zalando

CONTINUOUSLY DELIVER YOUR KUBERNETES INFRASTRUCTURE

KubeCon Copenhagen 2018

KubeCon | CloudNativeCon
Europe 2018

MIKKEL LARSEN

@mikkeloscar

2018-05-02

ZALANDO AT A GLANCE

~ 4.5 **billion EUR**

revenue 2017

> 200 million visits per month

> 23 million active customers

> 300.000 product choices

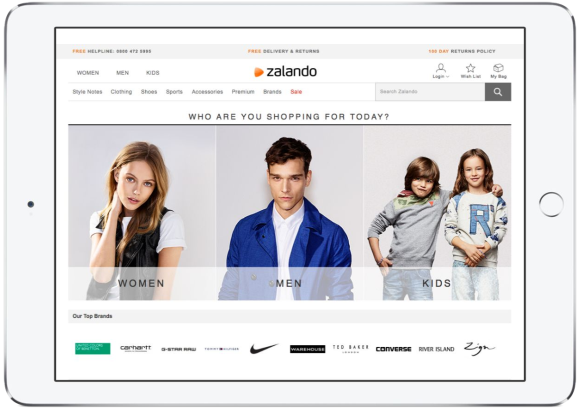
> 15.000 employees in Europe

> 70% of visits via mobile devices

~ 2.000 brands

15 countries

ZALANDO TECH



~ 2.000

Employees in Tech

> 200

Delivery teams

SCALE

366 Accounts

84 Clusters

INFRASTRUCTURE @ ZALANDO

STUPS

(toolset around AWS)

- AWS accounts per team.
- All instances must run the same AMI.
- PowerUser access to Production.
- You build it, you run EVERYTHING.

Kubernetes

- Clusters per product (multiple teams).
- Instances are not managed by teams.
- Hands off approach.
- A lot of stuff out of the box.

“PHILOSOPHY”

No pet clusters

We don’t want to tweak custom settings for 80 clusters.

Always provide the latest stable Kubernetes version

Oldest clusters were upgraded from v1.4 through v1.9.

Continuous and non-disruptive cluster updates

No maintenance windows.

“Fully” automated operations

Operators should only need to manually merge PRs.

CLUSTER SETUP

- Provisioned in AWS via Cloudformation.
- Etcd stack outside Kubernetes.
- Container Linux.
- Multi AZ worker nodes.
- HA control plane setup behind ELB.
- Cluster configuration stored in git.
- e2e tests run via Jenkins.
- Changes rolled out via ‘Cluster Lifecycle Manager’.

CLUSTER METADATA (CLUSTER-REGISTRY)

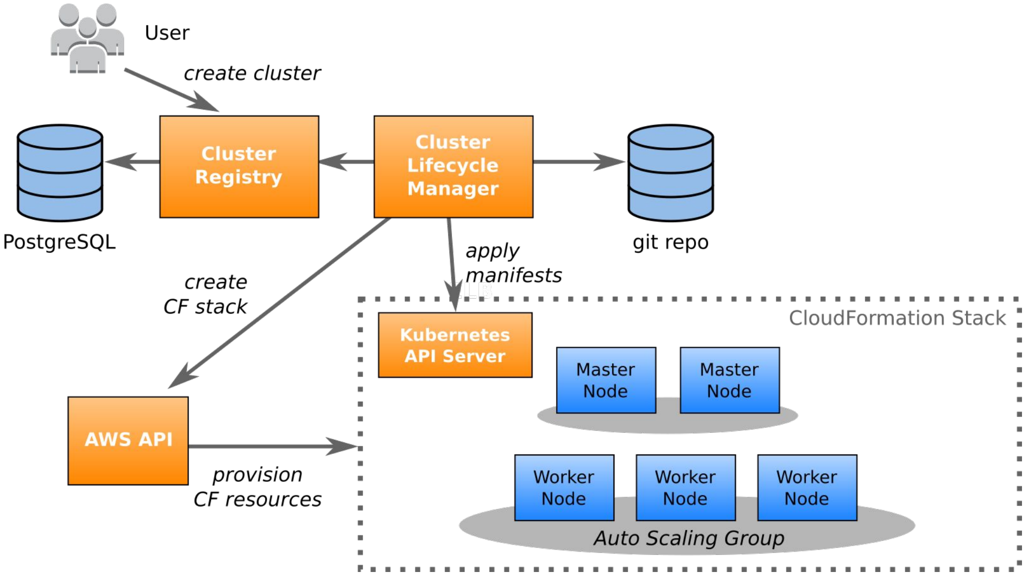
```yaml
clusters:
- id: “cluster-id”
  api_server_url: “https://cluster-id.example.org”
  config_items:
    Key: “value”
  environment: “test”
  region: “eu-central-1”
  lifecycle_status: “ready”
  node_pools:
  - name: “worker-pool”
    instance_type: “m5.large”
    min_size: 3
    max_size: 20
```

CLUSTER CONFIGURATION

```
cluster
├── cluster.yaml      # Kubernetes cluster stack
├── etcd-cluster.yaml # etcd cluster stack
├── manifests
│   ├── ...
├── master.clc.yaml  # userdata for master nodes
└── worker.clc.yaml  # userdata for worker nodes
```

