CLOUD COMPUTING AND DEVOPS

Rohan Tikotekar
VIIT IT C C3
RollNo.333056
PRN.22010060

Assignment No 5

Title: Write IaC using terraform to create EC2 machine on aws or azure or google cloud. **(Compulsory to use Input and output variable files)**

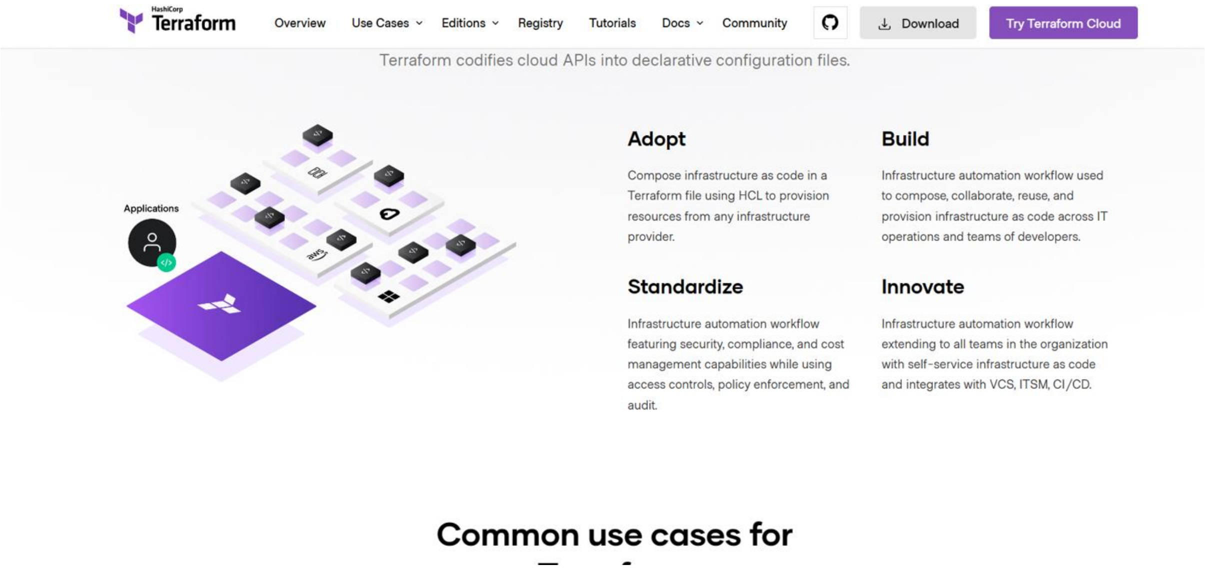
Theory:

1) What is Terraform
2) Step-by-step screenshot to install and configure Terraform.

3) Terraform script to create Infrastructure on any cloud platform (AWS or Azure or Google)

Implementation:

Download terraform



Terraform codifies cloud APIs into declarative configuration files.

Adopt

Compose infrastructure as code in a Terraform file using HCL to provision resources from any infrastructure provider.

Build

Infrastructure automation workflow used to compose, collaborate, reuse, and provision infrastructure as code across IT operations and teams of developers.

