

ChaOps

Chaos Engineering in a world of Dev**Ops**

Who am I?

DevOps* minded open, **Cloud** focused **Open Source** enthusiast with an increasingly growing passion for **Containers**, **Orchestrators**, **Microservices** and **Chaos Engineering**.

*(a lot) more Ops than Dev



Red Hat Premier Business Partner

ChaOps

Chaos Engineering in a world of DevOps

DevOps?

IMMIGRATION IS A LOT LIKE DEVOPS



YOU NEED CONTAINERS

**ONE DOES NOT
SIMPLY**



DO DEVOPS WITHOUT CULTURE CHANGE

BACK IN MY DAY,



**DEVOPS WERE
JUST DEVS**

THE 1ST RULE OF DEVOPS CLUB



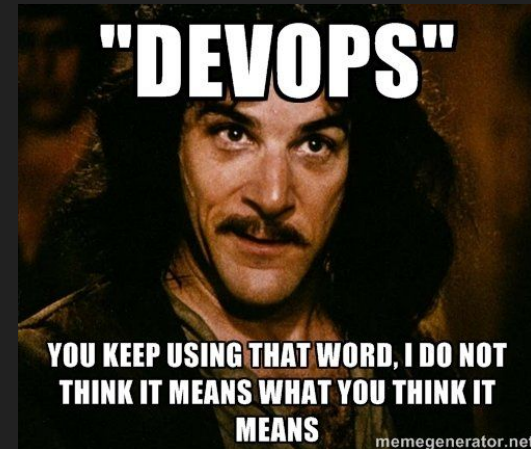
**IS TO CONTINUOUSLY DEPLOY
INTO PRODUCTION**

**AUTOMATIC BUILD, CONTINUOUS
TEST AND CONTINUOUS DEPLOY ISSUES ?**



WELCOME TO DEVOPS !!!

"DEVOPS"



**YOU KEEP USING THAT WORD, I DO NOT
THINK IT MEANS WHAT YOU THINK IT
MEANS**

What is DevOps NOT?

DevOps is NOT just tools...

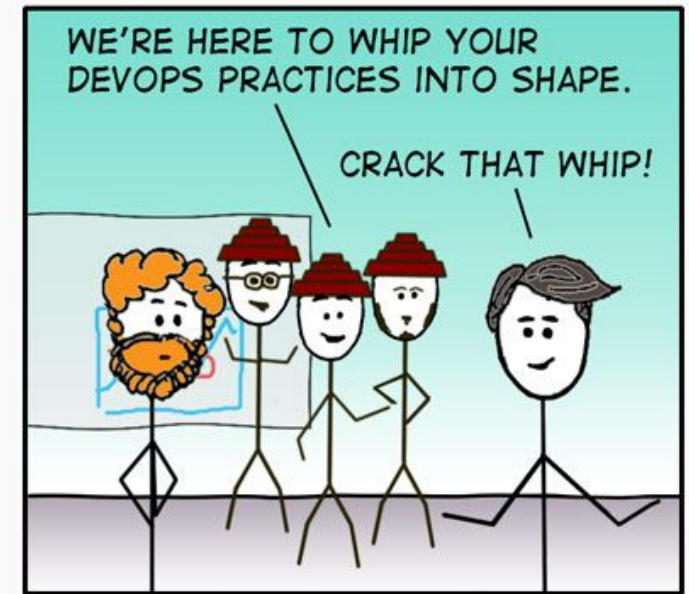
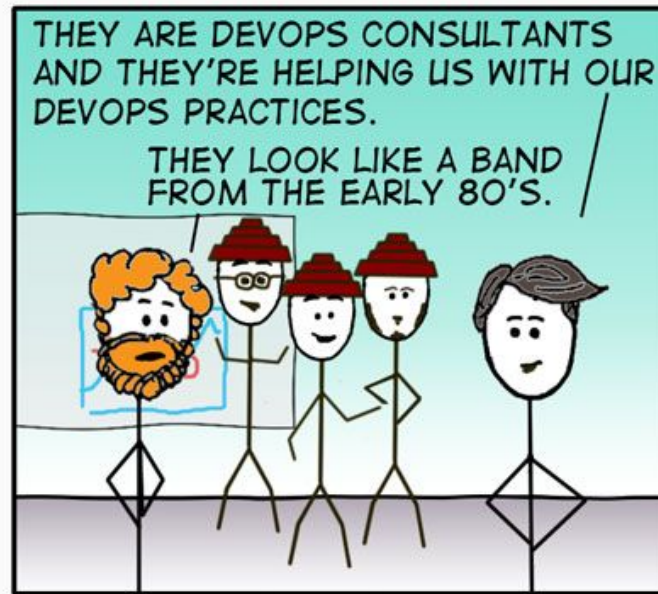
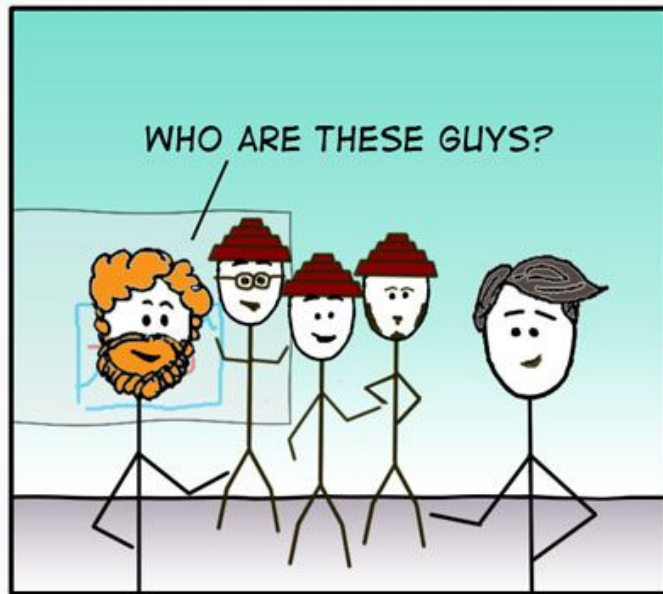
YOU FOOL!

