Bootstrapping databases in a single command: elastic provisioning for the win!

About Yelp

- *"Yelp is a website and mobile app that connects people with great local businesses"*
- Founded 2004
- More than just reviews!

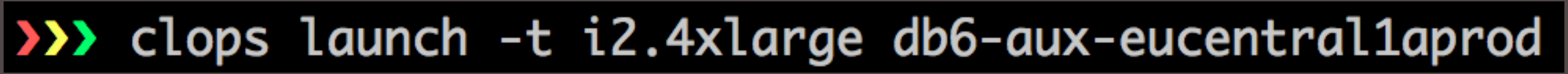
Infrastructure as Code <3

- our cloud deployment is getting larger and larger
- databases deployment can’t be a bottleneck
- we need to automate

Our Goal

- single command provisioning of a MySQL node
  - configured
  - replicating
  - monitored
  - in the load balancer

Our Goal



```
❯❯❯ clops launch -t i2.4xlarge db6-aux-eucentral1aprod
```

Our Toolbox

- Puppet
- Sensu
- Python
- xtrabackup
- Smartstack (Nerve + Synapse)

Puppet

- responsible for installing and configuring
  - MySQL
  - crons
  - kernels
  - mountpoints
  - monitoring
  - and more…
- spawning scripts

Nonrepeatable Infrastructure with Puppet

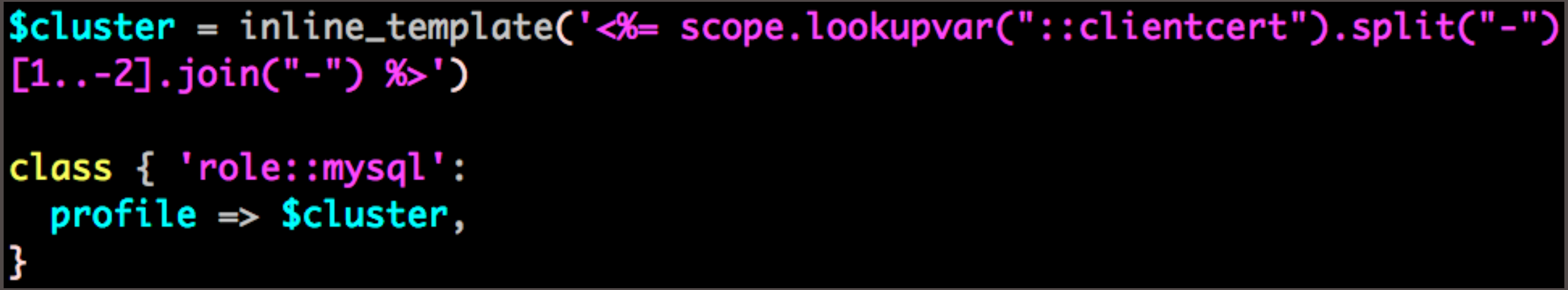
We had:

- Puppet 0.25 (2009)
- a node definition for each database
- no intuition from outside the node
- pets, not cattle

~~Non~~repeatable Infrastructure with Puppet

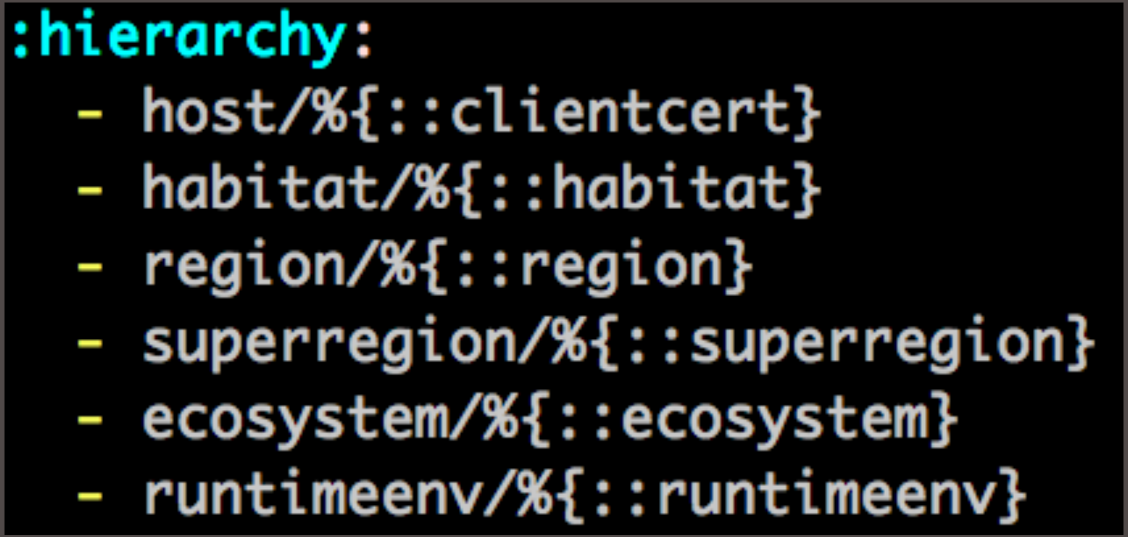
- facter all the things!
- pull information directly from hostnames:

```
db3-main-uswest1aprod.prod.yelpcorp.com => main
```



