Terraform: Understand Terraform Basics

➤ **Section 3 : Understand Terraform Basics**

   ➤ Handle Terraform and Provider Installation and Versioning

   ➤ Describe Plugin based Architecture

   ➤ Demonstrate using Multiple Providers

➤ **Handle Terraform and Provider Installation and Versioning :**

➤ Terraform configurations must declare which providers they require.

➤ Provider configurations belong in the root module of a Terraform configuration.

```
provider "google" {
 project = "acme-app"
 region  = "us-central1"
}
```

➤ **Handle Terraform and Provider Installation and Versioning :**

➤ **alias**: Multiple Provider Configurations

➤ Multiple configurations can be defined for the same provider.

➤ Primary reason of alias is to support multiple regions for a cloud platform.

```
# it as the default, and it can be referenced as `aws`.
provider "aws" {
  region = "us-east-1"
}

# reference this as `aws.west`.
provider "aws" {
  alias  = "west"
  region = "us-west-2"
}
```

➤ **Handle Terraform and Provider Installation and Versioning :**

➤ Selecting Alternate Provider Configurations

```
resource "aws_instance" "foo" {
  provider = aws.west
  # ...
}
```

```
module "aws_vpc" {
  source = "./aws_vpc"
  providers = {
    aws = aws.west
  }
}
```

➤ **Describe plug-in based architecture :**

➤ Terraform uses a plugin-based architecture to support hundreds of infrastructure and service providers.

➤ Initializing a configuration directory downloads and installs providers used in the configuration.
terraform init

➤ Format your configuration.
terraform fmt

➤ Validate your configuration.
terraform validate
Success! The configuration is valid.

See you in next lecture ...