Cloud Computing Applications and Services

Provisioning and Deployment

October 23, 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 in one of the VMs.

- 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 (Hashed) Passwords - `http://docs.ansible.com/ansible/latest/faq.html#how-do-i-generate-crypted-passwords-for-the-user-module`

Steps

VMs Setup

1. Create three VMs (see warm-up guide).
2. At VM1 install ansible with *sudo apt install ansible*.

Inventory and Provisioning

1. Create an Inventory for VM2.
2. Use the Ansible ping module to gather information from VM2.

   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.

   The command and inventory file should be executed at VM1. Do not forget to copy your private ssh key to VM1 as ansible will connect to VM2 with ssh.

At your inventory file you can specify the following host variable: *ansible_ssh_common_args='-o StrictHostKeyChecking=no'* to avoid the key host checking process.

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*. Check how to create hashed passwords on the documentation at this guide.
   (d) Add *tester* to *sudo* group
   (e) Prepare the user *tester* for SSH public key authentication
   (f) Make sure OpenNTPD and OpenSSH services are enabled and running
4. The provisioning steps should run at VM2.
5. The provisioner should be the local VM1. The playbook can be executed with *ansible-playbook -i "inventory_file" -u "username" playbook.yml*.
6. The "-K" flag can be used for ansible to request sudo password (if the user requires password for sudo at the machine where the provisioning is running).

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 and management
4. Explore the *authorized_key* module to handle public keys
5. Explore the *service* module for handling service state

Testing

1. Login (ssh) into VM2 and check if all the changes described in **Steps** are satisfied
   (a) Should be able to *login* and use *sudo* with the user *tester*
   You can check that you have configured correctly the user password with the command *su - tester*
   (b) Command *systemctl status ssh* should display active and enabled
   (c) Command *systemctl status openntpd* should display active and enabled

Swap Playbook

1. Launch VM3.
2. Create an Ansible Playbook that is able to deploy the *Swap* software stack in VMs 2 and 3.

   For example, the MySQL database will be deployed in VM2 and the Swap php application in VM3.

   Hints

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

   (b) Explore Ansible roles, templates, and handlers

   (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) Run the Swap server in background with:

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

Extra

1. Explore using ansible tags in your playbook for different roles.
   Then use the flags

   ```
   --tags and --skip-tags
   ```

   to run only a role, or exclude a role, when executing your playbook.

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.