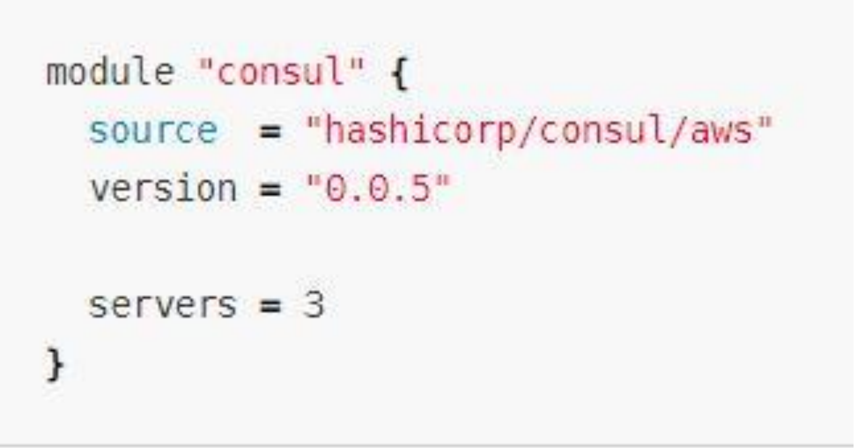
Terraform Modules

Terraform Modules - Introduction

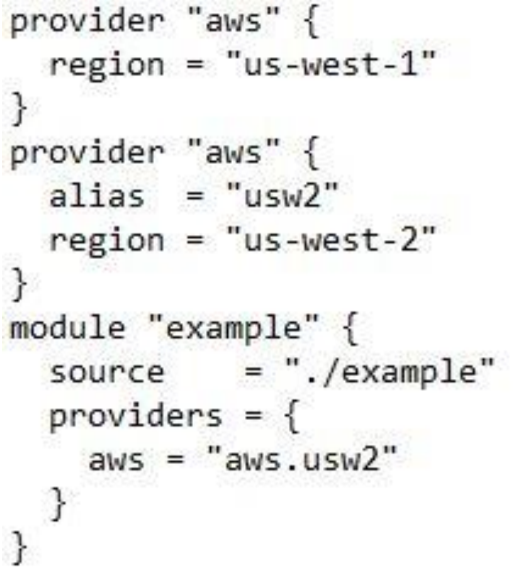
- A module is a container for multiple resources that are used together.
- Every Terraform configuration has at least one module, known as its root module, which consists of the resources defined in the .tf files in the main working directory.
- A module can call other modules, which lets you include the child module's resources into the configuration in a concise way.
- Modules can also be called multiple times, either within the same configuration or in separate configurations, allowing resource configurations to be packaged and re-used.

Terraform Module - Versions

- When using modules installed from a module registry, we recommend explicitly constraining the acceptable version numbers to avoid unexpected or unwanted changes.
- Version constraints are supported only for modules installed from a module registry, such as the public Terraform Registry or Terraform Cloud's private module registry.



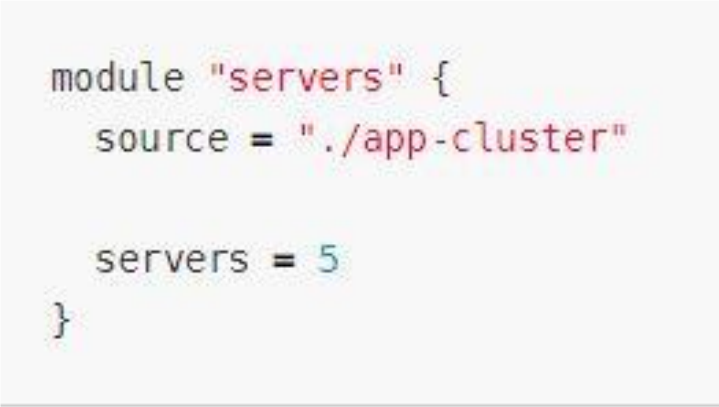
```
module "consul" {
  source  = "hashicorp/consul/aws"
  version = "0.0.5"

  servers = 3
}
```



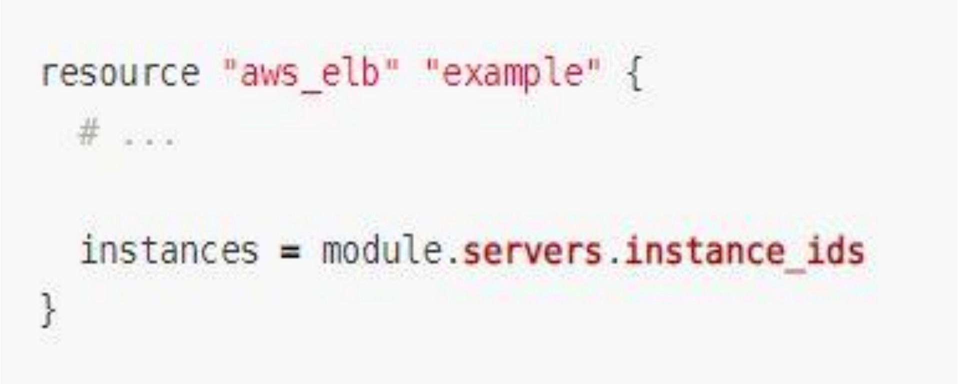
```
provider "aws" {
  region = "us-west-1"
}
provider "aws" {
  alias  = "usw2"
  region = "us-west-2"
}
module "example" {
  source    = "./example"
  providers = {
    aws = "aws.usw2"
  }
}
```

Terraform - Calling a Child Module

- To call a module means to include the contents of that module into the configuration with specific values for its input variables.
- Modules are called from within other modules using module blocks
- The resources defined in a module are encapsulated, so the calling module cannot access their attributes directly.
- However, the child module can declare output values to selectively export certain values to be accessed by the calling module.



```
module "servers" {
  source = "./app-cluster"

  servers = 5
}
```



```
resource "aws_elb" "example" {
  # ...

  instances = module.servers.instance_ids
}
```