

```

terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "~> 4.16"
    }
  }

  required_version = ">= 1.2.0"
}

provider "aws" {
  region = "us-west-2"
}

resource "aws_instance" "app_server" {
  ami          = "ami-830c94e3"
  instance_type = "t2.micro"

  tags = {
    Name = "ExampleAppServerInstance"
  }
}

```

#optional

The `terraform {}` block contains Terraform settings, including the required providers Terraform will use to provision your infrastructure. For each provider, the `source` attribute defines an optional hostname, a namespace, and the provider type. Terraform installs providers from the [Terraform Registry](#) by default.

Note - we can start from the provider

Provider

The provider block configures the specified provider, in this case `aws`. A provider is a plugin that Terraform uses to create and manage your resources.

You can use multiple provider blocks in your Terraform configuration to manage resources from different providers. You can even use different providers together

Resources

Use resource blocks to define components of your infrastructure. A resource might be a physical or virtual component such as an EC2 instance, or it can be a logical resource such as a Heroku application.

Resource blocks have two strings before the block: the resource type and the resource name.

The prefix of the type maps to the name of the provider. In the example configuration, Terraform manages the `aws_instance` resource with the `aws` provider. Together, the resource type and resource name form a unique ID for the resource.

Initialize the directory

When you create a new configuration — or check out an existing configuration from version control — you need to initialize the directory with `terraform init`.

Initializing a configuration directory downloads and installs the providers defined in the configuration, which in this case is the `aws` provider.