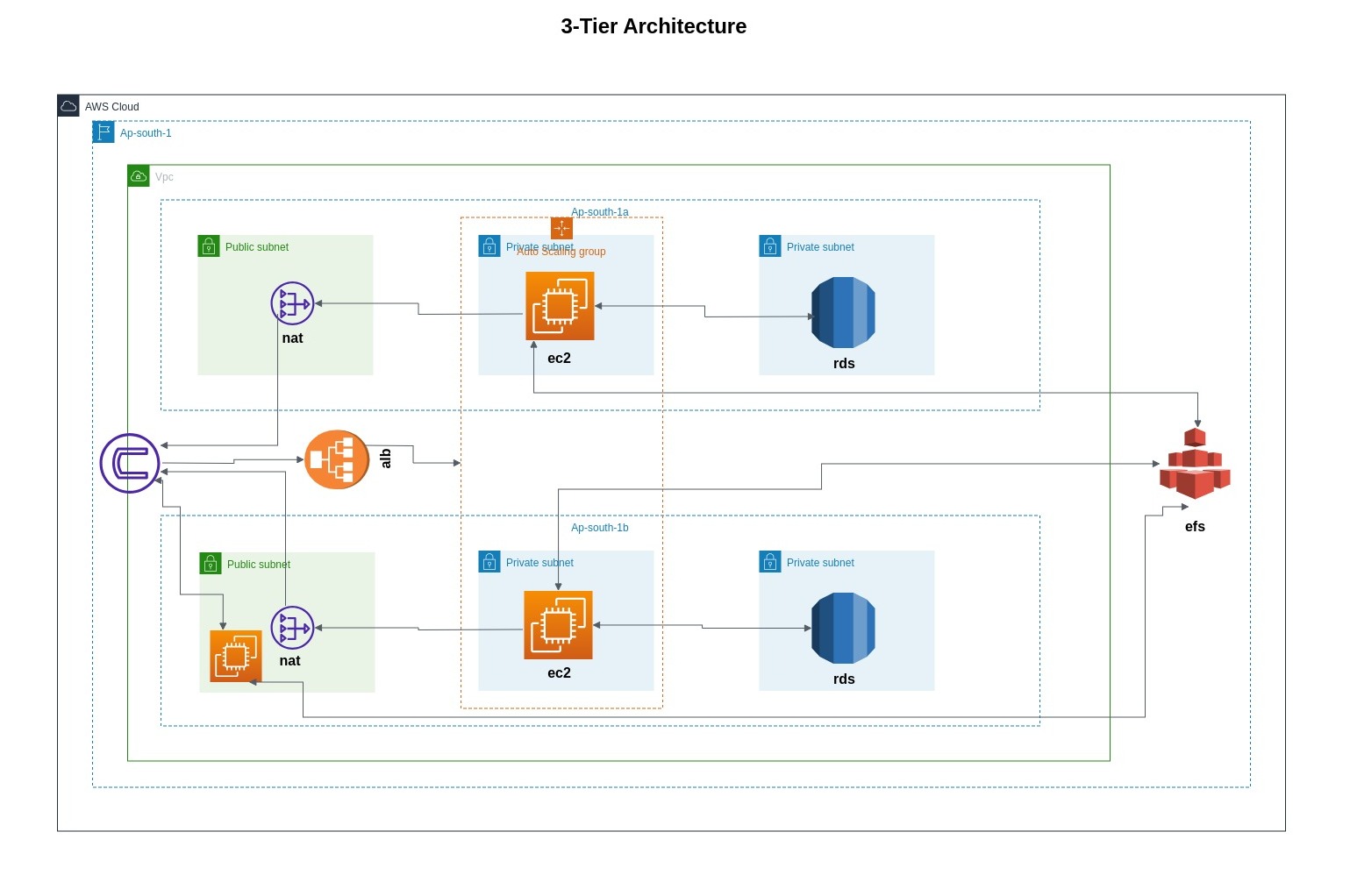
**Wordpress application deployment using Terraform**

**Architecture Diagram**

The 3-tier architecture used in this project consists of a web layer, an application layer, and a database layer. The infrastructure is provisioned using Terraform and consists of several components, including a VPC, subnets, security groups, EC2 instances, an RDS instance, an EFS file system, and an ALB. These components are configured to work together to provide a scalable and highly available infrastructure for hosting the WordPress application.



**AWS Services**

This project uses several AWS services to provision the infrastructure for hosting a WordPress application using a 3-tier architecture. The following AWS services are used:

* **Amazon VPC**: Used to create a virtual private cloud (VPC) for the infrastructure.
* **Amazon EC2**: Used to launch EC2 instances for the web and application layers.
* **Amazon RDS**: Used to create an RDS instance for the database layer.
* **Amazon EFS** Used to create an Elastic File System (EFS) for storing WordPress files.
* **Amazon ALB** Used to create an Application Load Balancer (ALB) for distributing traffic between the web and application layers.
* **Amazon CloudWatch** Used to monitor the infrastructure and collect logs and metrics.

