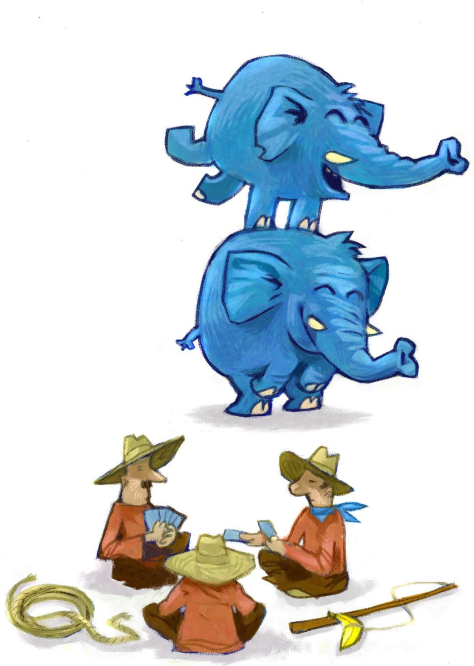
zalando



PG^3

PGDay'17 Russia
St. Petersburg

MURAT KABILOV

OLEKSII KLIUKIN

06-07-2017

ZALANDO AT A GLANCE

~3.6 billion EURO
revenue 2016

~200 million
visits per month

~20 million
active customers

~200,000
product choices

~2,000
brands

15
countries

>12,000
employees in Europe

50%
return rate across all categories

zalando

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

Evolution of database deployments

Manual deployments in data centers

- Wait for the hardware to be provisioned/configured
- Assign service IPs and create DNS entries
- Modify configuration files from templates:
  - postgresql.conf
  - recovery.conf
  - replica map
- Copy configuration to the destination systems
- Run scripts to create masters/replicas:

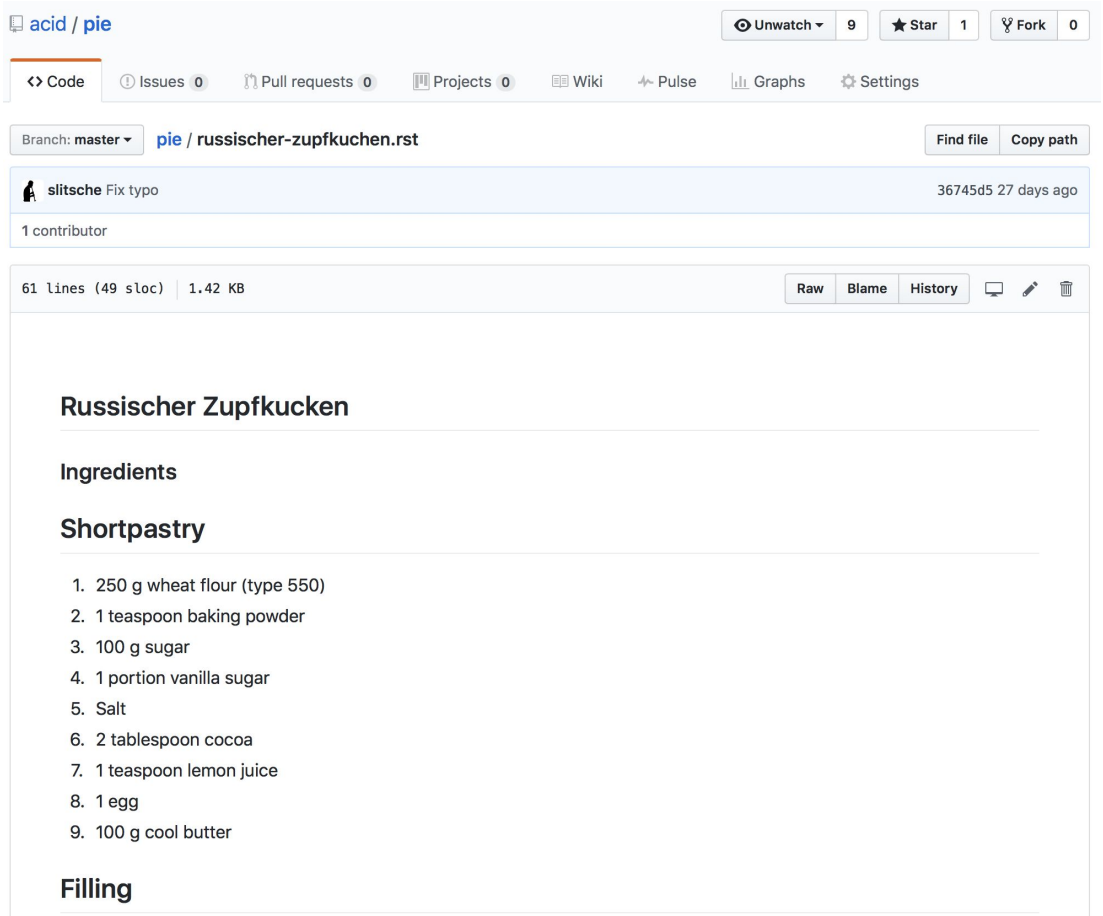
```
newdb2$ prepare_warmstdby.py -m newdb1.example.com
-n newdb -c
```

