zalando

PostgreSQL on Kubernetes

Jan Mußler

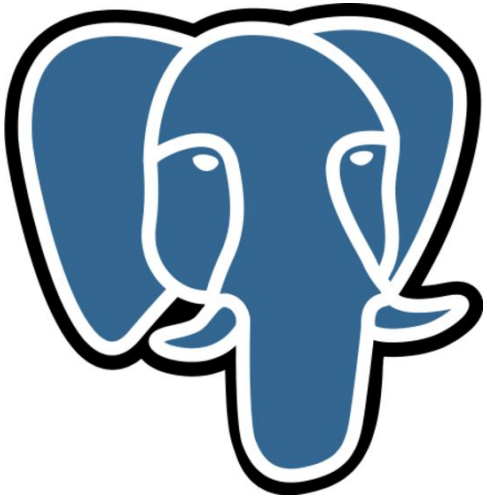
@JanMussler

12-07-2017

TABLE OF CONTENTS

zalando

PostgreSQL



Powerful, open-source, relational database

MVCC, lock free reads, great at concurrency

Transactional DDL!

Safe to modify tables in production **(yes!)**

Solid owner and role system for privileges

Highly customizable

Supports PL/pgSQL, Python, JS V8

ZALANDO AT A GLANCE

>300 databases

In data centers

>100

Cloud databases
Managed by DB team

>50

staging databases
in Kubernetes

as at May 2017

zalando

Running PostgreSQL in two data centers

Bare metal with LXC containers

Single Git repository with all configs

Database discovery service

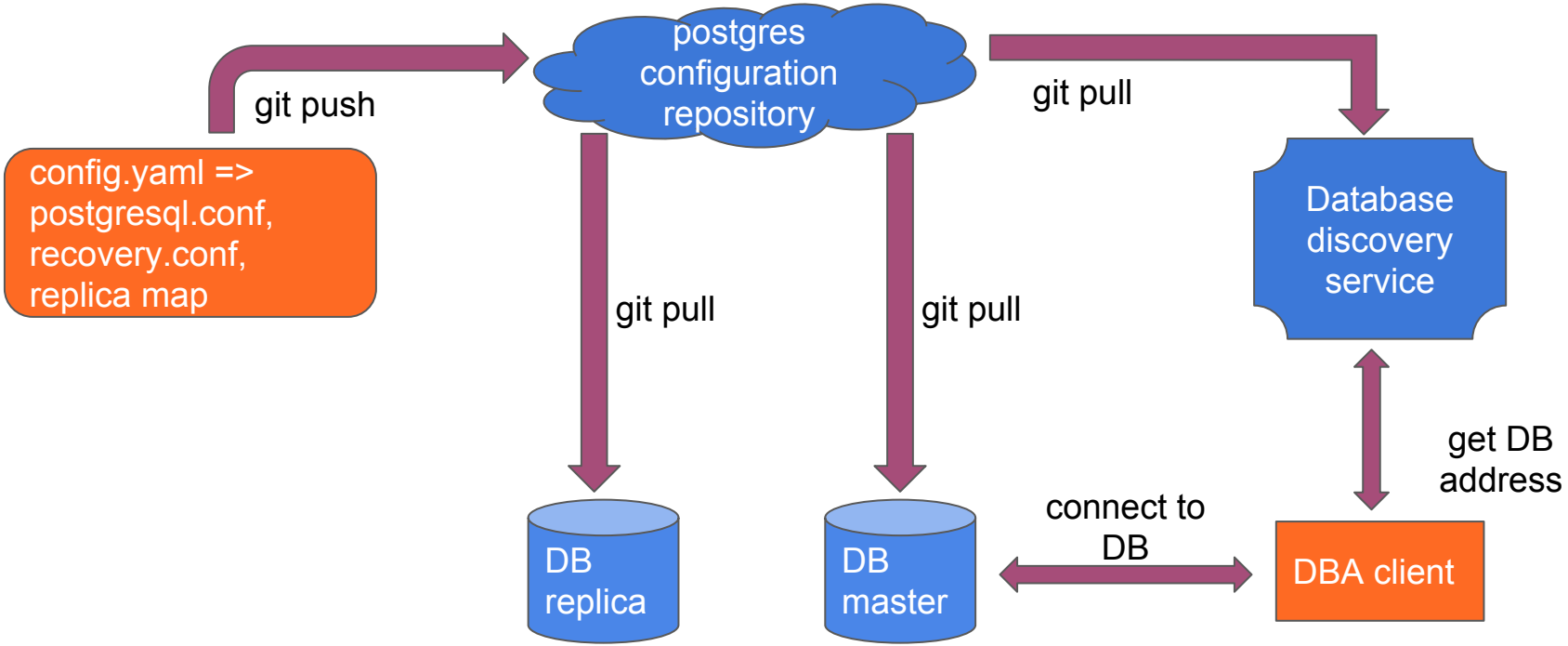
Script to initialize new nodes

Init from slave to lower impact

Time delayed slaves in one data center

PostgreSQL versions: 9.3+

Git-driven workflow in data centers



zalando

PostgreSQL on AWS

Faster database provisioning

Flexible hardware configurations

CPU, Memory, Storage, Price

Docker is enforced

Expected more node failures

Needs more automation

Patroni to the rescue

PostgreSQL management “daemon”

Platform independent using callback scripts

