MANAGING MULTI-DC MYSQL
INSTALLATION WITH CHEF

Tatyana Arenburg
14/04/2015 tata@wix.com

What's on the menu?

- System architecture overview
  - DB per service approach
  - Multi-instance MySQL
- MySQLer cookbook
  - Install & config MySQL instances
  - Create new replica
  - Automate credentials

How MySQLer will simplify your life?

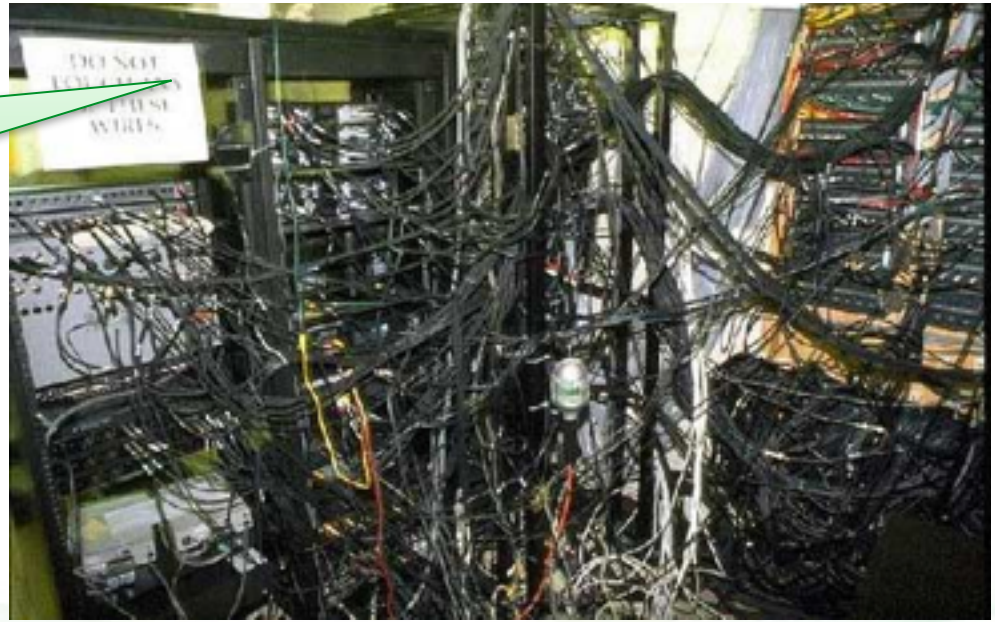
- Manage MySQL instances (multi or single)
- Create replica of existing server with just minutes of work
- 3 techniques to automate credentials.
- Handle MySQL upgrade and backward compatibility (distributions).

Some WIX numbers

- 7 Data Centres (both physical and cloud)
- ~ 500 Servers + 100s on demand cloud instances
- ~100 different services
- 150 MySQL instances of 30 databases
- ~7TB of production master data
- ~63M users over the globe

System Architecture

- Multi system architecture usually looks like:

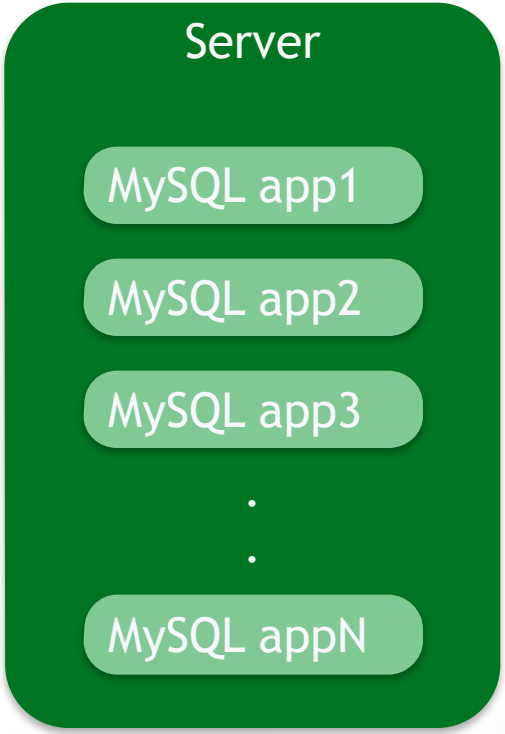


DB per service approach

- Each data set has it own requirements of:
  - Availability (how many 9's?, which DCs)
  - Durability (can data be lost in failure?)
  - Scalability (reads and writes throughput)
- Different working sets (memory)
- Simple data size planning