It's not about the tools, it's about the **CULTURE!**



<http://sandramiles.files.wordpress.com/2011/07/argument-cartoon.jpg>

DevOps is NOT just a project...



DevOps is NOT development + operations...

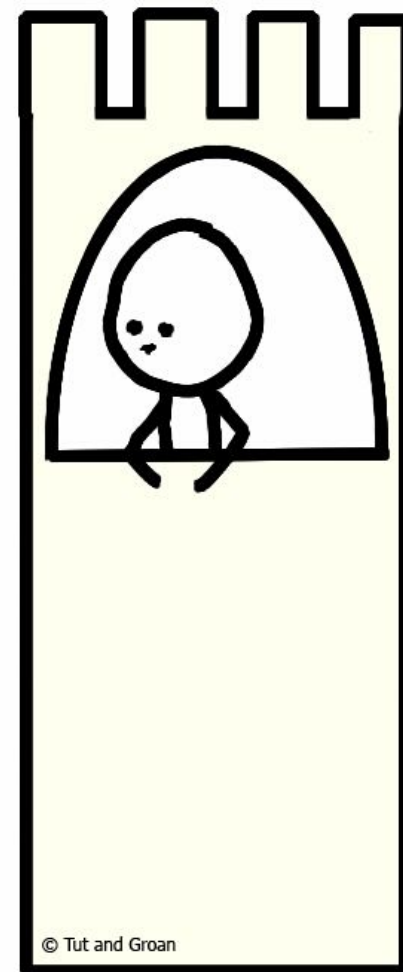


DevOps is NOT a separate team...

You come across
a bit snobby in
some of these...



Speak up-
the acoustics in
my ivory tower
are TERRIBLE



DevOps is NOT just copy/paste...

WE ARE NOT



NETFLIX

BATMANCOMIC..memegenerator.net

DevOps is NOT foolproof...

i wonder if this
will bounce



theycantalk.com



maybe that one will



ChaOps

Chaos Engineering in a world of DevOps

Chaos Engineering?

“Chaos Engineering is the discipline of experimenting on a distributed system in order to build confidence in the system’s capability to withstand turbulent conditions in production.”

- Netflix -

<http://principlesofchaos.org/>

Why tempt fate?

*“If it can go wrong,
it will...”*

- Murphy -



*“Anything that can go wrong,
will - at the worst possible moment.”*

- Finagle -







TESTING IS FOR WIMPS



**REAL MEN TEST IN
PRODUCTION**

Troll.me

How to start with Chaos Engineering?

Step 1: Define your “steady state”