Implemented in Python

Master election (using ETCD, …)

Zalando’s first open-source repo reaching 1000 ☆

Spilo: Packaging Patroni in Docker

Configuration via ENV variables

From self contained for AWS deployment

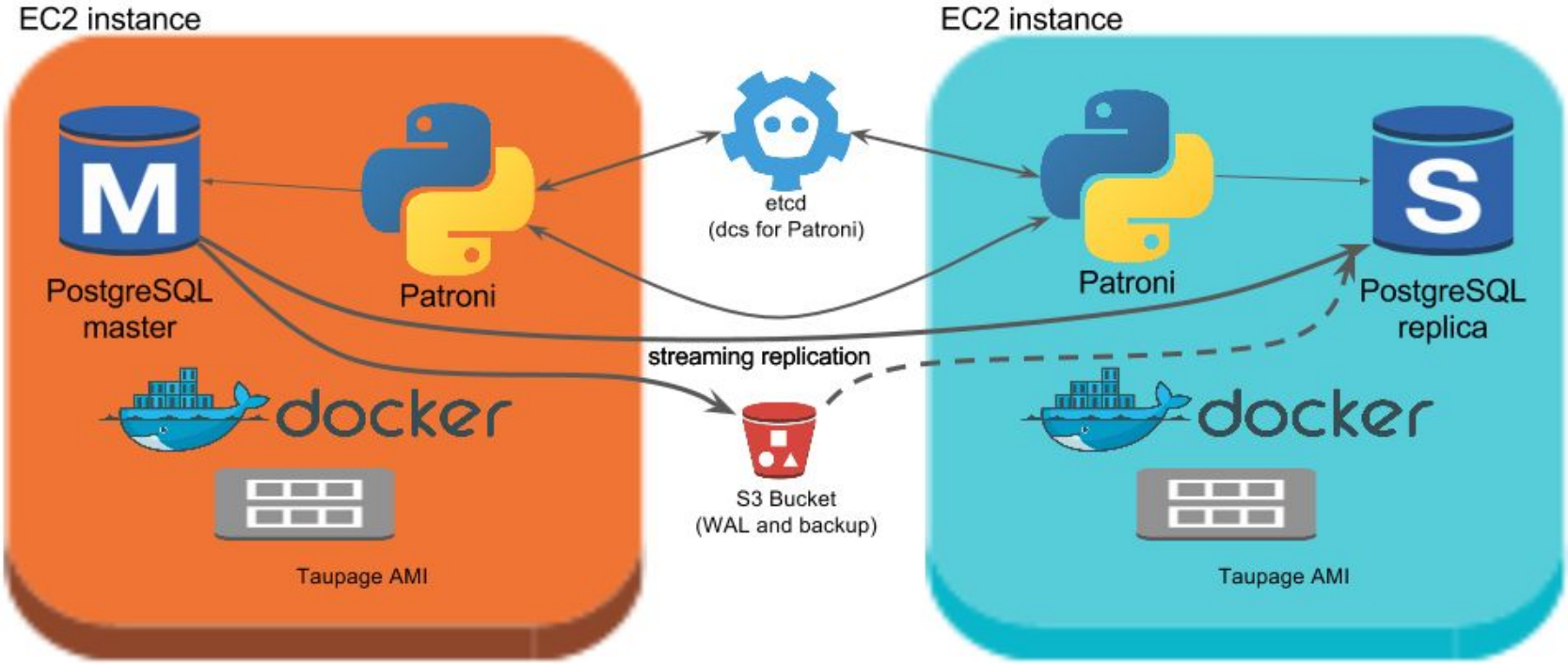
Including all PG versions since 9.3

```
# Install patroni and WAL-e
ENV PATRONIVERSION=1.2.5
ENV WALE_VERSION=1.0.3
RUN export DEBIAN_FRONTEND=noninteractive \
    export BUILD_PACKAGES="python3-pip" \
    && apt-get update \
    && apt-get install -y \
            # Required for wal-e
            daemontools lzop \
            # Required for /usr/local/bin/patroni
            python3 python3-setuptools python3-pystache python3-prettytable python3-six \
            ${BUILD_PACKAGES} \

    && pip3 install pip --upgrade \
    && pip3 install --upgrade patroni==$PATRONIVERSION \
            gcloud boto wal-e==$WALE_VERSION \

    # https://github.com/wal-e/wal-e/issues/318
    && sed -i 's/^\(    for i in range(0,\) num_retries):.*/\1 100):/g' /usr/local/lib/pytl

    # Clean up
    && apt-get purge -y ${BUILD_PACKAGES} \
    && apt-get autoremove -y \
    && apt-get clean \
    && rm -rf /var/lib/apt/lists/* /root/.cache


# install etcdctl
ENV ETCDVERSION 2.3.8
```