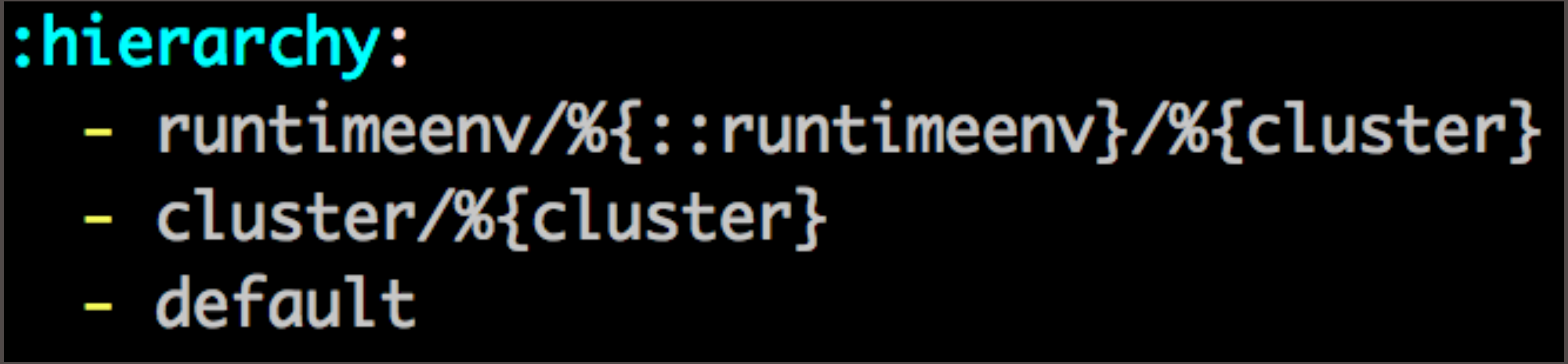
```
$cluster = inline_template('<%= scope.lookupvar("::clientcert").split("-")
[1..-2].join("-") %>')

class { 'role::mysql':
  profile => $cluster,
}
```

Hiera



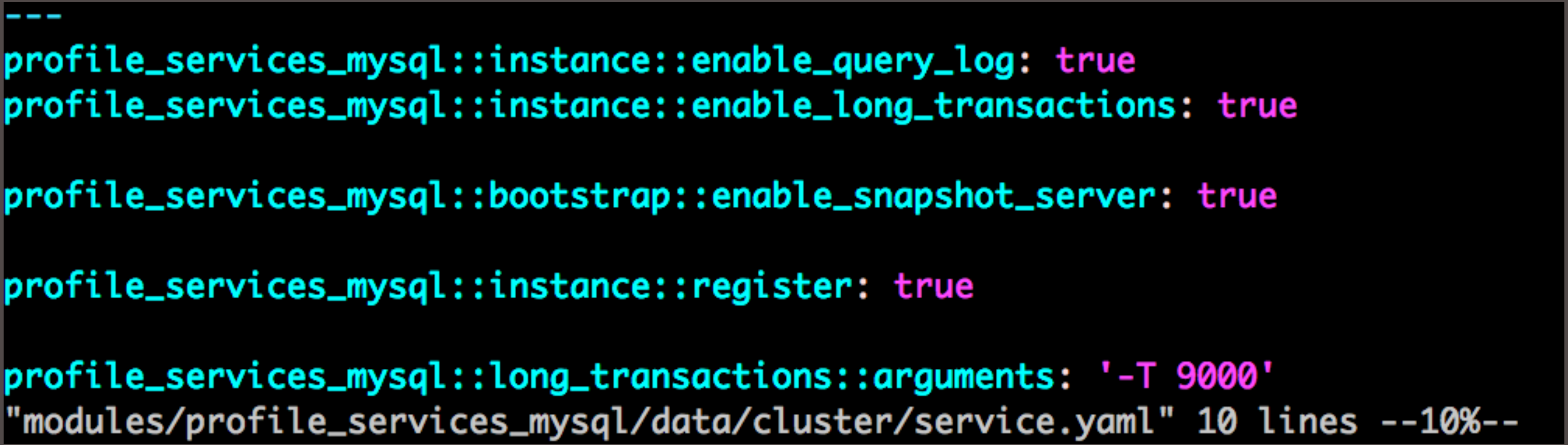
```
:hierarchy:
  - host/%{::clientcert}
  - habitat/%{::habitat}
  - region/%{::region}
  - superregion/%{::superregion}
  - ecosystem/%{::ecosystem}
  - runtimeenv/%{::runtimeenv}
```