- Modify psql_* aliases for quick access to the databases

```
psql_newdb_LIVE --slave -c 'VACUUM ANALYZE'
```

Maintenance and HA in data centers

- Reliable hardware and internal networks
- Only manual failovers, happening not so often
- Disruptive maintenance performed manually during off-hours
- Some extra servers, mostly reliance on 3 replicas to keep DBs alive
- Can’t continue if primary DC is down

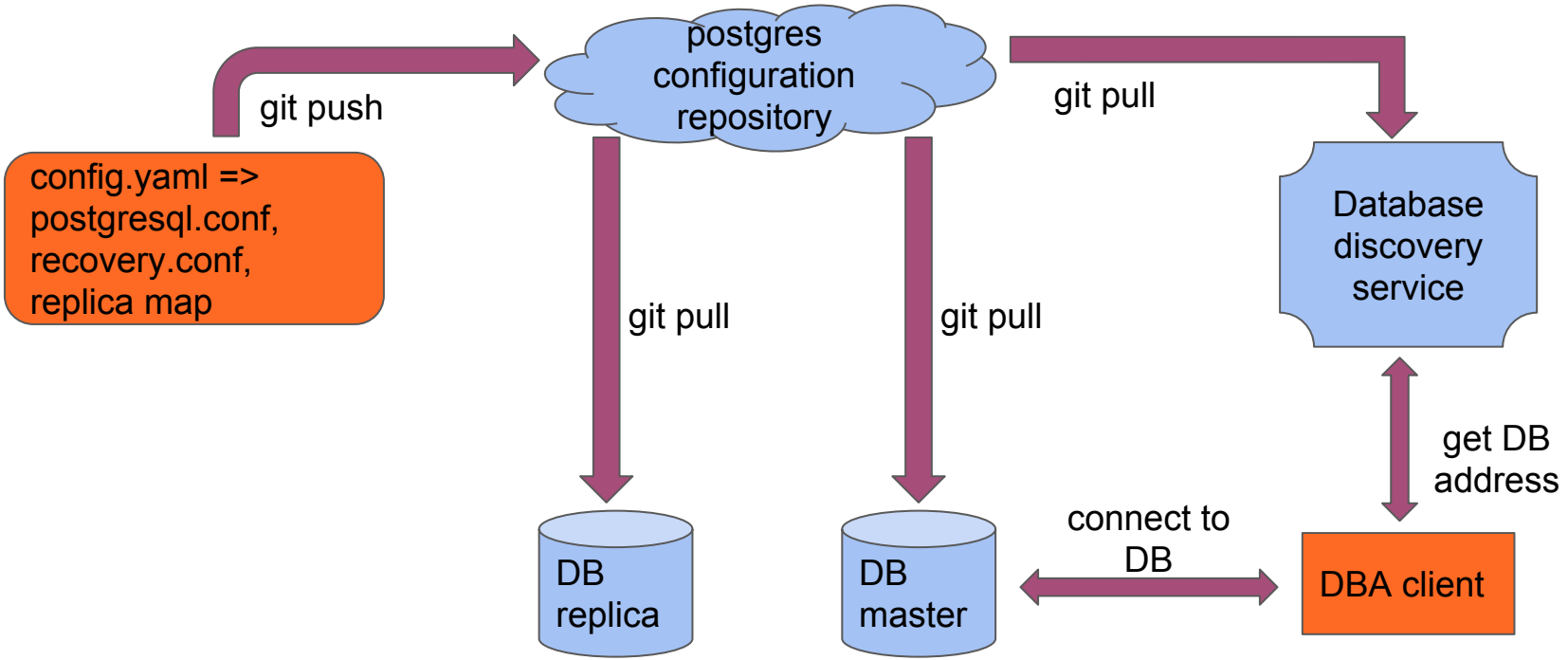

acid / pie
Unwatch 9
Star 1
Fork 0
Code
Issues 0
Pull requests 0
Projects 0
Wiki
Pulse
Graphs
Settings
Branch: master
pie / russischer-zupfkuchen.rst
Find file
Copy path
slitsche Fix typo
36745d5 27 days ago
1 contributor
61 lines (49 sloc) 1.42 KB
Raw
Blame
History
Russischer Zupfkucken
Ingredients
Shortpastry
1. 250 g wheat flour (type 550)
2. 1 teaspoon baking powder
3. 100 g sugar
4. 1 portion vanilla sugar
5. Salt
6. 2 tablespoon cocoa
7. 1 teaspoon lemon juice
8. 1 egg
9. 100 g cool butter
Filling

zalando

Add more automation

- YAML declarative configuration
- Store all configuration in git
- Update production servers with `git pull`
- RESTful discovery service to propagate connection information

