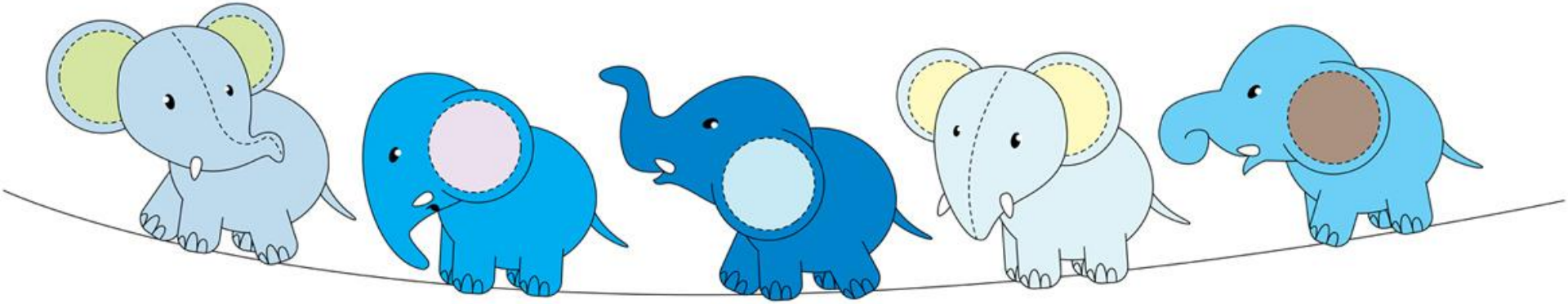


Spilo

Giving you control

Feike Steenbergen
PostgreSQL Meetup Berlin
18 June 2015

zalando

Spilo

- High Available PostgreSQL cluster
- Automatic failover
- Current implementation is for AWS
  - AutoScalingGroup: Dead instances replaced
  - Elastic Load Balancer: Route traffic to master
- Uses etcd for distributed consensus

Demo part 1

```
feike@fsteenbergen /tmp $ senza init spilo.yaml
Please select the project template
1) bgapp: Background app with single EC2 instance
2) postgresapp: HA Postgres app, which needs an S3 bucket to store WAL files
3) webapp: HTTP app with auto scaling, ELB and DNS
Please select (1-3): 2
Postgres WAL S3 bucket to use [zalando-spilo-app]:
EC2 instance type [t2.micro]:
Hosted Zone [acid.example.com]:
ETCD Discovery Domain [postgres.example.com]:
Database volume size (GB, 10 or more) [10]:
Database volume type (gp2, io1 or standard) [gp2]:
ID of the snapshot to populate EBS volume from []:
Filesystem for the data partition [ext4]:
Filesystem mount options (comma-separated) [noatime,nodiratime,nobarrier]:
Mint S3 bucket name []: nonsensical
Account key for your scalyr account []: nonsensical
Checking security group app-spilo.. OK
Checking S3 bucket zalando-spilo-app.. OK
Generating Senza definition file spilo.yaml.. OK
```

```
feike@fsteenbergen /tmp $ senza create spilo.yaml mycluster docker-registry.example.com/db/spilo:1.0
feike@fsteenbergen /tmp $ senza events spilo mycluster
```

Stack Name	Ver.	Resource Type	Resource ID	Status	Status Reason	Event Time
spilo	mycluster	CloudFormation::Stack	spilo-mycluster	CREATE_IN_PROGRESS	User Initiated	11s ago
spilo	mycluster	IAM::Role	PostgresAccessRole	CREATE_IN_PROGRESS		7s ago
spilo	mycluster	ElasticLoadBalancing::LoadBalancer	PostgresLoadBalancer	CREATE_IN_PROGRESS		7s ago
spilo	mycluster	ElasticLoadBalancing::LoadBalancer	PostgresLoadBalancer	CREATE_IN_PROGRESS	Resource creation Initiated	6s ago
spilo	mycluster	ElasticLoadBalancing::LoadBalancer	PostgresLoadBalancer	CREATE_COMPLETE		5s ago
spilo	mycluster	Route53::RecordSet	PostgresRoute53Record	CREATE_IN_PROGRESS		2s ago
spilo	mycluster	Route53::RecordSet	PostgresRoute53Record	CREATE_IN_PROGRESS	Resource creation Initiated	1s ago

```
feike@fsteenbergen /tmp $ senza resources spilo mycluster
```