Per-module Hiera



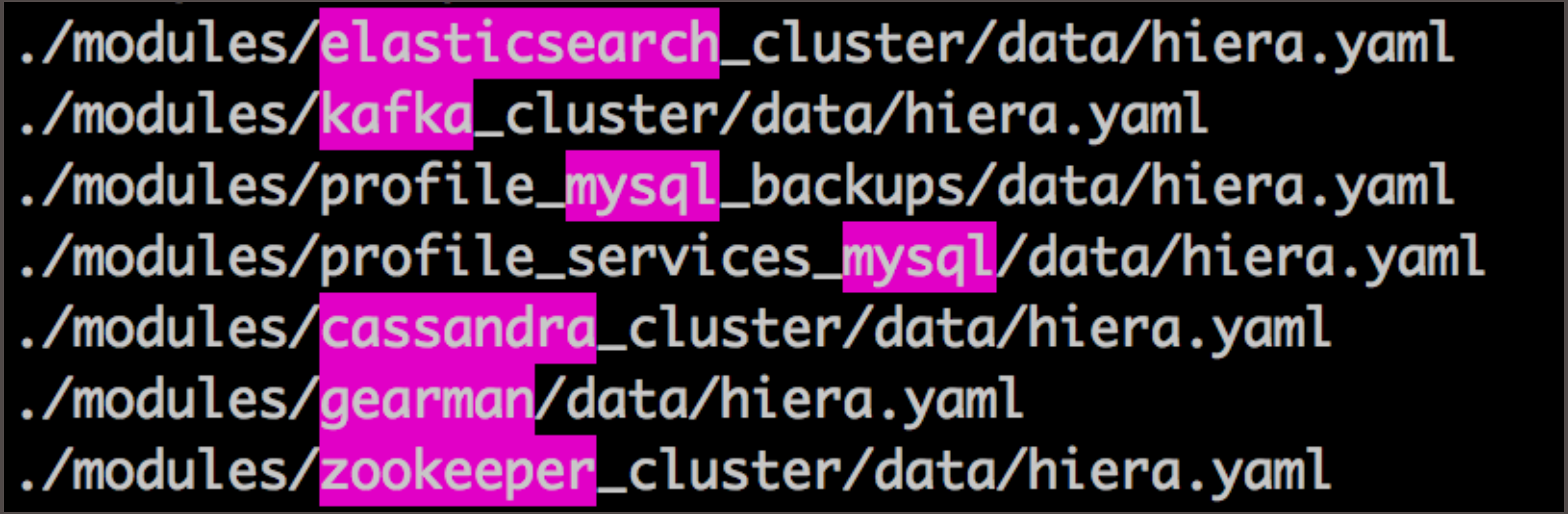
```
:hierarchy:
  - runtimeenv/%{::runtimeenv}/%{cluster}
  - cluster/%{cluster}
  - default
```

Per-module Hiera



```
---
profile_services_mysql::instance::enable_query_log: true
profile_services_mysql::instance::enable_long_transactions: true

profile_services_mysql::bootstrap::enable_snapshot_server: true

profile_services_mysql::instance::register: true

profile_services_mysql::long_transactions::arguments: '-T 9000'
"modules/profile_services_mysql/data/cluster/service.yaml" 10 lines --10%--
```

Per-module Hiera



```
./modules/elasticsearch_cluster/data/hiera.yaml
./modules/kafka_cluster/data/hiera.yaml
./modules/profile_mysql_backups/data/hiera.yaml
./modules/profile_services_mysql/data/hiera.yaml
./modules/cassandra_cluster/data/hiera.yaml
./modules/gearman/data/hiera.yaml
./modules/zookeeper_cluster/data/hiera.yaml
```

Packages

- stock Percona packages try to *do* things to my system
- that’s my job!
    - or Puppet’s
- solution: strip out initscripts + configs

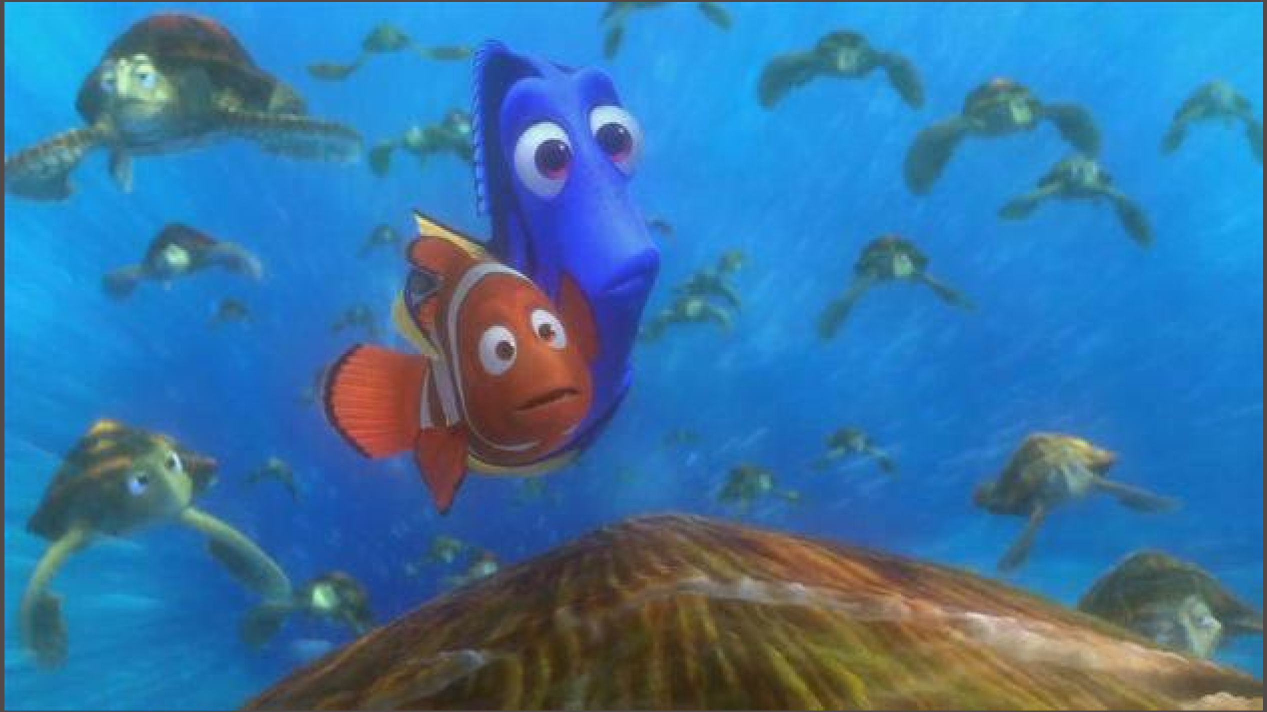
...with this groundwork, we

- install MySQL
- configure my.cnf
- set up Smartstack (packages, configs)
- set up cronjobs
- configure all OS services (e.g. ntpd, syslog-ng)
- configure Sensu

...but we still lack

- a dataset
- running MySQL
- load balancing

We need a buddy!



