

Lab 04

Instructions:

1. Paste all screenshots (highlighted in red) in a single Word document in the correct order
2. Name the document as **YourName-lab04**
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 2 sections in this lab. Each section has a different set of objectives. The sections are described below:

Section 1: Create multiple identical Linux VMs using count in one availability set

Section 2: Create multiple distinct Windows VMs using for_each in another availability set

You are not allowed to have more than **4 CPUs** for all your VMs combined in your Azure **Education** subscription. Please ensure that you remove any existing but unused VMs before proceeding with this lab.

Also, you are not allowed to create **more than 3 Public IP addresses** in your Azure **Free** subscription. Please adjust the quantity of VMs to create to no more than 3.

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:

- Create parametrized, scalable Terraform code to build any number of identical resources using the count meta-argument
- Deploy all virtual machines in a single availability set
- Validate, deploy, expand, analyze, and destroy infrastructure

Part 1: Prepare for the lab:

1. Copy the directory **lab03s3** as **lab04s1**
2. Change into **lab04s1**

Part 2: Update vmlinux-vars.tf file:

3. Define a new variable block for Linux availability set called `linux_avs`
4. Define a new variable block called `nb_count` with a value of 2. Leave rest of the settings intact.

SCREENSHOT of vmlinux-vars.tf showing the definition of the two variable blocks

Part 3: Update vmlinux-main.tf file:

5. Define a new resource block for availability set with 2 fault domains and 5 update domains. Use the `azurerm_availability_set` resource type.
6. Modify the VM resource block to include the count meta-argument in the VM name, network interface ID, computer name, and OS disk name
7. Modify the NIC resource block to include the count meta-argument in the NIC name, IP configuration name, and public IP address ID
8. Modify the Public IP resource block to include the count meta-argument in the public IP name

SCREENSHOT of vmlinux-main.tf

Part 4: Update outputs.tf file:

9. Define a new output block to display Linux availability set name
10. Modify the output values to reflect multiple resource creation for VMs, DNS labels, private IP addresses, and public IP addresses. Leave values for single resources (vnet, subnet, etc.) intact.

SCREENSHOT of outputs.tf

Part 5: Validate configuration:

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

Part 6: 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 7: Deploy infrastructure:

18. Deploy the infrastructure and monitor progress (**terraform apply**)
19. View Terraform log file and review details
20. View the content of terraform.tfstate file (**tail -20 terraform.tfstate**)

Part 8: Get information from Terraform state:

21. View and analyze state information (**terraform state list | nl**)
22. Display the output values (**terraform output**)

SCREENSHOT
SCREENSHOT

Part 9: Confirm resource creation in Azure:

23. Log in to the Azure Portal. Navigate to the resource group and confirm the names and types of all the resources created. Verify tags on the resources as well. **SCREENSHOT**

Part 10: Destroy all resources and verify:

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

You must run terraform destroy before moving to section 2.

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

Section 2

Objectives:

- Create parametrized, scalable Terraform scripts to build any number of distinct resources using the for_each meta-argument
- Deploy all virtual machines in a single availability set
- Validate, deploy, expand, analyze, and destroy infrastructure

Part 1: Prepare for the lab:

1. Copy the directory **lab04s1** as **lab04s2**
2. Change into **lab04s2**
3. Copy **vmlinux-vars.tf** as **vmwindows-vars.tf** [Leave vmlinux-vars.tf intact]
4. Copy **vmlinux-main.tf** as **vmwindows-main.tf** [Leave vmlinux-main.tf intact]

Part 2: Update vmwindows-vars.tf file:

5. Remove the locals block
6. Define a new variable block for availability set called windows_avs
7. Modify existing values to reflect Windows VM deployment. This includes the OS and disk information.

Windows OS:

Publisher: MicrosoftWindowsServer

Offer: WindowsServer

SKU: 2016-Datacenter

Version: latest

Disk Size: 128GB

Storage Account Type: StandardSSD_LRS

Caching: ReadWrite

8. Define a new variable block called windows_name with the following two key/pairs:

<HumberID>-w-vm1 = "Standard_B1s"

<HumberID>-w-vm2 = "Standard_B1ms"

9. Remove VM size variable block.

SCREENSHOT of vmwindows-vars.tf showing few lines before and after windows_name

Part 3: Update vmwindows-main.tf file:

10. Define a new resource block for Windows availability set with 2 fault domains and 5 update domains. Use the azure_rm_availability_set resource type.
11. Update the VM resource block to reflect Windows VM. Use the azure_rm_windows_virtual_machine resource type.
12. Modify the Windows VM resource block to set the for_each value to the variable created above. Update the VM name, network interface ID, computer name, and OS disk name accordingly. Add a sub-block called winrm_listener with protocol Http.
13. Modify the NIC resource block to set the for_each value to the variable created above. Update the NIC name, IP configuration name, and public IP address ID accordingly.
14. Update the NIC resource block to use subnet2/NSG2 as it contains a rule that allows RDP access (port 3389) into Windows.
15. Modify the Public IP resource block to set the for_each value to the variable created above. Update the Public IP name accordingly.

SCREENSHOT of vmwindows-main.tf

Part 4: Update outputs.tf file:

Do not change or remove any existing output blocks from this file.

16. Define a new output block to display Windows availability set name
17. Modify the output values to reflect multiple resource creation for VMs, DNS labels, private IP addresses, and public IP addresses. Leave values for single resources (vnet, subnet, etc.) intact.

SCREENSHOT of outputs.tf

Part 5: Validate configuration:

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

Part 6: Run simulation:

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

Part 7: Deploy infrastructure:

25. Deploy the infrastructure and monitor progress (**terraform apply**)
26. View Terraform log file and review details
27. View the content of terraform.tfstate file (**tail -20 terraform.tfstate**)

Part 8: Get information from Terraform state:

28. View and analyze state information (**terraform state list | nl**)
29. Display the output values (**terraform output**)

**SCREENSHOT
SCREENSHOT**

Part 9: Confirm resource creation in Azure:

30. Log in to the Azure Portal. Navigate to the resource group and confirm the quantity of each resource. Verify tags on the resources as well.

SCREENSHOT

Part 10: Destroy all resources and verify:

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

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

At the end of lab04s2, you will have 8 Terraform files: providers.tf, network-vars.tf, network-main.tf, vmlinux-vars.tf, vmlinux-main.tf, vmwindows-vars.tf, vmwindows-main.tf, and outputs.tf