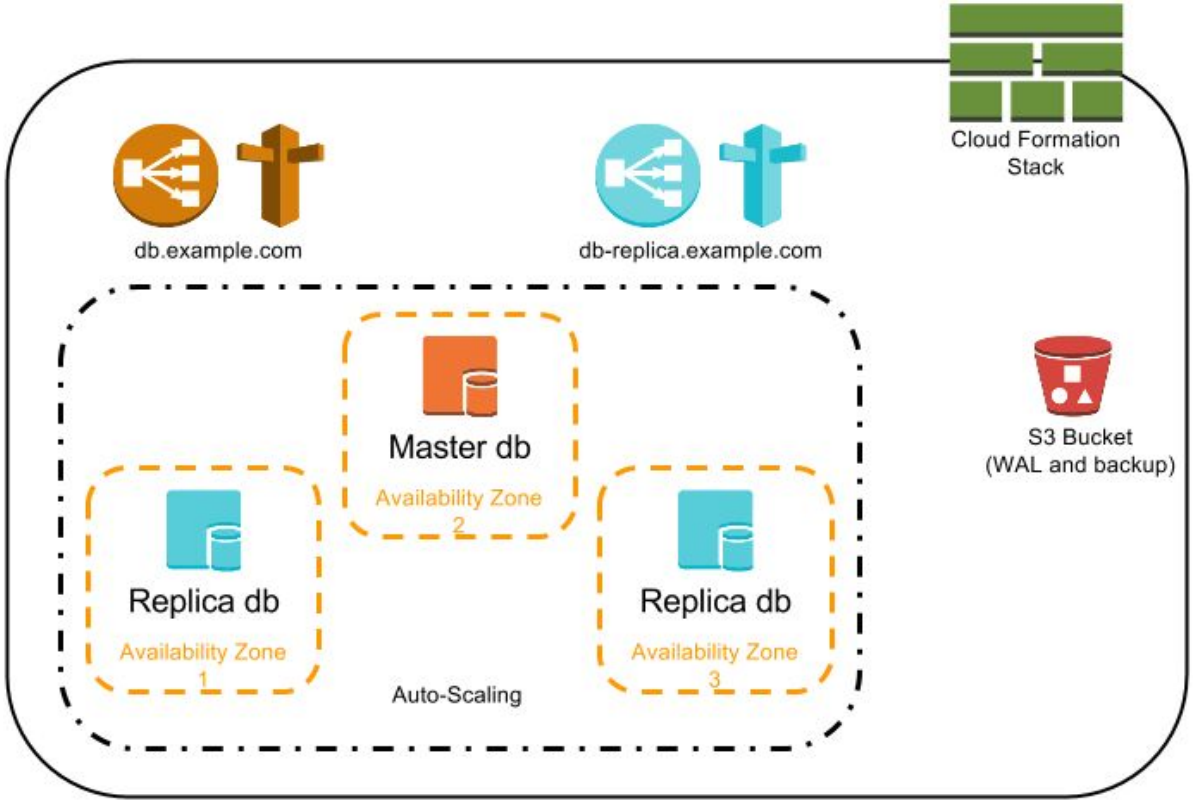
Try it locally!

```
docker run -p 8080:8080 registry.opensource.zalan.do/acid/spilo-9.6:1.2-p27
```