Snapshots, as a service

- just like any other SOA service!
- procedure:
  - find a buddy
  - download a snapshot from that buddy
- process should be identical to producing a backup for permanent storage

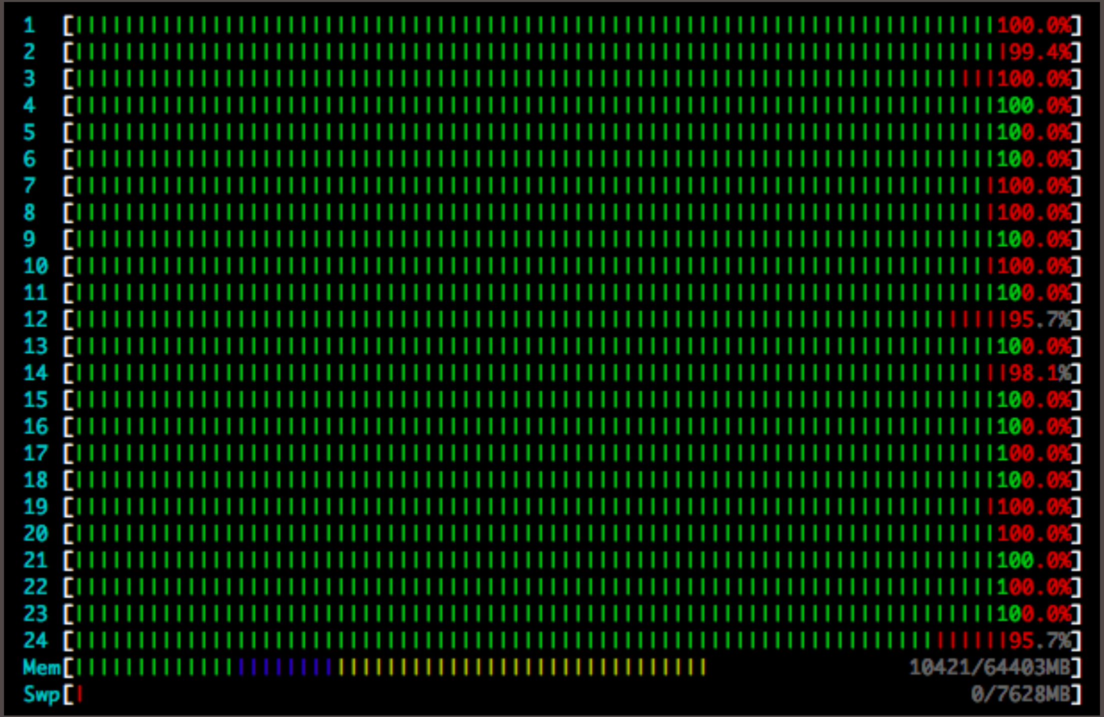
Design Considerations

- performance
- data integrity

High Performance MySQL (Snapshots)

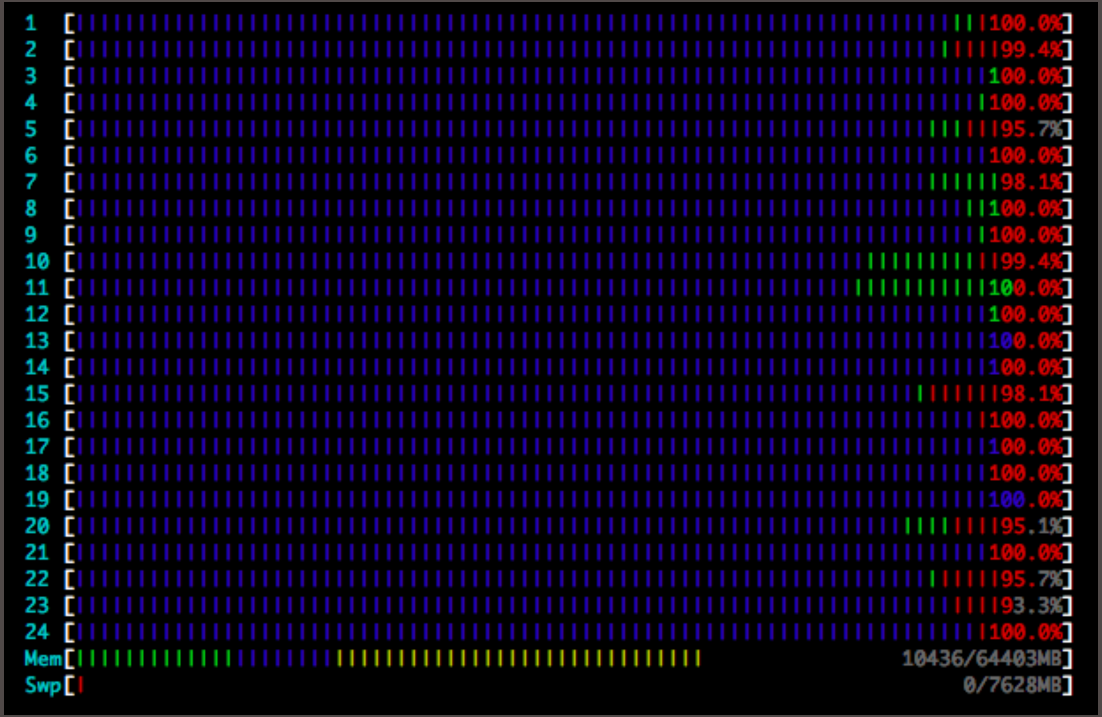
- we are running on a live MySQL node
- starving the application is a possibility
- we need to think about:
  - CPU
  - network
  - disk I/O