## Usage

1. Navigate to the project directory: cd wp-3tier-aws/src/infrastructure/

Main.tf

module "vpc\_module" {

  source = "./modules/vpc"

}

module "subnet\_module" {

  source   = "./modules/subnets"

  vpc\_id   = module.vpc\_module.vpc\_id

  zone\_one = var.zone\_one

  zone\_two = var.zone\_two

}

module "internet\_gateway\_module" {

  source = "./modules/internet-gateway"

  vpc\_id = module.vpc\_module.vpc\_id

}

module "nat\_gateway\_module" {

  source     = "./modules/nat-gateway"

  subnet\_ids = module.subnet\_module.subnet\_ids

}

module "route\_module" {

  source              = "./modules/route-table"

  subnet\_ids          = module.subnet\_module.subnet\_ids

  nat\_ids             = module.nat\_gateway\_module.nat\_ids

  vpc\_id              = module.vpc\_module.vpc\_id

  internet\_gateway\_id = module.internet\_gateway\_module.igw\_id

}

module "security\_group\_module" {

  source = "./modules/security-groups"

  vpc\_id = module.vpc\_module.vpc\_id

  my\_ip  = var.my\_ip

}

module "s3\_bucket" {

  source = "terraform-aws-modules/s3-bucket/aws"

  bucket = "my-s3-bucket"

  acl    = "private"

  control\_object\_ownership = true

  object\_ownership         = "ObjectWriter"

  versioning = {

    enabled = true

  }

}

module "database\_module" {

  source     = "./modules/database"

  subnet\_ids = module.subnet\_module.subnet\_ids

  db\_sg\_id   = module.security\_group\_module.db\_sg\_id

}

module "efs\_module" {

  source     = "./modules/efs"

  efs\_sg\_id  = module.security\_group\_module.efs\_sg\_id

  subnet\_ids = module.subnet\_module.subnet\_ids

}

module "ec2\_module" {

  source = "./modules/ec2"

  efs\_sg\_id        = module.security\_group\_module.efs\_sg\_id

  ssh\_sg\_id        = module.security\_group\_module.ssh\_sg\_id

  lb\_sg\_id         = module.security\_group\_module.lb\_sg\_id

  wb\_sg\_id         = module.security\_group\_module.wb\_sg\_id

  db\_sg\_id         = module.security\_group\_module.db\_sg\_id

  subnet\_ids       = module.subnet\_module.subnet\_ids

  wordpress\_tg\_arn = module.alb\_module.wordpress\_tg\_arn

  efs\_dns\_name     = module.efs\_module.efs\_dns\_name

  db               = module.database\_module.db

}

module "alb\_module" {

  source          = "./modules/load-balancer"

  vpc\_id          = module.vpc\_module.vpc\_id

  lb\_sg\_id        = module.security\_group\_module.lb\_sg\_id

  subnet\_ids      = module.subnet\_module.subnet\_ids

  auto\_scaling\_id = module.ec2\_module.auto\_scaling\_id

}

module "cloud\_watch\_module" {

  source            = "./modules/cloud-watch"

  auto\_scaling\_name = module.ec2\_module.auto\_scaling\_name

}

Variable.tf

variable "region" {

  type = string

}

variable "config" {

  type    = string

  default = "~/.aws/config"

}

variable "credentials" {

  type    = string

  default = "~/.aws/credentials"

}

variable "my\_ip" {

  description = "ip address"

  type        = string

}

variable "zone\_one" {

  description = "region"

  type        = string

}

variable "zone\_two" {

  description = "region"

  type        = string

}

Provider.tf

terraform {

  required\_providers {

    aws = {

      source = "hashicorp/aws"

    }

  }

}

provider "aws" {

  region                   = var.region

  shared\_config\_files      = [var.config]

  shared\_credentials\_files = [var.credentials]

}

Terraform.tfvars

my\_ip  = "35.170.239.160"

region = "us-east-1"

zone\_one = "us-east-1a"

zone\_two = "us-east-1b"

1. Initialize Terraform: terraform init -var-file = “terraform.tfvars”
2. Create a terraform.tfvars file with your AWS credentials and other required variables.
3. Terraform plan -var-file = “terraform.tfvars”
4. Terraform apply -var-file= “terraform.tfvars”

Resources has been provisioned successfully

This terraform code supports multiple regions in the terraform.tfvars we can choose any region based on the input it will be provision .

**Output**