Git-driven workflow in data centers

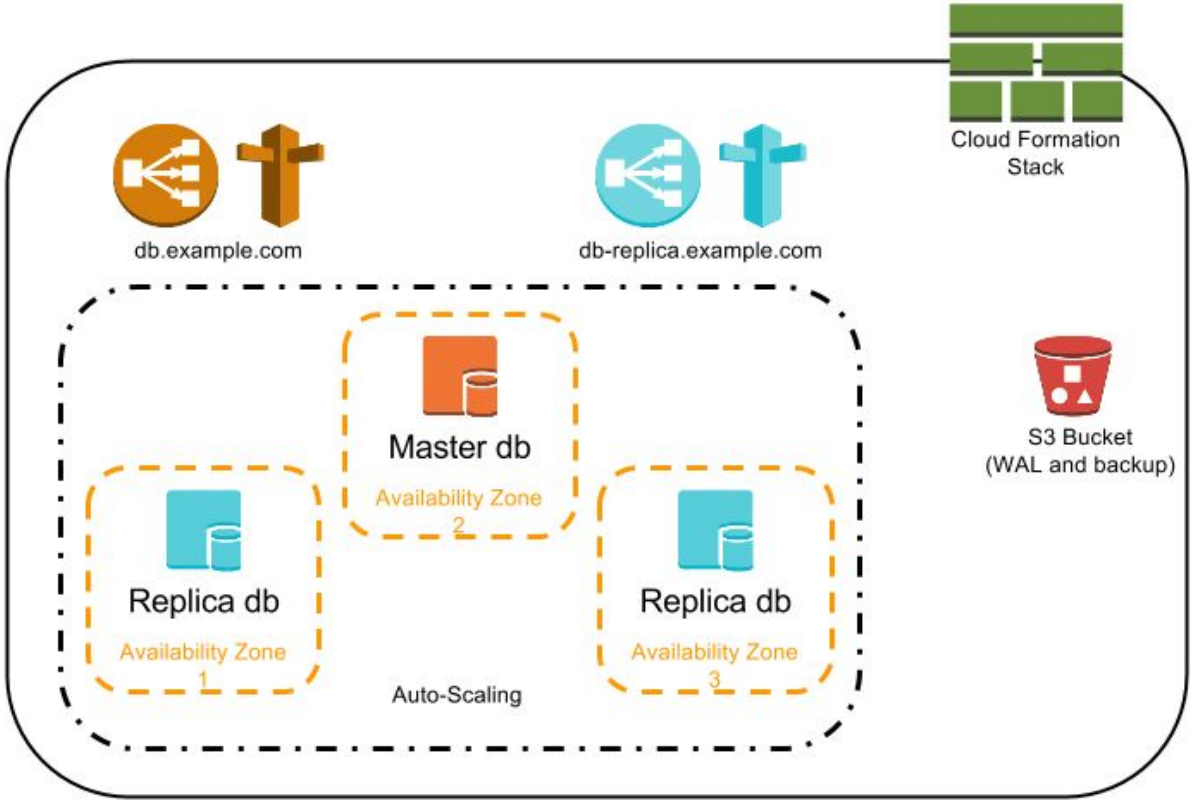


zalando

From traditional data centers to Amazon cloud

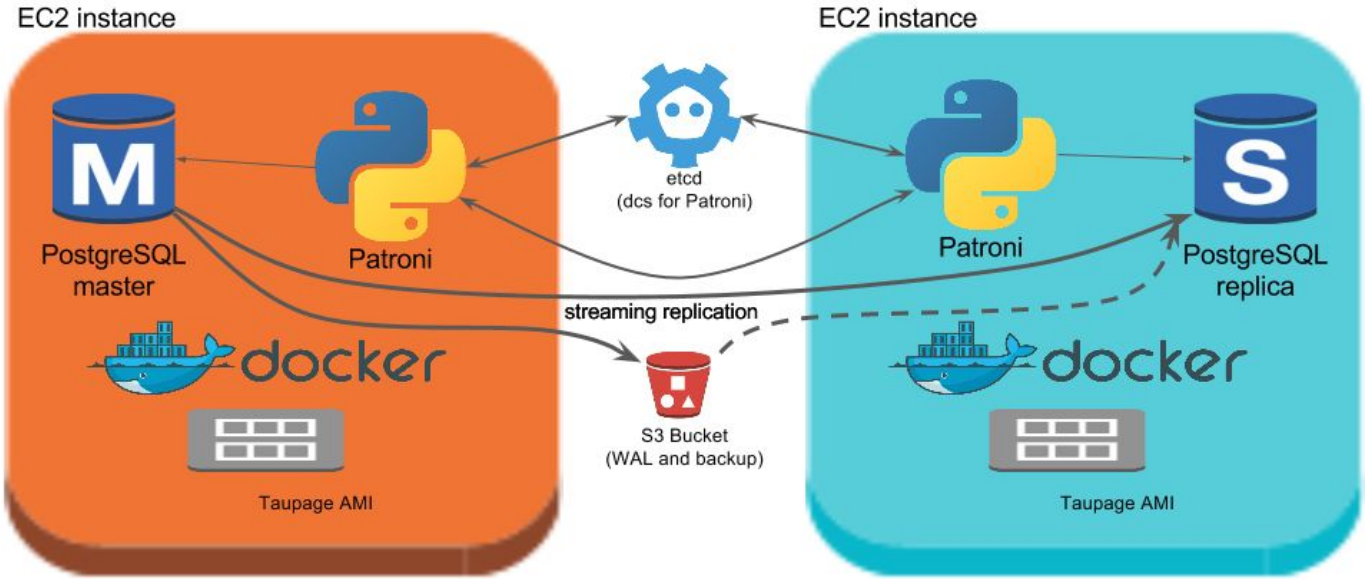
- No more waiting for the hardware (but mind the limits)
- Multi-DC (3 availability zones in Frankfurt)
- Based on STUPS: open-source Docker-based PAAS
- A unified tool (senza) for all deployments in the cloud
- Spilo: docker image for HA PostgreSQL databases
  - Declarative definition of database clusters (senza templates)
  - Automatic failovers and on-the-fly configuration changes
  - Based on Patroni: HA PostgreSQL
  - Stores cluster configuration and leader lock in Etcd (or Zookeeper, or Consul)
- Still manual deployments triggered by JIRA ticket

Spilo-based deployments in the cloud



zalando

Spilo-based deployments in the cloud



zalando

From databases in the cloud to the database as a service

Commercial Database as a service pros

- ‘One-click’ deployment of new databases
- Ability to choose cloud ‘hardware’
- Easy access to configuration and auto-tuning
- Storage auto-scaling
- Replication, PITR and automatic failover
- Maintenance windows for restarts and upgrades

