

jenkinsci/jnlp-slave ☆

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

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

HA and fault tolerant
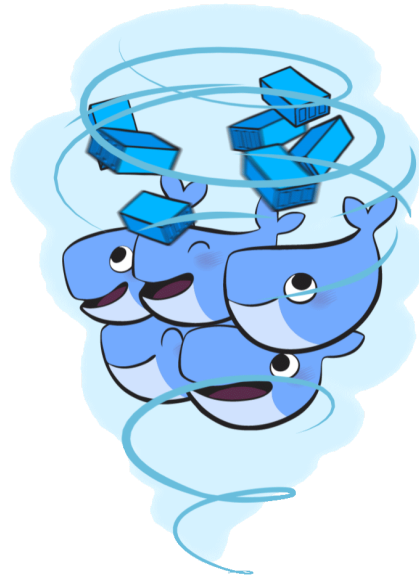
With Docker support of course

APACHE MESOS
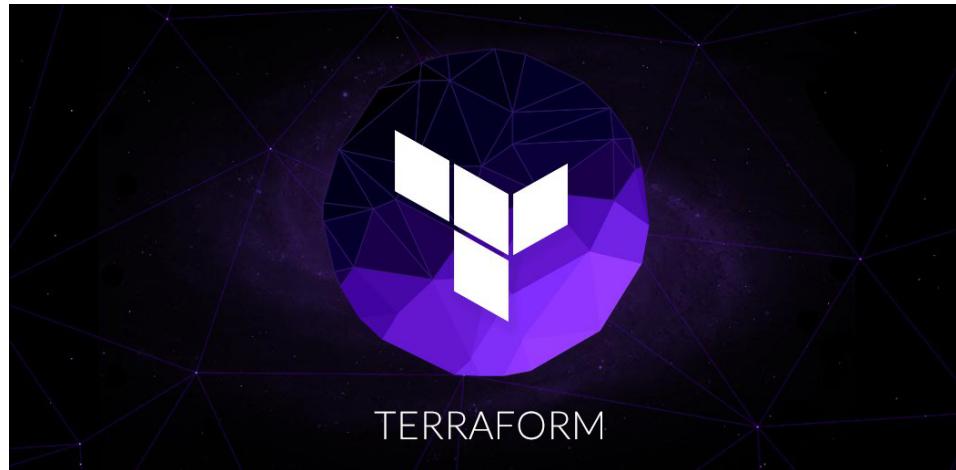

A distributed systems kernel

ALTERNATIVES

Docker Swarm / Kubernetes

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
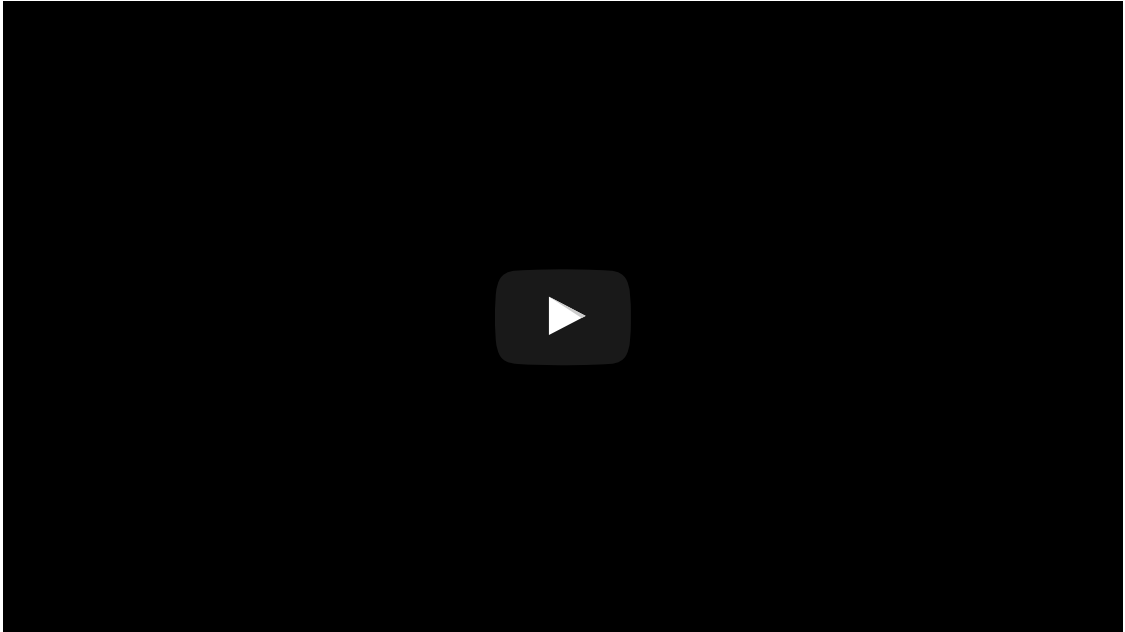 - Changing image id will restart all instances

