

MySQL HA on EC2

Percona Live February 2011

Yves Trudeau

Percona Inc

MySQLPerformanceBlog.com

About Us (and me)

- Former MySQL and Sun consultant
- <http://www.percona.com>
- <http://www.mysqlperformanceblog.com/>
- <http://www.bigdbahead.com>

Yves Trudeau, Ph.D.
Principal Consultant

Plan

- 1) The challenges
- 2) The solution
- 3) Building the solution
- 4) Other solutions

The challenges

1) Cost

- Avoid big standby instances

2) No Static IPs

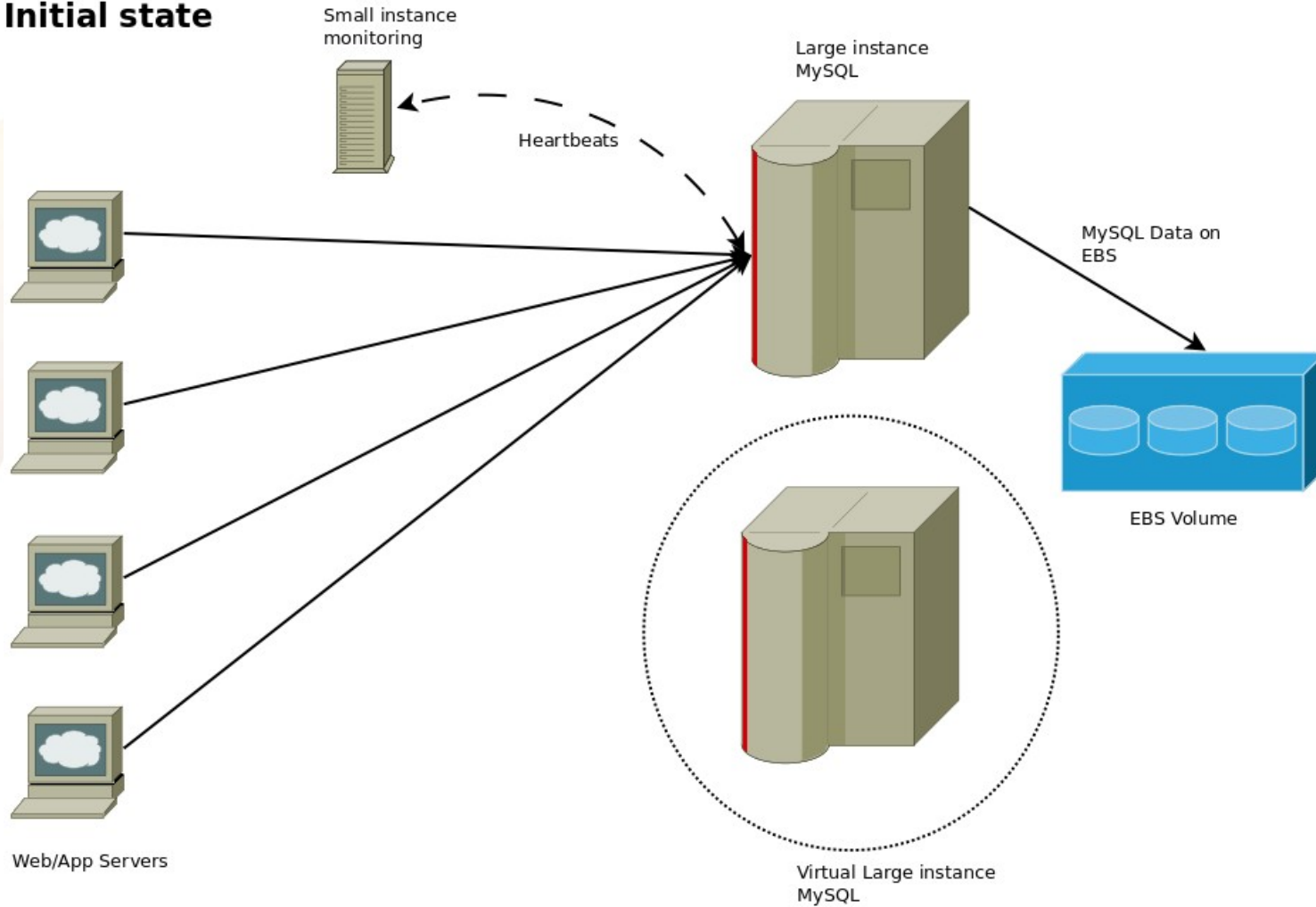
- How to manage?