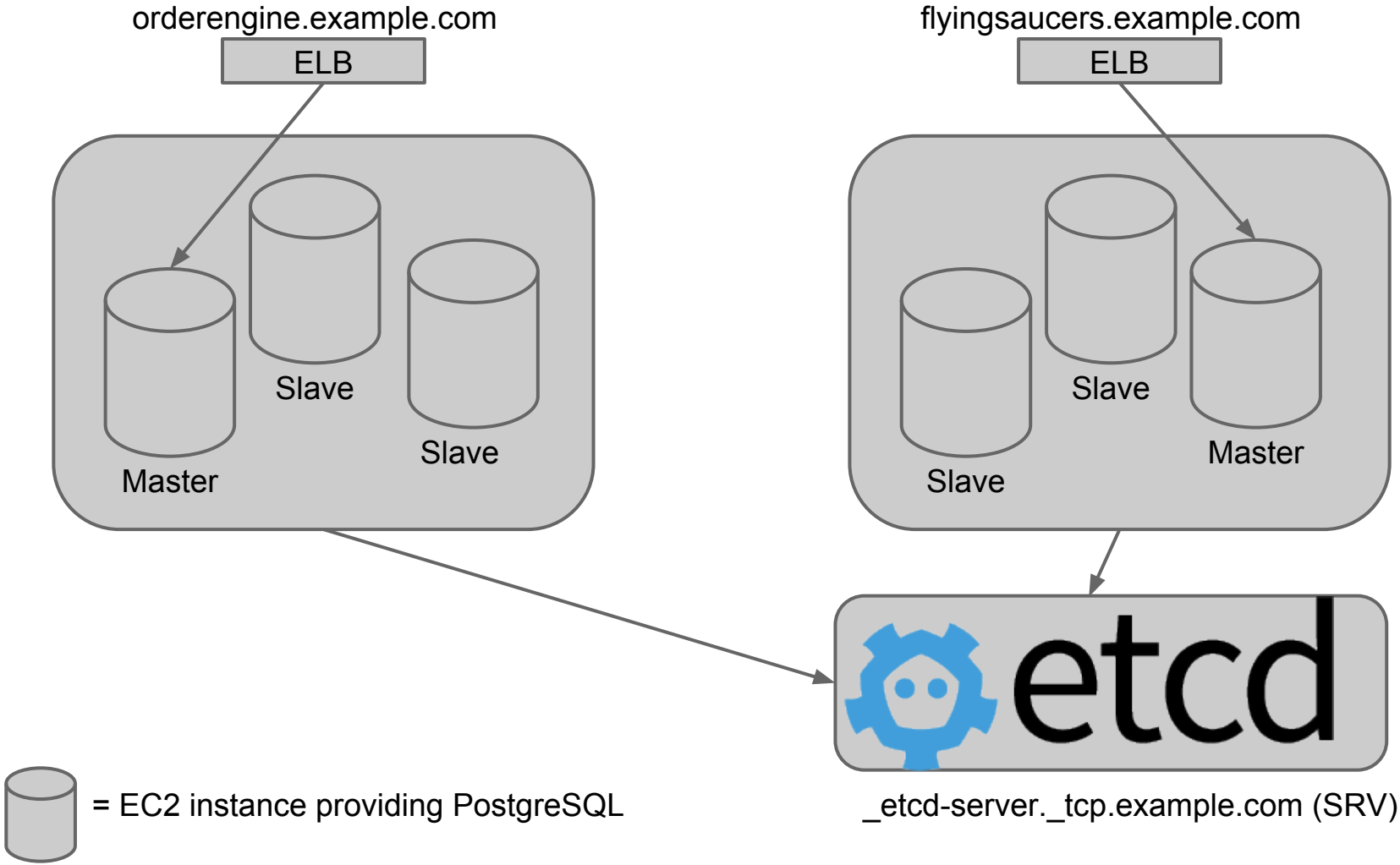
Stack Name	Ver.	Resource ID	Resource Type	Status	Created
spilo	mycluster	AppServer	AutoScaling::AutoScalingGroup	CREATE_COMPLETE	7m ago
spilo	mycluster	AppServerConfig	AutoScaling::LaunchConfiguration	CREATE_COMPLETE	10m ago
spilo	mycluster	AppServerInstanceProfile	IAM::InstanceProfile	CREATE_COMPLETE	10m ago
spilo	mycluster	AppServerScaleDown	AutoScaling::ScalingPolicy	CREATE_COMPLETE	7m ago
spilo	mycluster	AppServerScaleUp	AutoScaling::ScalingPolicy	CREATE_COMPLETE	7m ago
spilo	mycluster	PostgresAccessRole	IAM::Role	CREATE_COMPLETE	12m ago
spilo	mycluster	PostgresLoadBalancer	ElasticLoadBalancing::LoadBalancer	CREATE_COMPLETE	12m ago
spilo	mycluster	PostgresRoute53Record	Route53::RecordSet	CREATE_COMPLETE	10m ago