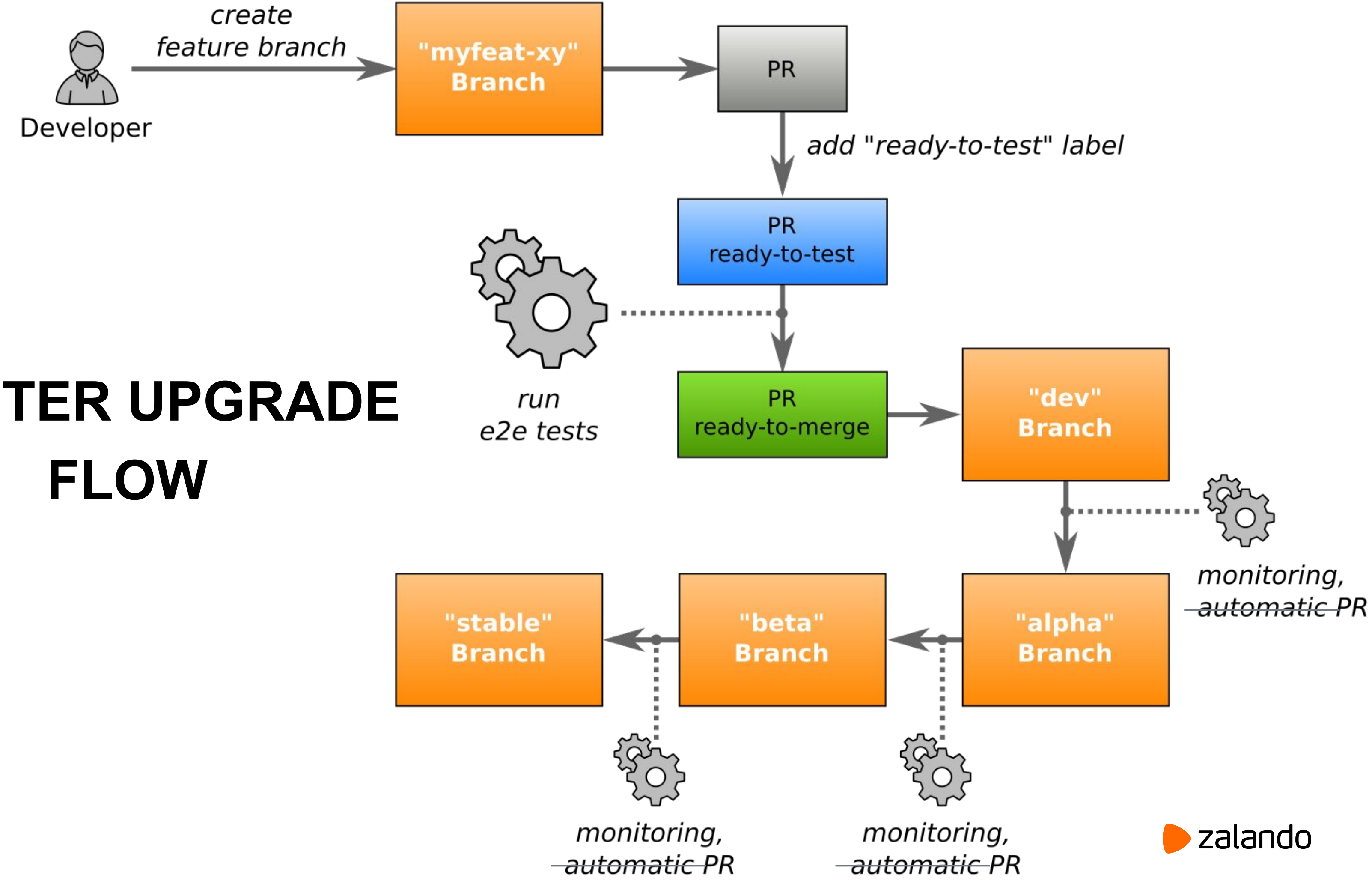
github.com/zalando-incubator/kubernetes-on-aws

CLUSTER LIFECYCLE MANAGER (CLM)



github.com/zalando-incubator/cluster-lifecycle-manager

CLUSTER UPGRADE FLOW

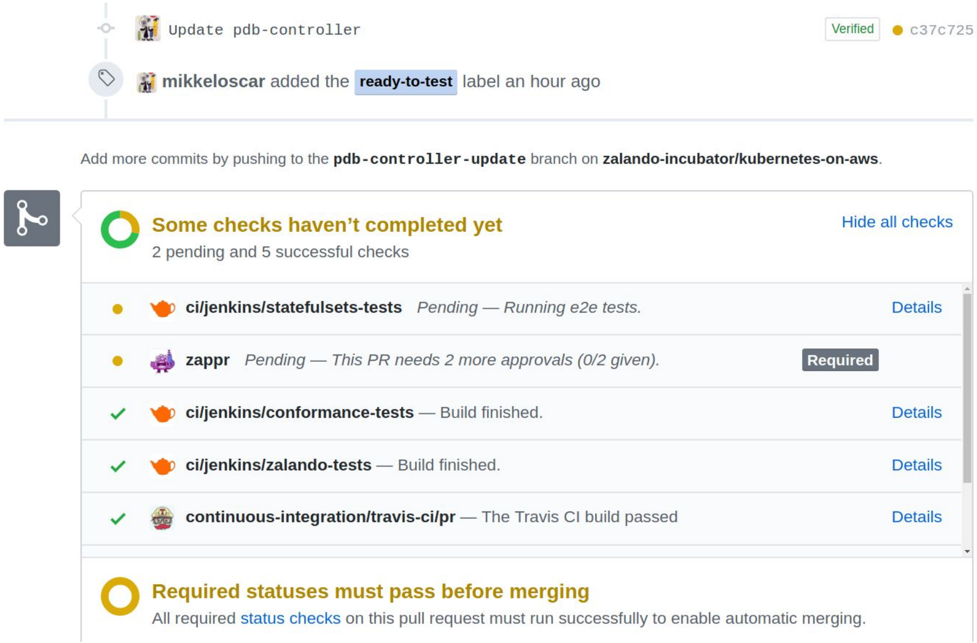


CLUSTER CHANNELS

Channel	Description	Clusters
`dev`	Development and playground clusters.	**3**
`alpha`	Main infrastructure cluster (**important to us**).	**1**
`beta`	Product clusters for the rest of the organization (prod/test).	**76+**

github.com/zalando-incubator/kubernetes-on-aws

E2E TESTS ON EVERY PR



Update pdb-controller Verified ● c37c725

mikkeloscar added the **ready-to-test** label an hour ago

Add more commits by pushing to the **pdb-controller-update** branch on **zalando-incubator/kubernetes-on-aws**.

Some checks haven't completed yet Hide all checks

2 pending and 5 successful checks

- ● **ci/jenkins/statefulsets-tests** *Pending — Running e2e tests.* Details
- ● **zappr** *Pending — This PR needs 2 more approvals (0/2 given).* **Required**
- ✓ **ci/jenkins/conformance-tests** — Build finished. Details
- ✓ **ci/jenkins/zalando-tests** — Build finished. Details
- ✓ **continuous-integration/travis-ci/pr** — The Travis CI build passed Details

Required statuses must pass before merging

All required status checks on this pull request must run successfully to enable automatic merging.

