

## Tugas:

1. Buatlah 1 main.tf untuk membuat 1 instance menggunakan terraform
2. Jelaskan step by step untuk membuat 1 instance menggunakan terraform

1. Install Terraform di laptop

<https://developer.hashicorp.com/terraform/install>

2. Konfigurasi AWS (

```
C:\Users\Valin>aws configure
AWS Access Key ID [*****TIZS]: |
```

Dari mana dapetin AWS Access Key ID?

Harus bikin di AWS Console.

Step-by-step bikin Access Key:

1. Login ke AWS Console
2. Klik nama akun kamu (kanan atas)
3. Pilih Security credentials
4. Scroll ke bagian Access keys
5. Klik Create access key
6. Pilih:
  - o Command Line Interface (CLI)
7. Klik Create access key

Namun, akun AWS yang saya gunakan merupakan company account dan proses login dilakukan menggunakan AWS Single Sign-On (SSO), sehingga user yang digunakan adalah Federated User.

Jadi solusi untuk ini adalah menggunakan AWS SSO Start URL dan SSO Region.

Namun karena admin IT di kantor tidak bisa dihubungi, jadi step ini di skip dulu untuk sementara waktu.

8. Buat main.tf

Buat satu folder project, lalu buat file bernama main.tf.

Isi file tersebut dengan konfigurasi berikut untuk membuat **1 instance EC2**:

```
provider "aws" {
    region = "us-east-1" # ganti sesuai region yang kamu mau
}

resource "aws_instance" "my_instance" {
    ami      = "ami-0c02fb55956c7d316" # ganti sesuai AMI yang ingin digunakan
    instance_type = "t2.micro"

    tags = {
        Name = "MyTerraformInstance"
    }
}
```

#### 9. Inisialisasi Terraform

Masuk ke folder project dan jalankan perintah:

**terraform init**

#### 10. Melihat Rencana Resource

Untuk melihat resource apa saja yang akan dibuat, jalankan:

**terraform plan**

#### 11. Menjalankan Terraform

**terraform apply**

Di step ini, terraform akan:

- Membuat EC2
- Menambahkan tag
- Menyimpan state di `terraform.tfstate`

Catatan...

Bagaimana cara Terraform “mengetahui” AWS account?

Di `main.tf` kamu ada bagian ini:

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

Di sini hanya region yang disebutkan, tapi provider AWS di Terraform sebenarnya menggunakan **credential** untuk login. Terraform AWS Provider mencari credential dari beberapa tempat, urutan prioritasnya kira-kira seperti ini:

- **Environment variables** (cara memberi “informasi global” ke program tanpa menulis langsung di kode). AWS dan Terraform menggunakan environment variables supaya program tahu credential dan region tanpa kamu tulis di main.tf atau file config lain.

- **AWS\_ACCESS\_KEY\_ID**

**AWS\_SECRET\_ACCESS\_KEY**

Opsional: **AWS\_SESSION\_TOKEN** (kalau pakai temporary credentials)

- Shared credentials file (~/.aws/credentials di Linux/Mac atau %UserProfile%\.aws\credentials di Windows)
- AWS CLI SSO atau profile (misal aws configure sso atau aws configure --profile myprofile)
- Instance role / EC2 IAM role (kalau Terraform dijalankan langsung dari EC2 yang punya role)

Jadi sebenarnya Terraform tidak perlu kamu tulis AWS Account ID di main.tf, karena yang penting adalah credential AWS yang memberikan akses ke account tertentu.

#### **Jadi bagaimana AWS Account dikenali?**

- AWS Account ID itu **terkandung dalam credential** yang kamu pakai (access key + secret key).
- Saat Terraform menjalankan terraform apply, provider AWS akan **autentikasi ke AWS API** menggunakan credential itu. AWS API akan otomatis tahu ke account mana credential tersebut terkait.
- Jadi, “AWS account” ditentukan **oleh kredensial**, bukan oleh main.tf.

#### ◆ **Intinya:**

- Terraform **tidak perlu AWS Account ID** di main.tf.
- Terraform “mengetahui” account dari **credential yang digunakan**:
  - Access Key + Secret Key → langsung kenal account
  - SSO / temporary credential → otomatis kenal account