①

TERRAFORM

→ Hashicorp Language

Why Terraform ? → competitors { pulumi, crossplane }
→ Universal approach

- AWS (CLOUD FORMATION)
- AZURE (RESOURCE MGR)
- GCP [CLOUD DEPLOYMENT MGR (CDM)]
- OPENSTACK

HCL] → Converted as AWS API or Azure API ...

Github CODESPACE } sandbox or container env github provides for free (60 hrs)

>dev 2 CPU, 4 GB RAM

main.tf ← holds main configuration of terraform script

2

```
provider "aws" {
      region = "us-east-1"
}
```

← this step will validate if terraform has access to "aws" (i.e, aws api)

example

```
resource "aws-instance" "example" {
      ami = "A12x345"
      instance-type = "t3.micro" # specify appropriate AMI ID
}
```

⟷

terraform init
- → initiates terraform configuration (.terraform)
- → it reads from main.tf
- → through aws access keys it connects to "aws api"
- → downloads provider

terraform plan
- ← like a dry-run
- ← shows the resources it is going to create

terraform apply ← if all the details (ami, instance type ...) are correct then it creates ec2-instance.

Note : Hashicorp HCL
Hashicorp Terraform } plugins in vs code

Key_pairs on AWS side } useful to login to instance

PROJECT 1 : Create ec2-instance using terraform

sol
before creating ec2 using terraform,
lets revise on what is required to
create an EC2 instance manually

1. AWS account, IAM → terraform uses this to talk to AWS
2. Region → You must choose where the server will run
3. AMI → This is the OS Image of your server
   No AMI = No EC2
4. Instance type → This decides CPU & RAM
5. Network (VPC + subnet) → Your EC2 must live inside the network
   → default : default VPC
6. Security Group (Firewall) → Controls :
   - which ports are open
   - who can access EC2
7. Key Pair (optional) → SSH into server
   → without this, you cannot login to E

DAY 2

TERRAFORM

1. Understanding Providers x Resources
2. Variables x Outputs in Terraform
3. Conditional Expressions x Functions
4. Debugging x Formatting Terraform Files

PROVIDERS

↳ plugin / medium for terraform to create infrastructure.

- **official**
  ↓ Hashicorp actually maintains
- **Partner**
  ↓ maintained by partner
  ex: alibaba; oracle
- **community**
  ↓
  - no official backing
  - opensource

IQ How do you setup terraform in multi region? → using alias

How do you setup terraform in multi cloud? or multi provider?
↳ refer documentation and write code block for each provider.

VARIABLES

↳ used as parameters to pass values

- Input variable
- Output Variable → give the results once terraform apply is successful

example : <Input Variables>

```
resource "aws-instance" "example-instance" {
    ami = var.ami_id          ← calling a variable
    instance_type = var.instance_type
```

(define variables either in seperate file or in main.tf)

```
variable "ami_id" {
    description = "EC2 AMI ID"  ← not mandatory
    type        = string
}
                    name of the variable
variable "instance-type"
    description = "EC2 Instance Type"
    type        = string
    default     = "t3.micro"
}
```

example :

```
output "public-ip" {
    description = "Public IP address of EC2"
    value = aws.instance.example_instance.public_ip
}
```

you are telling terraform to give me that information

real time ↓

- → main.tf
- → provider.tf
- → output.tf
- → input.tf
- → terraform.tfvars ✱✱

→ actual value of variables

Adv

easy to change the variable values / pass the values in this file.

ex: dev → value 1
stage → value 2
prod → value 3

} easy to pass different values

if you change the name of tfvars file, then

```
terraform apply example.tfvars
```

CONDITIONAL EXPRESSIONS

↳ like if/else

syntax :

$\boxed{\text{condition ? true\_val : false\_val}}$

example :

```
production_subnet_cidr = 10.0.1.0/24
dev_subnet_cidr = 10.0.2.0/24
```

$\underbrace{\text{cidr\_blocks = var.environment == "production" ?}}_{\text{condition}}$

$\underbrace{\text{[var.production\_subnet\_cidr]}}_{\text{true}}$: $\underbrace{\text{[var.dev\_subnet.cidr]}}_{\text{false}}$

BUILT-IN FUNCTIONS

example :

```
output "my_map" {
  value = map(var.keys, var.values)
}
```

↳ here, this built-in function provides output in key-value format

ex: "name" = "Esha"
"age" = 25