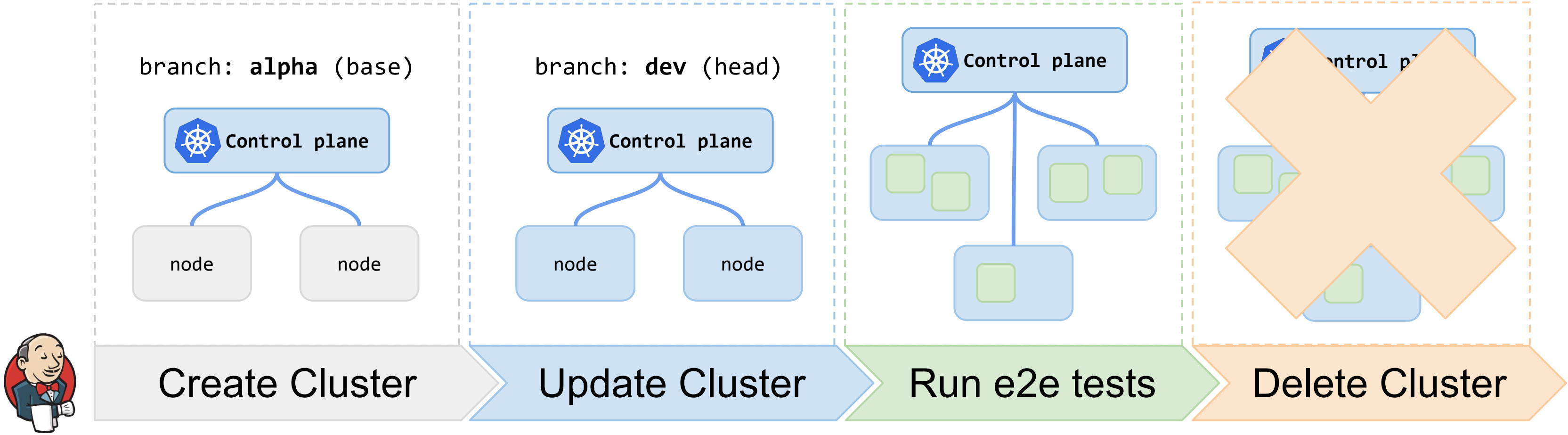
github.com/zalando-incubator/kubernetes-on-aws

E2E TESTS

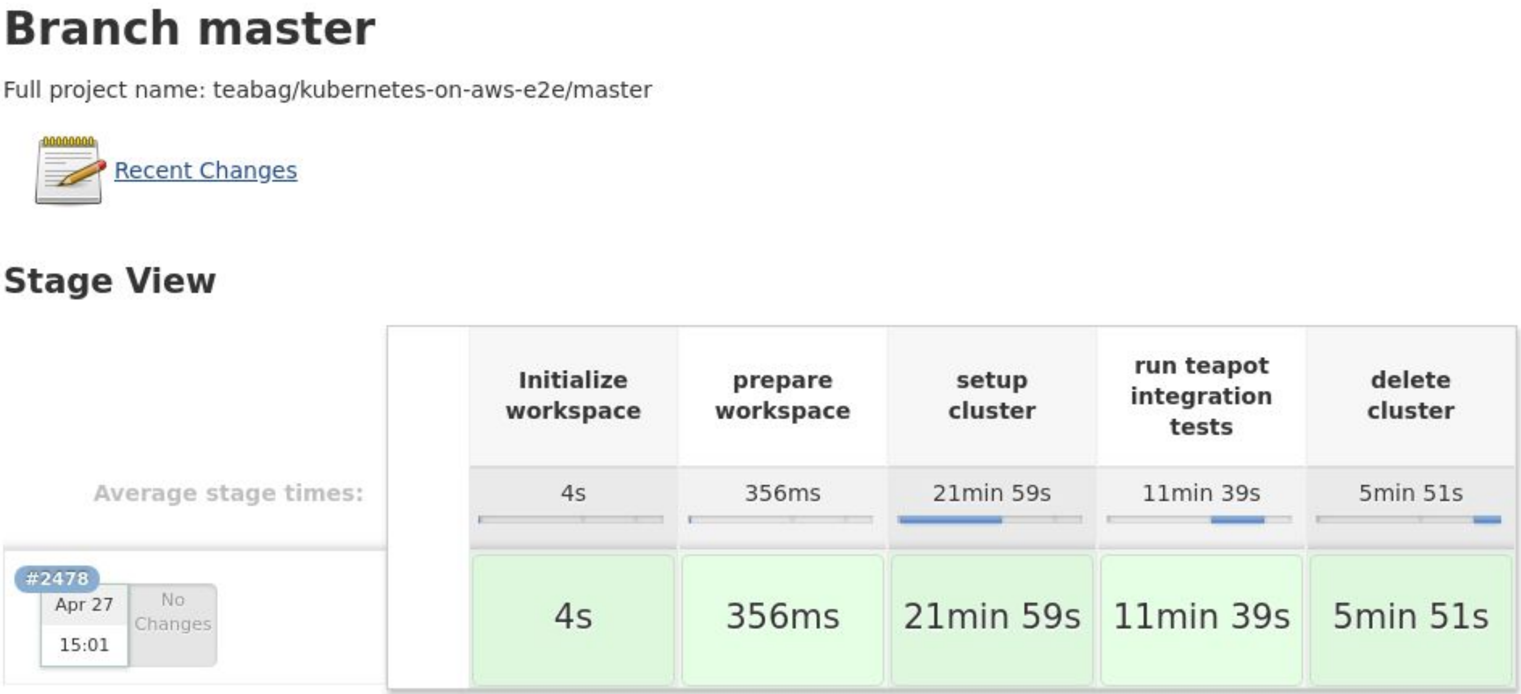
✓	**Conformance Tests** Upstream Kubernetes e2e conformance tests	**144**
✓	**StatefulSet Tests** Rolling update of stateful sets including volume mounting	**2**
✓	**Zalando Tests (custom)** Custom tests for ingress, external-dns, PSP etc.	**4**

