


The unbreakable, scalable elephant -  
Patroni automation with Ansible



## Who we are



### The Company

- > Founded in 2010
- > More than 70 specialists
- > Specialized in the Middleware Infrastructure
  - > The invisible part of IT
- > Customers in Switzerland and all over Europe

### Our Offer

- > Consulting
- > Service Level Agreements (SLA)
- > Trainings
- > License Management



## About me



**Julia Gugel**

Consultant

+41 78 320 43 07

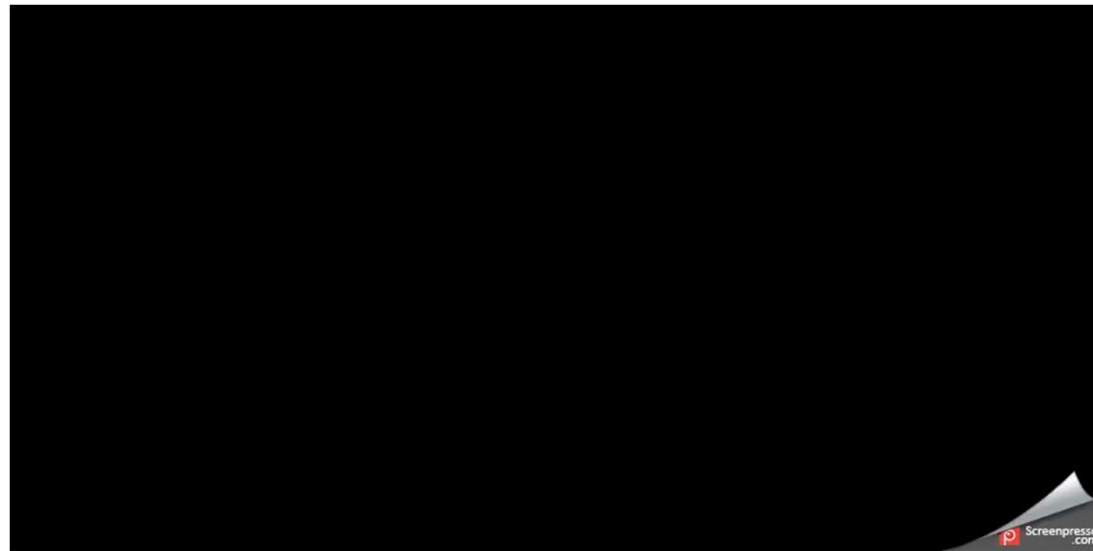
julia.gugel[at]dbi-services.com



# Agenda



1. Ansible
2. Patroni
3. Little helpers
4. Demo
5. Conclusion

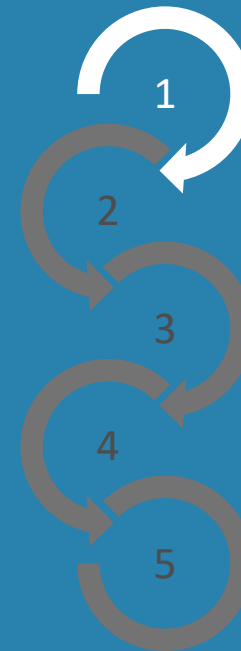


# Ansible



- > What is Ansible?
- > Playbooks, Modules & Co.
- > How to run a playbook

The unbreakable, scalable elephant



# Ansible

## What is Ansible?



### Overview

- > Simple IT automation engine
- > Designed for multi-tier deployments
- > Does not use agents
- > Pushing small programs "Ansible modules" to the nodes
- > Playbooks written in YAML
  - > No need to know the commands to accomplish tasks



# Ansible

Playbooks, Modules & Co.