Commercial Database as a service cons

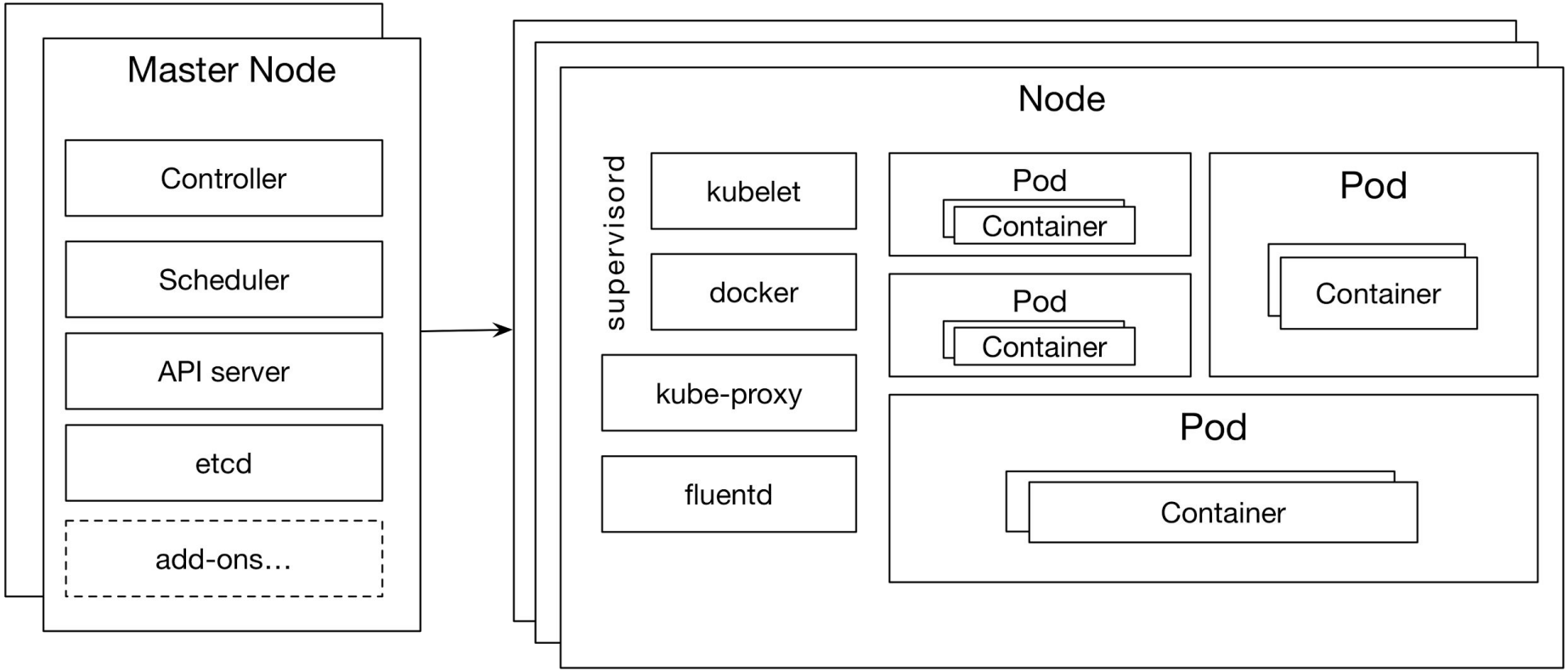
- Closed-source, custom backend implementation
- Vendor lock and limited possibilities to migrate
- Premium price
- Cannot fine-tune resource limits (memory, cpu)
- Non-stock running PostgreSQL
- Not always providing latest versions

Can we have all pros of PostgreSQL as a service without the cons?

Kubernetes introduction

- Container orchestration
- Cluster-wide application scheduling and autoscaling
- Application deployments automation
- Abstracts bare metal and most cloud providers (google, aws, azure, etc)
- Declarative description of resources and deployments
- Rich metadata (versions, labels, annotations)
- Supported by open-source community