compression nom nom nom



```
1  [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
2  [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||99.4%]
3  [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
4  [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
5  [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
6  [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
7  [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
8  [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
9  [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
10 [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
11 [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
12 [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||95.7%]
13 [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
14 [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||98.1%]
15 [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
16 [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
17 [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
18 [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
19 [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
20 [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
21 [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
22 [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
23 [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
24 [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||95.7%]
Mem[|||||||||||||||||||||||||||||||||||||||||||||||||||                10421/64403MB]
Swp[|                                                                       0/7628MB]
```

being nicer helps



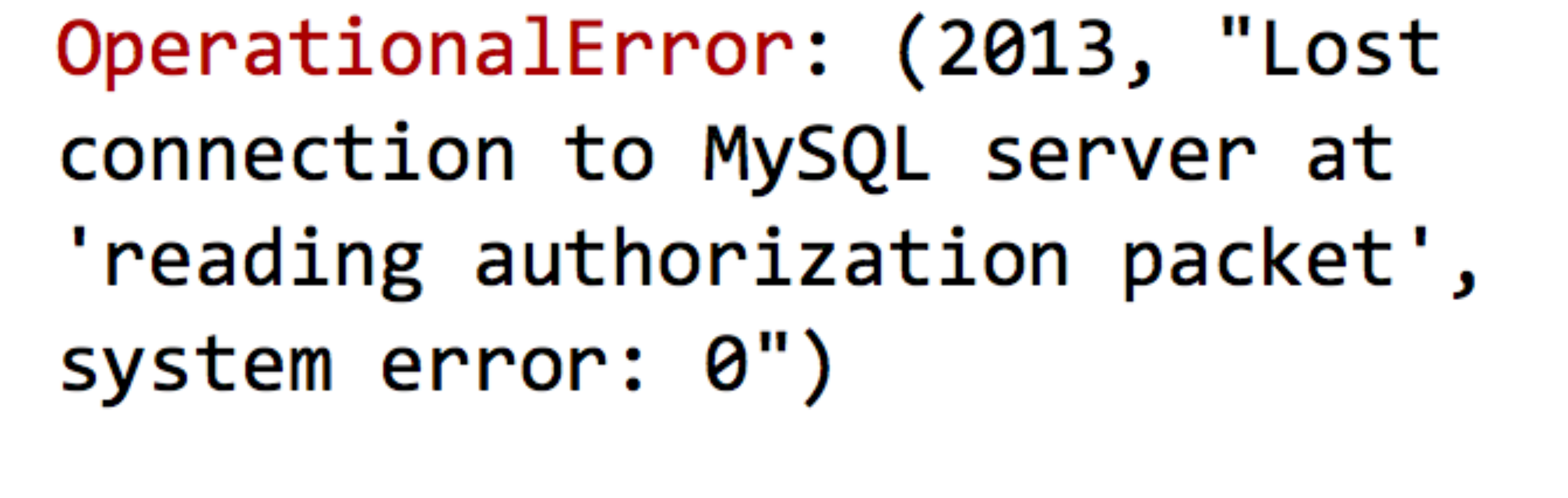
```
1  [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
2  [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||99.4%]
3  [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
4  [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
5  [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||95.7%]
6  [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
7  [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||98.1%]
8  [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
9  [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
10 [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||99.4%]
11 [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
12 [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
13 [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
14 [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
15 [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||98.1%]
16 [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
17 [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
18 [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
19 [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
20 [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||95.1%]
21 [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
22 [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||95.7%]
23 [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||93.3%]
24 [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
Mem[||||||||||||||||||||||||||||||||||||||||||||||||||            10436/64403MB]
Swp[|                                                                  0/7628MB]
```

