

**Lab: HashiCorp Configuration Language**

Terraform is written in HCL (HashiCorp Configuration Language) and is designed to be both human and machine readable. HCL is built using code configuration blocks which typically follow the following syntax:

```
1 # Template
2 <BLOCK TYPE> "<BLOCK LABEL>" "<BLOCK LABEL>" {
3   # Block body
4   <IDENTIFIER> = <EXPRESSION> # Argument
5 }
6
7 # AWS EC2 Example
8 resource "aws_instance" "web_server" { # BLOCK
9   ami           = "ami-04d29b6f966df1537" # Argument
10  instance_type = var.instance_type # Argument with value as expression (Variable value r
11 }
```

Terraform Code Configuration block types include:

- Terraform Settings Block
- Terraform Provider Block
- Terraform Resource Block
- Terraform Data Block
- Terraform Input Variables Block
- Terraform Local Variables Block
- Terraform Output Values Block
- Terraform Modules Block

We will be utilizing Terraform Provider, Terraform Resource, Data and Input Variables Blocks in this lab. This course will go through each of these configuration blocks in more detail throughout the course.

- Task 1: Connect to the Student Workstation
- Task 2: Verify Terraform installation
- Task 3: Update Terraform Configuration to include EC2 instance
- Task 4: Use the Terraform CLI to Get Help
- Task 5: Apply your Configuration
- Task 6: Verify EC2 Server in AWS Management Console



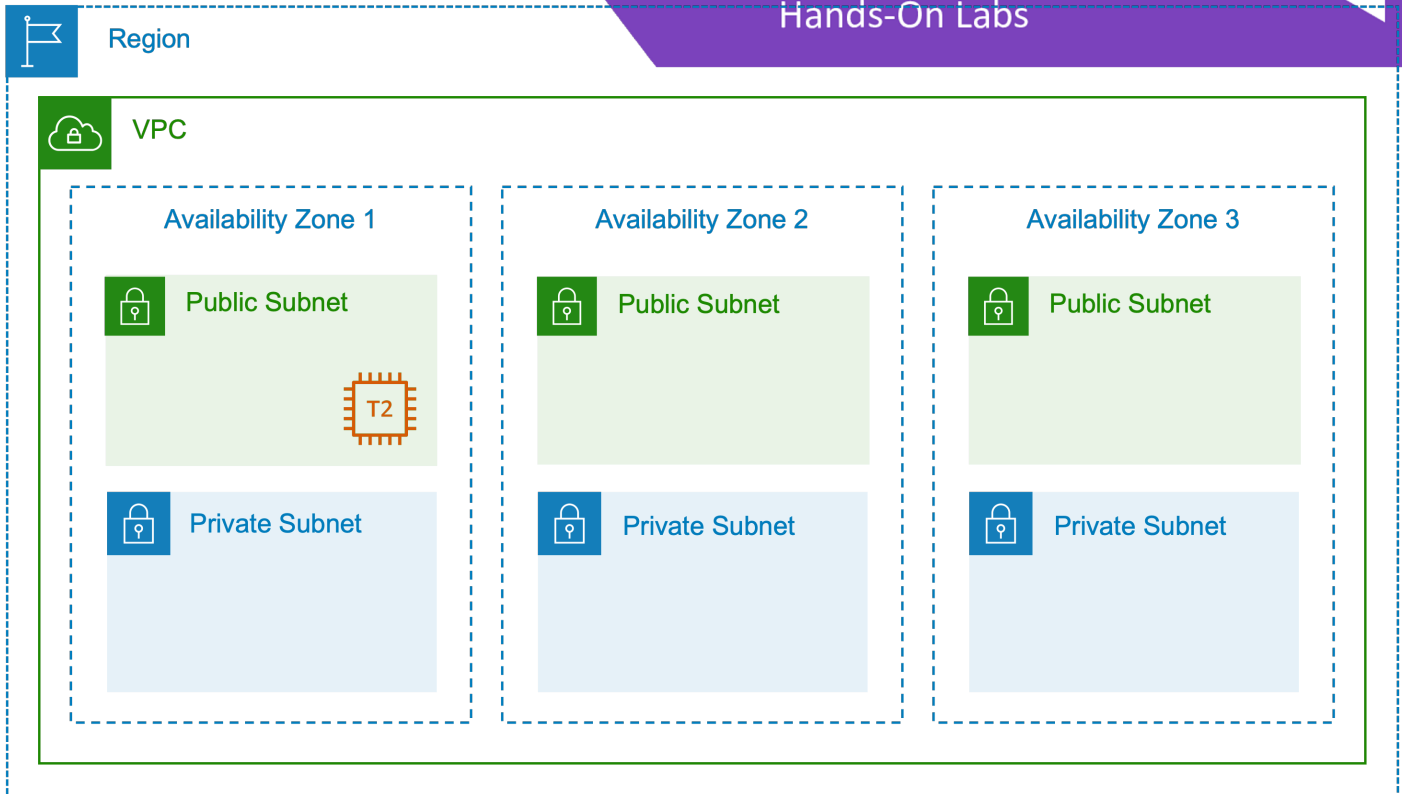


Figure 1: AWS Application Infrastructure Buildout

Task 1: Connect to the Student Workstation

In the previous lab, you learned how to connect to your workstation with either VSCode, SSH, or the web-based client.