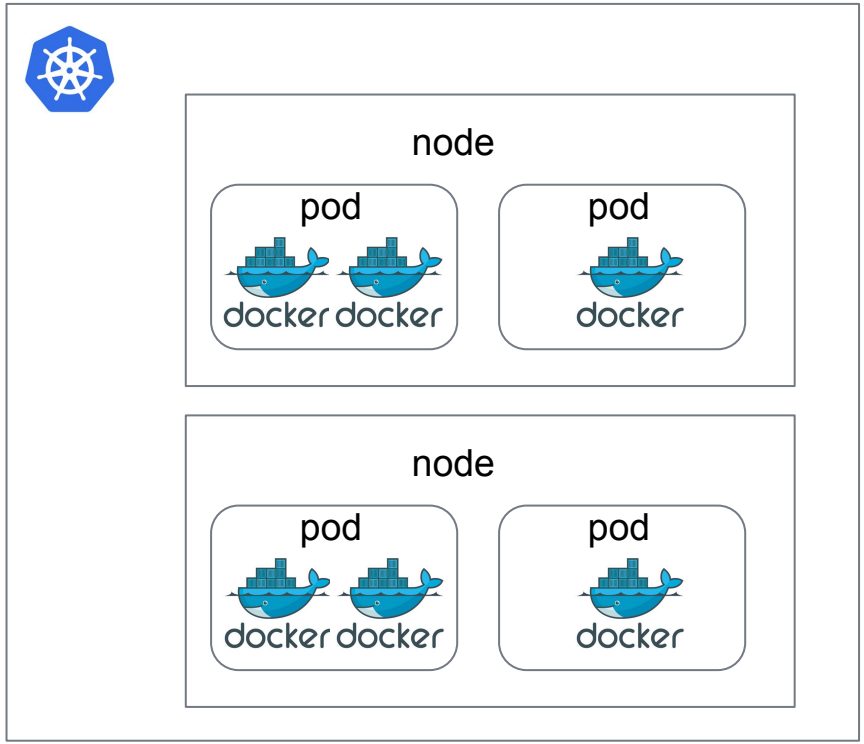
Kubernetes overview



zalando

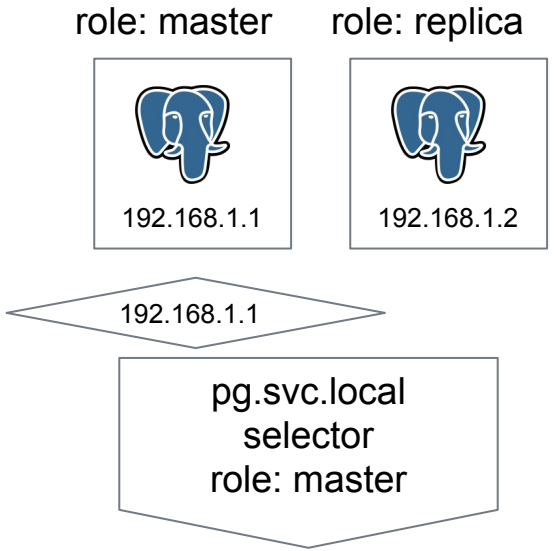
Nodes and pods

- Nodes map to servers
- Pods are scheduled on nodes
- Scheduling is controlled by resource limits
- Pods are owned by applications
- Each pod runs one or more containers
- Pods share node resources



Services and Endpoints

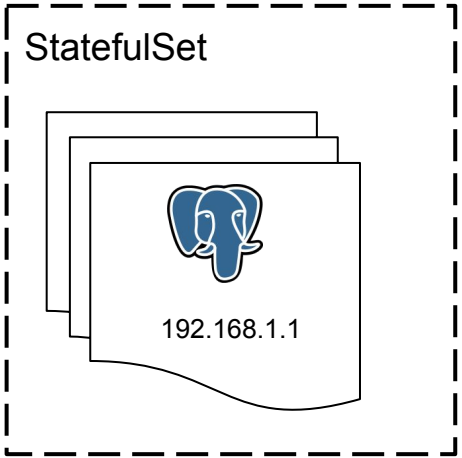
- Connect to applications running on pods
- Choose a pod with selectors
- Endpoints contain ip addresses/ports
- Endpoints can be managed separately



zalando

Running stateful applications

- Persistent Volumes
- Persistent Volume Claims
- Stateful sets: Pods + Persistent Volume Claims
- Pod keeps the attached Volume and IP
- Guaranteed number of Pods



ThirdPartyResources (TPR)

- A place to store user-defined metadata
- Cluster-wide
- Kubernetes 1.7+: Custom Resource Definition (CRD)

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

Cluster manifest with Patroni configuration

```
apiVersion: acid.zalan.do/v1
kind: Postgresql
metadata:
…
spec:
  allowedSourceRanges:
  - 192.168.1.0/24
  numberOfInstances: 2
  postgresql:
    version: "9.6"
  patroni:
    maximum_lag_on_failover: 33554432
    initdb:
      data-checksums: true
    pg_hba:
      - hostnossl all all all reject
      - hostssl all +team all pam
      - hostssl all all all md5
  teamId: acid
  volume:
    size: 10Gi
```

Deploying databases in Kubernetes

- Writing manifests manually
- Helm charts
- Operators

```
Name: patroni
Component: patroni
ImagePullPolicy: IfNotPresent
Spilo:
 Image:
registry.opensource.zalan.do/acid/spilo-9.6
 Version: 1.2-p17
Replicas: 5
Resources:
 Cpu: 100m
 Memory: 512Mi
Credentials:
 Superuser: tea
 Admin: cola
 Standby: pinacolada
Etcd:
 Host: # fill-in value for etcd host use the
discovery parameter
 Discovery:
persistentVolume:
 size: 1G
 mountPath: "/home/postgres/data"
 accessModes:
   - ReadWriteOnce
```

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

Kelsey Hightower
@kelseyhightower

Following

If managed services are off the table consider deploying to virtual machines. Learn how to do upgrades, tuning, backups, AND recovery.

