**1. VPC (Virtual Private Cloud)**

**Purpose**: Creates a **highly available VPC** with public/private subnets, NAT gateways, and security groups.

**Terraform Module**:

module "vpc" {

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

version = ">=3.0"

name = "enterprise-vpc"

cidr = "10.0.0.0/16"

azs = ["us-east-1a", "us-east-1b"]

public\_subnets = ["10.0.1.0/24", "10.0.2.0/24"]

private\_subnets = ["10.0.3.0/24", "10.0.4.0/24"]

enable\_nat\_gateway = true

enable\_dns\_support = true

}

✅ **Used in enterprises for network segmentation, high availability, and security.**

**2. EC2 Auto Scaling Group (ASG)**

**Purpose**: Deploys and manages scalable **EC2 instances with Auto Scaling**.

**Terraform Module**:

module "asg" {

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

name = "web-asg"

min\_size = 2

max\_size = 10

desired\_capacity = 2

vpc\_zone\_identifier = module.vpc.private\_subnets

launch\_template = {

id = module.ec2\_lt.id

version = "$Latest"

}

}

✅ **Used in production to maintain highly available workloads.**

**3. ALB (Application Load Balancer)**

**Purpose**: Creates an **Application Load Balancer (ALB)** for routing traffic.

**Terraform Module**:

module "alb" {

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

name = "enterprise-alb"

load\_balancer\_type = "application"

vpc\_id = module.vpc.vpc\_id

subnets = module.vpc.public\_subnets

target\_groups = [

{

name = "web-tg"

backend\_protocol = "HTTP"

backend\_port = 80

target\_type = "instance"

}

]

}

✅ **Enterprises use ALBs for handling web traffic and ensuring fault tolerance.**

**4. RDS (Relational Database Service)**

**Purpose**: Deploys a **highly available AWS RDS database**.

**Terraform Module**:

module "rds" {

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

identifier = "prod-db"

engine = "mysql"

engine\_version = "8.0"

instance\_class = "db.t3.medium"

allocated\_storage = 100

db\_name = "mydb"

username = "admin"

password = "SuperSecret123"

multi\_az = true

publicly\_accessible = false

}

✅ **Critical for enterprise applications requiring managed databases.**

**5. S3 with IAM Access Control**

**Purpose**: Creates an **S3 bucket with proper IAM access control**.

**Terraform Module**:

module "s3\_bucket" {

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

bucket = "enterprise-data-bucket"

acl = "private"

versioning = {

enabled = true

}

lifecycle\_rule = [{

id = "expire\_old\_files"

enabled = true

expiration = { days = 365 }

}]

}

✅ **Used for storing logs, backups, and data lakes in enterprise environments.**

**6. AWS Lambda with CloudWatch Logs**

**Purpose**: Creates a **serverless Lambda function** with CloudWatch logging.

**Terraform Module**:

module "lambda" {

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

function\_name = "enterprise-lambda"

handler = "index.handler"

runtime = "python3.9"

source\_path = "./lambda\_code"

}

✅ **Enterprises use Lambda for event-driven automation and microservices.**

**7. EKS (Elastic Kubernetes Service)**

**Purpose**: Deploys a **scalable Kubernetes cluster** on AWS.

**Terraform Module**:

module "eks" {

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

cluster\_name = "enterprise-eks"

cluster\_version = "1.27"

vpc\_id = module.vpc.vpc\_id

subnet\_ids = module.vpc.private\_subnets

}

✅ **Used by enterprises to manage containerized applications.**

**8. CloudFront with S3 Origin**

**Purpose**: Deploys a **global CDN with S3 as an origin**.

**Terraform Module**:

module "cloudfront" {

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

distribution\_name = "enterprise-cdn"

origin = [{

domain\_name = module.s3\_bucket.bucket\_regional\_domain\_name

origin\_id = "s3-origin"

}]

}

✅ **Used for accelerating website performance globally.**

**9. IAM Role & Policies**

**Purpose**: Creates **secure IAM roles and policies** for AWS services.

**Terraform Module**:

module "iam\_role" {

source = "terraform-aws-modules/iam/aws//modules/iam-assumable-role"

role\_name = "enterprise-role"

trusted\_role\_services = ["ec2.amazonaws.com"]

custom\_role\_policy\_arns = ["arn:aws:iam::aws:policy/AmazonS3FullAccess"]

}

✅ **Essential for fine-grained access control in enterprises.**

**10. AWS Backup (Automated Snapshots)**

**Purpose**: Configures **automated backups** for AWS services.

**Terraform Module**:

module "backup" {

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

plan\_name = "enterprise-backup-plan"

rule = [{

rule\_name = "daily-backup"

target\_vault\_name = "default"

schedule = "cron(0 12 \* \* ? \*)"

}]

}

✅ **Used in enterprises for compliance and disaster recovery.**