# Terraform 101

## Useful Documentations

* <https://developer.hashicorp.com/terraform/language>
  + Documentation for Terraform HCL
    - Understand the different code blocks (e.g. provider, resource, data, variables, output)
* <https://registry.terraform.io/providers/hashicorp/aws/latest/docs>
  + This page shows how the provider can be configured
  + On the left sidebar, navigate the different AWS resources that can be provisioned and managed.

## Setup

* AWS CLI installed
  + Default credentials configured
* Terraform CLI installed

## 

## Activities

### Activity 0: EC2 default VPC

In an empty directory, create the file `main.tf`. All code shall be written within this file.

#### Pre-requisites

1. Create provider block
   1. Use the latest AWS provider
   2. Specify `ap-southeast-1` as the region
2. Run `terraform init`

#### Steps

1. Create an EC2 Instance resource block
   1. Use the latest amazon linux 2 ami image
   2. Tag the resource with a name
2. Plan and apply
   1. Run `terraform fmt`
   2. Run `terraform validate`
   3. Run `terraform plan`
   4. Run `terraform apply`
3. Validate
   1. EC2 instance is created with specified image and name

#### Prompt

* Following the `terraform init`, what files appeared in the directory? How much space did it take?
* What changed in your code after `terraform fmt`?
* From `terraform plan` against your configuration files, what resources do you expect to be created/updated/deleted?
* Where does terraform store and maintain the references between the desired resource (from your configuration code) and the actual resource (in the cloud)?
* How is Terraform being permissioned to interact with AWS? Where are the credentials coming from?

### Activity A: EC2 in non-default VPC

#### Pre-requisites

1. Decide on a non-default VPC. Note the VPC id.
2. Decide on a public subnet of the same VPC. Note the Subnet id.

#### Steps

1. (Re)create SSH keys via EC2 console
2. Create Security Group resource block
   1. Place within the non-default VPC
   2. Allows 22/tcp inbound from 0.0.0.0/0
3. Create an EC2 Instance resource block
   1. Place within the public subnet
   2. Attach the SSH key from before
   3. Attach the Security group from before
4. Plan and apply
5. Validate
   1. From your local machine, ssh into the machine using your private key through the public IP

#### Prompt

* Manually via the console, change the tagged name of your resources
  + Can this be detected using the ‘terraform plan’?
  + Can this be corrected using the `terraform apply` command?
* Run the terraform command to destroy all provisioned resources

#### Hints

* [Terraform AWS Provider: Instance](https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/instance)
* [Terraform AWS Provider: Security Groups](https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/security_group)

### Activity B: S3 Bucket

#### Steps

1. Create an S3 bucket with versioning enabled
2. (Via S3 Console) Add a tag to the bucket
3. Plan and apply
4. Validate

### Activity C: Refactoring

#### Steps

1. Create `name\_prefix` variable block
   1. Set your name as the default value
   2. Interpolate this value
2. Create `environment` variable block
   1. Set a default value (e.g. ‘development’)
   2. Use this as a tag value
3. Create an output block
   1. Set the instance public ip as the value
4. Decompose the single `main.tf` file into multiple files
   1. provider.tf
   2. main.tf
   3. securitygroups.tf
   4. variables.tf
   5. outputs.tf
5. Plan and apply

#### Prompt

* What is the terraform command to show only the outputs?

#### Hints

* [Terraform Docs: Variables and Outputs](https://developer.hashicorp.com/terraform/language/values)
* [Terraform Docs: String interpolation](https://developer.hashicorp.com/terraform/language/expressions/strings#interpolation)
* [Best Practices](https://www.terraform-best-practices.com/code-structure)

### 

### Activity D: Latest AMI Image

1. Use a data block to retrieve latest amazon linux 2 ami image
2. Update the existing ec2 instance to use the referenced image
3. Plan and apply
4. Validate

#### Hints

* [Terraform Docs: Data Sources](https://developer.hashicorp.com/terraform/language/data-sources)