3) No VIP / No Broadcast

- How to reach services?

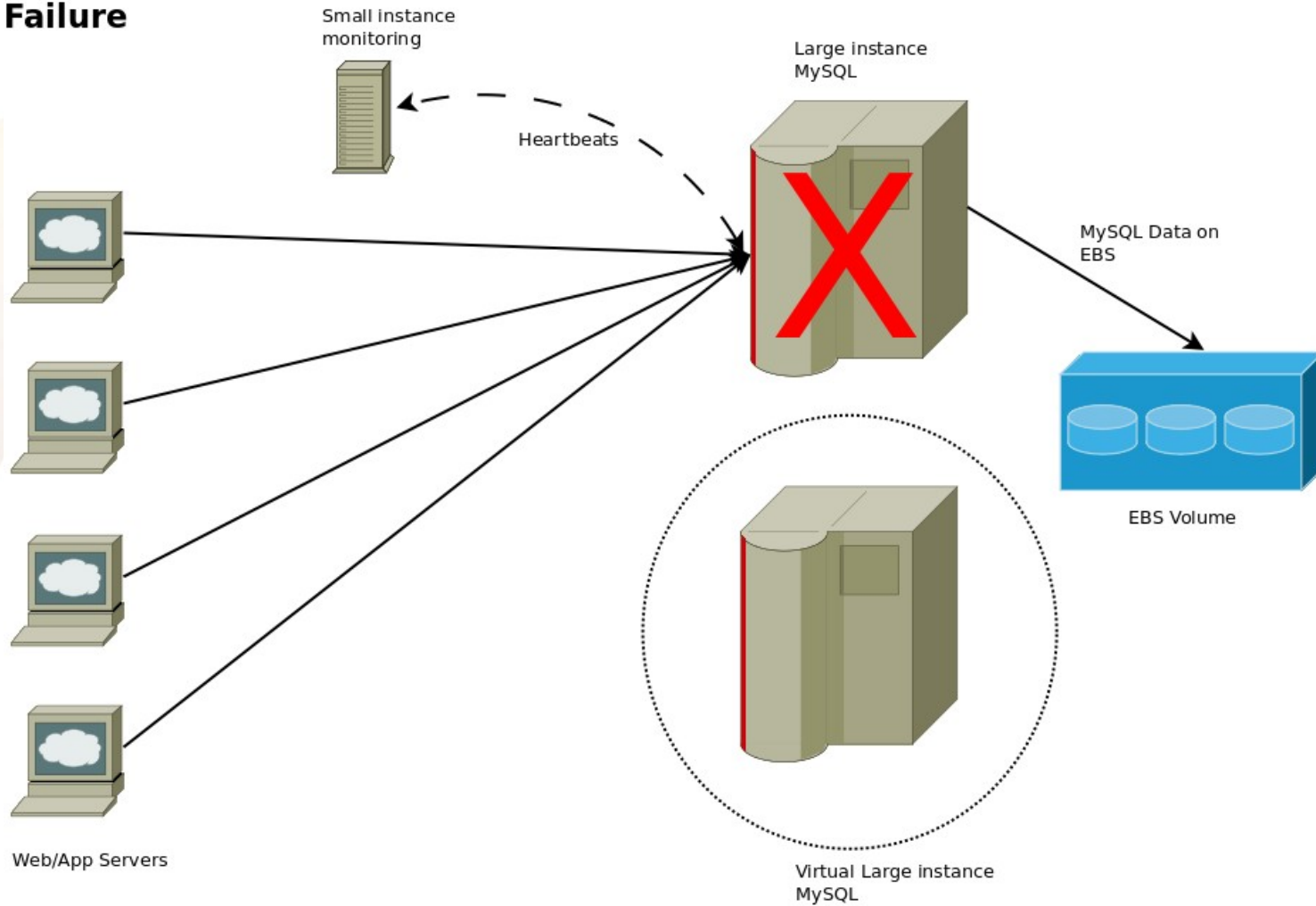
The solution

Initial state



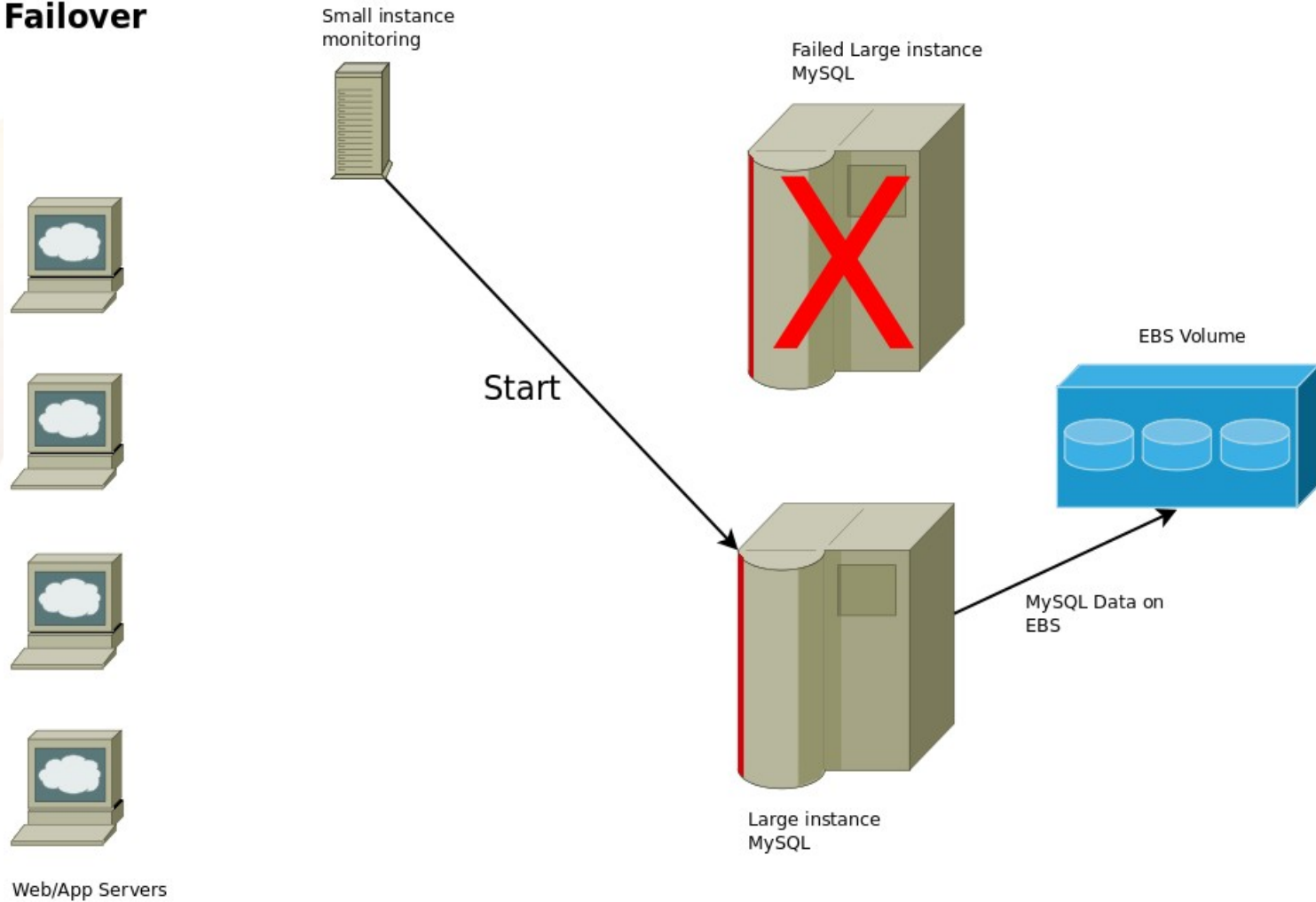
The solution

Failure



The solution

Failover



The solution

Reconfiguring service consumer

Small instance monitoring

Failed Large instance MySQL

Heartbeats

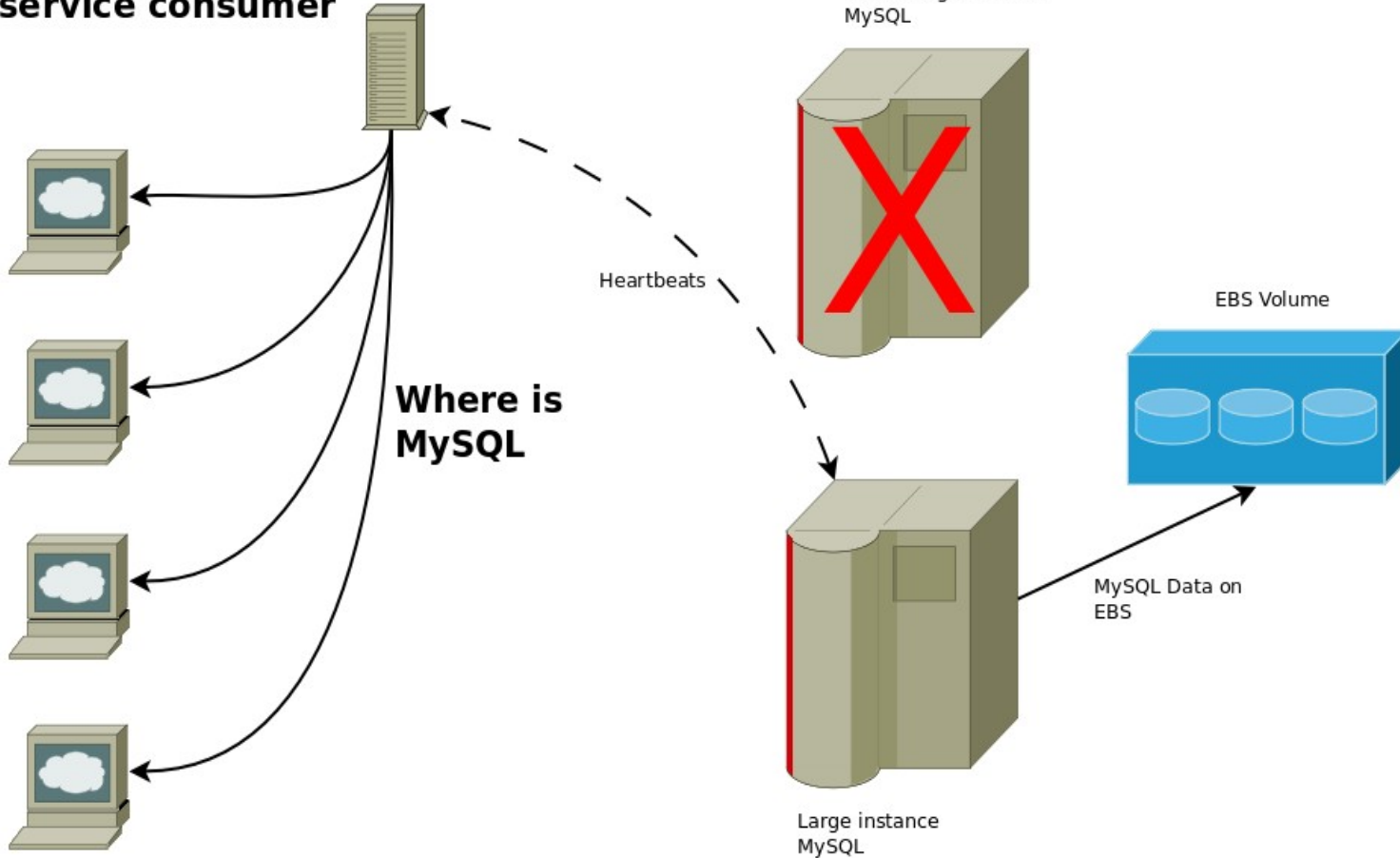
Where is MySQL

EBS Volume

MySQL Data on EBS

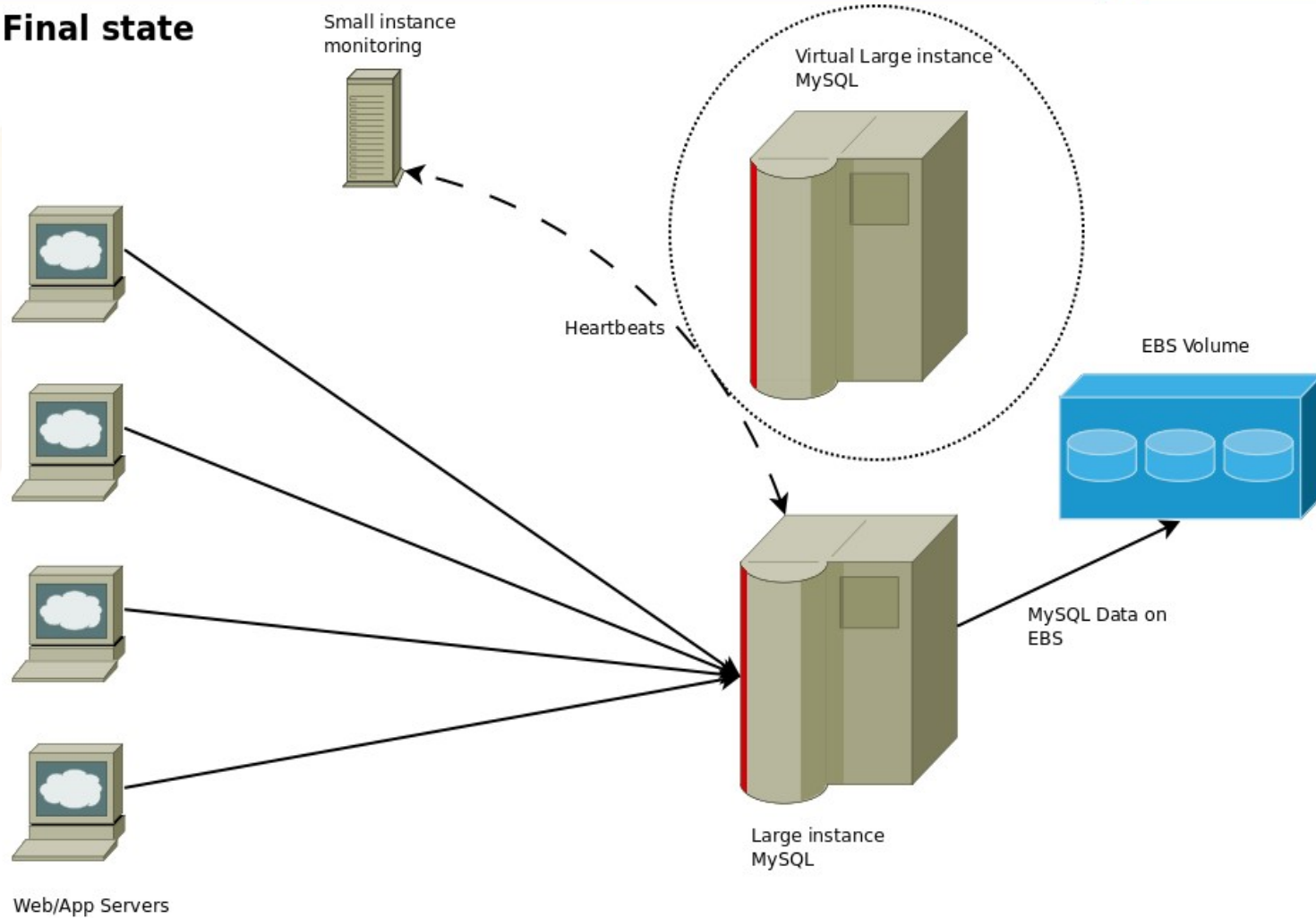
Large instance MySQL

Web/App Servers



The solution

Final state



Technology pieces

- Cluster manager
 - Pacemaker (other could likely work too)
- Unicast communication
 - Lack of broadcast or multicast
 - Setup cluster partner IP for new host