## Inventory

- > Set of Hosts

## Task

- > Call to an Ansible module

## Module

- > Unit of code which is implemented by Ansible on the host

## Play

- > One or more tasks executed on a particular host

## Playbooks

- > One or more plays
- > Plays may be executed on similar or different hosts



# Ansible

## Playbooks, Modules & Co.



### Roles

- > Group the resources according to particular functionality
- > Directories for
  - > Variables
  - > Templates
  - > Tasks
  - > .....
- > Specify roles within playbooks to use them



### Templates

- > Transfers templated files to remote hosts
- > Templating happens on the Ansible controller before the task is sent
- > Stored in j2 format





# Ansible

## How to run a playbook



## One simple command to run a playbook

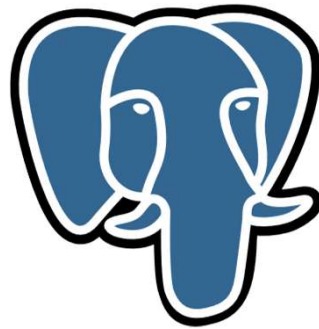
```
[root@ansible]:/home/ansible/roles# ansible-playbook -i ../patroni patroni/site.yml
PLAY [patroni-servers] *****

TASK [Gathering Facts] *****
ok: [192.168.22.112]
ok: [192.168.22.113]
ok: [192.168.22.111]
```

.....and all the magic starts.....

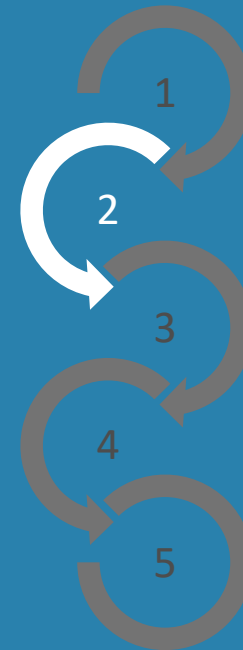


# Patroni



- > What is Patroni?
- > Requirements

The unbreakable, scalable elephant



## Patroni

What is Patroni?



Zalando's Python-based PostgreSQL controller

Template to create high-availability solutions

> Far from being a one-size-fits-all or plug-and-play replication system

**Fork of Governor with plenty of new features**

> Support for Consul or Zookeeper

> Dynamical reconfiguration of all cluster members at once

> Support for watchdog on Linux

In active development and accepts contributions



## Patroni

What is Patroni?



### Components

- > One writable node (primary/leader)
- > Multiple read-only replicas (secondary)

Supports manual and automatic failovers



### Problem:

- > How to know who is leader? Is the leader present? Who gets the new leader?



# Patroni

## Requirements



## Packages

```
[root@patroni2 ~]# yum install etcd haproxy libyaml python
[root@patroni2 ~]# yum install python-psycopg2
[root@patroni2 ~]# pip install psycopg2-binary
[root@patroni2 ~]# pip install psycopg2>=2.5.4
```

## A distributed key value store (dcs)

> etcd, consul, zookeeper.....

```
[root@patroni2 ~]# pip install patroni[etcd]
```

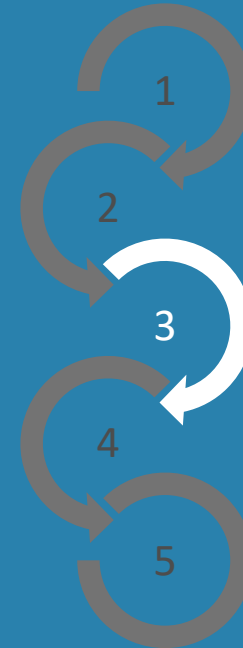


## Little helpers

- > etcd
- > HAProxy
- > watchdog



The unbreakable, scalable elephant



18.10.2019

Page 14

## Little helpers

etcd

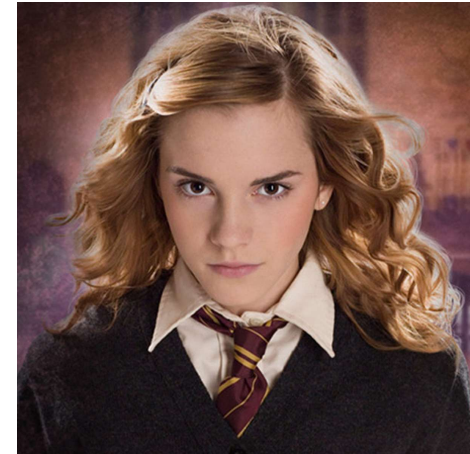


### What is etcd?

- > Distributed Key value store to store data across distributed systems
- > Stores information about the Cluster status, available nodes and resources
- > Needs an uneven number of members
- > Leader is elected by the other members