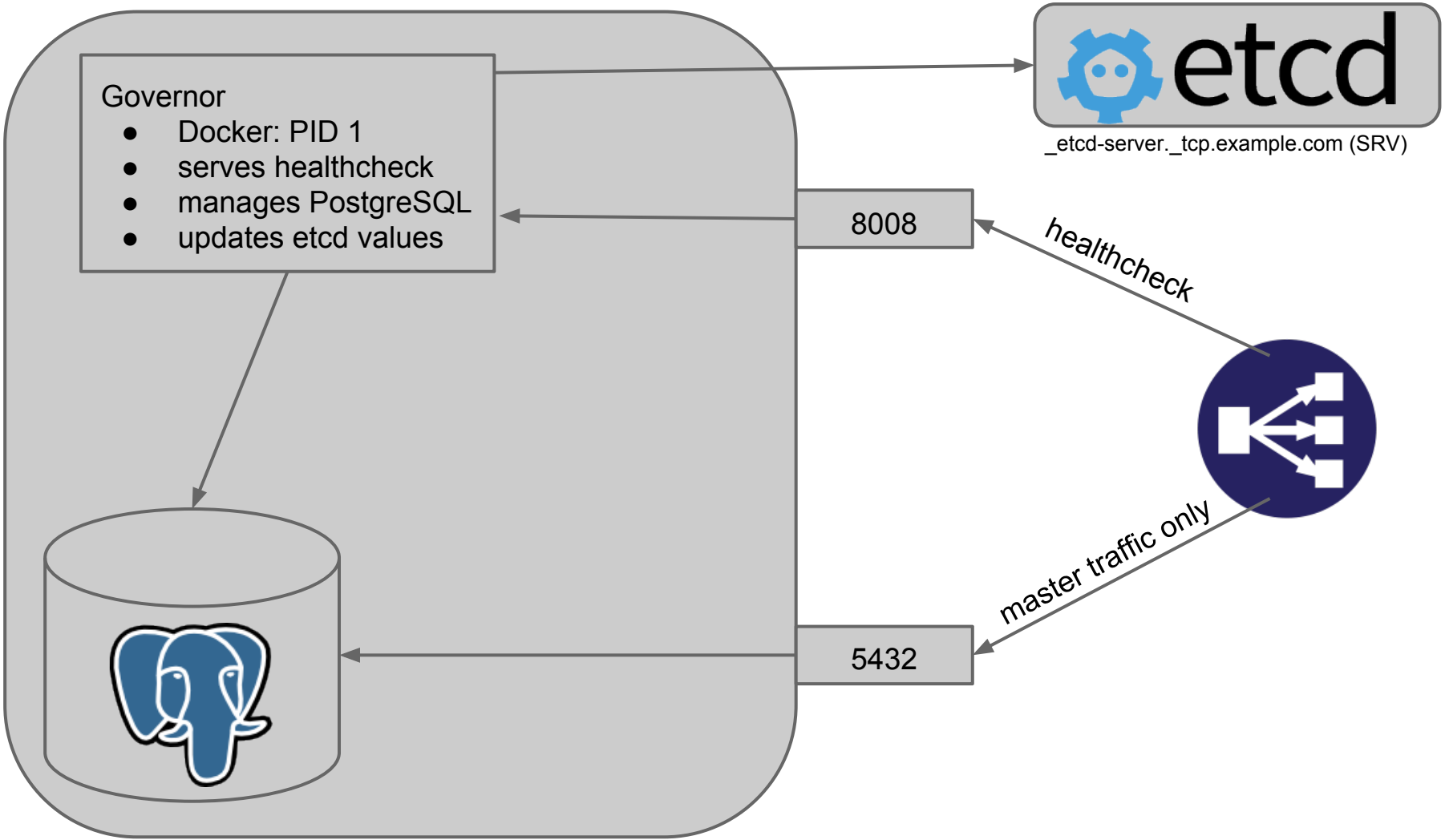
Why build Spilo?

- Less human interaction
  - when creating a cluster
  - during failover
- Freedom and control
  - No dba needed to create a new cluster
  - No restrictions on usage of the cluster
  - Shifts responsibility to the teams
- Flexibility
  - Can run anywhere (your own datacenter, cloud)

Etcd?