Step 2: Design for failure



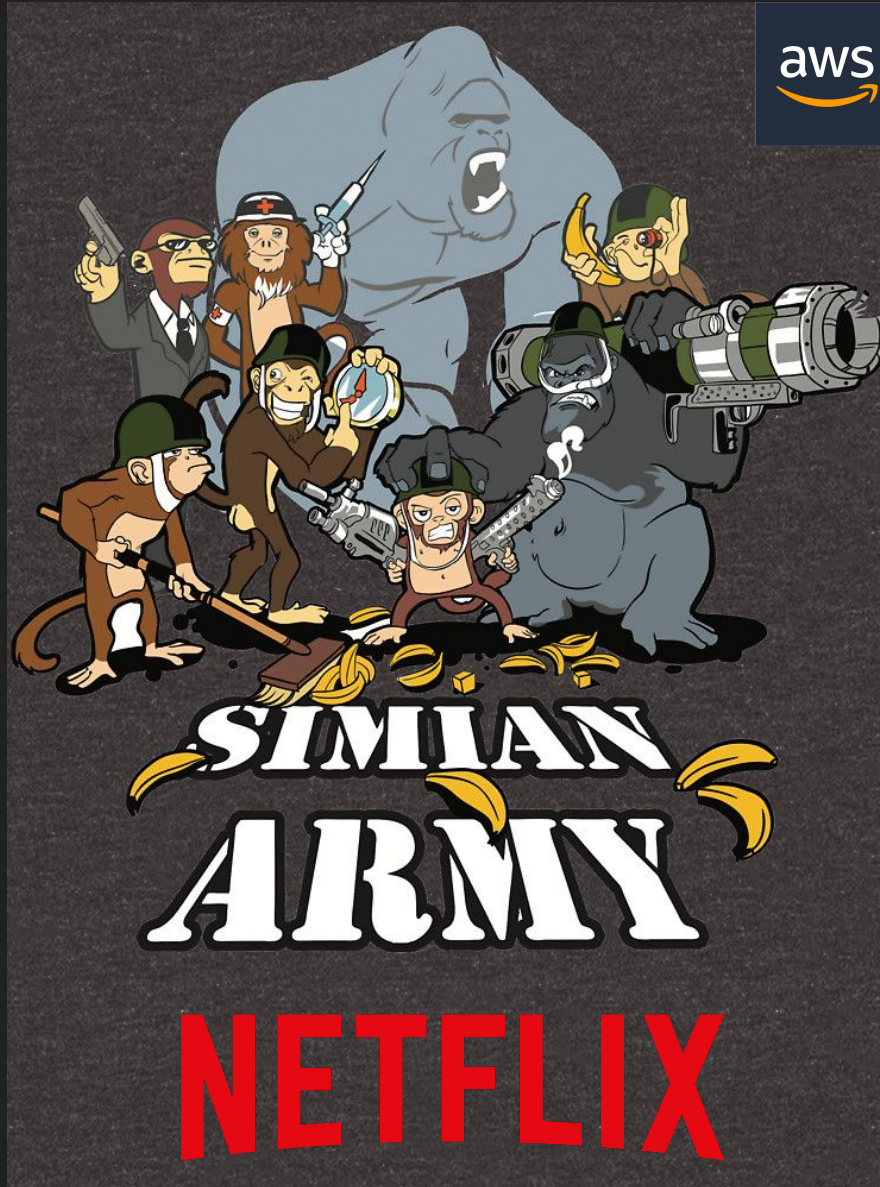
Step 3: Start small



Step 4: Learn & Improve



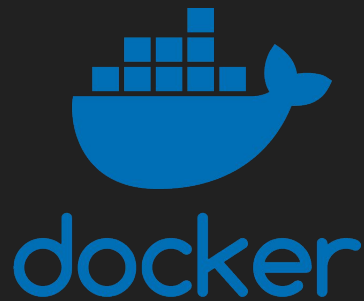
Chaos Engineering tools?



ChaOps

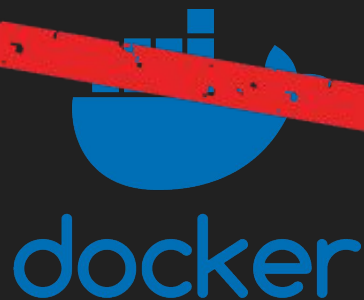
Designing for failure

Immutable Infrastructure





Immutable Infrastructure



Infrastructure as Code



Infrastructure as Code



Time IT

Bestand

Bewerken

Beheer

Rapportage

Extra

Help

Infrastructure as Code



```
MacSteven:scalable-openshift trescst$ ansible-playbook -i inventory/ec2.py destroy.yml -e @vars/eu.yml
```

```
PLAY [tag_master_eu[0]] *****
```

```
TASK [Delete DNS record] *****
changed: [35.159.0.118]
```

```
PLAY [localhost] *****
```

```
TASK [Get all health checks] *****
ok: [localhost]
```

```
TASK [Remove demo app DNS route] *****
changed: [localhost] => (item={u'HealthCheckConfig': {u'FullyQualifiedDomainName': u'autoscaler.eu.gluo.io', u'ResourcePath': u'/', u'Inverted': False, u'MeasureLatency': False, u'RequestInterval': 30, u'Type': u'HTTP', u'Port': 80, u'FailureThreshold': 3}, u'CallerReference': u'88efa5dc-45cb-4631-925b-875e5240a463', u'HealthCheckVersion': 1, u'Id': u'4753c64c-54bc-4959-8407-3604cfdcf730'}})
skipping: [localhost] => (item={u'HealthCheckConfig': {u'FullyQualifiedDomainName': u'autoscaler.eu.gluo.io', u'Inverted': False, u'MeasureLatency': False, u'RequestInterval': 30, u'Type': u'HTTP', u'Port': 80, u'FailureThreshold': 1}, u'HealthCheckVersion': 2, u'Id': u'bf5ff847-86c2-437a-a6f9-4a7faa5ab5b5'}})
```

```
TASK [Remove demo app health check] *****
changed: [localhost]
```

```
TASK [Remove node lifecycle hooks] *****
changed: [localhost] => (item={u'group': u'asg-nodes', u'name': u'scaleup-nodes'})
changed: [localhost] => (item={u'group': u'asg-nodes', u'name': u'scaledown-nodes'})
changed: [localhost] => (item={u'group': u'asg-master', u'name': u'scaleup-masters'})
changed: [localhost] => (item={u'group': u'asg-master', u'name': u'scaledown-masters'})
```

```
TASK [Delete auto scaling groups] *****
changed: [localhost] => (item=asg-master)
changed: [localhost] => (item=asg-nodes)
```