Spilo deployment on AWS



Spilo-based deployments in the cloud



zalando

Why not AWS RDS or Aurora PostgreSQL

Not an easy answer :)

Full control

- Independent of cloud provider
- Real super user available
- Custom extensions
- Streaming/WAL replication in and out
- Local storage not supported on RDS

Costs? Cost of development? ...

zalando

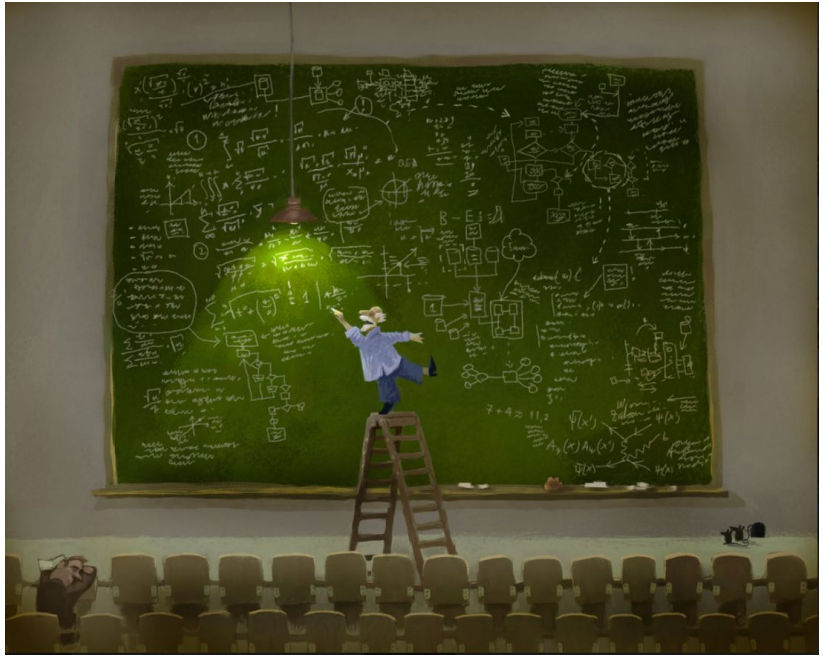
Costs: RDS vs Spilo

	RDS Multi AZ + 100GB	Spilo (2x EC2 + 100GB gp2)
m4.large	335€	219€
m4.2xlarge	1300€	750€
m4.4xlarge	2600€	1460€

+ ELB Traffic costs for Spilo (RDS can be VPC internal)

PostgreSQL as a Service

Goals



Automation

- Get rid of Jira and users waiting
- Convenient way to get new cluster
- Enable users to modify cluster setup

Integration

- Works with deployment pipeline
- User and Application user provisioning
- Monitoring out of the box

Kubernetes


Kelsey Hightower
@kelseyhightower
Following
Kelsey's guide to running traditional databases on Kubernetes. Strongly consider using a managed service.
6:56 PM - 20 Jan 2017
92 Retweets 175 Likes

The Stateful Set

```yaml
apiVersion: apps/v1beta1
kind: StatefulSet
metadata:
  labels:
    application: spilo
  name: zmon-nc-10-3
  namespace: default
spec:
  replicas: 2
  selector:
    matchLabels:
      application: spilo
  serviceName: zmon-nc-10-3
  template:
    metadata:
    spec:
      containers:
      - env:
        - name: ETCD_HOST
          value: etcd.o.z.d
        image: r.o.z.d/acid/spiloprivate-10:1.3-p2
```

Building block for stateful services

- Stable host or pod names
- Stable assignment of volumes
- Scale factor
- Ordering during upstart and scaling

Useful for: Cassandra, ETCD, PostgreSQL, ...

zalando

Storage on Kubernetes

```
spec:
  containers:
  - env:
    [ .. ]
    volumeMounts:
    - mountPath: /home/postgres/pgdata
      name: pgdata

[..]

volumeClaimTemplates:
  - metadata:
      annotations:
        volume.alpha.kubernetes.io/storage-class: default
      name: pgdata
    spec:
      accessModes:
      - ReadWriteOnce
      resources:
        requests:
          storage: 10Gi
```