RUNNING E2E TESTS

`Testing **dev** to **alpha** upgrade`



RUNNING E2E TESTS



Branch master

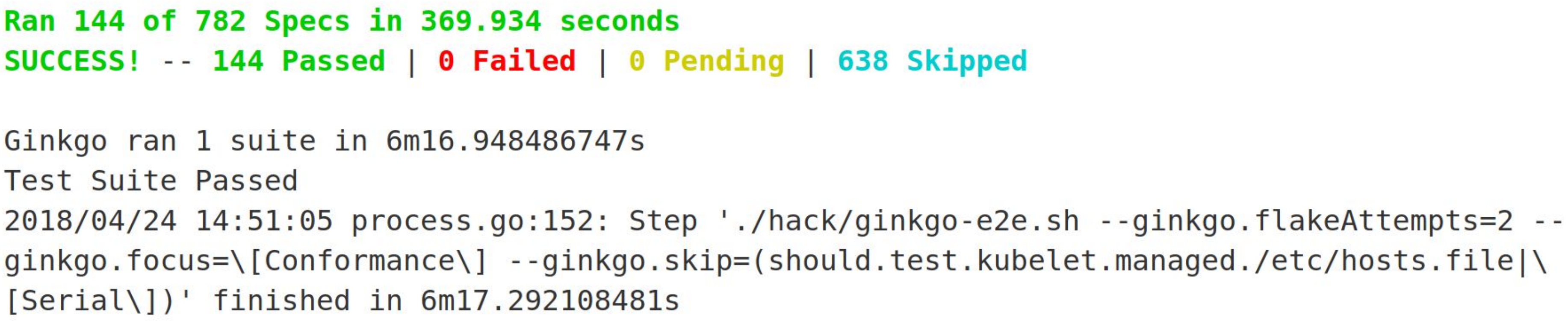
Full project name: teabag/kubernetes-on-aws-e2e/master

Recent Changes

Stage View

	Initialize workspace	prepare workspace	setup cluster	run teapot integration tests	delete cluster
Average stage times:	4s	356ms	21min 59s	11min 39s	5min 51s
#2478 Apr 27 15:01 No Changes	4s	356ms	21min 59s	11min 39s	5min 51s

RUNNING E2E TESTS



```
Ran 144 of 782 Specs in 369.934 seconds
SUCCESS! -- 144 Passed | 0 Failed | 0 Pending | 638 Skipped

Ginkgo ran 1 suite in 6m16.948486747s
Test Suite Passed
2018/04/24 14:51:05 process.go:152: Step './hack/ginkgo-e2e.sh --ginkgo.flakeAttempts=2 --
ginkgo.focus=\[Conformance\] --ginkgo.skip=(should.test.kubelet.managed./etc/hosts.file|\
[Serial\])' finished in 6m17.292108481s
```

RUNNING E2E TESTS

Running Kubernetes Conformance tests

```
# Run all Conformance tests except *serial* tests
$ docker run -v $HOME/.kube/config:/kubeconfig \
  mikkeloscar/kubernetes-e2e:latest -p \
  -focus "\[Conformance\]" -skip "\[Serial\]" /e2e.test
```

Select the type of tests to run (Conformance)

Skip tests that can’t be run in parallel

github.com/mikkeloscar/kubernetes-e2e

RUNNING E2E TESTS

Running Kubernetes Statefulset tests

```
# basic statefulset tests
$ docker run -v $HOME/.kube/config:/kubeconfig \
  mikkeloscar/kubernetes-e2e:latest -p \
  -focus "\[StatefulSetBasic\]" /e2e.test

# Test running a StatefulSet with PVCs (test volume attachment)
$ docker run -v $HOME/.kube/config:/kubeconfig \
  mikkeloscar/kubernetes-e2e:latest -p \
  -focus "\[Feature:StatefulSet\]\s\[Slow\].*redis" /e2e.test
```

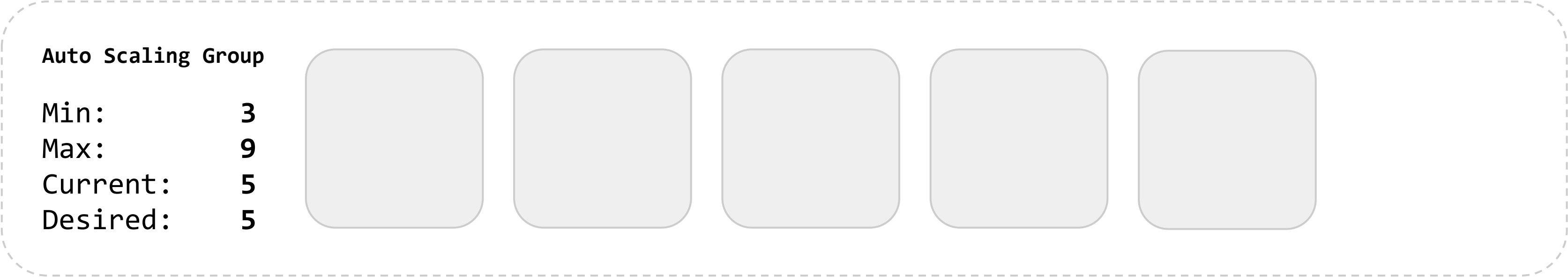
github.com/mikkeloscar/kubernetes-e2e

HINTS FOR RUNNING E2E TESTS

- Run with **`-flakeAttempts=2`**
- Update e2e image for every **major** release of Kubernetes!
- Disable broken e2e tests using **`-skip`**!

