

Utilizing Ansible to Manage a Highly Available MySQL Environment

MYSQL – ANSIBLE - MHA

Presented by ALKIN TEZUYSAL , MIKLOS SZEL

April 14 2015

Pythian[®]
love your data

About : Alkin Tezuysal

- ◆ Currently Team Manager, Pythian
- ◆ 20 years experience in RDBMS world
- ◆ Acted as Sr. DBA to Supervisor roles
- ◆ Previously worked
 - ◆ ebay Intl, Supervisor
 - ◆ Bank of America, Sr. DBA
 - ◆ AT&T, Sr. DBA
- ◆ How to Contact:
 - ◆ Email: tezuysal@pythian.com
 - ◆ Twitter: @ask_dba
 - ◆ Linkedin: tr.linkedin.com/in/askdba/en

About : Miklos 'Mukka' Szel

- ◆ Currently MySQL Consultant, Pythian
- ◆ 10+ years experience System and Database world
- ◆ Acted as Sr. DBA, Sr. Ops, Developer
- ◆ Previously worked
 - ◆ Own startup
 - ◆ ISP
 - ◆ Walt Disney
- ◆ How to Contact:
 - ◆ Email: szel@pythian.com
 - ◆ Twitter: [@42mukka](https://twitter.com/@42mukka)
 - ◆ LinkedIn: hu.linkedin.com/in/miklosszel/en

Ansible in a Nutshell

- ✓ Simple
- ✓ Agentless
- ✓ Easy to manage
- ✓ Idempotent
- ✓ 100% Python
- ✓ Configuration management
- ✓ Uses ssh

For more: http://docs.ansible.com/intro_installation.html

MHA in a Nutshell

- ✓ Automated Master failover
- ✓ Interactive Master failover
- ✓ Non-interactive Master failover
- ✓ Online Master switching
- ✓ Easy to configure
- ✓ Open Source HA utility

For more: <https://code.google.com/p/mysql-master-ha/>

MySQL in a Nutshell

- ✓ De-facto standard for web, e-commerce and startup companies
- ✓ Easy to manage and use
- ✓ Why we all are here

For more:

<http://www.oracle.com/us/products/mysql/overview/index.html>

Back to Ansible

- 1) Installation
- 2) Configuration
- 3) Best practices
- 4) Demo on how to use with MySQL-MHA

Installation - I

- Verify your version of Python, install pip:

```
$ sudo easy_install pip
```

- Add Python modules:

```
$ sudo pip install paramiko PyYAML Jinja2 httplib2
```

- Install Ansible via pip :

```
$ sudo pip install ansible
```

- Install Ansible via yum:

```
# install the epel-release RPM if needed on CentOS, RHEL, or Scientific Linux
```

```
$ sudo yum install ansible
```

- On OSX :

```
$ brew install ansible
```

For more:

http://docs.ansible.com/intro_installation.html

Installation - II

- Set hosts to test:

```
$ echo "127.0.0.1" > ~/ansible_hosts  
$ export ANSIBLE_HOSTS=~/ansible_hosts
```

- Test your installation:

```
$ ansible all -m ping --ask-pass  
SSH password:  
127.0.0.1 | success >> {  
  "changed": false,  
  "ping": "pong"  
}
```

Configuration- I

➤ Setup Config file:

- ANSIBLE_CONFIG (an environment variable)
- ansible.cfg (in the current directory)
- ansible.cfg (in the home directory)
- /etc/ansible/ansible.cfg

• ask_pass

This controls whether an Ansible playbook should prompt for a ssh password by default.

The default behavior is no:

`ask_pass=True`

For more: http://docs.ansible.com/intro_configuration.html

Configuration- II

- Inventory file : Define hosts and groups
- Default file : /etc/ansible/hosts

[databases]

54.167.36.187 mysql_role=master server_id=1

23.22.187.189 mysql_role=slave server_id=2 backup_src=54.167.36.187

54.87.24.162 mysql_role=slave server_id=3 backup_src=54.167.36.187

[system]

54.167.36.187 mycnf=plsc_s1 mysql_role=master nickname=dev-plsc-m

23.22.187.189 mycnf=plsc_s2 mysql_role=slave nickname=dev-plsc-s1

54.87.24.162 mysql_role=slave nickname=dev-plsc-s2ample:

- Dynamic Inventory :
 - AWS EC2 External Inventory Script (ec2.py)
- ```
ansible-playbook -i "localhost," ansible/playbooks/launch_ec2.yml
/ec2.py --list
```

For more: [http://docs.ansible.com/intro\\_inventory.html](http://docs.ansible.com/intro_inventory.html) ,  
[http://docs.ansible.com/intro\\_dynamic\\_inventory.html#example-aws-ec2-external-inventory-script](http://docs.ansible.com/intro_dynamic_inventory.html#example-aws-ec2-external-inventory-script)

# Configuration- III

- Playbooks
  - YAML format
  - One or more “plays” in a list
  - Executes tasks
  - Supports hosts
  - Supports roles
  - Supports tags
  - Supports groups
  - Supports remote\_user
  - Provides idempotent runs against host groups
  - Uses ansible modules

Example:

```
$ ansible-playbook playbook.yml --list-hosts
$ ansible-playbook playbook.yml --list-tasks
$ ansible-playbook playbook.yml -f 10
$ ansible-playbook playbook.yml --tags=setup,configure --list-tasks
$ ansible-playbook playbook.yml --tags=always,setup,configure --list-tasks
```

*For more: [http://docs.ansible.com/playbooks\\_intro.html](http://docs.ansible.com/playbooks_intro.html)*

# Configuration- IV

- Ansible built-in modules
  - Providing robust functions
  - Runtime or via playbook
  - Can be customized or developed

Example:

Adhoc

```
$ansible -i hosts.yml dbbservers -m service -a "name=mysql state=started"
```

```
$ansible -i hosts.yml webbservers -m command -a "/sbin/reboot -t now"
```

Playbooks

```
- name: reboot the servers
```

```
 command: /sbin/reboot -t now
```

*For more: <http://docs.ansible.com/modules.html>*

# Configuration- IV (cntd)

- MySQL Modules
  - mysql\_db → Add remove databases to remote hosts
  - mysql\_replication → Manage MySQL replication
  - mysql\_user → MySQL user management
  - mysql\_variables → Manage global variables
- Others Modules
  - Mongodb, postgresql
  - Package managers (yum, apt)
  - Service
  - File
  - Template

*For more:* [http://docs.ansible.com/list\\_of\\_database\\_modules.html#mysql](http://docs.ansible.com/list_of_database_modules.html#mysql)

# Best practices – Ansible Vault

- ansible-vault
  - Key management (aes256 encryption)
    - Create new encrypted vault  
\$ ansible-vault create system.yml
    - Editing encrypted key  
\$ ansible-vault edit system.yml
    - Rekeying  
\$ ansible-vault rekey system.yml
    - Encrypt unencrypted files  
\$ ansible-vault encrypt system.yml
    - Decrypt encrypted files  
\$ ansible-vault decrypt system.yml
    - Viewing decrypted files  
\$ ansible-vault view system.yml
    - Playbook integration  
\$ ansible-playbook -i clusters/test setup.yml --ask-vault-pass
    - File integration  
\$ ansible-playbook -i clusters/test setup.yml --vault-password-file ~/.vault\_pass.txt

*For more:* [https://docs.ansible.com/playbooks\\_vault.html](https://docs.ansible.com/playbooks_vault.html)

