<Brian W. Lee>

CSC 2510: DevOps

Lab 13 – Ansible I

General Instructions

Using your book and previous lecture material, fill out this assignment sheet. **Use blue text to signify your answers.** You should utilize online resources and your book to answer these questions as well.

Submission Instructions

To submit, **change the name in the header** and save this document as a PDF. Attach your PDF document to the iLearn dropbox.

Lab Instructions

In this lab we will be automating the installation and configuration of Apache and the installation of your `index.html` file. We could take this a step further and automate the provisioning of your VMs. However, that is outside the scope of this lab. Additionally, using and understanding Ansible documentation will be very important for this lab. All linked documentation pages show an example that is very similar to what you need to accomplish in this lab. Also note the that there is a Troubleshooting Guide section with some common problems.

1. Your `csc-2510` VM will now be known as your *primary* VM.
2. Clone your infrastructure repository to your primary VM.
3. Use *apt-get update*
4. Use *apt-get upgrade*
5. Use *apt-get install ansible*
6. Make a new VM following the instructions below.
    a. Click on the VM you created during Lab 12 (`csc-2510-apache`).
    b. Click the `Create Similar` button at the top of the page.
    c. Change the `Name` to `csc-2510-ansible-demo`.
    d. Click `Create` or `Done`.
7. Start your new `csc-2510-ansible-demo` VM but do not connect to it. Make note of the internal IP address of your VM.
8. From your primary VM, do the following:
    a. Follow the instructions here: https://souvikhaldar.medium.com/how-to-add-the-public-ssh-key-to-a-gcp-virtual-machine-ef5703e8e596 to setup an SSH key on your GCP account for use during this and future VMs.

b. Use the following command to create an SSH connection from your primary VM to your new VM. Be sure to replace `<internal_ip_addr>` with the internal IP of your new VM.

```
ssh -i ~/.ssh/gcp_devops <internal_ip_addr>
```

9. After confirming that the SSH connection is established and working type `exit`.
   a. This should put you back into your Primary VM's terminal.
10. In your infrastructure repository do the following (with regular commands):
    a. We must first create a host file. This file will store information regarding every *host* (or server) that we need to connect to. Ansible allows us to write several config files as either `.cfg` or `.yml`. We will be writing all files as `.yml` for consistency and for IDE support.
       i. Create a file at `ansible_inventory/hosts.yml`.
       ii. In this file, you will need to create something similar to what is shown in the Ansible documentation for a basic inventory file. For more information on Ansible host files view the Ansible documentation.
       iii. We only need 1 group and 1 host. Name your 1 group `devops` and name your 1 host `demo`. You should use your internal IP address for your VM, as this rarely changes. You will not need any `vars` sections at this time.
          1. **Tip:** YML files hate tabs. You ***must*** use spaces for indentation.
    b. We now need to create an Ansible config file so that ansible knows where to find our inventory file. We will also set host key checking to false.
       i. Create a file at `ansible.cfg`.
       ii. Add the following to the file. Note that there is no indentation.

```
[defaults]
inventory = ansible_inventory
host_key_checking = false
```

    c. Now we need to create our first Ansible playbook.
       i. Create a file at `example.yml`.
       ii. Add the following to your file.

```
---
- hosts: all
  become: no
  become_method: sudo
```

       iii. Make note at what each of these commands are doing, or rather what they are setting up. What would happen if we changed `hosts: all` to `hosts: devops` or to `hosts: demo`? What does `become` do?
    d. We've created our playbook but it currently doesn't *do* anything. Let's add some commands to install the packages we need, `python-apt` and `apache2`. Use the Ansible documentation on the package module to add this.
          1. **Tip:** You will need to add a `tasks` section to your playbook.
          2. **Tip:** You will need to use `become: yes` to make this work. Why is that?

e. Next, we need to copy over our config files from Lab 12. Before we do that let's restructure our repository so that it makes more sense.
    i. Restructure your repository so that it looks like this:

```
<repository root>/
│
├── ansible_inventory/
│   └── hosts.yml
│
├── files/
│   ├── config/
│   │   └── apache.conf
│   │
│   └── site/
│       └── index.html
│
├── ansible.cfg
└── example.yml
```

    ii. Use the Ansible documentation for the copy module to copy the `apache.conf` and `index.html` files to the correct locations. Be sure to set the owner, group, and permissions (or mode) correctly.

f. Now, we need to setup the symbolic link. Use the Ansible documentation for the file module to do this.
    i. **Tip:** use the `state: link` option.

g. The last command we need to do, is restart the apache2 service. You can do this using the Ansible documentation for the service module.

11. Finally, we need to run the Ansible playbook. Use the following command to run the playbook.

```
ansible-playbook --private-key ~/.ssh/gcp_devops example.yml
```

12. Assuming your script ran without errors, you should now have an accessible webpage just like on your VM from Lab 12.
13. Remember to commit and tag your final submission.

Troubleshooting Guide

- If you have confirmed that your ssh connection works manually, but the Ansible playbook returns an `unreachable` or `connection refused`, try restarting your VMs (and giving them some extra time to warm-up). This issue appeared frequently while writing this lab with no clear cause.
- Did you add `become: yes` to the right tasks?
- Did you use tabs? Tabs are bad.
- Is your YML formatted correctly? This site can help you check general YML syntax.
- Did you format your YML correctly for Ansible specifically? Examples of completed playbooks can be found here.
- Are you running the Ansible command from the right location?

<Brian W. Lee>

Lab Questions

1. (5) Ansible has a large community that creates roles called Ansible Galaxy. What are Ansible roles and how are they useful? Ansible Galaxy is large public repository of Ansible roles, which the roles define a set of tasks that you want to run on your target hosts. It is useful to utilize the roles on the Playbooks for automated projects.
2. (5) List 3 alternative IaC tools to Ansible. Terraform, Saltstack, Puppet