Build resilience into your applications



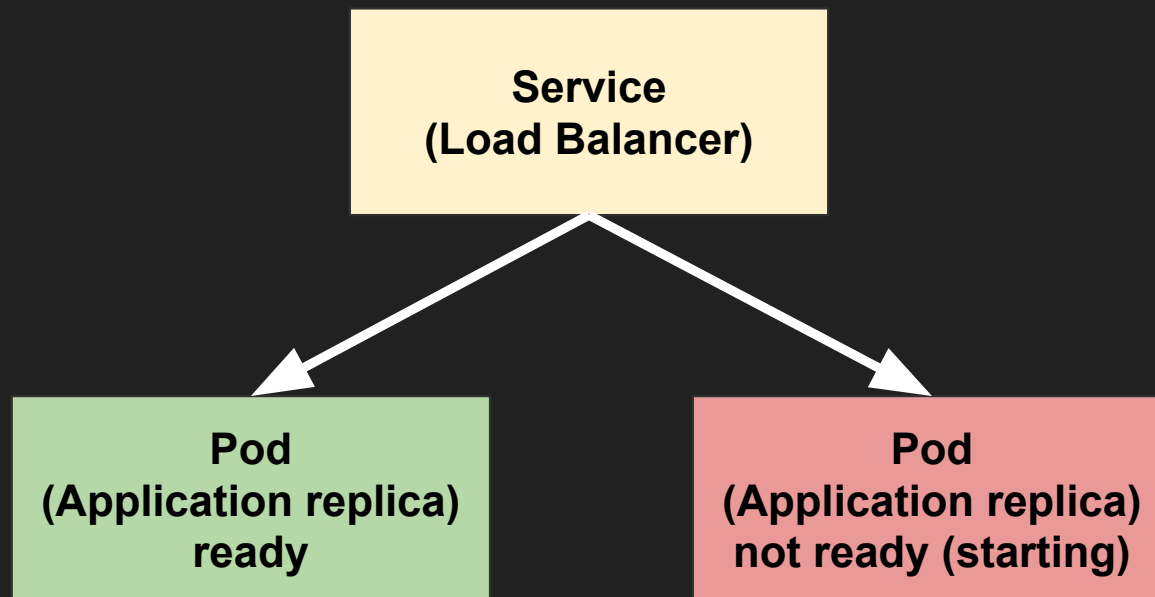
Build resilience into your applications



Readiness Checks

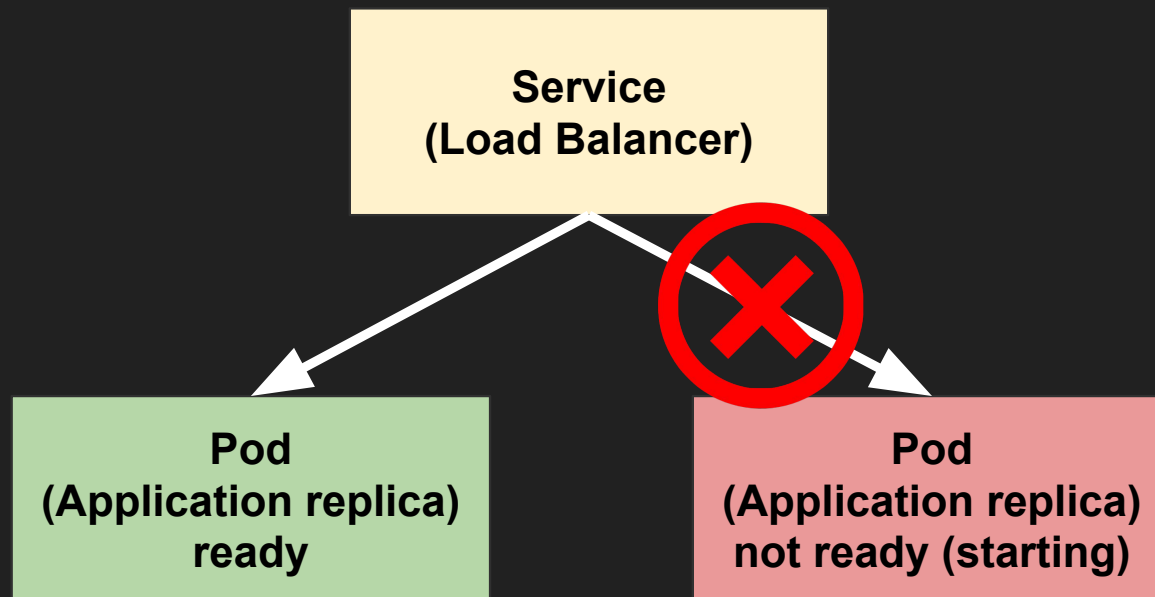


Readiness Checks