UPGRADING NODES

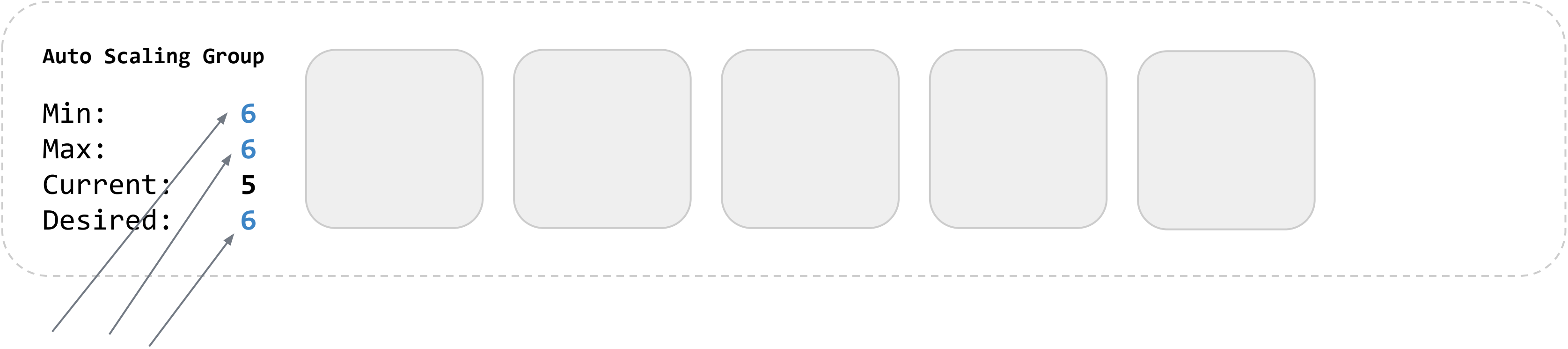
NAÏVE NODE UPGRADE STRATEGY



Auto Scaling Group

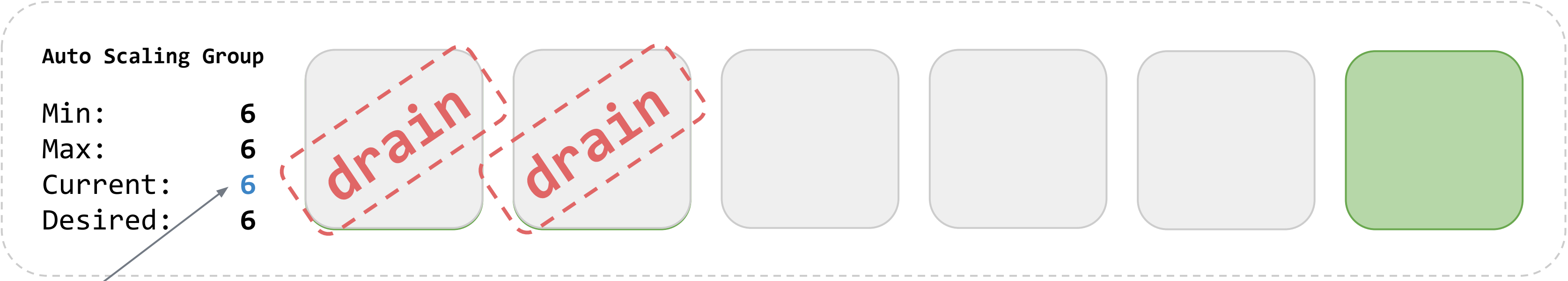
Min:	3
Max:	9
Current:	5
Desired:	5

NAÏVE NODE UPGRADE STRATEGY



Set ASG size to current + 1

NAÏVE NODE UPGRADE STRATEGY



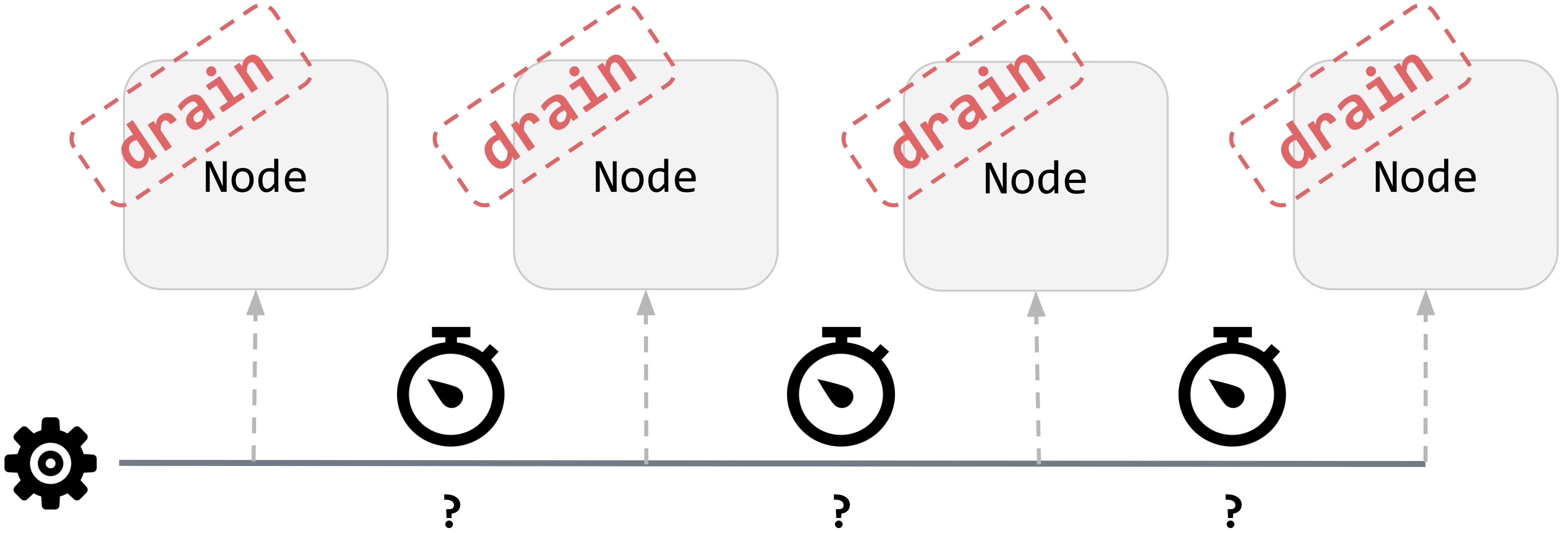
Get a new instance

THE NAÏVE STRATEGY WAS ‘FINE’

© Nick Takayama

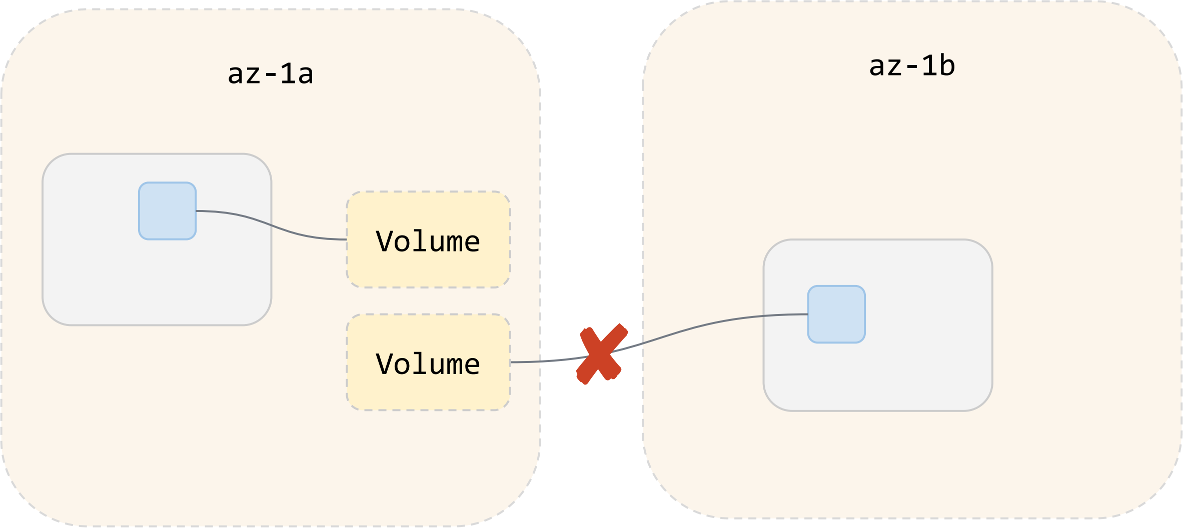
PROBLEMS WITH THE NAÏVE STRATEGY

How long do we wait between draining nodes?



