Terraform Starter Module Kit

Production-ready project structure for AWS

Project Structure

```
terraform-project/
  environments/
    dev/
      main.tf          # Calls modules with dev values
      variables.tf     # Variable declarations
      terraform.tfvars # Dev-specific values
      backend.tf       # S3 state: dev/terraform.tfstate
    staging/
      main.tf / variables.tf / terraform.tfvars / backend.tf
    prod/
      main.tf / variables.tf / terraform.tfvars / backend.tf
  modules/
    vpc/
      main.tf          # VPC, subnets, NAT, IGW, routes
      variables.tf     # CIDR, AZs, tags
      outputs.tf       # vpc_id, subnet_ids, sg_ids
    ec2/
      main.tf          # Instance, ASG, launch template
      variables.tf     # AMI, type, key_name
      outputs.tf       # instance_id, public_ip
    rds/
      main.tf          # DB instance, subnet group, params
      variables.tf     # Engine, size, credentials
      outputs.tf       # endpoint, port
  .gitignore
  README.md
```

Backend Config (backend.tf)

```
terraform {
  backend "s3" {
    bucket         = "mycompany-terraform-state"
    key            = "dev/terraform.tfstate"
    region         = "ap-south-1"
    dynamodb_table = "terraform-locks"
    encrypt        = true
  }
}
```

Best Practices

- One module per resource group (VPC, compute, database)

- Remote state in S3 + DynamoDB lock - never local
- Separate state per environment - never share
- Pin provider versions with required_providers
- Tag everything: Name, Environment, Team, ManagedBy=terraform
- Run fmt + validate in CI before plan
- Review plan in PR before apply
- Never apply from local machine in production
- Use moved blocks when renaming resources
- Mark sensitive outputs: sensitive = true

.gitignore

```
.terraform/
*.tfstate
*.tfstate.backup
*.tfvars  # if contains secrets
.terraform.lock.hcl
```