- A HA distributed key/value store
- We use it to provide distributed consensus
- Not without issues
- Alternatives:
  - Apache ZooKeeper
  - Consul by HashiCorp
  - doozerd


orderengine.example.com
ELB
Master
Slave
Slave
flyingsaucers.example.com
ELB
Slave
Slave
Master
etcd
= EC2 instance providing PostgreSQL
_etcd-server._tcp.example.com (SRV)


Governor
Docker: PID 1
serves healthcheck
manages PostgreSQL
updates etcd values
etcd
_etcd-server._tcp.example.com (SRV)
8008
healthcheck
master traffic only
5432

Demo part 2

```
feike@fsteenbergen /tmp $ senza instances spilo meetup
Stack Name|Ver.  |Resource ID|Instance ID|Public IP|Private IP    |State      |LB Status      |Launched
spilo      meetup AppServer   i-01b103ab            172.31.144.92  RUNNING     OUT_OF_SERVICE   2h ago
spilo      meetup AppServer   i-40df81ea            172.31.170.44  RUNNING     IN_SERVICE       2h ago
spilo      meetup AppServer   i-60b27699            172.31.131.208 RUNNING     OUT_OF_SERVICE  35m ago
spilo      meetup AppServer   i-849f5b7d                           TERMINATED                   2h ago
feike@fsteenbergen /tmp $ aws ec2 terminate-instances --instance-ids i-40df81ea
                     |                     |            | 172.31.144.92   | postgresql_d94d9df8e100
 2015-06-18 12:43:52 | 2015-06-18 13:26:05 |       5034 | 172.31.131.208 +| postgresql_21379e4ede0f+
                     |                     |            | 172.31.144.92   | postgresql_d94d9df8e100
2015-06-18 15:26:05 Unable to connect to PostgreSQL
2015-06-18 15:26:07 Unable to connect to PostgreSQL
2015-06-18 15:26:08 Unable to connect to PostgreSQL
2015-06-18 15:26:09 Unable to connect to PostgreSQL
2015-06-18 15:26:10 Unable to connect to PostgreSQL
2015-06-18 15:26:11 Unable to connect to PostgreSQL
2015-06-18 15:26:12 Unable to connect to PostgreSQL
2015-06-18 15:26:14 Unable to connect to PostgreSQL
2015-06-18 15:26:15 Unable to connect to PostgreSQL
2015-06-18 15:26:16 Unable to connect to PostgreSQL
2015-06-18 15:26:17 Unable to connect to PostgreSQL
2015-06-18 15:26:18 Unable to connect to PostgreSQL
2015-06-18 15:26:19 Unable to connect to PostgreSQL
 2015-06-18 12:49:40 | 2015-06-18 13:26:21 |       5036 | 172.31.131.208  | postgresql_b280748b2f72+
                     |                     |            |                 | postgresql_d94d9df8e100
 2015-06-18 12:49:40 | 2015-06-18 13:26:23 |       5038 | 172.31.131.208  | postgresql_b280748b2f72+
                     |                     |            |                 | postgresql_d94d9df8e100
```

Leader key in etcd for a cluster

```
feike@fsteenbergen /tmp $ curl -s "http://localhost:22379/v2/keys/service/mycluster/leader"
{
  "action": "get",
  "node": {
    "createdIndex": 3427317,
    "expiration": "2015-06-18T13:39:09.694280263Z",
    "key": "/service/mycluster/leader",
    "modifiedIndex": 3429356,
    "ttl": 22,
    "value": "postgresql_62cc29af75ee"
  }
}
```

```
feike@fsteenbergen /tmp $ curl -s "http://localhost:22379/v2/keys/service/mycluster/members" | jq --sort-k
{
  "action": "get",
  "node": {
    "createdIndex": 3427309,
    "dir": true,
    "key": "/service/mycluster/members",
    "modifiedIndex": 3427309,
    "nodes": [
      {
        "createdIndex": 3430347,
        "expiration": "2015-06-18T13:49:51.96333074Z",
        "key": "/service/mycluster/members/postgresql_62cc29af75ee",
        "modifiedIndex": 3430347,
        "ttl": 47,
        "value": "postgres://standby:standby@172.31.134.64:5432/postgres?application_name=http://172.31.13
      },
      {
        "createdIndex": 3430932,
        "expiration": "2015-06-18T14:48:56.821179877Z",
        "key": "/service/mycluster/members/postgresql_31f63c364fee",
        "modifiedIndex": 3430932,
        "ttl": 3591,
        "value": "postgres://standby:standby@172.31.170.252:5432/postgres?application_name=http://172.31.1
      },
      {
```

Roadmap

- Features
  - Multi-master replication
  - Implement `pg_rewind` to reuse old masters
  - Use different distributed consensus implementations
- Tooling
  - Manual failover
  - Restore from backup
  - PostgreSQL upgrade

Thanks!

- The governor: https://github.com/zalando/governor
- Spilo (implements Governor on AWS) https://github.com/zalando/spilo/
- Our tech blog https://tech.zalando.com/


zalando