# Best practices – Templates

- Use of templates

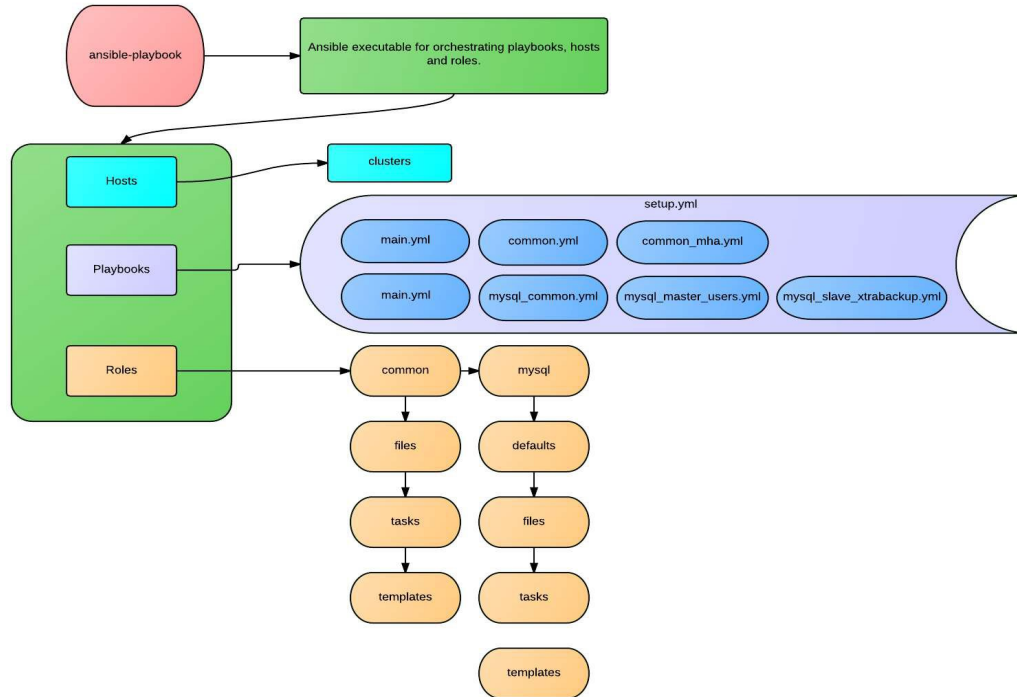
```
MHA cluster configuration
generated by Ansible, do not modify by hand.
config for cluster: {{cluster_id}}
[server default]
user={{my_mha_user}}
password={{my_mha_pass}}
repl_user={{my_repl_user}}
repl_password={{my_repl_pass}}
ssh_user=root
manager_workdir=/var/log/masterha/{{cluster_id}}
remote_workdir=/var/log/masterha/{{cluster_id}}
...
#my.cnf
innodb_buffer_pool_size = {{ (ansible_memtotal_mb * 70 / 100) |round|int }}M
...
```

For more: [https://github.com/mszel-pythian/ansible\\_mha/blob/master/roles/common/templates/etc/cluster.cnf](https://github.com/mszel-pythian/ansible_mha/blob/master/roles/common/templates/etc/cluster.cnf)

# MySQL Use Cases

- Installation and deployments
- MySQL configuration management
- User management
- DB task management
  - Backup and Restores
  - Bootstrapping
- Database change management
- Database pool management
- Orchestration with other services

# Ansible Workflow



# Demo — [https://github.com/mszel-pythian/ansible\\_mha](https://github.com/mszel-pythian/ansible_mha)

- Show configuration files and sample setup
- Demonstrate launching ec2 mysql instances configuring MHA and failing over with Ansible
  - SYSTEM SETUP(common role)
    - Fine tune system variables (sysctl)
    - Install system packages and mha
    - Create configuration files based on templates
    - Create users based on public keys under files/users
  - MYSQL PART(mysql role)
    - Install mysql binaries
    - Create my.cnf based on template (exception handling)
    - Install mysql users (replica,system users, monitor app users), remove passwordless users
    - Build slave automatically with xtrabackup streaming

# Demo – (cntd)

roles/common/tasks/common\_mha.yml

```
- yum: name=/root/mha4mysql-manager-0.56-0.el6.noarch.rpm state=present
 when: "mysql_role == 'mha_manager'"
```

TASK: [common | yum name=/root/mha4mysql-manager-0.56-0.el6.noarch.rpm state=present] \*\*\*

skipping: [54.163.66.141]

skipping: [54.161.225.49]

skipping: [54.166.98.113]

**changed: [54.157.195.13]**

[...]

roles/mysql/tasks/main.yml:

```
- include: mysql_slave_xtrabackup.yml
 when: "bootstrap_enabled and mysql_role != 'master'"
 tags: slave
```

TASK: [mysql | Streaming the backup from {{ backup\_src }} to {{ inventory\_hostname }}] \*\*\*

**skipping: [54.161.225.49]**

changed: [54.166.98.113]

changed: [54.163.66.141]

# Credits – Q & A

- Derek Downey @derek\_downey Principal Consultant, Pythian
- Rick Pizzi DBA, Bravofly
- Narcis Pillao Devops, eDreams