### What is it used for in combination with Patroni?

- > Postgres nodes make use of etcd to keep the Postgres Cluster up and running



## Little helpers

### HAProxy



### What is HAProxy?

- > Solution for high availability, load balancing and proxying for TCP/HTTP-based applications
- > De-facto standard open source load balancer
- > Shipped with most mainstream Linux distributions
- > Often deployed by default in cloud platforms

### What is it used for in combination with Patroni?

- > Giving your application a single endpoint for connecting to the cluster's leader





## Little helpers

### watchdog



### What is watchdog?

- > Daemon/subsystem used to monitor the basic health of a machine
- > Device that triggers a system reboot if it detects
  - > The system hangs
  - > No more free memory
- > Can also trigger a script, before triggering a reboot

### What is it used for in combination with Patroni?

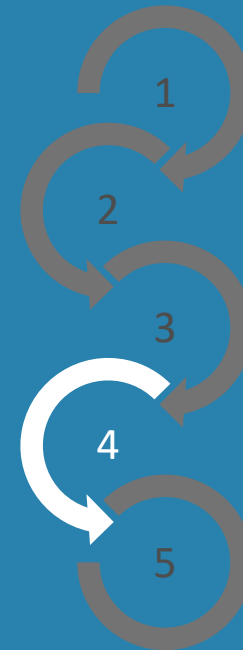
- > Automatic reboot when server gets unavailable due to failures