Using less CPU helps even more

- compression
  - use quicklz
- encryption

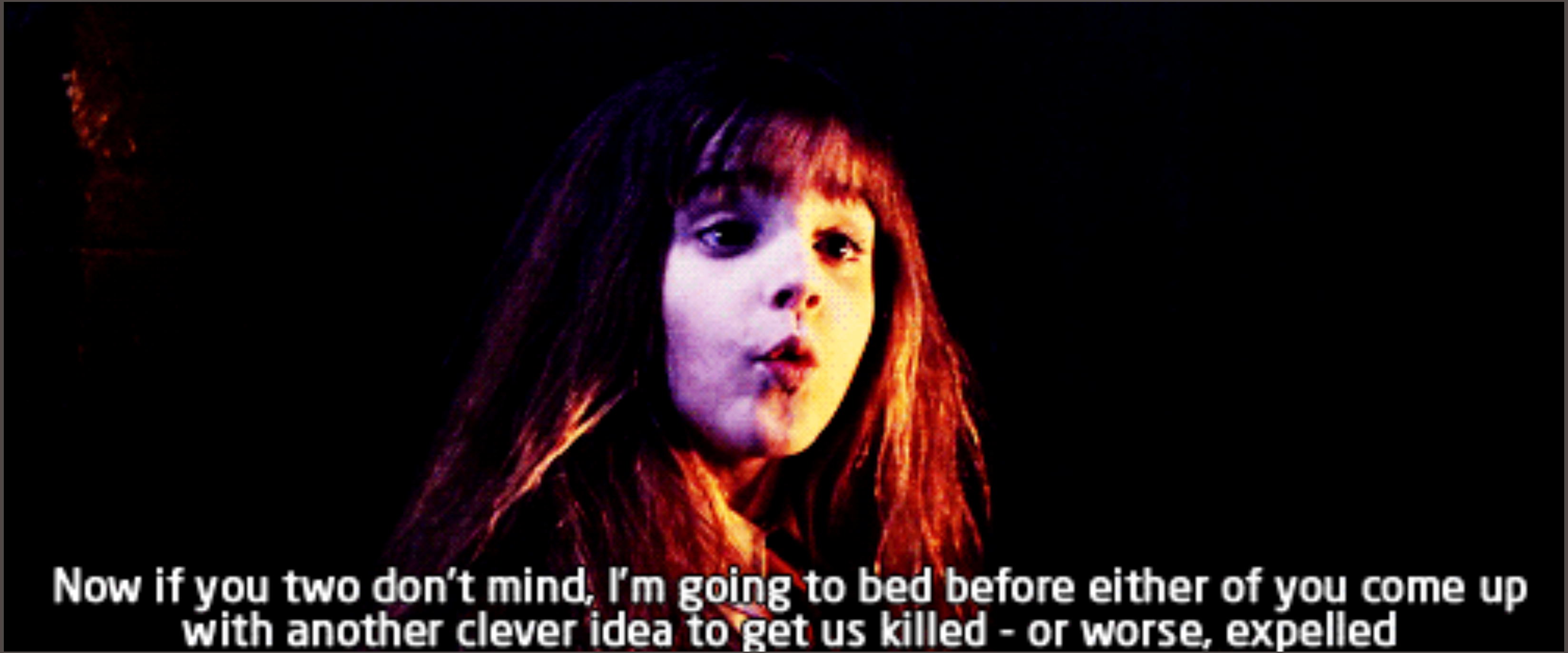
  hardware acceleration :) :)
- xtrabackup

Who likes?



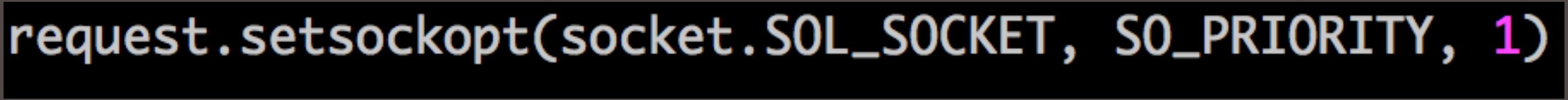
```
OperationalError: (2013, "Lost
connection to MySQL server at
'reading authorization packet',
system error: 0")
```

Let’s find our priorities



Now if you two don’t mind, I’m going to bed before either of you come up with another clever idea to get us killed - or worse, expelled

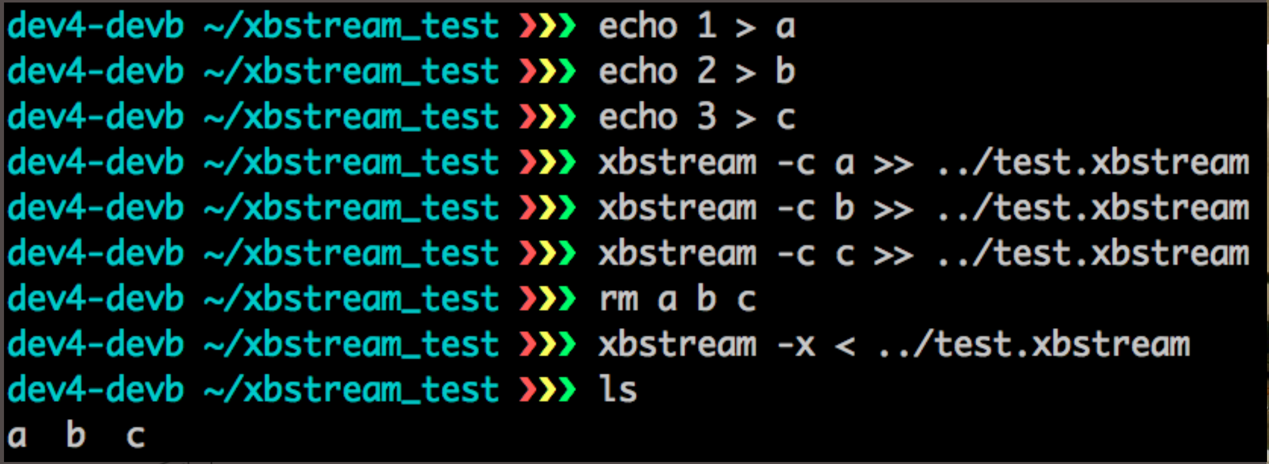
Let’s find our priorities



```
request.setsockopt(socket.SOL_SOCKET, SO_PRIORITY, 1)
```

Extra data?

