Terraform Modules

Terraform Modules

What is a Module?

- A **module** is a container for multiple Terraform resources used together
- It represents a **logical piece of infrastructure**, not individual resources

Root Module

- All `.tf` files in the working directory form the **root module**
- Terraform executes this module when running `terraform plan` or `terraform apply`

Child Modules

- The root module can call **child modules**
- Child modules receive **input variables** and return **output values**

Why Terraform Modules Matter?

- Improve **reusability** and **consistency**
- Reduce **code duplication**
- Allow infrastructure to be described in terms of **architecture** (VPC, App, DB)

How to Write a Terraform Module?

1. Module File Structure

```
modules/
└── vpc/
    ├── main.tf        # Resources
    ├── variables.tf   # Input variables
    ├── outputs.tf     # Output values
```

2. main.tf (Resources)

- Defines the infrastructure resources
- Example: VPC, Subnets, Security Groups

3. variables.tf (Inputs)

- Declares configurable values
- Makes the module reusable
- Example: CIDR block, region, environment

4. outputs.tf (Outputs)

- Exposes values to other modules or root module
- Example: VPC ID, Subnet IDs

How to use a Terraform Module?

1. Using a Local Module

```
module "network" {

  source = "./modules/vpc"

  cidr_block = "10.0.0.0/16"

  env        = "dev"

}
```

- `source` points to a **local directory**
- Common for **learning, labs, and internal projects**
- Easy to modify and debug

How to use a Terraform Module?

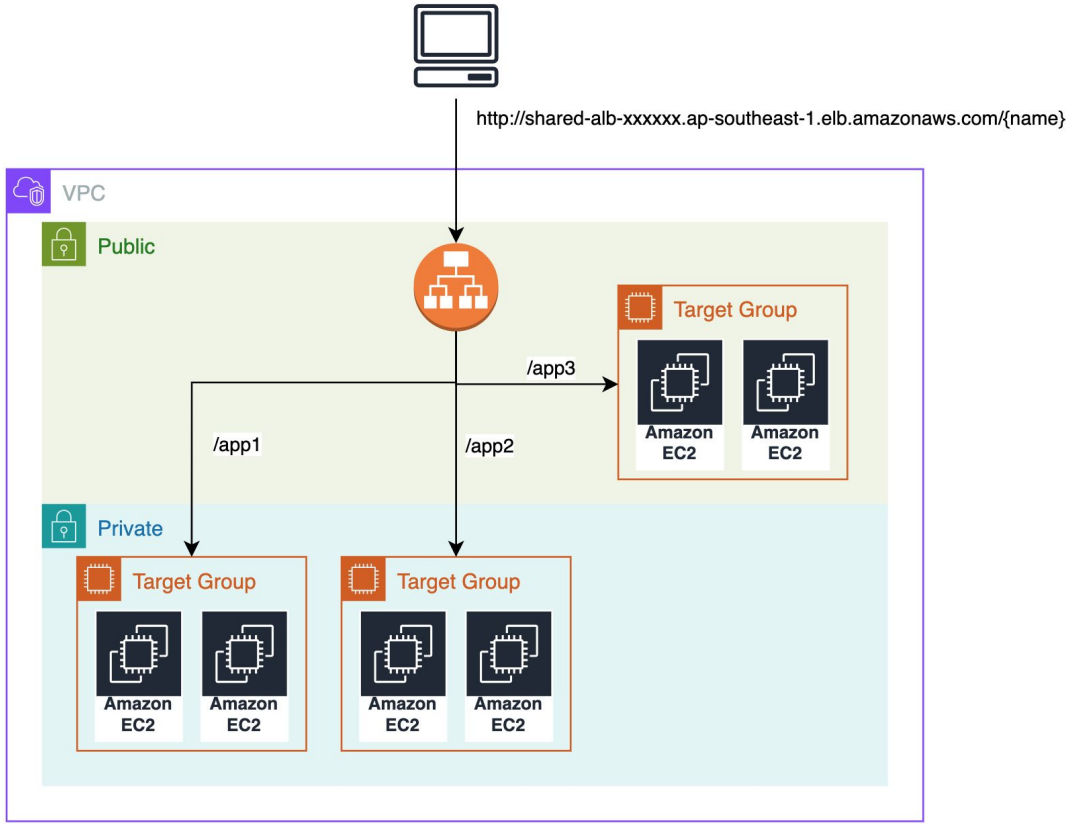
2. Using a Remote Module

```
module "network" {

  source = "terraform-aws-modules/vpc/aws"

  version = "~> 5.0"

  cidr_block = "10.0.0.0/16"

  env        = "dev"

}
```

- Module stored in **Terraform Registry / Git repository**
- `version` ensures **stability**
- Best for **production and shared teams**

Activity

Create a Web App module using Terraform


http://shared-alb-xxxxxx.ap-southeast-1.elb.amazonaws.com/{name}
VPC
Public
Target Group
Amazon EC2
Amazon EC2
/app3
/app1
/app2
Private
Target Group
Amazon EC2
Amazon EC2
Target Group
Amazon EC2
Amazon EC2

Guidance

Terraform Module resources

There are the resources you would create using Terraform within your module.

- EC2 instance
- Security groups
- Target groups
    - Attachment of EC2s to target groups
- Load balancer listener rules
-

Shared Environment

These resources should create out of the terraform module(Prerequisites).

- VPC
- Application load balancer
    - Default listener rule on port 80

Solution

Code Repo: https://github.com/chathra222/sltc-lab-3