## Demo



- > Starting position
- > Adding a node
- > Removing a node



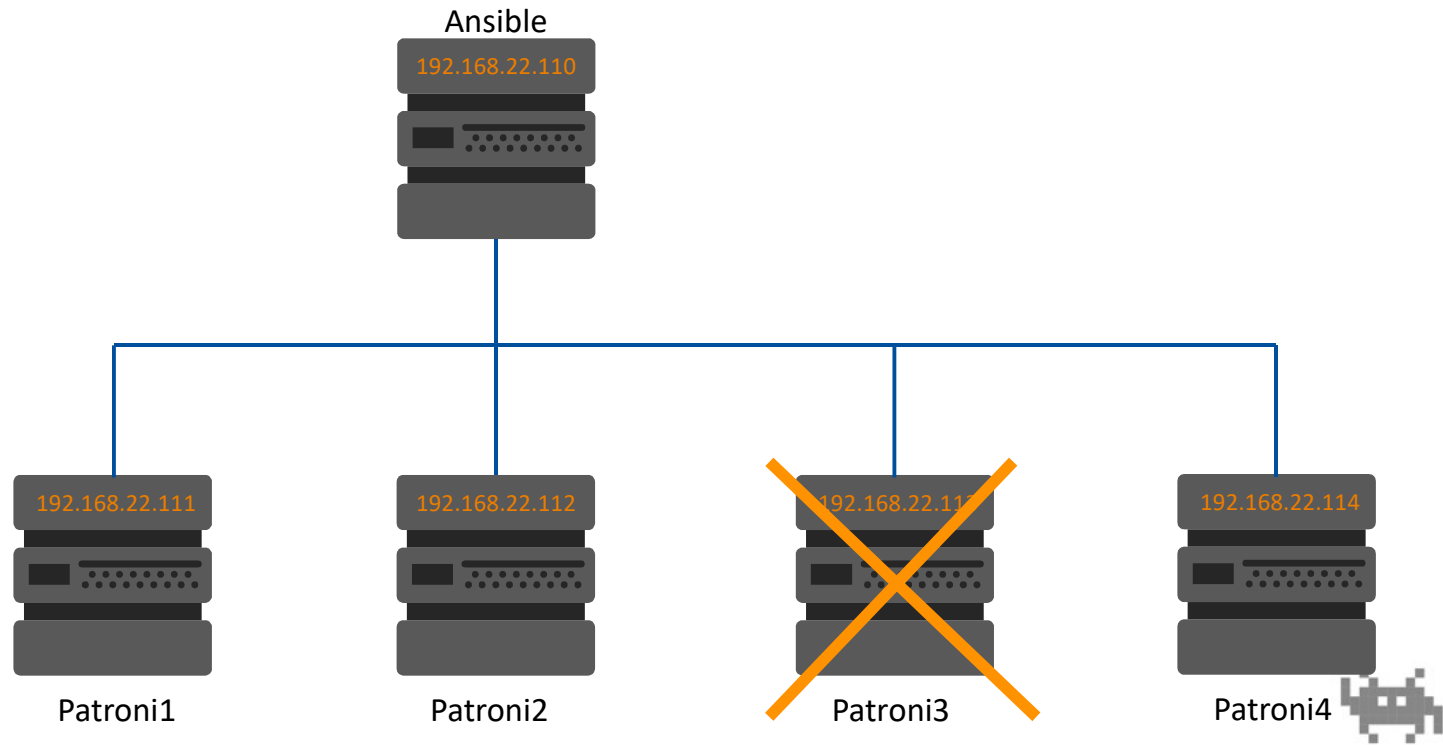
The unbreakable, scalable elephant

18.10.2019

Page 18

# Demo

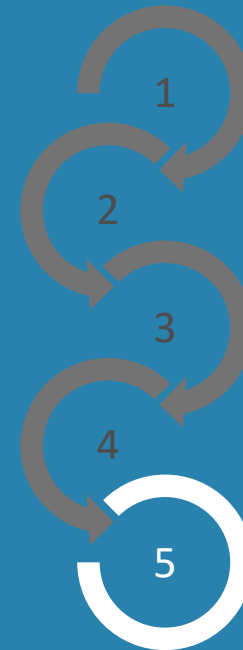
Starting position



## Conclusion

- > Lessons learned
- > Advantages and drawbacks

The unbreakable, scalable elephant



## Conclusion

### Lessons learned



Did you recognize this?

```
TASK [etcd_change : Stop service patroni, if started] *****
ok: [192.168.22.114]
changed: [192.168.22.112]
changed: [192.168.22.113]
changed: [192.168.22.111]

TASK [etcd_change : Stop service etcd, if started] *****
changed: [192.168.22.112]
ok: [192.168.22.114]
changed: [192.168.22.113]
changed: [192.168.22.111]
```



## Conclusion

### Lessons learned



And that?

```
TASK [etcd_change : Start service etcd] *****
changed: [192.168.22.112]
changed: [192.168.22.113]
changed: [192.168.22.111]
changed: [192.168.22.114]

TASK [etcd_change : Start service patroni] *****
changed: [192.168.22.111]
changed: [192.168.22.113]
changed: [192.168.22.112]
changed: [192.168.22.114]
```



Conclusion  
Lessons learned



And now?

# Alexander Kukushkin



## Conclusion

Lessons learned



And now?



