Provisioning and Deployment

System Deployment and Benchmarking

2020/2021

The main goal of this guide is to understand how to provision systems and deploy services in an automatic and reproducible fashion.

For the exercises described next, the following tools must be installed,

- Ansible - `http://ansible.com`

Ansible documentation is available at:

- Ansible - `http://docs.ansible.com/ansible/`
- Ansible Getting Started - `https://docs.ansible.com/ansible/latest/user_guide/intro_getting_started.html#intro-getting-started`
- Ansible Encrypted Passwords - `http://docs.ansible.com/ansible/latest/faq.html#how-do-i-generate-crypted-passwords-for-the-user-module`

Steps

Ansible Setup

1. Launch a local VM using Ubuntu 20.04.
2. At the VM install ansible with *sudo apt install ansible*.

Google Cloud Platform (GCP) Setup

1. Add a public key to GCP so it can be made available in the instance. Be sure to add the corresponding private key to the VM running Ansible.
2. Manually Create one GCP VM instance (Ubuntu 20.04)

Inventory and Provisioning

1. Create an Inventory that contemplates the GCP VM.

2. Use the Ansible ping module to gather information from the VM.

   e.g., ansible -i "inventory_file" -u "username" all -m ping

   Note that you do not need to use the default inventory file at /etc/ansible/hosts. Also "all" can be replaced with the inventory group name.

3. Create an Ansible Playbook that is able to:

   (a) Update system packages
   (b) Install vim-nox, openntpd, and sudo
   (c) Create a user called *tester* with password *123456*
   (d) Add *tester* to *sudo* group
   (e) Prepare the user *tester* for SSH public key authentication
   (f) Disable *root* and *password* authentication for OpenSSH
   (g) Make sure OpenNTPD and OpenSSH services are enabled and running

4. The provisioning steps should run at the GCP VM.

5. The provisioner should be the local VM. The playbook can be executed with *ansible-playbook -i "inventory_file" -u "username" playbook.yml*

Hints

1. Explore the *apt* module for updating the system

2. Explore the *apt* module for installing packages

3. Explore the *user* module for user creation

4. Explore the *authorized_key* module to handle public keys

5. Explore the *copy* module for OpenSSH configuration

6. Explore the *service* module for handling service state

Testing

1. Login (ssh) into the VM and check if all the changes described in **Steps** are satisfied

   (a) Should be able to *login* and use *sudo* with the user *tester*
   Sudo password should be asked to this user.
   (b) Command *systemctl status ssh* should display active and enabled
   (c) Command *systemctl status openntpd* should display active and enabled

Swap Playbook

1. Create an Ansible Playbook that is able to deploy the *Swap* application in two GCP Virtual Machines.

   The MySQL database will be deployed in one of the VMs (VM1) and the Swap php application in the other VM (VM2).

   #### Hints

   (a) Read again the practical guide *Case-study application: Swap* for further instructions on how to deploy Swap.

   (b) Explore Ansible roles, templates, files, handlers, and tags.

   (c) Create at least one role for common VM configurations, another for the database deployment, and another one for the Swap app deployment.

   (d) Explore the *apt, apt_repository, copy, service, shell, git, lineinfile, mysql_db, mysql_user* ansible modules.

   (e) Look at examples of ansible playbooks at `https://github.com/ansible/ansible-examples`.

   (f) The internal IP of the VMs should be used for the database connections.

   (g) GCP firewall needs to be configured so that clients can access port 8080 on VM2.

   (h) Run the Swap server with:

   ```
   nohup php artisan serve --host=0.0.0.0 --port:8080 > app_out.log 2>&1 &
   ```

Extra

1. Explore dynamic inventories for GCP - `https://docs.ansible.com/ansible/latest/scenario_guides/guide_gce.html`

Learning outcomes

Experiment systems provisioning and configuration management workflows with Ansible Develop playbooks that hold reproducible provisioning recipes. Understand the importance of task automation and self documentation.