One you've connected, make sure you've navigated to the `/workstation/terraform` directory. This is where we'll do all of our work for this training.

Task 2: Verify Terraform installation

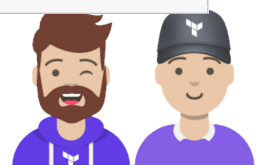
Step 1.2.1

Run the following command to check the Terraform version:

```
1 terraform -version
```

You should see:

```
1 Terraform v1.0.8
```





Task 3: Update Terraform Configuration to include EC2 instance Hands-On Labs

Step 1.3.1

In the /workstation/terraform directory, edit the file titled `main.tf` to create an AWS EC2 instance within one of the our public subnets.

Your final `main.tf` file should look similar to this with different values:

```
1 provider "aws" {
2   access_key = "<YOUR_ACCESSKEY>"
3   secret_key = "<YOUR_SECRETKEY>"
4   region     = "<REGION>"
5 }
6
7 resource "aws_instance" "web" {
8   ami           = "<AMI>"
9   instance_type = "t2.micro"
10
11   subnet_id          = "<SUBNET>"
12   vpc_security_group_ids = ["<SECURITY_GROUP>"]
13
14   tags = {
15     "Identity" = "<IDENTITY>"
16   }
17 }
```

Don't forget to save the file before moving on!

Task 4: Use the Terraform CLI to Get Help

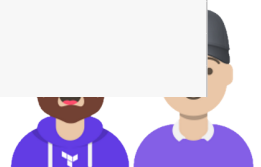
Step 1.4.1

Execute the following command to display available commands:

```
1 terraform -help
```

```
1 Usage: terraform [-version] [-help] <command> [args]
2
3 The available commands for execution are listed below.
4 The most common, useful commands are shown first, followed by
5 less common or more advanced commands. If you're just getting
6 started with Terraform, stick with the common commands. For the
7 other commands, please read the help and docs before usage.
8
9 Common commands:
10   apply           Builds or changes infrastructure
11   console         Interactive console for Terraform interpolations
12   destroy         Destroy Terraform-managed infrastructure
13   env            Workspace management
14   fmt            Rewrites config files to canonical format
```

Created by Gabe Maentz and Bryan Krausen





15	<code>get</code>	Download and install modules for the configuration
16	<code>graph</code>	Create a visual graph of Terraform resources
17	<code>import</code>	Import existing infrastructure into Terraform
18	<code>init</code>	Initialize a Terraform working directory
19	<code>output</code>	Read an output from a state file
20	<code>plan</code>	Generate and show an execution plan
21		
22	<code>...</code>	

- (full output truncated for sake of brevity in this guide)

Or, you can use short-hand:

```
1 terraform -h
```

Step 1.4.2

Navigate to the Terraform directory and initialize Terraform

```
1 cd /workstation/terraform
```

```
1 terraform init
```

```
1 Initializing provider plugins...
2 ...
3
4 Terraform has been successfully initialized!
```

Step 1.4.3

Get help on the `plan` command and then run it:

```
1 terraform -h plan
```

```
1 terraform plan
```

Task 5: Apply your Configuration

Step 1.5.1

Run the `terraform apply` command to generate real resources in AWS

```
1 terraform apply
```

You will be prompted to confirm the changes before they're applied. Respond with `yes`.





Task 6: Verify EC2 Server in AWS Management Console Hands-On Labs

Login to AWS Management Console -> Services -> EC2 to verify newly created EC2 instance

The screenshot displays the AWS Management Console interface. On the left, the navigation menu is visible with categories like 'Instances', 'Images', 'Elastic Block Store', 'Network & Security', and 'Load Balancing'. The main content area shows the 'Instances (1/1)' page. A table lists the instance 'Terraform HOL Server' with ID 'i-040805d3f31a4ea4f', state 'Running', type 't2.micro', and status '2/2 checks passed'. Below the table, the 'Instance: i-040805d3f31a4ea4f (Terraform HOL Server)' details are expanded, showing tabs for 'Details', 'Security', 'Networking', 'Storage', 'Status checks', 'Monitoring', and 'Tags'. The 'Details' tab is active, displaying a grid of instance information:

Instance summary		
Instance ID i-040805d3f31a4ea4f (Terraform HOL Server)	Public IPv4 address 34.201.205.45 open address	Private IPv4 addresses 10.0.101.171
IPv6 address -	Instance state Running	Public IPv4 DNS -
Private IPv4 DNS ip-10-0-101-171.ec2.internal	Instance type t2.micro	Elastic IP addresses -
VPC ID vpc-029c8f08991bad7fe (demo_vpc)	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more	IAM Role -
Subnet ID subnet-0698af9fe82d2cf7e (public_subnet_1)		
Instance details		
Platform Ubuntu (Inferred)	AMI ID ami-036490d46656c4818	Monitoring disabled
Platform details Linux/UNIX	AMI name ubuntu/images/hvm-ssd/ubuntu-focal-20.04-amd64-server-20211001	Termination protection Disabled

Figure 2: AWS EC2 Server

References

Terraform Configuration Terraform Configuraiton Syntax