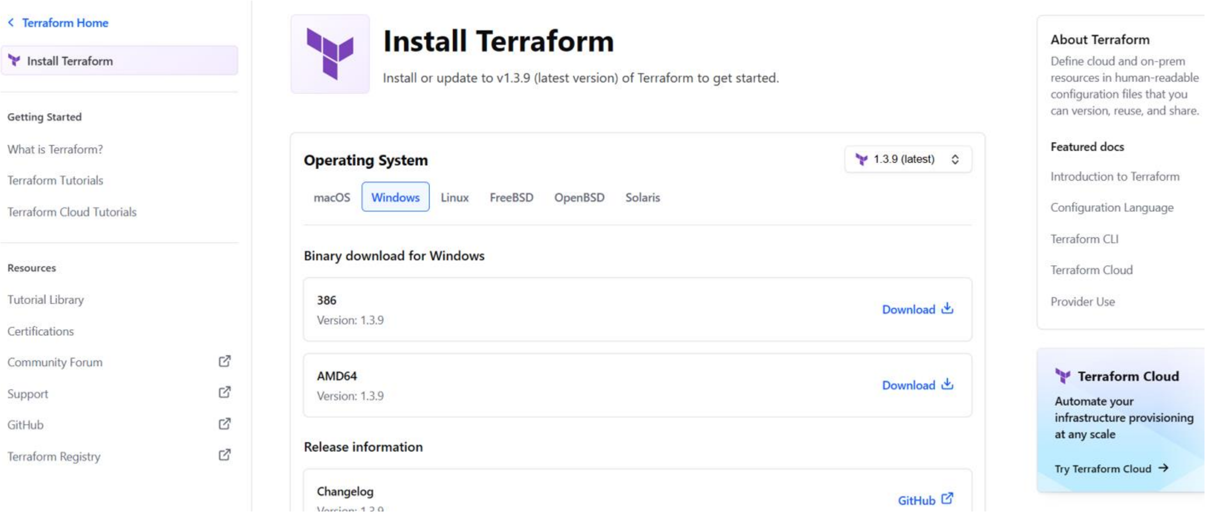
Standardize

Infrastructure automation workflow featuring security, compliance, and cost management capabilities while using access controls, policy enforcement, and audit.

Innovate

Infrastructure automation workflow extending to all teams in the organization with self-service infrastructure as code and integrates with VCS, ITSM, CI/CD.

Common use cases for



Install Terraform

Install or update to v1.3.9 (latest version) of Terraform to get started.

Operating System 1.3.9 (latest)

macOS | Windows | Linux | FreeBSD | OpenBSD | Solaris

Binary download for Windows

386
Version: 1.3.9 — Download

AMD64
Version: 1.3.9 — Download

Release information

Changelog — GitHub

About Terraform

Define cloud and on-prem resources in human-readable configuration files that you can version, reuse, and share.

Featured docs

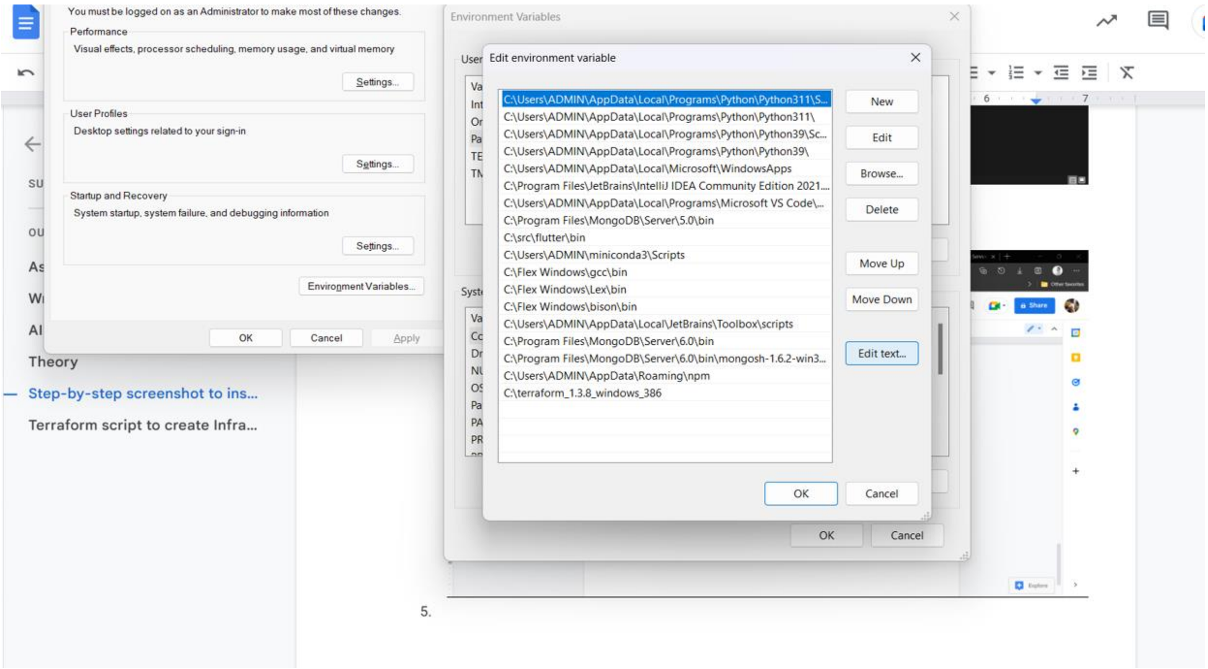
Introduction to Terraform
Configuration Language
Terraform CLI
Terraform Cloud
Provider Use

