**Usecase**

1. A VPC with subnets across 2 availability zones.  
2. Two EC2 instances (web servers) in an Auto Scaling Group with a load balancer distributing traffic between them.  
3. Security groups to allow HTTP traffic from the internet to the load balancer and SSH traffic to the web servers only from a specific CIDR block.  
4. Use Terraform modules to encapsulate reusable components for VPC, Auto Scaling Group, and Security Groups.  
5. The EC2 instances should have a provisioner to install NGINX on each instance after it's launched.  
6. Use a remote backend (such as S3) to store the Terraform state securely.  
7. Fetch an existing AMI for the EC2 instances dynamically using a data source.  
8. Ensure the instance type of the EC2 instances can be dynamically selected based on an environment variable (like dev, prod).  
9. Set up an output that will return the DNS name of the load balancer.

---------------------------------------------------------------------------------------------------------------------------------------

**Install Terraform in Local (Windows)**

* Browse – <https://developer.hashicorp.com/terraform/downloads> > Windows > Download
* Extract zip file to a folder (C:\Terraform).
* System Properties > Environment Variables > System Variables > Path > Add path – C:\Terraform
* Verify

$ terraform -version

---------------------------------------------------------------------------------------------------------------------------------------

**Configure AWS CLI in local**

* Install AWS CLI & verify

$ aws --version

* Create user

IAM > Users > Create user

Username – admin > Check – Provide access to AWS Console > Choose – I want to create IAM user

* Generate access key

IAM > User > username (admin) > Create access key > Select – Command Line Interface > Next

* Create keypair

EC2 > Key pairs > Create key pair

Name – terraform-key > Key pair type – RSA > Private key file format – .pem

* Configure AWS

$ aws configure

AWS Access Key ID [\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*AGRT]: <access-key>

AWS Secret Access Key [\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*2s5C]: <secret-access-key>

Default region name [d]:

Default output format [d]:

Verify

$ aws sts get-caller-identity

**Create S3 bucket & Dynamo DB table for Remote Backend**

* Create S3 bucket

$ aws s3api create-bucket \

  --bucket my-terraform-state-bucket \

  --region us-east-1 \

--create-bucket-configuration LocationConstraint= us-east-1

* Create Dynamo Table

$ aws dynamodb create-table \

--table-name terraform-lock \

--attribute-definitions AttributeName=LockID,AttributeType=S \

--key-schema AttributeName=LockID, KeyType=HASH \

--provisioned-throughput ReadCapacityUnits=5,WriteCapacityUnits=5

---------------------------------------------------------------------------------------------------------------------------------------

**Initialize & Set workspace**

* Initialize working directory

$ terraform init

* Create workspaces

$ terraform workspace new dev

$ terraform workspace new prod

* Verify

$ terraform workspace new list

dev

prod

---------------------------------------------------------------------------------------------------------------------------------------

**Create resources for dev environment**

* Select workspace

$ terraform workspace select dev

* Verify

$ terraform workspace show

dev

* Check execution plan

$ terraform plan -var-file="terraform-dev.tfvars"

* Apply changes

$ terraform apply -var-file="terraform-dev.tfvars"

* Verify

Browse – <ALB-URL>