Volumes and Volume claims

Auto provisioning of volumes in the Cloud

EBS volumes in our case

No resize support in K8S itself

Volumes per failure domain (AZ)

Local storage in 1.7

zalando

Third party resources

Store user-defined data in K8S

Create your own “types”

Big YAML “blob”

Watchable by applications via K8S API

Changes in 1.7 to Custom Resource Definition

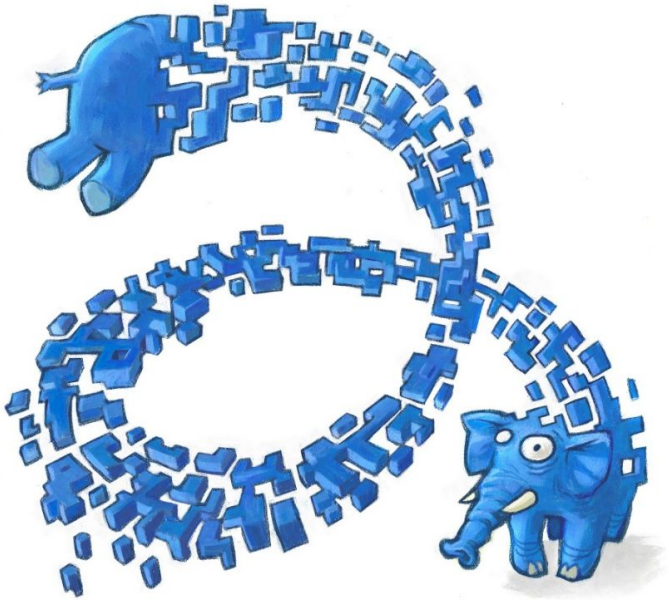
Cluster manifest with PostgreSQL configuration

```yaml
apiVersion: acid.zalan.do/v1
kind: Postgresql
metadata:
  labels:
    team: acid
  name: acid-newdatabase-01
  namespace: default
spec:
  allowedSourceRanges:
  - 192.168.1.0/24
  numberOfInstances: 2
  postgresql:
    version: "9.6"
    parameters:
      shared_buffers: 512MB
      max_connections: 128
  teamId: acid
  volume:
    size: 10Gi
```

payload

zalando

The “postgres-operator”