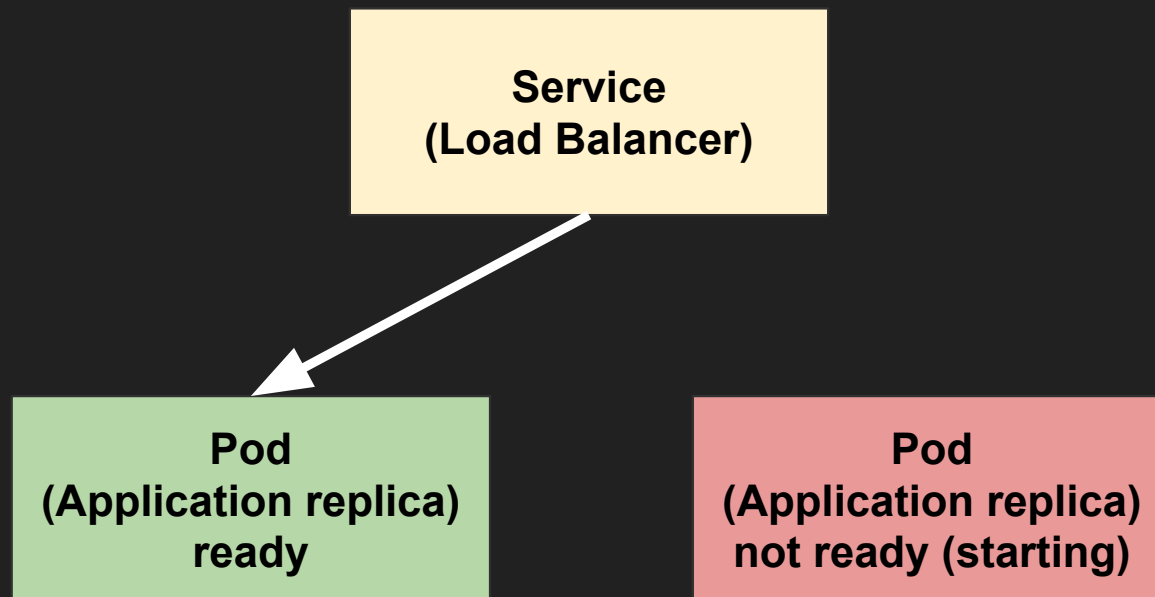
Without readiness check

Readiness Checks



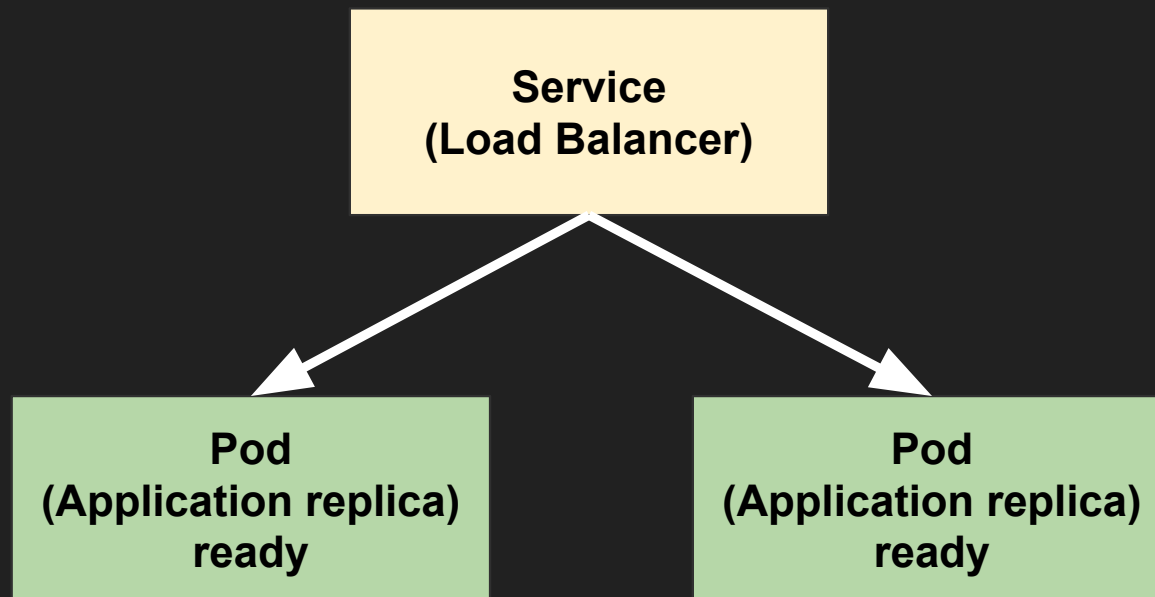
Without readiness check: 50% failure (when 2nd replica starting)

