RHCE Practice Exam 2

To work through this exam, you need a total of five servers that are running RHEL 8 or Centos 8. One server needs to be configured as the control host, and the other four servers should be configured as managed servers, using the names ansible1.example.com through ansible4.example.com. The IP addresses used on the managed servers are not important; you can pick anything that matches your configuration. Make sure the servers meet the following requirements:

- 1 GB of RAM.
- 20 GB of disk space on the primary disk /dev/sda.
- 5 GB of disk space on the secondary disk /dev/sdb, which only exists on ansible1 and ansible2.
- The root user account configured with the password "password" on each of the servers.
- The control server with a user account "ansible". SSH public and private keys have been generated for this user. No further configuration has been done yet.

In the assignments in this exam, you'll need to create scripts and yaml files. Make sure that all these scripts are stored in the directory /home/ansible.

Common Tasks

1. Configure the control host with a static inventory, as well as the ansible.cfg configuration file. In the static inventory, configure the following host groups:
   a. Group test with ansible1.example.com as a member
   b. Group dev with ansible2.example.com as a member
   c. Group prod with ansible3 and ansible4 as members
   d. A group server, with groups dev and prod as members

   Ensure that hosts can be reached through their FQDN, but also by using the short name (so ansible1.example.com as well as ansible1).
2. Create a playbook with the name setupreposerver.yml to set up the control host as a repository host. Make sure this host meets the following requirements, which must be done by the playbook.
   a. The RHEL 8 installation ISO is loop-mounted on the directory /var/ftp/repo.
   b. The firewalld service is disabled.
   c. The vsftpd service is started as well as enabled and allows anonymous user access to the /var/ftp/repo directory.
3. Create a script that configures the managed servers as repository clients to the repository server that you have set up in the previous task. This script must use ad-hoc commands and perform the following tasks:
   a. Disable any currently existing repository.
   b. Enable access to the BaseOS repository on control.example.com.
   c. Enable access to the AppStream repository on control.example.com.
4. Create a script with the name setuphosts.sh that uses ad hoc commands to complete configuration on the managed servers. This includes:

   a. Installing Python
   b. Creating a user with the name ansible
   c. Creating a sudo configuration that allows user ansible to run tasks with root privileges
   d. Using an ad-hoc command to call the appropriate module to test connectivity to the remote hosts

Exam 2 Specific Tasks

1. Set up the Ansible configuration file such that the directory /home/ansible/roles is used as the default location to store Ansible roles. Make sure that other locations are also still available.
2. Create a playbook with the name setupstorage.yml that accomplishes the following tasks:
   a. On all servers that have a second hard drive, create a partition with a size of 5 GiB.
   b. Use this partition to set up an LVM volume group with the name vgdata that uses physical extents with a size of 8 MiB.
   c. In the vgdata volume group, create a logical volume with the name lvdata and a size of 1 GiB.
   d. Format this logical volume with the Ext3 file system.
   e. Ensure the volume is mounted persistently on the directory /data.
3. Create a playbook with the name packagefacts.yml that gathers facts about packages that are installed on your managed nodes. Have the playbook generate a report with the name /root/packages.txt. In this report, package versions should be printed for the packages listed next. Make sure the report is printed in the format packagename=version, such as zlib=1.2.11. Do this for the following packages:
   a. kernel
   b. bash
   c. glibc
4. Create a vault-encrypted password file with the name cloudpass.yml. In this file, set the variable CLOUDID to the value myid, and set the variable CLOUDPASS to the password cloudpass. Encrypt the vault file with the password cloudsecret. Store this password in the file vaultpass.txt in such a way that it can be used while using this cloudpass.yml file in a playbook.
   Next, create a playbook with the name usevault.yml. This playbook should import the variables that are set in the cloudpass.yml file and use them to create a clear text readable file with the name /root/cloudcreds.txt. In this file, the variables and their values should be listed in the VARNAME=value format, like CLOUDPASS=cloudpass. Ensure this playbook can use the vault password file that you have created.
5. Install the geerlingguy.nginx role and the geerlingguy.docker role from Ansible Galaxy. Make sure they are stored in /home/ansible/roles. Create a playbook with the name start-galaxy-roles that starts both of them.
6. Create a role to configure the Apache web server. It should start and enable the Apache service, and it should open the firewall to allow external access. Next, create a sample index.html in the directory /var/www/html that is based on a template file. When it is accessed, the visitor should see a welcome to HOSTNAME at IPADDRESS,

where HOSTNAME and IPADDRESS are replaced with the current host name and IP address. Write a playbook with the name runweb.yml that runs all tasks in this role on servers in the group test. Before any task in the role is executed, the playbook should ensure the firewalld service is started on the managed hosts.

7. Create a playbook with the name create_users.yml. This playbook should create users based on the input file users_pass.yml. Manually create this file, and ensure it has the following contents:

   ```
   users:
   - name: linda
     password: password
     department: profs
   - name: lisa
     password: secret
     department: profs
   - name: anna
     password: geheim
     department: students
   ```

   On servers that are in the prod group, users who have the department set to profs should be created, and the department should be set as a secondary group to the user. Also make sure that the password that is specified in users_pass.yml is set as a SHA256-encrypted password while creating the users.