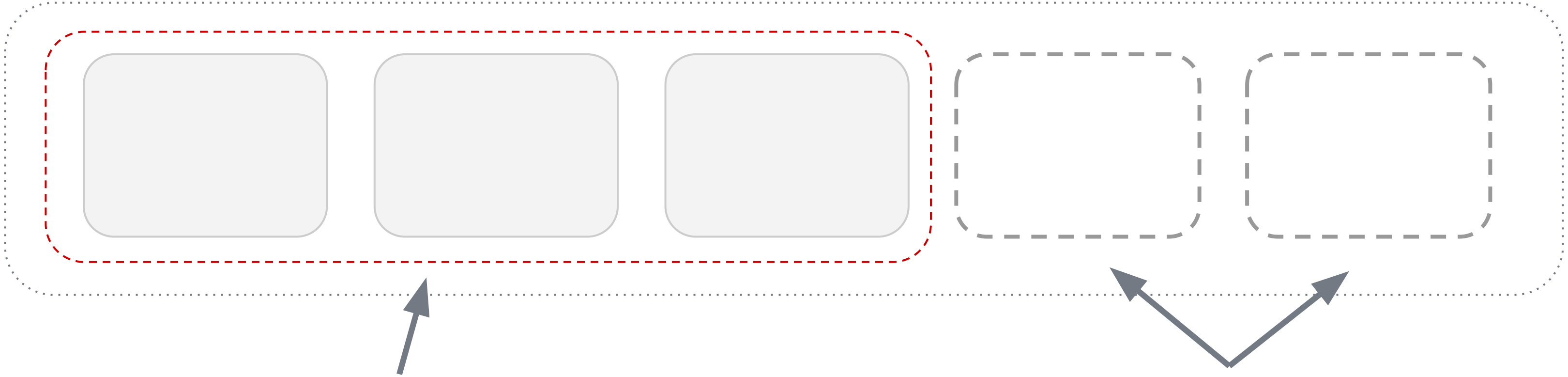
PROBLEMS WITH THE NAÏVE STRATEGY

Volumes are per Availability Zone!



PROBLEMS WITH THE NAÏVE STRATEGY

No autoscaling during rolling upgrade!