7:01 PM - 20 Jan 2017

11 Retweets **27** Likes

Kubernetes Operators

An Operator represents human operational knowledge in software to reliably manage an application.

Postgres Operator

- Defines and watches PostgreSQL third-party resource type
- Creates new PostgreSQL clusters
- Keeps clusters in sync with TPR manifest
- Handles updates and deletes

Kubernetes objects controlled by the operator

- Stateful sets

  Manage pods, makes sure a persistent volume gets reattached to the pod after restart

- Persistent volume claims

  Provide storage for the PostgreSQL database

- Services

  Database access from the applications

- Endpoints

  Direct queries to the master pod

Layer by layer

- Operator starts pods with Spilo docker image
- Operator provides environment variables to Spilo
- Spilo generates Patroni configuration
- Patroni creates roles and configures PostgreSQL
- Patroni makes sure there is only one master
- Patroni uses Etcd to keep the leader lock
- Patroni creates roles and applies configuration
- Patroni changes service endpoints on failover
- Operator makes sure all Kubernetes objects are in sync

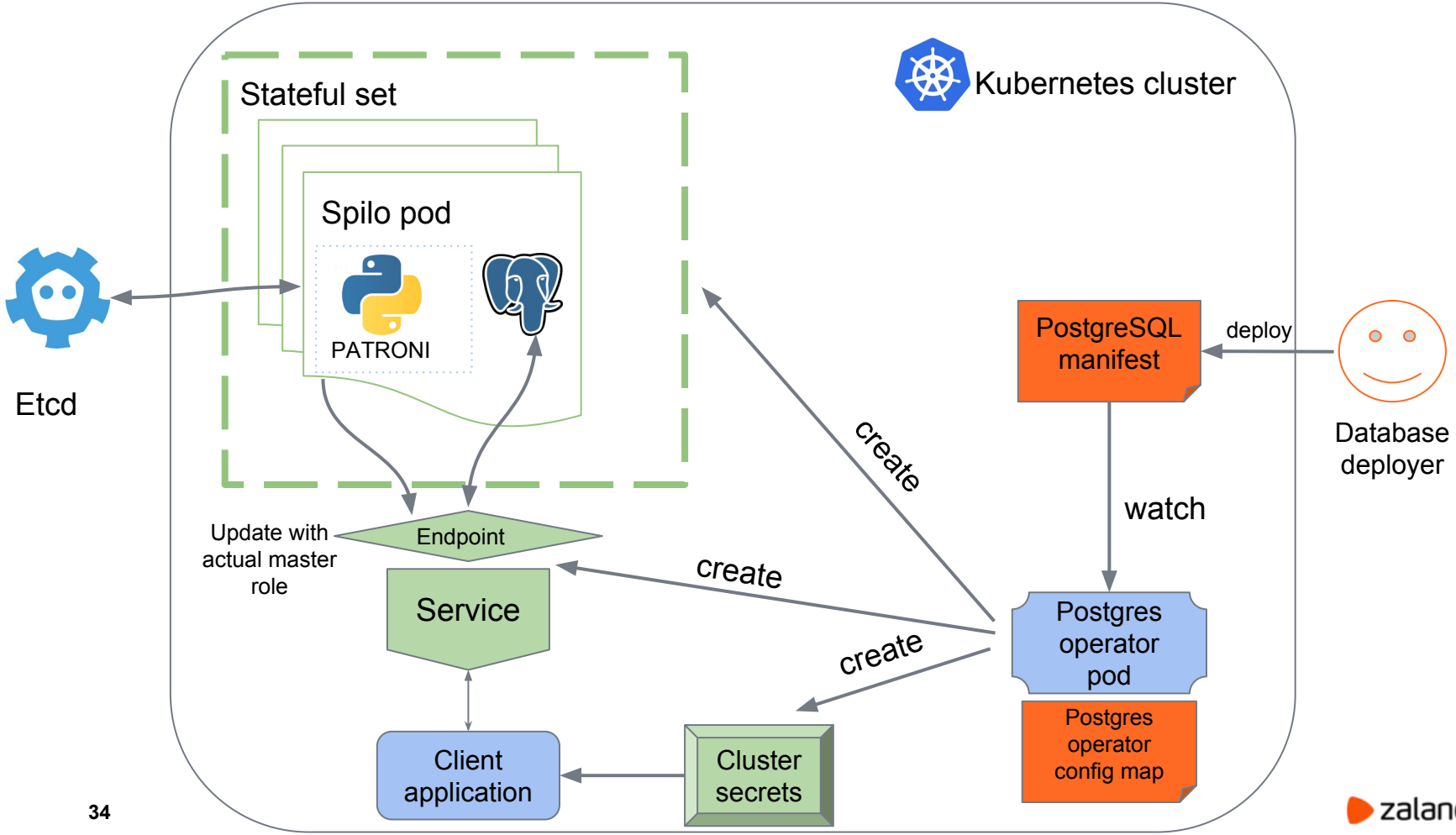
Operator configures pods to run Spilo

```
spec:
      containers:
      - env:
        - name: SCOPE
          value: acid-newdatabase-01
        - name: PGROOT
          value: /home/postgres/pgdata/pgroot
        - name: ETCD_HOST
          value: etcd-server.etcd.example.com:2379
...
        - name: SPILO_CONFIGURATION
          value:
'{"postgresql":{"bin_dir":"/usr/lib/postgresql/9.6/bin"},"bootstrap":{"initdb":[{"auth-host":"md5"},{"auth-loca
l":"trust"}],"users":{"human":{"password":"","options":["CREATEDB","NOLOGIN"]}},"pg_hba":["hostnossl
            all all all reject","hostssl   all +team all pam","hostssl   all all
            all md5"],"dcs":{}}}'
        - name: WAL_S3_BUCKET
          value: eu-central-1-spilo-example
        image: registry.opensource.zalan.do/acid/spiloprivate-10:1.3-p2
```