Readiness Checks



With readiness check

Readiness Checks



With readiness check

Liveness Checks



Liveness Checks



**Pod
(Application replica)
ready**

Liveness Checks



**Pod
(Application replica)
ready**

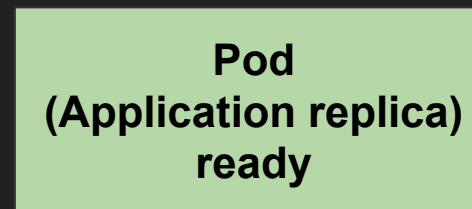
Liveness Checks



**Pod
(Application replica)
frozen**

Without liveness check: nothing happens... (your custom monitoring might pick it up at some point, or if it doesn't your users will start complaining)

Liveness Checks



**With liveness check: frozen pod is killed, and new one automatically started
(self healing without manual intervention)**

Self-healing systems (orchestrator)



DEMO

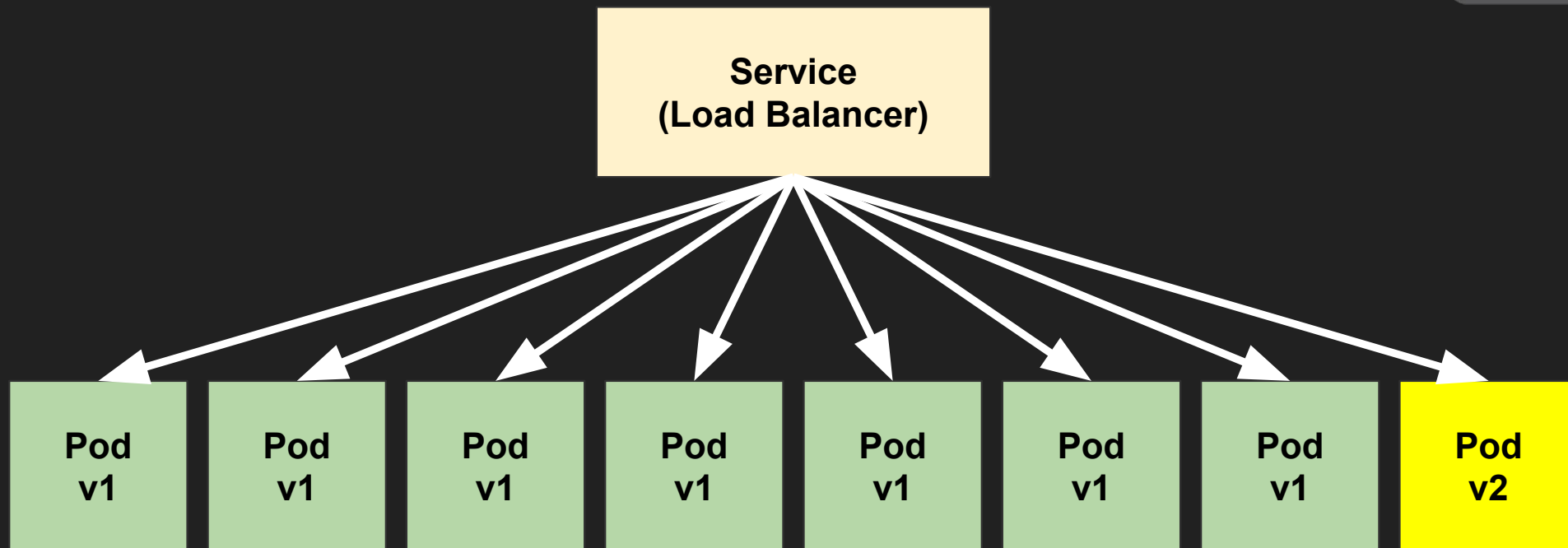
Self-healing systems (orchestrator)



Canary Releases



Canary Releases



ChaOps

Unleash the chaos monkeys!