Terraform Cloud

Automate your infrastructure provisioning at any scale

Try Terraform Cloud →

Add path



You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage, and virtual memory

User Profiles

Desktop settings related to your sign-in

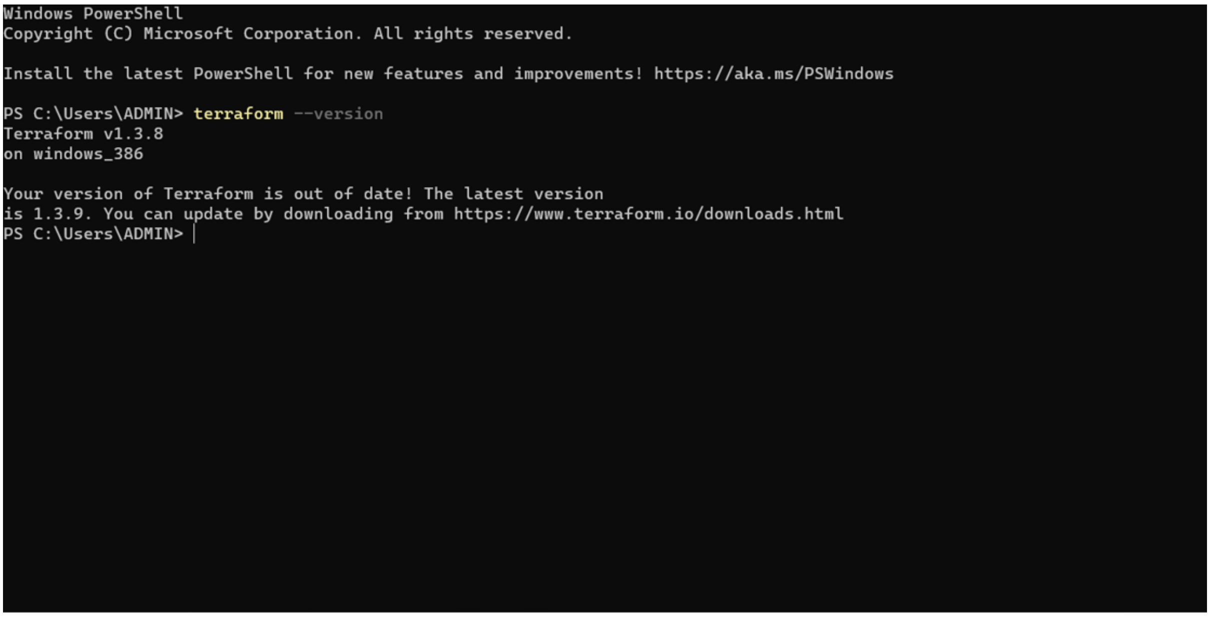
Startup and Recovery

System startup, system failure, and debugging information

Environment Variables

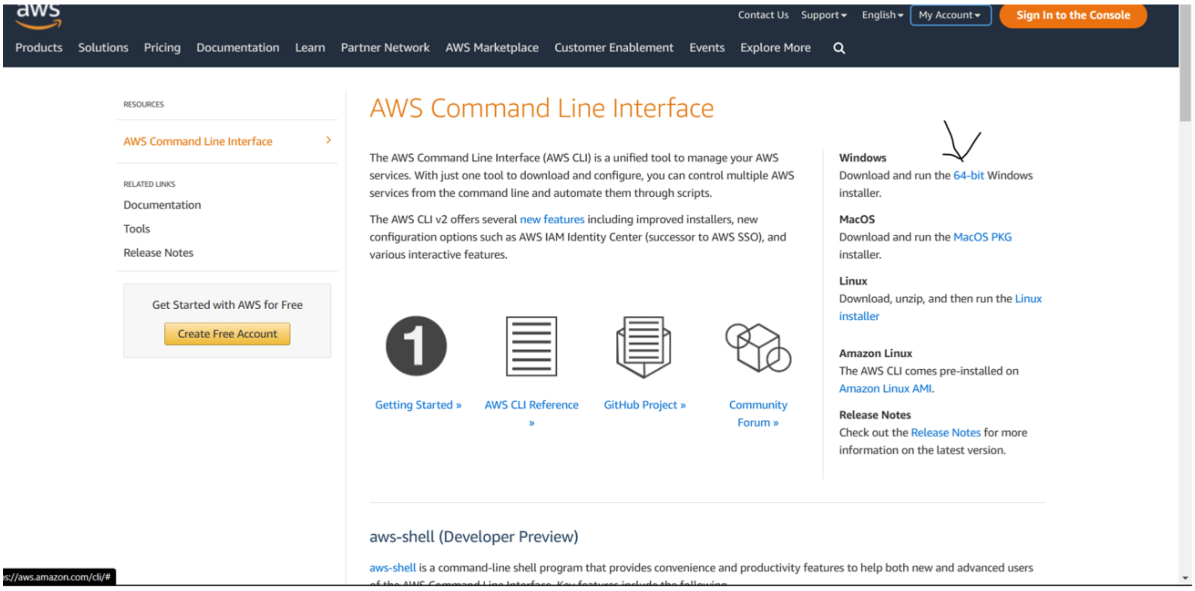
Edit environment variable

```
C:\Users\ADMIN\AppData\Local\Programs\Python\Python311\S...
C:\Users\ADMIN\AppData\Local\Programs\Python\Python311\
C:\Users\ADMIN\AppData\Local\Programs\Python\Python39\Sc...
C:\Users\ADMIN\AppData\Local\Programs\Python\Python39\
C:\Users\ADMIN\AppData\Local\Microsoft\WindowsApps
C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021....
C:\Users\ADMIN\AppData\Local\Programs\Microsoft VS Code\...
C:\Program Files\MongoDB\Server\5.0\bin
C:\src\flutter\bin
C:\Users\ADMIN\miniconda3\Scripts
C:\Flex Windows\gcc\bin
C:\Flex Windows\Lex\bin
C:\Flex Windows\bison\bin
C:\Users\ADMIN\AppData\Local\JetBrains\Toolbox\scripts