Spilo configures Patroni

```
scope: acid-newdatabase-01
postgresql:
  data_dir: /home/postgres/pgdata/pgroot/data
  bin_dir: /usr/lib/postgresql/9.6/bin
...
  callbacks:
    on_restart: /callback_role.py
    on_role_change: /callback_role.py
...
bootstrap:
  initdb:
    - auth-host: md5
    - auth-local: trust
    pg_hba:
    - hostssl replication standby 0.0.0.0/0 md5
    - hostnossl all all all reject
    - hostssl   all +team all pam
    - hostssl   all all all md5
  users:
    human:
      options:
      - CREATEDB
      - NOLOGIN
      password: ''
etcd:
  host: etcd-server.etcd.example.com:2379
```


zalando


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

zalando

Operator features

- Automatic creation of users (using external REST service)
- Oauth2 authentication via PAM
- Optional load-balancer for external access
- Resize EBS volumes (on AWS)
- Act on multiple clusters in parallel
- Infrastructure roles

Security

- Service accounts
- allowedSourceRanges
- Role secrets
- PAM Oauth2 for human users

Operator configuration

```
service_account_name: operator
teams_api_url: http://fake-teams-api.default.svc.cluster.local
workers: "4"
enable_load_balancer: "true"
db_hosted_zone: db.example.com
dns_name_format: '{cluster}.{team}.staging.{hostedzone}'
docker_image: registry.opensource.zalan.do/acid/spiloprivate-9.6:1.2-p4
etcd_host: etcd-client.default.svc.cluster.local:2379
infrastructure_roles_secret_name: postgresql-infrastructure-roles
oauth_token_secret_name: postgresql-operator
pam_configuration: |
  https://info.example.com/oauth2/tokeninfo?access_token= uid realm=/employees
pam_role_name: zalandos
replication_username: replication
```

Accessing logs

- CSV foreign data wrapper on the host
- kubectl logs
- Export to external log storage (i.e. scalyr)

LIVE

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

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

Upcoming operator features

- Snapshots: create new cluster based on the existing one
- PITR
- Reduce number of Pod restarts (e.g. during nodes upgrades)
- Operator API: stats, debug
- Paused clusters

- 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!