 - Reconfiguring service consumers

Pacemaker

- Allow port 694 udp within security group
- Heartbeat or Corosync for communication, Unicast based
- Scripts type resource on the monitoring host (type *anything*)
- Localization weight favoring the large instance

Pacemaker Heartbeat config

- /etc/ha.cf

```
autojoin any  
ucast eth0 184.73.96.54  
warntime 5  
deadtime 15  
initdead 60  
keepalive 2  
crm respawn
```

Pacemaker resources config

```
root@monitor:~# crm configure
crm(live)configure# primitive mysql ocf:heartbeat:anything \
    params binfile="/usr/local/bin/mysql"
    pidfile="/var/run/heartbeat/mysql.pid" \
    op start interval="0" timeout="300s"
crm(live)configure# location loc-1 mysql 1000: hamysql
crm(live)configure# location loc-2 mysql 1: monitor
crm(live)configure# commit
```

EC2 stuff

- Custom EC2 Images
 - Pre-configured
 - Pacemaker/Heartbeat
 - MySQL
- AWS tools
 - Rich API, needed to manage instances
 - Parsing difficult → helper scripts

Failover steps

- 1) Detect failure
- 2) Kill the faulty instances (stonith like)
- 3) Free the shared resource
- 4) Launch the new instance
- 5) Reassign shared resource
- 6) Reconfigure heartbeat
- 7) Spread connection info

ec2-run-instance

- User data script
 - Dynamically generated
 - Run at instance startup
 - Seed the new instance with cluster partner IP

```
#!/bin/bash