Application to manage PostgreSQL clusters

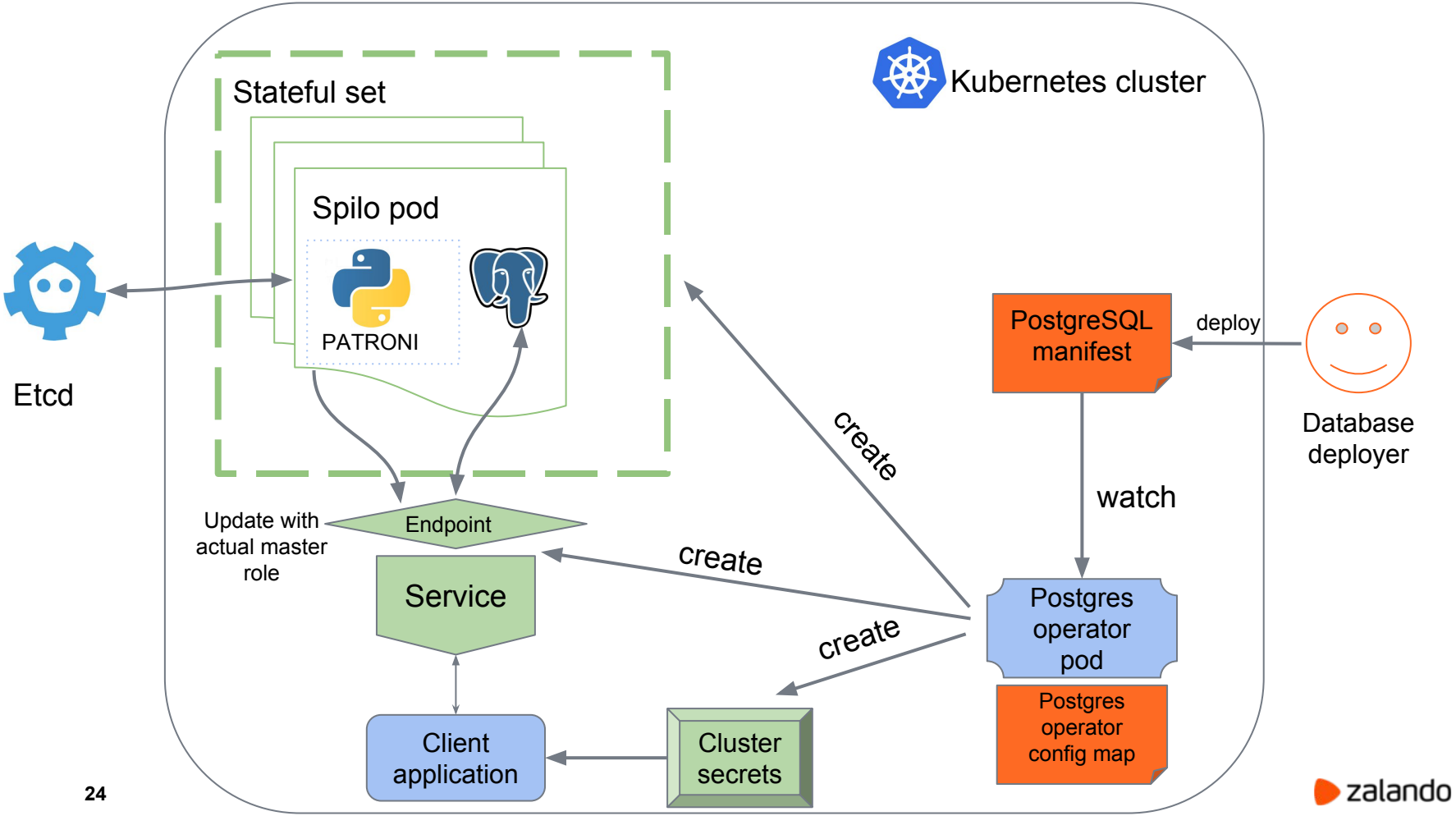
Observes 3rd party “postgres” manifests

Spawns and modifies new clusters

Syncs and provisions roles

Handles volume resize, incl. Resize2fs

Also responsible for e.g. updating Docker images


Kubernetes cluster
Stateful set
Spilo pod
PATRONI
Etcd
PostgreSQL manifest
deploy
Database deployer
create
watch
Update with actual master role
Endpoint
Service
create
create
Postgres operator pod
Postgres operator config map
Client application
Cluster secrets

zalando

Create a new PostgreSQL cluster

Cluster YAML definition

```yaml
kind: "Postgresql"
apiVersion: "acid.zalan.do/v1"

metadata:
    name: "guild-24x7-new-cluster"
    namespace: "default"
    labels:
        team: guild-24x7

spec:
    teamId: "guild-24x7"
    volume:
        size: "10Gi"
    numberOfInstances: 1

    postgresql:
        version: "9.6"

    allowedSourceRanges:
    # IP ranges to access your cluster go here
```

Cluster configuration

Database Name	new-cluster
Owning team	guild-24x7 ▾
PostgerSQL Version	9.6 ▾
DNS Name:	new-cluster.acid.staging.
Number of instances	1
Replica Load Balancer	☐ Enable Replica ELB
Volume Size	10 Gi
Office/DC IP ranges	☐ Datacenter 1 ☐ Datacenter 2 ☐ Berlin
AWS NAT IPs	1. / 32 2. / 32 3. / 32
Odd Host	IP / 32

Copy Definiton | Create Cluster

zalando

Waiting for master to become available

PostgreSQL cluster status **zmon-nc-10-3** edit

Cluster YAML definition

```yaml
apiVersion: acid.zalan.do/v1
kind: Postgresql
metadata:
  creationTimestamp: '2017-06-07T16:31:33Z'
  labels:
    team: zmon
  name: zmon-nc-10-3
  namespace: default
spec:
  allowedSourceRanges:
  numberOfInstances: 1
  postgresql:
    version: '10'
  teamId: zmon
  volume:
    size: 10Gi
status: Running
```

