Lab: Terraform Provider Installation

Terraform relies on plugins called “providers” to interact with remote systems and expand functionality. Terraform configurations must declare which providers they require so that Terraform can install and use them. This is performed within a Terraform configuration block.

- Task 1: Check Terraform version
- Task 2: How to require specific versions of both Terraform and providers.

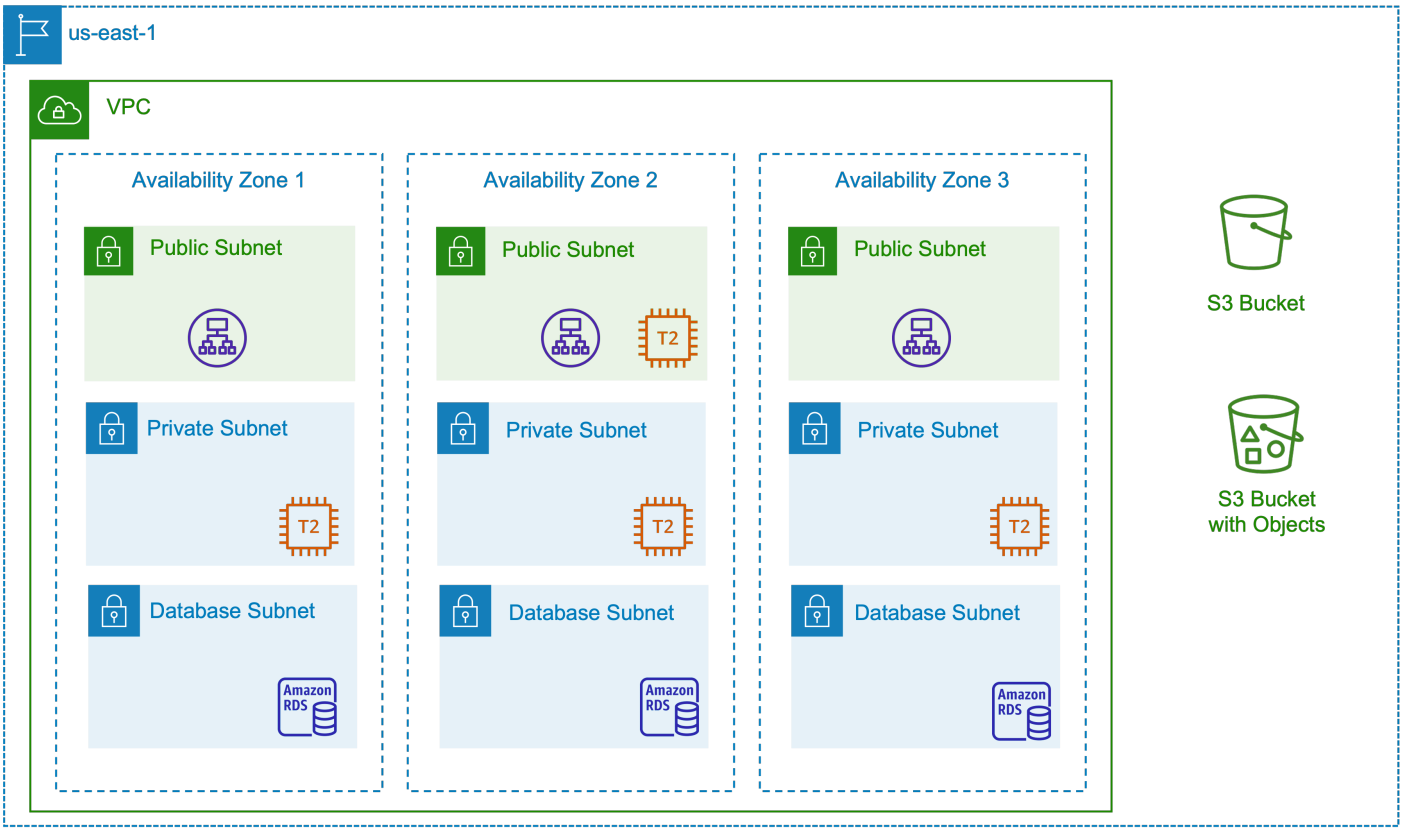

Figure 1: AWS Application Infrastructure Buildout

Task 1: Check Terraform version

Run the following command to check the Terraform version:

```
terraform -version
```

You should see:

```
Terraform v1.0.8
```

Task 2: Require specific versions of Terraform

Create a file titled `terraform.tf` to define a minimum version of Terraform to be used.

`terraform.tf`

```
terraform {
  required_version = ">= 0.15.0"
}
```

This informs Terraform that it must be at least of version `0.15.0` to run the code. If Terraform is an earlier version it will throw an error.

Update the `required_version` line to `"=1.0.0"` and run `terraform init`.

What happened when you changed the version requirement?

> Note: Make sure the required version is set back to `">=0.15.0"` and you > have run `terraform init` again before proceeding with this lab.

At this point you have now stated that Terraform can only run this code if its own version is `0.15.0` or greater. Terraform and providers can both have their versions set through terraform code. In the next step you will use the AWS provider, and set the provider version in a way that is very similar to how you did for Terraform.

> Note: You can always find the latest version of a provider on its > registry page at https://registry.terraform.io.

To begin you need to let Terraform know to use the provider through a `required_providers` block in the `provider.tf` file as seen below.

`#terraform.tf`

```
terraform {
  required_version = ">= 0.15.0"
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "~> 3.0"
    }
  }
}
```

Now that you have told Terraform to use this new provider you will have to run the `init` command. This will cause Terraform to notice the change and download the correct version if it was not already downloaded.

```
terraform init
```

By default Terraform will always pull the latest provider if no version is set. However setting a version provides a way to ensure your Terraform code remains working in the event a newer version introduces a change that would not work with your existing code. To have more strict controls over the version you may want to require a specific version

(e.g. `required_version = "= 0.15.0"`) or use the ~>operator to only allow the right-most version number to increment.

To check the terraform version and provider version installed via `terraform init` run the `terraform version` command.