bacula-ansible Documentation

Release 0.1

Andraz Brodnik

Oct 26, 2017

Contents

Contents:

CHAPTER 1

Intro

This is an ansible playbook that makes deploying bacula easy. It has a role for:

- Bacula director
- Bacula file daemon (director client)
- Bacula storage daemon
- Multiple file daemons (clients)

Features

- Built-in distribution detection (can be easily extended for RHEL or different versions of Ubuntu)
- When client are installed the configuration that director needs is pushed to the director.
- Support for encrypted backups (you have to generate keys manually with the script)
- Special multiline var for each client is possible.

Limitations

- It's only tested for the standard small setup with the director, storage, and director filedaemon setup. (with multiple standard clients)
- Can't be installed on windows and no python boxes (limitations of ansible)

Plans

- Add Almir web interface
- Add AWS script for testing

- Register this project on Galaxy
- Support for TLS communication
- Multiple directors and stuff

How can you help

- Test if it works
- Talk to me (brodul [/.at/.] brodul.org)
- Improve documentation

How to use for your environment

I would recommend you that you to reuse the roles.

see this two links:

https://groups.google.com/forum/#!msg/ansible-project/vgc2bFQgzmE/0SDKwCniPjgJ http://www.ansibleworks.com/docs/playbooks_roles.html#id7

If you have other design patterns about reusage of ansible configuration please contact brodul on Freenode IRC server.

UPDATE:

http://galaxy.ansibleworks.com/ is released will port this playbook there

> **Warning:** This is still in development.

How to test

I recommend you use a public cloud for testing anything other will go also. You should easily adapt to your testing infrastructure.

Add at least 3 instances (2 clients and 1 director/storage). I use the micro instances on Amazon Web services EC2.

On your computer start ssh-agent and add your key with ssh-add. Make sure that the agent propegates the key to the connected host.

Connect to the director instance. Install:

```
- python-dev
- python-virtualenv
- git
- openssl
```

Clone this.

Change directory to the cloned dir. Make a virtual-env in the cloned dir with:

```
virtualenv .
```

Activate virtualenv and install ansible:

```
. bin/activate
pip install ansible
```

Set the correct hostnames in the testing inventory file. See example in the EC2 you need to setup the internal hostnames and domains.

Then you need to generate the master key pair and the client pairs.

For the generation of master keys run make_key_pair.sh.

Then set the group_vars/testing file. See example.

Then you need to generate the key pairs for each client in the system. The first argument needs to be the hostname of a client without the domain. See example:

```
sh make_fd_pair.sh ip-172-31-7-237
```

Then run:

```
ansible-playbook   testing-director-storage.yml -i testing  -u ubuntu
```

This will configure the director/storage.

Then run:

```
ansible-playbook   testing-clients.yml -i testing  -u ubuntu
```

This will configure the director/storage.