- raw xtrabackup isn’t enough
- xbstream is concatenable!



```
dev4-devb ~/xbstream_test ❯❯❯ echo 1 > a
dev4-devb ~/xbstream_test ❯❯❯ echo 2 > b
dev4-devb ~/xbstream_test ❯❯❯ echo 3 > c
dev4-devb ~/xbstream_test ❯❯❯ xbstream -c a >> ../test.xbstream
dev4-devb ~/xbstream_test ❯❯❯ xbstream -c b >> ../test.xbstream
dev4-devb ~/xbstream_test ❯❯❯ xbstream -c c >> ../test.xbstream
dev4-devb ~/xbstream_test ❯❯❯ rm a b c
dev4-devb ~/xbstream_test ❯❯❯ xbstream -x < ../test.xbstream
dev4-devb ~/xbstream_test ❯❯❯ ls
a  b  c
```

Client side work

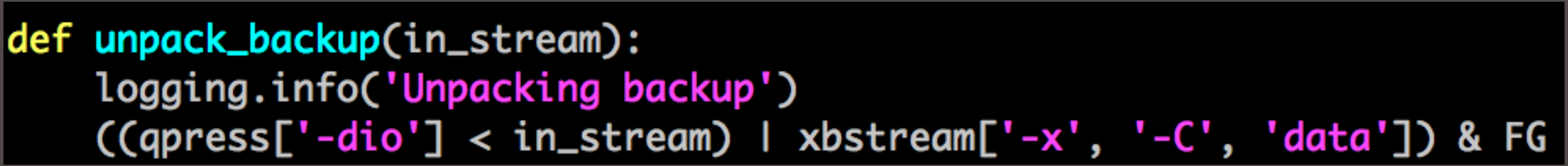
- very typical xtrabackup workflow:
  - unpack
  - prepare
  - configure replication

Staying safe

- How do we ensure that we never mess up a working MySQL node?

Staying safe

- choose your language carefully
  - something with real error handling (bash is out)
  - Ruby has open4 (but I haven’t used it)
  - Python’s plumbum library is nice



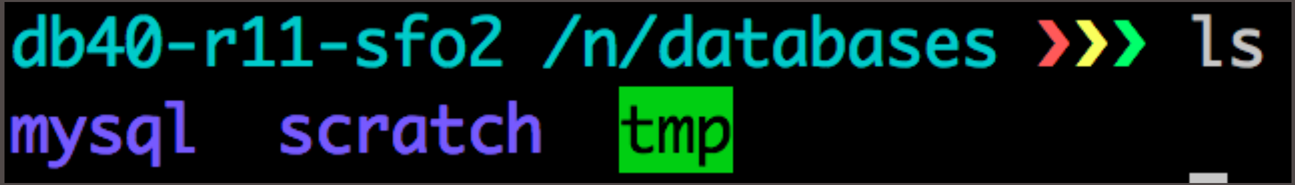
```
def unpack_backup(in_stream):
    logging.info('Unpacking backup')
    ((qpress['-dio'] < in_stream) | xbstream['-x', '-C', 'data']) & FG
```

Staying safe

- CI workflow
  - enables testing before deployment
  - allows isolation of dependencies

Staying safe

- establish a “no-go” zone around your data

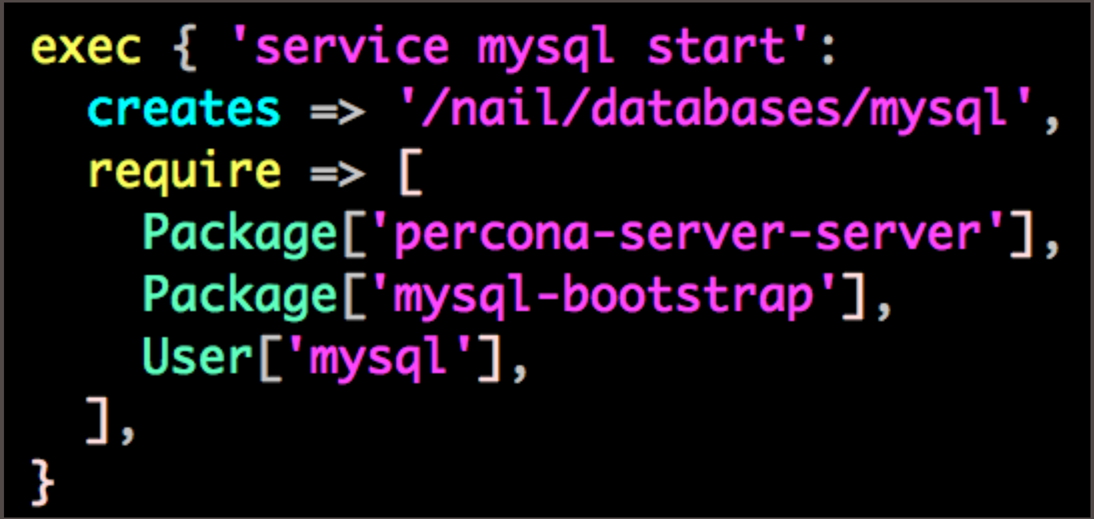


```
db40-r11-sfo2 /n/databases ››› ls
mysql  scratch  tmp
```

- bootstrap script
  - respects the no-go zone
  - works in a temporary directory