Container



Availability Zone

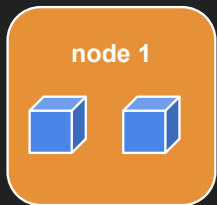
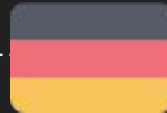


Region

<http://demo.gluo.io>



<http://demo.gluo.io>
<http://demo.eu.gluo.io>



eu-central-1a



eu-central-1b



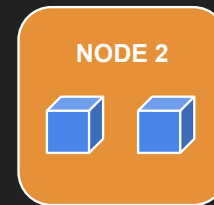
eu-central-1c



<http://demo.gluo.io>
<http://demo.us.gluo.io>



us-west-1a



us-west-1b



us-west-1c



Region

<http://demo.gluo.io>



<http://demo.gluo.io>
<http://demo.eu.gluo.io>



node 1



eu-central-1a

node 2



eu-central-1b

node 3



eu-central-1c



<http://chaops.io>
<http://us.chaops.io>



node 1



us-west-1a

node 2

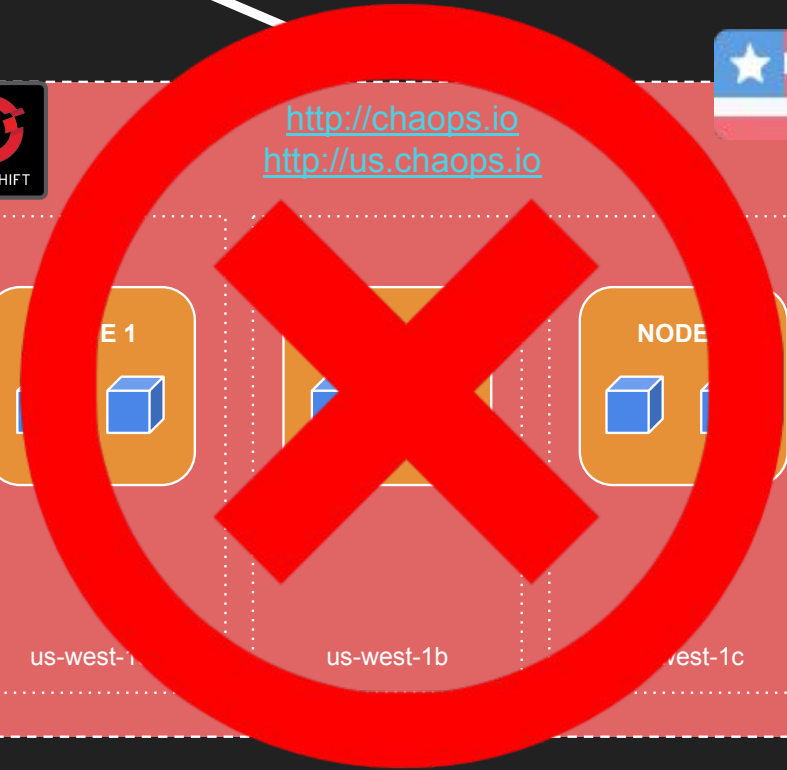


us-west-1b

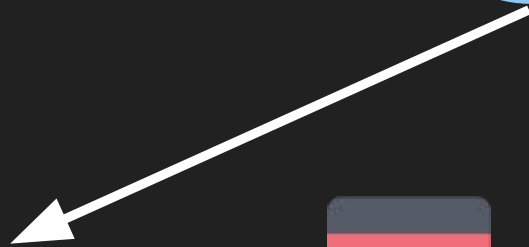
node 3



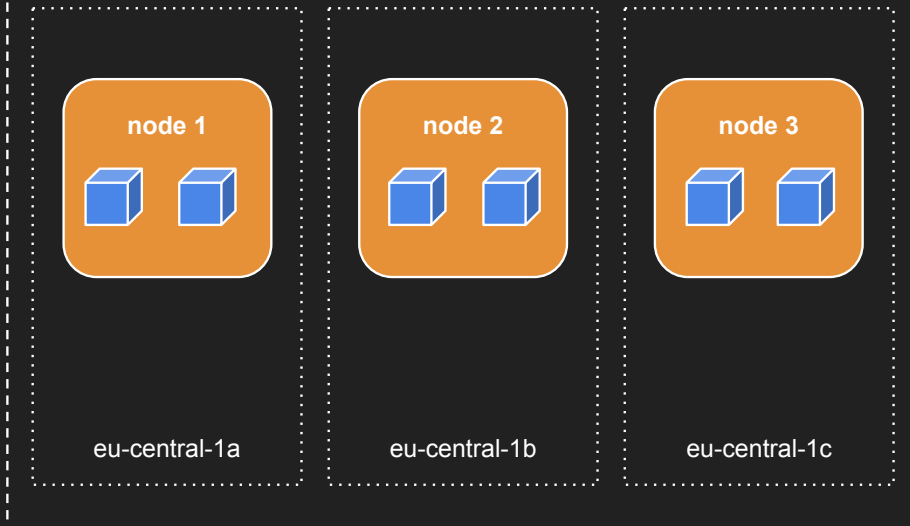
us-west-1c



<http://demo.gluo.io>



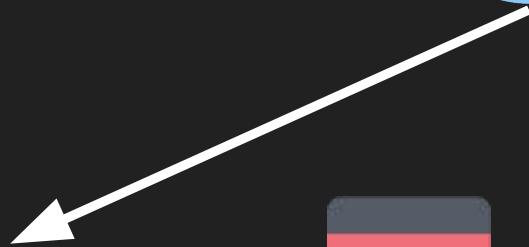
<http://demo.gluo.io>
<http://demo.eu.gluo.io>