@DEVOPS_BORAT
DevOps Borat
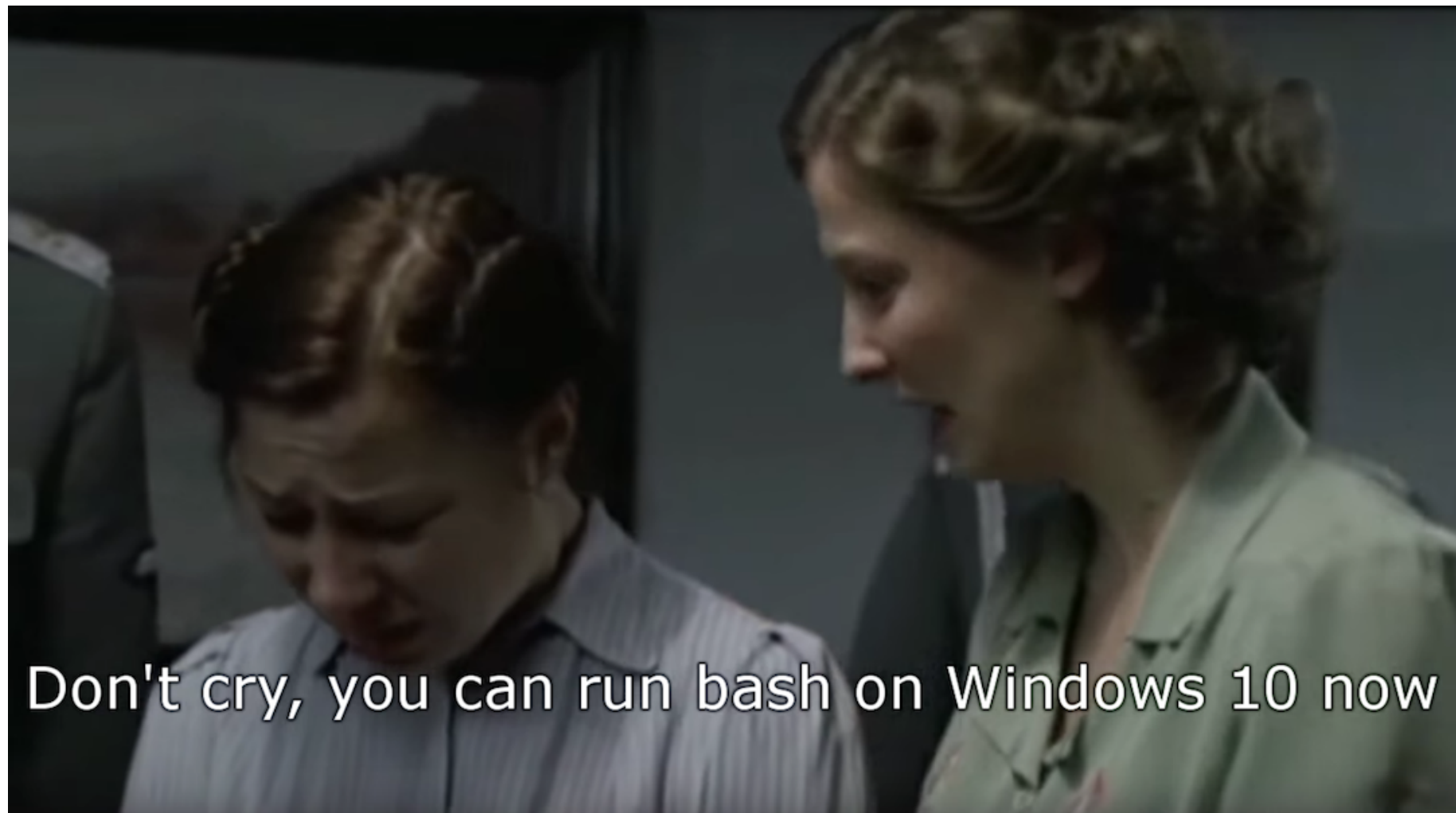
To make error is human. To propagate error to all server in automatic way is #devops.

IF YOU HAVEN'T AUTOMATICALLY DESTROYED SOMETHING BY MISTAKE,

YOU ARE NOT AUTOMATING ENOUGH

If you never used Docker in production, leave the room now

STORAGE

Handling distributed storage

Servers can start in any host of the cluster

And they can move when they are restarted

DOCKER VOLUME PLUGINS

- Flocker
- GlusterFS
- NFS
- EBS

KUBERNETES

- GCE disks
- Flocker
- GlusterFS
- NFS
- EBS

SIDEKICK CONTAINER

A privileged container that manages mounting for other containers

Can execute commands in the host and other containers

A lot of magic happening with `nsenter`

IN OUR CASE

Sidekick container

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

Supporting EBS (AWS) and external NFS

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

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?

CPU

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

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

Mesos/Kubernetes CPU translates into Docker `--cpu-shares`

NETWORKING

Multiple services running in the same ports

Must redirect from random ports in the host

Services running in one host need to access services in other hosts

NETWORKING: SERVICE DISCOVERY

DNS is not great, caching can happen at multiple levels

`marathon-lb` uses `haproxy` and Marathon API

A typical `nginx` reverse proxy is also easy to setup

NETWORKING: SOFTWARE DEFINED NETWORKS

Create new custom networks on top of physical networks

Allow grouping containers in subnets

NETWORKING: SOFTWARE DEFINED NETWORKS

Battlefield: Calico, Flannel, Weave and Docker Overlay Network

http://chunqi.li/2015/11/15/Battlefield-Calico-Flannel-Weave-and-Docker-Overlay-Network/

SCALING

New and interesting problems

LOGGING

Running ElasticSearch as a cluster service, and the ELK stack

Docker configured to log to syslog

Logstash redirecting syslog to ElasticSearch

Embedded Kibana dashboard in CloudBees Jenkins Operations Center

AWS

Resource limits: VPCs, S3 snapshots, some instance sizes

Rate limits: affect the whole account

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

UPGRADES / MAINTENANCE

Moving containers from hosts

Draining hosts

Rolling updates

Blue/Green deployment

Immutable infrastructure

THANKS

csanchez.org

csanchez

carlossg

cloudbees