Lab 07

Instructions:

1. Paste all screenshots (highlighted in red) in a single Word document in the correct order
2. Name the document as **YourName-lab07**
3. Submit the document as an attachment in Bb under Labs
4. Use a WSL terminal for all activities

Lab submissions must be made by the due date (as indicated on the Critical Path). Each day thereafter will incur a **10%** deduction from the earned marks, up to a maximum of **3 days**. Submissions beyond this deadline will receive a grade of **Zero**.

Lab Objectives:

There are 4 sections in this lab as described below:

Section 1: Build a combination of Linux and Windows virtual machines using Terraform
Section 2: Configure Ansible defaults
Section 3: Configure Ansible inventory
Section 4: Run ad-hoc commands and confirm output

WARNING

Code generated by ChatGPT or a similar generative AI tool, and copied and pasted without making the **right** modifications will result in a **ZERO** for that **entire section**.

Section 1

Objectives:

- Modify the existing Terraform code to build 2 CentOS and 1 Windows 2019 virtual machines as Ansible managed nodes
- Validate, deploy, and test infrastructure

Part 1: Prepare for the lab:

1. Copy folder **lab06** as **lab07s1**
2. Change into **lab07s1**
3. Modify appropriate Terraform variables, main, and outputs files to create infrastructure to support the following 3 virtual machines:

Hostname	**(VM size)**	**Operating System**
<HumberID>-c-vm1	(Standard_B1s)	CentOS 8.2
<HumberID>-c-vm2	(Standard_B1s)	CentOS 8.2
<HumberID>-w-vm1	(Standard_B1s)	Windows 2019

4. Use the following information for CentOS image:

   Publisher: OpenLogic
   Offer: CentOS
   Sku: 8.2 (or 8_2)
   Version: Latest

Part 2: Validate configuration:

5. Validate the configuration to ensure there are no errors or typos in the file (**terraform validate**)
6. Fix any issues in the Terraform files if reported
7. Re-run the validation until no errors are reported (**terraform validate**)

Part 3: Format configuration:

8. Format all Terraform configuration files (**terraform fmt -recursive**)

Part 4: Run simulation:

9. Perform a dry run (**terraform plan**)
10. Review output and ensure all configuration is as per requirements. Observe the resources with +, -, or -/+ signs.
11. Fix any issues in the Terraform files if reported
12. Redo the dry run until no errors are reported (**terraform plan**)

Part 5: Deploy infrastructure:

13. Deploy the infrastructure and monitor progress (**terraform apply**)

Part 6: Get information from Terraform state:

14. View and analyze state information (**terraform state list | nl**) **SCREENSHOT**
15. Display the output values (**terraform output**) **SCREENSHOT**

Part 7: Test access to Linux VMs:

16. Run the **ssh** command on the automation VM one at a time as follows to test access to the managed nodes using their FQDNs:

$ **ssh -o StrictHostKeyChecking=False <HumberID>@<HumberID>-c-vm1.<azure_region>.cloudapp.azure.com** SCREENSHOT

$ **ssh -o StrictHostKeyChecking=False <HumberID>@<HumberID>-c-vm2.<azure_region>.cloudapp.azure.com** SCREENSHOT

You should be able to log in to each machine as the <**HumberID>** user without being prompted for a password.

Part 8: Test access to Windows VMs:

17. Use the RDP program on your personal Windows computer to test connection to the following Windows managed node with username **winadm** and password **Winadm!23**: SCREENSHOT

<HumberID>-w-vm1.<azure_region>.cloudapp.azure.com

Part 9: Install Ansible on automation/control node:

18. Install Ansible:

$ **sudo apt install ansible**

19. Check Ansible version:

$ **ansible --version**

20. Create the following directory to be used as Ansible home:

$ **mkdir ~/automation/ansible -p**

============================= End of Section 1 ================================

Section 2

Objectives:

- Set configuration defaults for Ansible on the automation/control node

Part 1: Set configuration defaults:

1. Create a file called **ansible.cfg** under **~/automation/ansible** using a text editor of your choice and define two sections using the information provided below:

A defaults section:
inventory set to ~/automation/ansible/hosts
host_key_checking set to false
deprecation_warnings set to false

A privilege_escalation section:
become set to true
become_method set to sudo
become_user set to root
become_ask_pass set to false

2. Run **ansible-config view** SCREENSHOT

=============================== End of Section 2 ===============================

Section 3

Objectives:

- Set up and view Ansible inventory on the automation/control node

Part 1: Set up and view host inventory:

1. Create **~/automation/ansible/hosts** file with the following content:
   a. A group called **linux** with FQDNs of <HumberID>-c-vm1 and <HumberID>-c-vm2 as members
   b. A group called **windows** with FQDN of <HumberID>-w-vm1 as a member
   c. A group called **os** with **linux** and **windows** groups as children

2. Add the following to the inventory hosts file:

   [windows:vars]
   ansible_user set to winadm
   ansible_password set to “Winadm!23"
   ansible_connection set to winrm
   ansible_port set to 5985
   ansible_winrm_transport set to ntlm
   ansible_winrm_server_cert_validation set to ignore

3. Run **ansible-inventory --graph** SCREENSHOT

============================== End of Section 3 ===============================

Section 4

Objectives:

- Use common Ansible modules to perform ad-hoc operations

Part 1: Run the following ad-hoc commands (SCREENSHOT of all):

1. Use the **ping** module to test network connectivity to Linux managed nodes
2. Use the **shell** module to display the current date on <HumberID>-c-vm1 only
3. Use the **setup** module to display the configuration of <HumberID>-w-vm1
4. Use the **user** module to create a user account called **user2000** with UID 2000 on both <HumberID>-c-vm1 and <HumberID>-c-vm2
5. Use the **user** module to delete the user account **user2000** from <HumberID>-c-vm2
6. Use the **win_user** module to create a user account called **winuser2000** on <HumberID>-w-vm1
7. Use the **win_user** module to delete the user account **winuser2000** from <HumberID>-w-vm1
8. Use the **copy** module to (a) add content "Managed by Ansible" to local file **test_file** and then (b) upload it to the **linux** inventory group in the **/tmp** directory
9. Use the **command** module to display the content of the **/etc/motd** file from the **linux** inventory group nodes
10. Use the **file** module to create a directory called **testdir** on localhost
11. Use the **apt** module to install the package called **cifs-utils** on localhost and then remove it
12. Use the **shell** module to display the value of the variable USER from both Linux nodes

============================= End of Section 4 ===============================