sudo hostname hamysql
sudo perl -pi -e "s/ucast eth0 (\d+)(\.\d+){3}/ucast eth0 10.220.230.18/g" /etc/ha.d/ha.cf
```

```
# to eventually be added to the ebs image
sudo perl -pi -e 's/bind-address/#bind-address/g' /etc/mysql/my.cnf
sudo service mysql restart
sleep 5
sudo /etc/init.d/heartbeat start
```

Heartbeat reconfigure

- Once the IP of the new instance is known
 - Replace the IP in ha.cf
 - Tell Heartbeat to reload its config.

Spreading connection info

- Configuring consumers
 - All in same security group
 - ssh based remote execution

```
TMPFILE=`mktemp`  
ec2-describe-instances -K $PK -C $CERT | /usr/local/bin/filtre_instances.pl | grep  
$CLIENT_SECURITY_GROUP > $TMPFILE  
  
while read line  
do  
    IP=`echo $line | cut -d'|' -f2`  
    ssh -i /usr/local/bin/update_mysql ubuntu@$IP sudo ./updated_xinetd.sh  
$MYSQL_IP  
done < $TMPFILE  
  
rm $TMPFILE
```

Pitfalls

- Do not move resources with Pacemaker...
- A type of instance can be unavailable at a given site!
- Timing... affect failover time

Want more!

<http://www.mysqlperformanceblog.com/2010/06/17/high-availability-for-mysql-on-amazon-ec2-part-1-intro/>

Other solutions

- RDS, pros/cons, only MySQL
- Continuent
- RightScale (EIP based)
- Scalarium
- Scalebase (beta)
- Other?

Questions

Questions?

Yves Trudeau: yves@percona.com

<http://www.percona.com/>

<http://www.mysqlperformanceblog.com/>