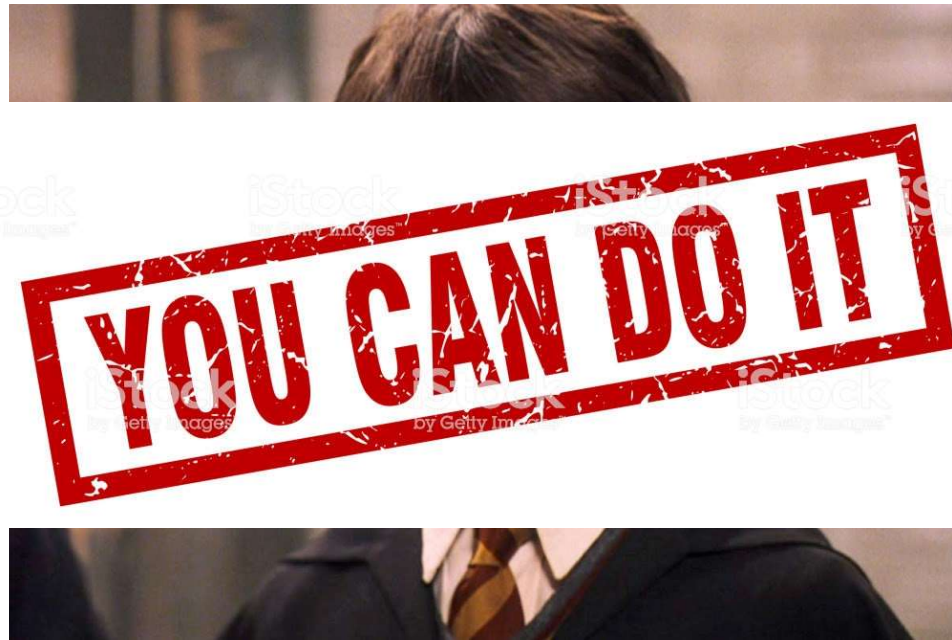


# Conclusion

Lessons learned



And now?



## Conclusion

### Lessons learned



### To improve it:

- > Correct the entry in the etcd.conf:

```
name: patroni4
data-dir: /u02/pgdata/etcd
initial-advertise-peer-urls: http://192.168.22.114:2380
listen-peer-urls: http://192.168.22.114:2380
listen-client-urls: http://192.168.22.114:2379,http://localhost:2379
advertise-client-urls: http://192.168.22.114:2379
initial-cluster:
patroni1=http://192.168.22.111:2380,patroni2=http://192.168.22.112:2380,patr
oni3=http://192.168.22.113:2380,patroni4=http://192.168.22.114:2380
```

- > Add the new host to the etcd cluster:

```
postgres@patroni1:/home/postgres/ [PG1]$ etcdctl member add patroni4
http://192.168.22.114:2380
```



## Conclusion

### Lessons learned



### To improve it:

- > Correct the entry in the etcd.conf:

```
name: patroni4
data-dir: /u02/pgdata/etcd
initial-advertise-peer-urls: http://192.168.22.114:2380
listen-peer-urls: http://192.168.22.114:2380
listen-client-urls: http://192.168.22.114:2379,http://localhost:2379
advertise-client-urls: http://192.168.22.114:2379
initial-cluster-state: 'existing'
initial-cluster:
patroni1=http://192.168.22.111:2380,patroni2=http://192.168.22.112:2380, \
patroni3=http://192.168.22.113:2380,patroni4=http://192.168.22.114:2380
```

- > Add the new host to the etcd cluster:

```
postgres@patroni1:/home/postgres/ [PG1]$ etcdctl member add patroni4
http://192.168.22.114:2380
```



## Conclusion

Lessons learned



### To make it better and High Available:

- > Start etcd service

```
root@patroni4:/home/postgres/ [PG1]$ systemctl start etcd
```

- > Change the patroni.yml on the new host

```
etcd:  
  host: 127.0.0.1:2379
```

- > Start patroni service

```
root@patroni4:/home/postgres/ [PG1]$ systemctl start patroni
```



## Conclusion

Lessons learned



### To make it better and High Available:

- > Start etcd service

```
root@patroni4:/home/postgres/ [PG1]$ systemctl start etcd
```

- > Change the patroni.yml on the new host

```
etcd:  
  hosts: 192.168.22.111:2379,192.168.22.112:2379 \  
         ,192.168.22.113:2379,192.168.22.114:2379
```

- > Start patroni service

```
root@patroni4:/home/postgres/ [PG1]$ systemctl start patroni
```



## Conclusion

### Advantages and drawbacks



- ✓ Scale up and down is easy and fast.
- ✓ Ansible playbooks make an admin's life easier / reduce human errors.
- ✓ Ability to reduce common risks.



- ✗ Pre work tasks take a lot of time.
- ✗ It is not unbreakable, but close to.
- ✗ Time mismatch is critical.



## Conclusion

Advantages and Drawbacks





QUESTIONS?

Please do ask



GETTING GREAT PEOPLE TOGETHER