Multi Instance MySQL

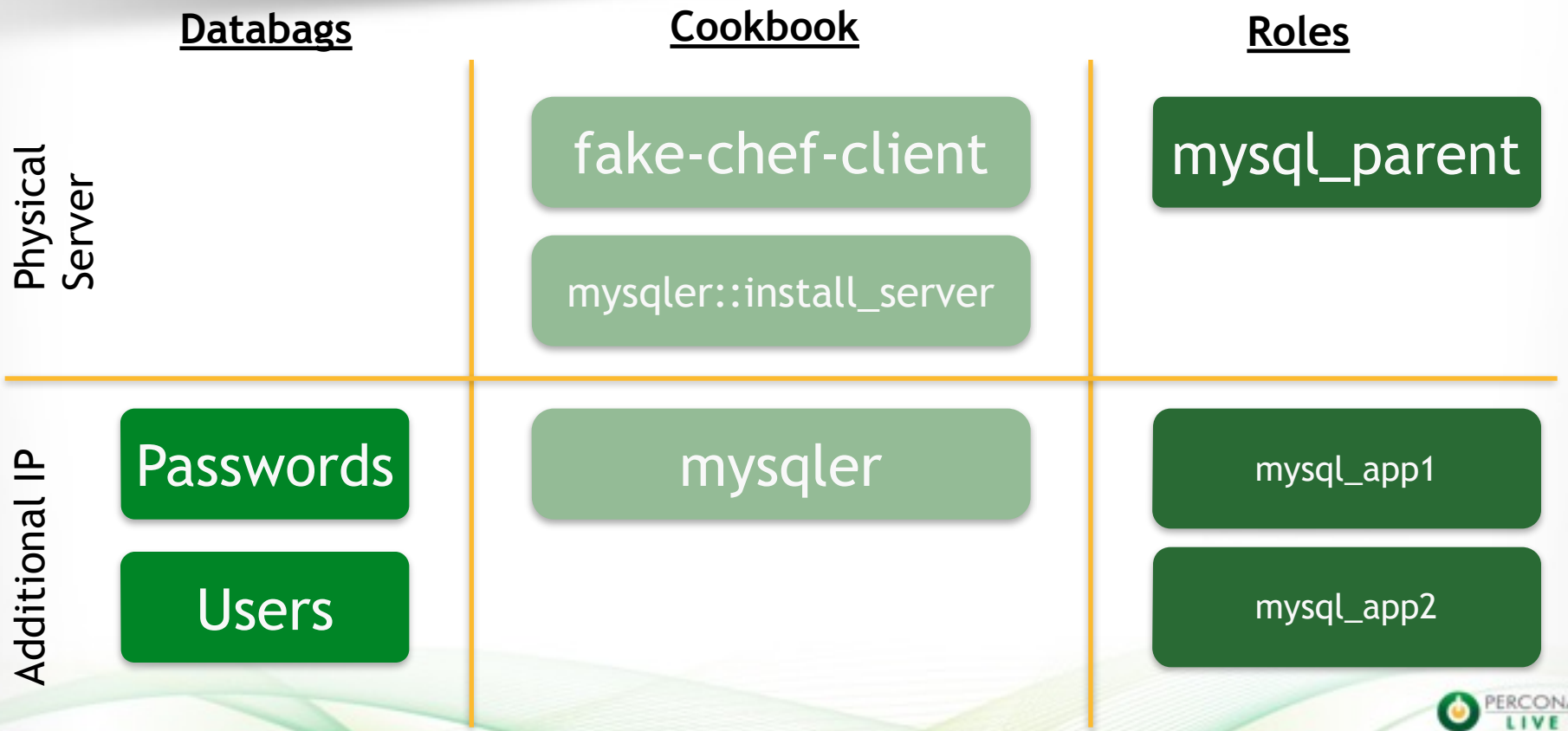
- Pros:
  - Increase flexibility
  - Decrease costs
- Cons:
  - Complex management
- How?
  - fake-chef-client cookbook



Fake-chef-client

- What?
  - Run chef client for each IP/instance
- How?
  - Add additional config files
  - Run chef client in the background

Our Chef Toolkit

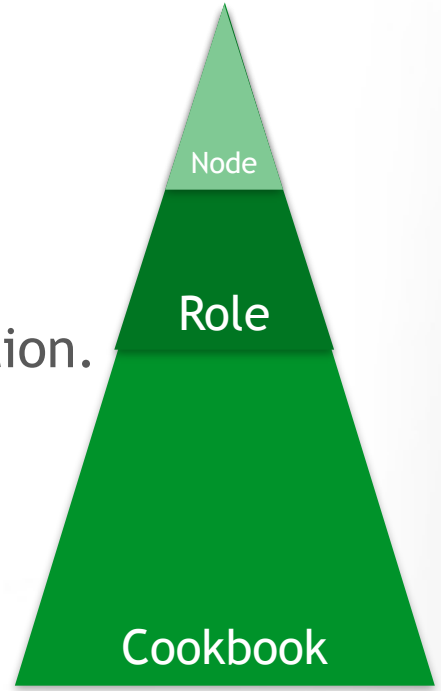


Steps to create new functioning replica

- Step I
  - Install binaries
  - Configure instance
- Step II
  - Get data from another replica/master
  - Add grants
- Step III
  - Add monitoring

Installation and configuration of MySQL instances

- Wrapper Cookbook Attributes
  - Tunings, relevant to all servers
- Role
  - App_name if multi-instance is used
  - Default innodb_buffer_pool_size for your application.
  - Auto_increment_increment
- Node
  - innodb_buffer_pool_size -if differs from role
  - Auto_increment_offset



* Default only

Building new replica - Rebuild strategies

- Dump
  - Schema changes before data transfer.
  - Start io_slave together with dump start.
  - No need in additional software.
- Xtrabackup
  - Replica is synced immediately after rebuild
  - Requires additional software
- Data snapshots

Using snapshots to create new replica

- Potential problems
  - Forgotten snapshot can leave you with no available disk space on source DB
  - Not properly flushed data can waste your time, and you will need to start all over again.

Automated credentials

- Local and monitoring users
  - Predefined passwords
  - Created from encrypted databag
  - or created from template

```
{
  "id": "mysql",
  "users": {
    "root": {
      "password": "root_password",
      "grants": {
        "db": "*",
        "actions": "SELECT"
      },
      "sources": [ "localhost" ]
    }
  }
}
```

Automated credentials

- Application credentials
  - Generated passwords
  - Created from "users" data-bag

*Also used to find the servers by application

```
{
 "id": "app_name",
 "users": {
  "app1": {
    "db_name": "db1",
    "table_names": ["*"],
    "privileges": {
      "ro": "SELECT",
      "rw": "SELECT, INSERT",
    },
   "action": "create"
  }
 }
}
```

Summary

- Defining system architecture
- Create wrapper cookbook
- Add users and applications to data-bags
- Choose rebuild strategy

Q&A