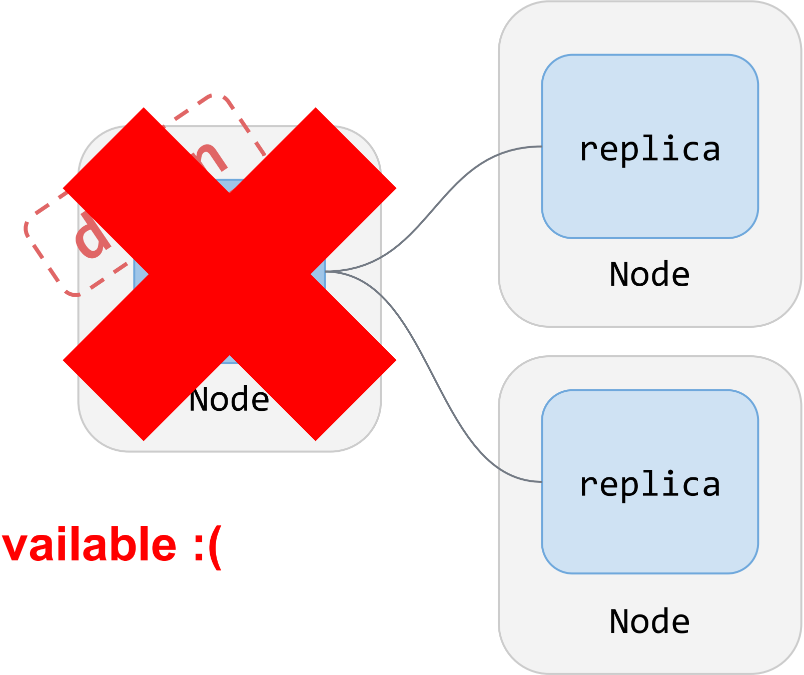


Fixed ASG size

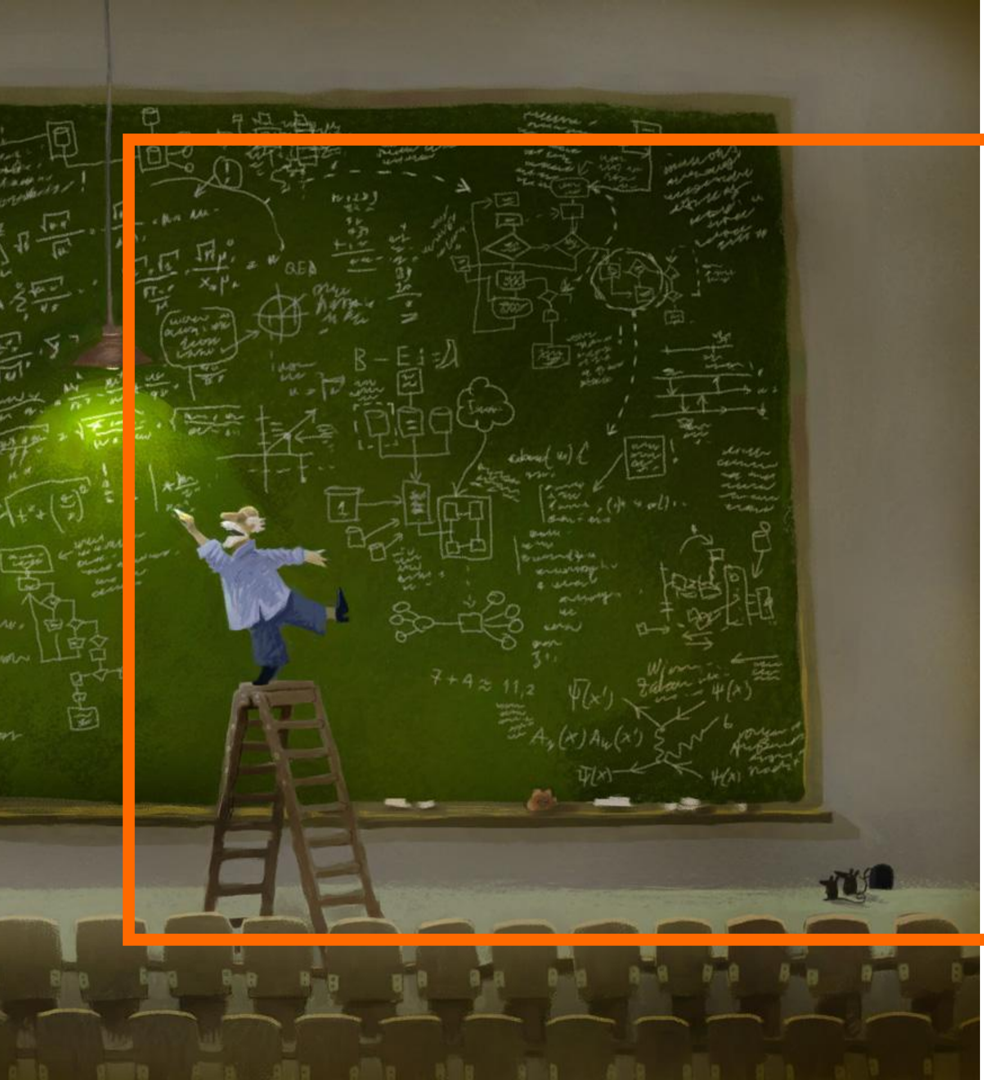
Pending nodes

PROBLEMS WITH THE NAÏVE STRATEGY

What about stateful applications like Postgres?

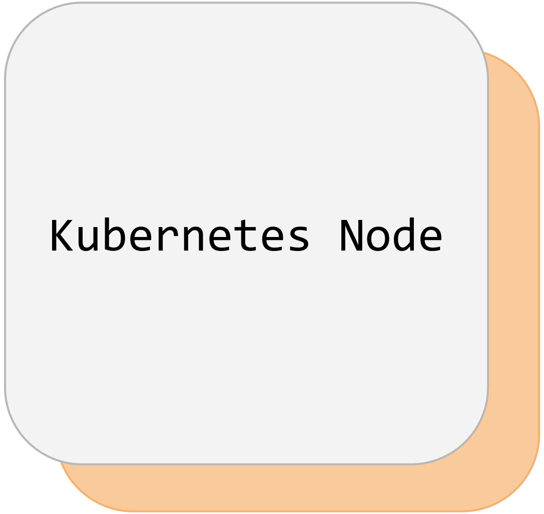


Postgres cluster unavailable :(



DESIGNING A NEW UPGRADE STRATEGY

NODES READY?



Kubernetes Node

NODES READY?

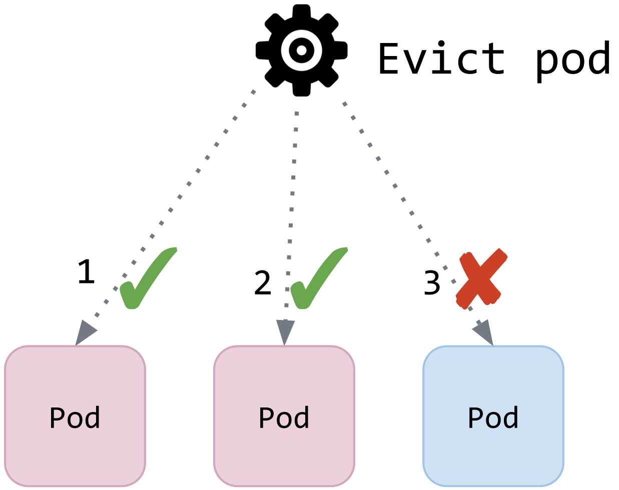
Kubernetes Node

EC2 Instance

- Kubelet is reporting NodeReady

- Instance ‘InService’ in ASG.
- (Instance ‘InService’ in ELB.)

POD DISRUPTION BUDGETS

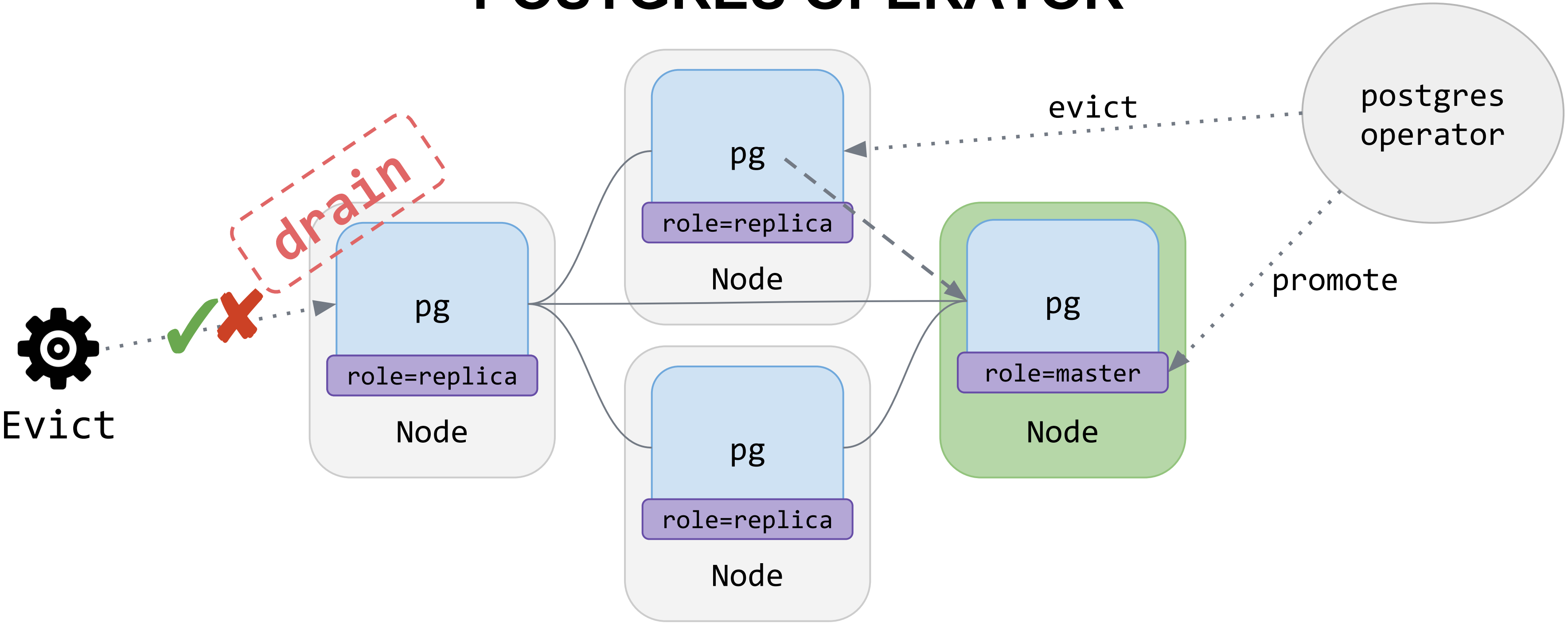


```
apiVersion: policy/v1beta1
kind: PodDisruptionBudget
metadata:
  name: "my-app"
spec:
  minAvailable: 1
  selector:
    matchLabels:
      application: “my-app”
```