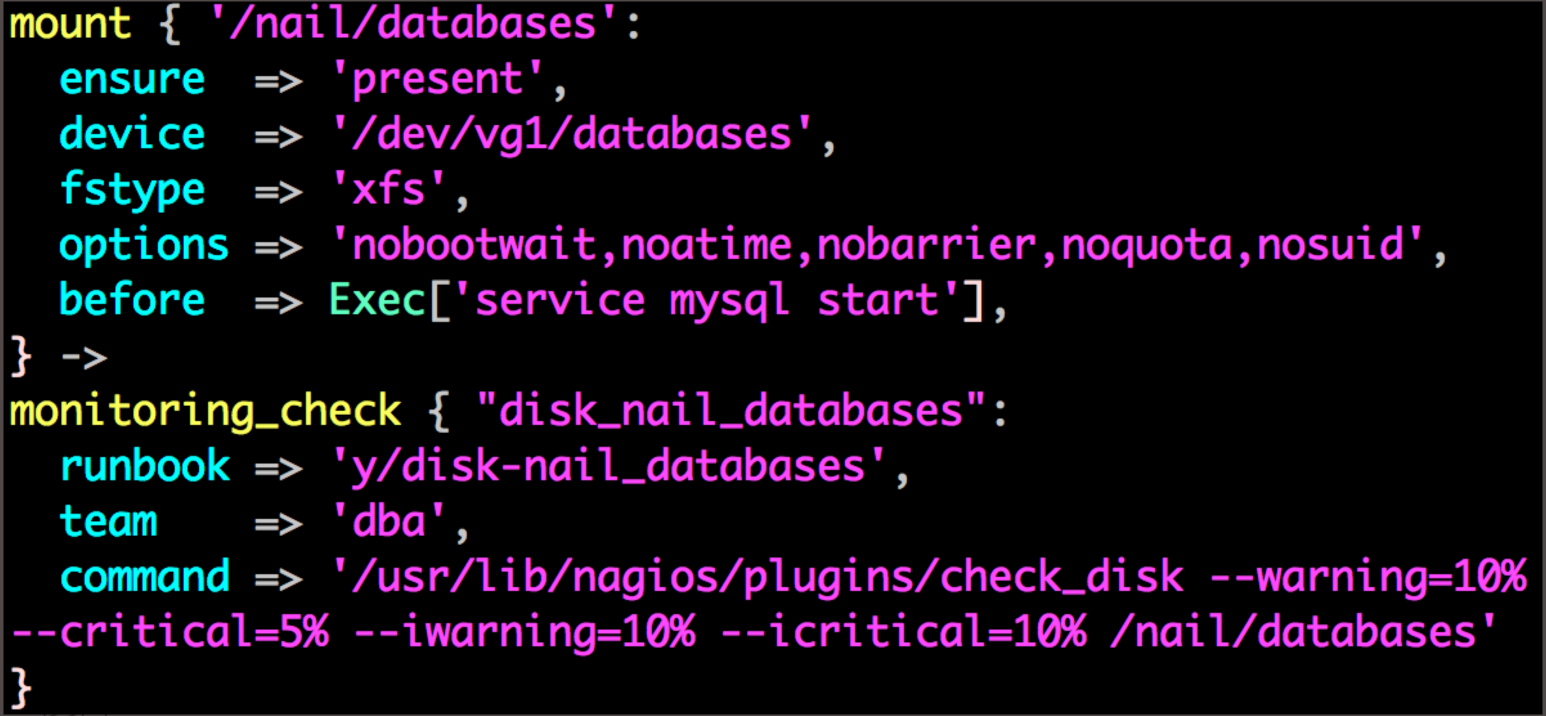
Tying it together

We execute our script from Puppet:



```
exec { 'service mysql start':
  creates => '/nail/databases/mysql',
  require => [
    Package['percona-server-server'],
    Package['mysql-bootstrap'],
    User['mysql'],
  ],
}
```

Stop! Sensu time!



```
mount { '/nail/databases':
  ensure  => 'present',
  device  => '/dev/vg1/databases',
  fstype  => 'xfs',
  options => 'nobootwait,noatime,nobarrier,noquota,nosuid',
  before  => Exec['service mysql start'],
} ->
monitoring_check { "disk_nail_databases":
  runbook => 'y/disk-nail_databases',
  team    => 'dba',
  command => '/usr/lib/nagios/plugins/check_disk --warning=10%
--critical=5% --iwarning=10% --icritical=10% /nail/databases'
}
```

Load balancer automation

- healthy nodes register themselves in Nerve
- on the other end, Synapse
  - configures HAProxy
  - drops a JSON file on disk

A bit of philosophy

- configuration management is awesome!
  - but be careful with it
- sometimes there is a better tool for the job
  - integrate that tool with your CM
- for more info, watch Charity Majors’s talk from ChefConf!

A bit of philosophy

- decentralized coordination works!
- eventual consistency makes your life better

A bit of philosophy

- we succeeded!
  - deploying new databases is
    - faster
    - easier
    - more reliable than ever
- most importantly: non-DBAs can spin up databases

Questions?

Follow me @hashbrowncipher

Charity’s talk at ChefConf 2015:

There and Back Again: How We Drank the Chef Kool-Aid, Sobered Up, and Learned to Cook Responsibly

http://goo.gl/94UQS8

Craig Dunn on Puppet roles and profiles:

http://www.craigdunn.org/2012/05/239/

yelpengineers

@yelpengineering

engineeringblog.yelp.com

github.com/yelp

Come see us at booth 402!