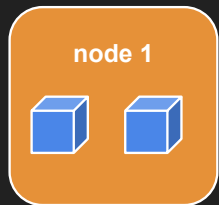
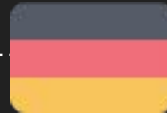


Availability Zone

<http://demo.gluo.io>



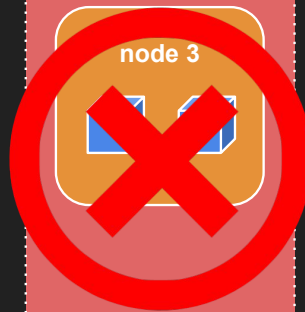
<http://demo.gluo.io>
<http://demo.eu.gluo.io>



eu-central-1a

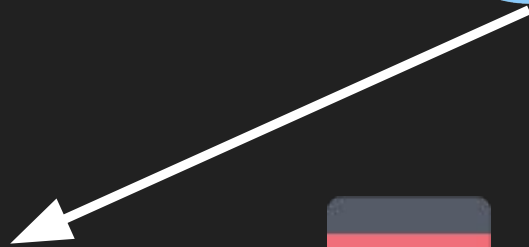


eu-central-1b



eu-central-1c

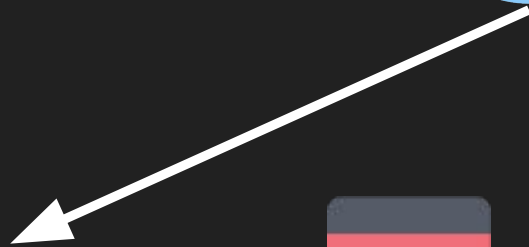
<http://demo.gluo.io>





Container

<http://demo.gluo.io>



<http://demo.gluo.io>
<http://demo.eu.gluo.io>



node 1



node 2



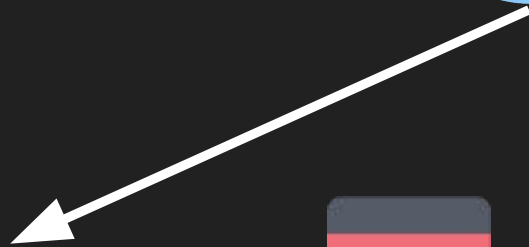
node 3



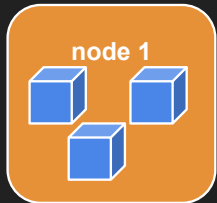
eu-central-1a

eu-central-1b

<http://demo.gluo.io>



<http://demo.gluo.io>
<http://demo.eu.gluo.io>



eu-central-1a

eu-central-1b

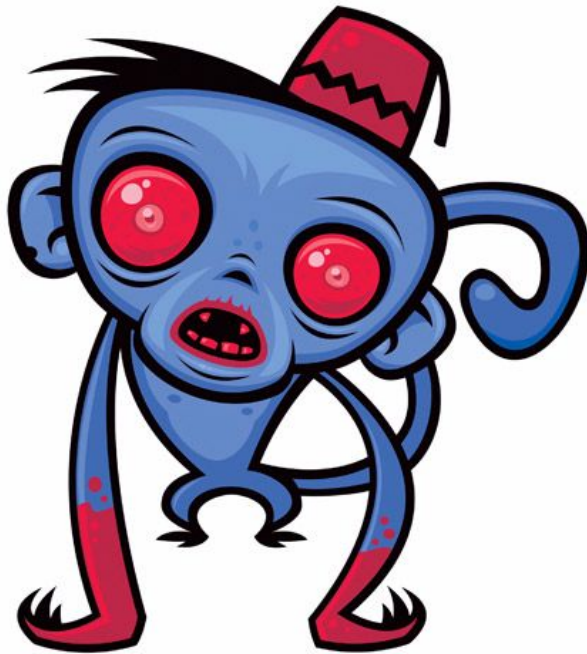


Container

CHAOS MONKEY



EXPECT FAILURE



**SYSADMIN
BADASS**

Chaos Zombie vs Sysadmin

ChaOps

Takeaways

- 1. Design for failure (in all layers: network, infrastructure, application,...)**
- 2. Embrace failure (learn from it!)**
- 3. DevOps tools/mindset enable Chaos Engineering**
- 4. Start small, grow gradually (not everybody is Netflix and that is OK!)**
- 5. Don't fight the chaos zombies by yourself, use tools that help you fight them!**

steven@gluo.be



Red Hat Premier Business Partner