github.com/mikkeloscar/pdb-controller

STATEFUL WORKLOADS (POSTGRES)

POSTGRES OPERATOR



github.com/zalando-incubator/postgres-operator

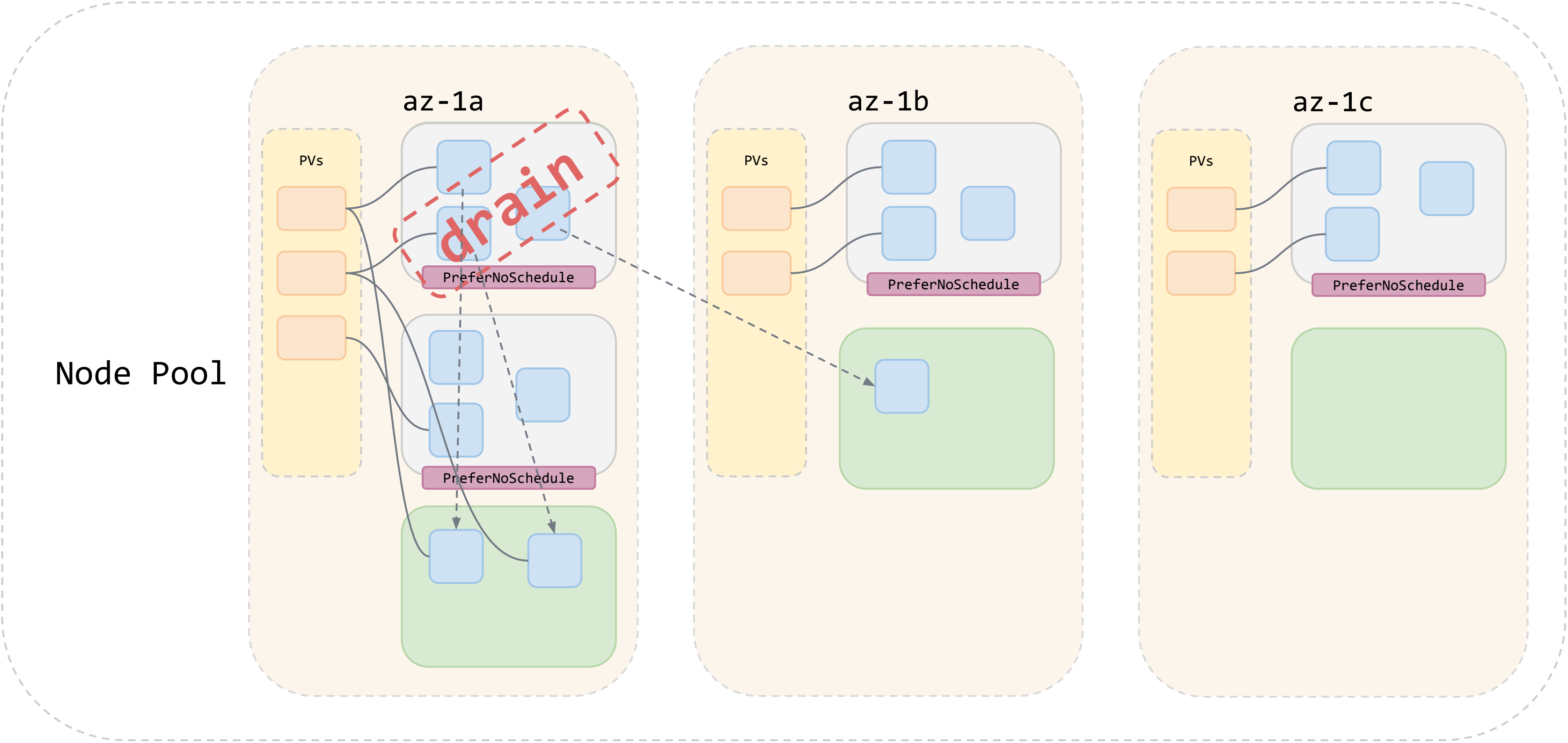
zalando

POSTGRES OPERATOR

```
apiVersion: policy/v1beta1
kind: PodDisruptionBudget
metadata:
  name: "postgres-cluster"
spec:
  minAvailable: 1
  selector:
    matchLabels:
      application: “postgres-cluster”
      role: “master”
```

github.com/zalando-incubator/postgres-operator

ROLLING UPGRADE OF NODES



‘THIS IS FINE’

A few times where the continuous delivery **wasn’t** fine

© KC Green

1. Broke flannel network in main infrastructure cluster because we didn’t test upgrade path.

2. Took down internal docker registry when updating too many clusters in parallel and rolled nodes without kubelet running.

OPEN SOURCE

Cluster Lifecycle Manager
github.com/zalando-incubator/cluster-lifecycle-manager

Kubernetes on AWS
github.com/zalando-incubator/kubernetes-on-aws

AWS ALB Ingress controller
github.com/zalando-incubator/kube-ingress-aws-controller

Skipper HTTP Router & Ingress controller
github.com/zalando/skipper

External DNS
github.com/kubernetes-incubator/external-dns

Pod Disruption Budget Controller
github.com/mikkeloscar/pdb-controller

Postgres Operator
github.com/zalando-incubator/postgres-operator

TAK

(Thank you)

zalando

MIKKEL LARSEN

SOFTWARE ENGINEER

PLATFORM INFRASTRUCTURE

mikkel.larsen@zalando.de

@mikkeloscar

Illustrations by @01k, @kcgreenn, @ntakayama

2018-05-02