Lab 06

Instructions:

1. Paste all screenshots (highlighted in red) in a single Word document in the correct order
2. Name the document as **YourName-lab06**
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 is 1 section in this lab. The section is described below:

Section 1: Create separate child modules for (1) resource group, (2) network, (3) 2 Linux VMs, and (4) 2 Windows VMs, and call them from the root module.

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**.

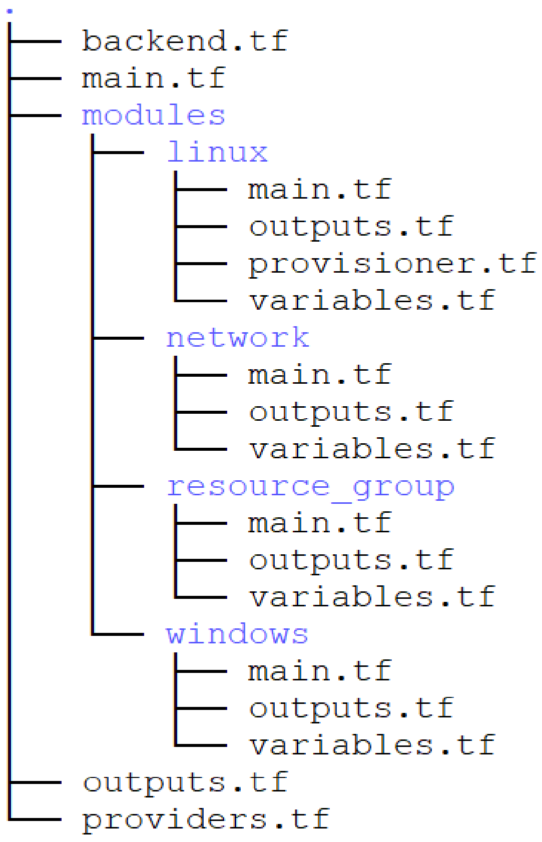
Section 1

Objectives:

- Modularize Terraform configuration by creating separate child modules for (1) resource groups, (2) networking, (3) Linux VMs with provisioner, and (4) Windows VMs (without provisioner), and call them from the root module.
- Ensure that output values are properly defined for the child and root modules
- Validate, deploy, analyze, and destroy infrastructure

Part 1: Prepare for the lab:

1. Create folder called **lab06** under **automation/terraform**
2. Change into **lab06**
3. Create the file structure as shown in the image under **lab06**:



```
.
├── backend.tf
├── main.tf
├── modules
│   ├── linux
│   │   ├── main.tf
│   │   ├── outputs.tf
│   │   ├── provisioner.tf
│   │   └── variables.tf
│   ├── network
│   │   ├── main.tf
│   │   ├── outputs.tf
│   │   └── variables.tf
│   ├── resource_group
│   │   ├── main.tf
│   │   ├── outputs.tf
│   │   └── variables.tf
│   └── windows
│       ├── main.tf
│       ├── outputs.tf
│       └── variables.tf
├── outputs.tf
└── providers.tf
```

Part 2: Create resource group module:

Hint: Use lab05s2 as a reference.

4. **Define** three resource group blocks in modules/resource_group/main.tf file: one for networking, one for Linux, and one for Windows.
5. **Declare** variable blocks for resource group names and location in modules/resource_group/variables.tf file.
6. **Define** three output blocks for resource groups in modules/resource_group/outputs.tf file.
7. **Define** a module block for resource groups in root module’s main.tf file and define variables that are declared in modules/resource_group/variables.tf file.
8. **Define** root module’s outputs.tf file to display the resource group names in the output.

Part 3: Initialize Terraform and validate configuration:

9. Initialize Terraform (**terraform init**)
10. Validate the configuration to ensure there are no errors or typos in the file (**terraform validate**)

11. Fix any issues in the Terraform files if reported
12. Re-run the validation until no errors are reported (**terraform validate**)

Part 4: Format configuration:

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

Part 5: Run simulation:

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

Part 6: Deploy infrastructure:

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

Part 7: Get information from Terraform state:

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

Part 8: Destroy all resources and verify:

21. Destroy all the resources (**terraform destroy**)
22. Verify deletion (**terraform state list | nl**)
23. View the content of terraform.tfstate file (**tail -20 terraform.tfstate**)

Part 9: Create networking module:

Hint: Use lab05s2 as a reference.

1. **Define** resource blocks for networking (1 virtual network, 2 subnets, and 2 network security groups) in modules/network/main.tf file. Deploy these resources in the resource group defined earlier for networking.
2. **Declare** variable blocks for networking in modules/network/variables.tf file.
3. **Define** output blocks for networking (names and address spaces for the virtual network and subnets, and names of the network security groups) in modules/network/outputs.tf file.
4. **Define** a module block for networking in root module’s main.tf file and define variables that are declared in modules/network/variables.tf file.
5. **Update** the root module’s outputs.tf file to include the names and address spaces for the virtual network and subnets, and names of the network security groups.
6. Test by executing steps provided above (parts 3 to 8).

terraform init, terraform validate, terraform plan, terraform apply, **terraform state list | nl (screenshot)**, **terraform output (screenshot)**, terraform destroy, and **terraform state list | nl (screenshot)**.

Part 10: Create Linux virtual machine module:

Hint: Use lab05s2 as a reference.

1. **Define** resource blocks for 2 Linux virtual machines (availability set, virtual machines, network interface cards, and public IP addresses) in modules/linux/main.tf file. Deploy these resources in the resource group defined earlier for Linux virtual machines.
2. Define the provisioner block for Linux virtual machines in modules/linux/provisioner.tf file.
3. **Declare** variable blocks for Linux virtual machines in modules/linux/variables.tf file.
4. **Define** output blocks for Linux virtual machines (availability set, virtual machines, private IP addresses, and public IP addresses) in modules/linux/outputs.tf file.
5. **Define** a module block for Linux virtual machines in root module's main.tf file and define variables that are declared in modules/linux/variables.tf file.
6. **Update** the root module's outputs.tf file to include the hostnames, FQDNs, and private and public IP addresses for Linux virtual machines.
7. Test by executing steps provided above (parts 3 to 8).

terraform init, terraform validate, terraform plan, terraform apply, **terraform state list | nl (screenshot)**, **terraform output (screenshot)**, terraform destroy, and **terraform state list | nl (screenshot)**.

Part 11: Create Windows virtual machine module:

Hint: Use lab05s2 as a reference.

1. **Define** resource blocks for 2 Windows virtual machines (availability set, virtual machines, network interface cards, and public IP addresses) in modules/windows/main.tf file. Deploy these resources in the resource group defined earlier for Windows virtual machines.
2. **Declare** variable blocks for Windows virtual machines in modules/windows/variables.tf file.
3. **Define** output blocks for Windows virtual machines (availability set, virtual machines, private IP addresses, and public IP addresses) in modules/windows/outputs.tf file.
4. **Define** a module block for Windows virtual machines in root module's main.tf file and define variables that are declared in modules/windows/variables.tf file.
5. **Update** the root module's outputs.tf file to include the hostnames, FQDNs, and private and public IP addresses for Windows virtual machines.
6. Test by executing steps provided above (parts 3 to 8).

terraform init, terraform validate, terraform plan, terraform apply, **terraform state list | nl (screenshot)**, **terraform output (screenshot)**, terraform destroy, and **terraform state list | nl (screenshot)**.

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