Checking status of Cluster

- Waiting for master to become available
- First PostgreSQL cluster container spawned
- StatefulSet created
- PostgreSQL 3rd party object created
- Create request successful

zalando

Cluster create completed

PostgreSQL cluster status **zmon-nc-10-3** edit

Cluster YAML definition

```yaml
apiVersion: acid.zalan.do/v1
kind: Postgresql
metadata:
  creationTimestamp: '2017-06-07T16:31:33Z'
  labels:
    team: zmon
  name: zmon-nc-10-3
  namespace: default
spec:
  allowedSourceRanges:
    -
  numberOfInstances: 1
  postgresql:
    version: '10'
  teamId: zmon
  volume:
    size: 10Gi
status: Running
```

Checking status of Cluster

PostgreSQL ready: **nc-10-3.zmon** (DNS may be slow)

PostgreSQL master available, label is attached

First PostgreSQL cluster container spawned

StatefulSet created

PostgreSQL 3rd party object created

Create request successful

zalando

Deployment to production

YAML file → Git commit → Deployment → K8S apply

zalando

Employee and application roles

Employee roles get created based on owning teams

Employee login via oauth2 tokens, no credential sync, MFA

Application roles:

- Declared in manifest
- Credentials stored in kubernetes secret
- Applications can mount secret
- Credentials don’t leave the system

Monitoring

Integration with ZMON

My Team's clusters (2)

Team	Instances	Name	Monitoring
zmon	1	zmon-nc-10-3	Node metrics Diskspace Tables Indexes
zmon	2	zmon-new-cluster-replica-lb	Node metrics Diskspace Tables Indexes

All clusters (169)

search:

Team	Instances	Name	Monitoring
ACID	2	acid-testcluster	Node metrics Diskspace Tables Indexes
ACID	2	acid-testcluster-42	Node metrics Diskspace Tables Indexes
DBAAS	3	contentbrokerpicasso	Node metrics Diskspace Tables Indexes
Gerrymanders	2	prodstoreaxiom13	Node metrics Diskspace Tables Indexes
Gerrymanders	3	productstoreaxiom1	Node metrics Diskspace Tables Indexes
Gerrymanders	2	productstoreaxiom5	Node metrics Diskspace Tables Indexes
Gerrymanders	3	productstoreaxiom9	Node metrics Diskspace Tables Indexes

zalando

Monitoring with pgview.web

CPU (Cores: 4)

Idle %	System %	User %	IOWait	Load 1	Load 5	Load 15
42.4	19.8	30.5	7.4	1.58	0.53	0.29

Memory

Total	Free	Buffers	Cached	Dirty	Commit Limit	Commited As
32.16GB	1.140GB	5.294GB	22.60GB	114.5MB	B	B

Partitions

Device	await	read	write	Free	Total	Path	Size
data	9.37	8960	132352	8.271GB	11.49GB	/home/postgres/pgdata/pgroot/data	542.9MB
wal	9.37	8960	132352	8.271GB	11.49GB	/home/postgres/pgdata/pgroot/data/pg_xlog	201.4MB

Processes 3 / 6 of maximum 100

Play Non Backends

PID	Lock	Type	utime	stime	read	write	age	DB	User	Query
59		logger	0	0	0	0				
201		checkpointer	0	0	0	0				
202		writer	0	0	0	0				
203		stats collector	0	0	0	0				
312		backend	0	0	0	0		postgres	postgres	idle in transaction (aborted)
383		backend	68	31	0	108864	5	postgres	postgres	insert into test select id from generate_series(1, 100000000) id;
445		wal writer	0	0	0	0				
446		autovacuum launcher	0	0	0	0				
447		archiver	0	0	0	0				last was 000000030000000600000044
466	383	backend	0	0	0	0	1	postgres	postgres	lock table test;

zalando

EC2 Instance Metrics via ZMON

zalando

Learnings on AWS: not only credits!

EBS IOPS credits

CPU Credits (t2)

Start/Stop to change instance type
(EBS backed only!)

Prepare for instance stops:
Retirement, rolling upgrades,...

zalando

- Postgres Operator

  github.com/zalando-incubator/postgres-operator

- Spilo HA Docker image

  github.com/zalando/spilo

- Patroni

  github.com/zalando/patroni

- Web-based monitoring via Postgres background worker

  github.com/CyberDem0n/bg_mon

- PAM Oauth2 for PostgreSQL

  https://github.com/zalando-incubator/pam-oauth2

- Postgres Operator blog post

  https://jobs.zalando.com/tech/blog/postgresql-in-a-time-of-kubernetes/

Thank you!