Code Submission and Docker Images for Linux VMs

Source code as Git project

All files including source codes, scripts, samples, notes, and instruction as well as this document are put together as a Git project. It is uploaded and accessible through GitHub. Run the following command to download

git clone https://github.com/lzjiangjeff/GR-tech-assignment.git

Linux Virtual Machine Image

2 Linux virtual machines created with Docker engine are submitted and uploaded to DockerHub. Using Docker pull command to download from DockerHub.

docker pull lzjiangjeff/gr-tech-assignment

They can also be built from scratch with Docker build command. Details are provided in the following section,

Virtualization Tools and Configuration Management Information

Virtualization tool to create the Linux virtual machine

The virtualization tool used to create the Linux virtual machine is Docker. The following table summarized the version information of the tools used and the Linux virtual machine it creates

Tools/Components/VMs	Version	Description
Local workstation	CentOS Linux release 7.5.1804 (Core)	● The local workstation on which all work are done ● It is also the Ansible server on which all playbooks are executed
Virtualization tool	Docker 18.06.1-ce, build e68fc7a	Tool to create and run Linux VMs
Linux virtual machine created	Ubuntu 16.04.5 LTS	Ubuntu 16.04 LTS container for both mysql1.local and backup1.local
Ansible	2.4.2.0	

Create, Launch, and Destroy Linux Virtual Machine

1. Create Linux VM with Dockerfile

   Linux virtual machine is created with Docker engine with Dockerfile.

   Source code file: ***docker/Dockerfile***

   Command to run: ***docker build –t YOUR-TAG-NAME - -file docker/Dockerfile***

   A pre-built image is also uploaded and available for download/pull from DockerHub registry:

   docker pull lzjiangjeff/gr-tech-assignment

2. Start Linux VM

   Target Linux VM is launched by running docker container.

   Launch mysql1.local:

   docker run -d -p 22221:22 --name mysql-container --hostname=mysql1.local IMAGE

   Launch mysql1.local

   docker run –d -p 22222:22 --name backup-container --hostname=backup1.local IMAGE

3. Accessing Running Linux VM

   Running Linux VMs can be accessed through SSH e.g.

   ssh -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null -p 22221 root@localhost

   ssh -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null -p 22222 root@localhost

   Default password is ***securepasswd***

4. Stop/Destroy Linux VM

   a) Stop running Linux VM with Docker command: ***docker stop CONTAINER_NAME***

   b) Destroy Linux VM with Docker command: ***docker rm CONTAINER_NAME***

Ansible Server Configuration and Assumptions

Assumptions

I assume user who runs my code has root level privilege to the machine, e.g. user has the permission to set and change firewall and/or system services settings.

Adding hosts group to Ansible server hosts configuration file

The following setting needs to add to ***/etc/ansible/hosts*** file to register 2 new Linux VM

[docker-hosts]

mysql1 ansible_host=localhost ansible_port=22221 ansible_ssh_user=root ansible_ssh_pass=securepasswd

backup1 ansible_host=localhost ansible_port=22222 ansible_ssh_user=root ansible_ssh_pass=securepasswd

A sample hosts file, ***hosts.sample***, is included in source code and available on GitHub repository

Disable host key checking

For simplicity, we can avoid checking host key for each SSH session by disabling strict host key verification. But it is not mandatory. A sample ***ansible.cfg.sample*** file is included in source code and available in GitHub repository

Ansible code to install, configure and start mysql on mysql1.local

Code is implemented in an Ansible role called ***install_mysql***. Run the following command to execute this Ansible playbook role

ansible-playbook assignment.yml --tags=install_mysql

The following 5 tasks are defined and executed

1. Install Mysql server
2. Configure Mysql server (setting port, data folder, log folder, error log, etc…)
3. Start Mysql the first time
4. Set the root password
5. Restart Mysql server

Output

After execution, on Linux VM mysql1.local, Mysql server is installed, configured, and running. Mysql server root user password is set.

Ansible code to create cron backup job for Mysql server on mysql1.local

Code is implemented in an Ansible role called ***backup_mysql***. Run the following command to execute this Ansible playbook role

ansible-playbook assignment.yml --tags=backup_mysql

The following 3 tasks are defined and will be executed by this role

1. Create backup directory
2. Create cron job
3. Restart Cron daemon

Output

After execution, on Linux VM mysql1.local, a daily cron job is set to run a script, mysql-backup, to backup Mysql server databases and store the backup dump file (as well as its MD5 checksum file) on the Linux VM backup1.local

Test cases to verify the backup

Code is implemented in an Ansible role called ***verify_backup***. Run the following command to execute the Ansible playbook role

ansible-playbook assignment.yml --tags=verify_backup

The following 4 tasks are defined and will be executed by this role

1. Copy good test files only if when test_mode is set to true
2. Copy bad test files only if when test_mode and test_bad are both set to true
3. Create shell script
4. Verify checksum

Output

During execution, all backup dump files are validated against the MD5 checksum to make sure the backup dump file is not corrupted. If test_mode is set to true, additionally it will copy all good sample files to target VM regardless whether or not there are existing backup dump files. If test_bad is also set to true, it will also copy bad sample files to target VM. Execution will fail as expected