Week 2

Objectives

At the end of week2, you should be

- Use attributes, variables, locals and outputs in terraform
- Use different datatypes in terraform
- Use conditionals and loops in terraform
- Use functions In terraform
- Use terraform format and validate commands
- Use taints on manually modified resources

Topics

- Attributes, outputs, locals and variables
- Different variable assignment approaches
- Datatypes in terraform
- Datasources
- Conditionals and loops
- Functions in terraform
- TF format and validate
- Taints and Graphs
- Splat expressions

Suggested Learning Materials

Go through the below materials/contents in the order they are listed

Udemy

- Udemy Course – Section 4 - Read, Generate and Modify Configurations Tainting on version v0.15.2 and above – replace flag
- Tainting on version v0.15.2 and above – replace flag

TF Documentation

- Variables and outputs
- Expressions
- Functions
- Datasources

Practice Lab Questions

Note: Please create a repo and check-in all your work so that they can be tracked by the facilitators

Azure Users

1. Create a linux virtual machine

- Provider reference – Linux virtual machine
- Use variables to store the azurerm_linux_virtual_machine resource fields – size and admin_username
- Use locals to store the azurerm_virtual_network resource fields – location and resource_group_name
- Use outputs to get the value of the azurerm_linux_virtual_machine attributes – id, private_ip_addresses, public_ip_address

2. Create 3 identical virtual machines using the count variable(same config as in challenge1)

- The name of the virtual machine should have the count.index value appended as a suffix. (example – myvm-1, myvm-2 and so on)

3. Create different number of vms based on environment(same config as in challenge1)

- Create a map variable numberOfInstances to store the number of machines needed for each environment. Example (dev=1, test=3)
- Create a variable environment to store the name of the environment to be used
- Based on the value of the variable "environment", the instance count should change (per example above, if you pass dev as the value to env, 1 vm should be created; if it is test, 3 should be created).

4. Use terraform graph command to create a graph for the challenge 3 and visualize it.

AWS Users

1. Create an EC2 instance

- Provider reference – AWS instance
- Use variables to store the aws_instance resource fields – ami and instance_type
- Use locals to store the aws_subnet resource fields – availability_zone
- Use outputs to get the value of the aws_instance attributes – arn, public_ip

2. Create 3 identical ec2 using the count variable(same config as in challenge1)

- The name of the ec2 instance should have the count.index value appended as a suffix. (example – myvm-1, myvm-2 and so on)

3. Create different number of ec2 based on environment(same config as in challenge1)

- Create a map variable numberOfInstances to store the number of machines needed for each environment. Example (dev=1, test=3)
- Create a variable environment to store the name of the environment to be used
- Based on the value of the variable "environment", the instance count should change (per example above, if you pass dev as the value to env, 1 ec2 instance should be created; if it is test, 3 should be created).

4. Use terraform graph command to create a graph for the challenge 3 and visualize it.

Useful Links

- Udemy Course - HashiCorp Certified: Terraform Associate 2023
- OReilly Book - Terraform: Up and Running, 3rd Edition