C:\Program Files\MongoDB\Server\6.0\bin
C:\Program Files\MongoDB\Server\6.0\bin\mongosh-1.6.2-win3...
C:\Users\ADMIN\AppData\Roaming\npm
C:\terraform_1.3.8_windows_386
```



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\ADMIN> terraform --version
Terraform v1.3.8
on windows_386

Your version of Terraform is out of date! The latest version
is 1.3.9. You can update by downloading from https://www.terraform.io/downloads.html
PS C:\Users\ADMIN> |
```

Download aws cmd



AWS Command Line Interface

The AWS Command Line Interface (AWS CLI) is a unified tool to manage your AWS services. With just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts.

The AWS CLI v2 offers several new features including improved installers, new configuration options such as AWS IAM Identity Center (successor to AWS SSO), and various interactive features.

Windows
Download and run the 64-bit Windows installer.

MacOS
Download and run the MacOS PKG installer.

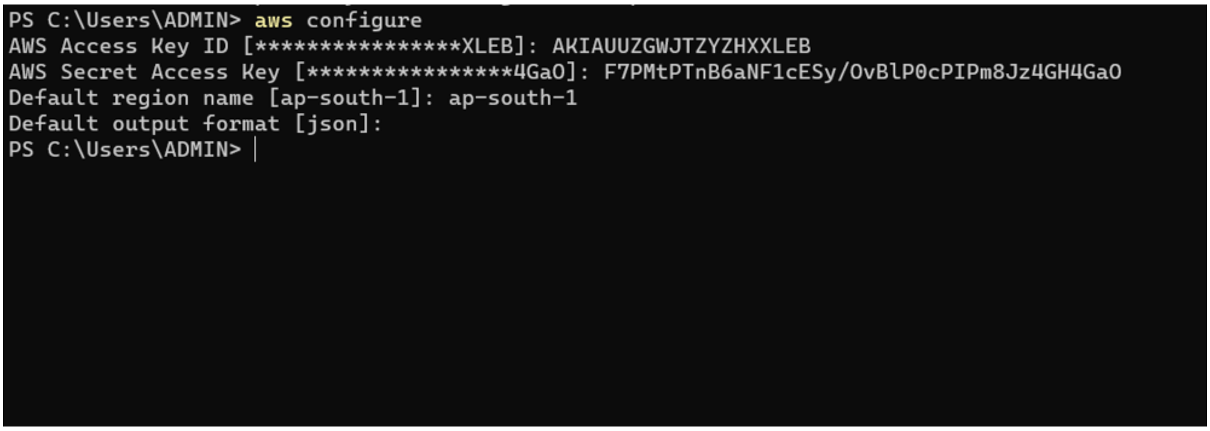
Linux
Download, unzip, and then run the Linux installer

Amazon Linux
The AWS CLI comes pre-installed on Amazon Linux AMI.

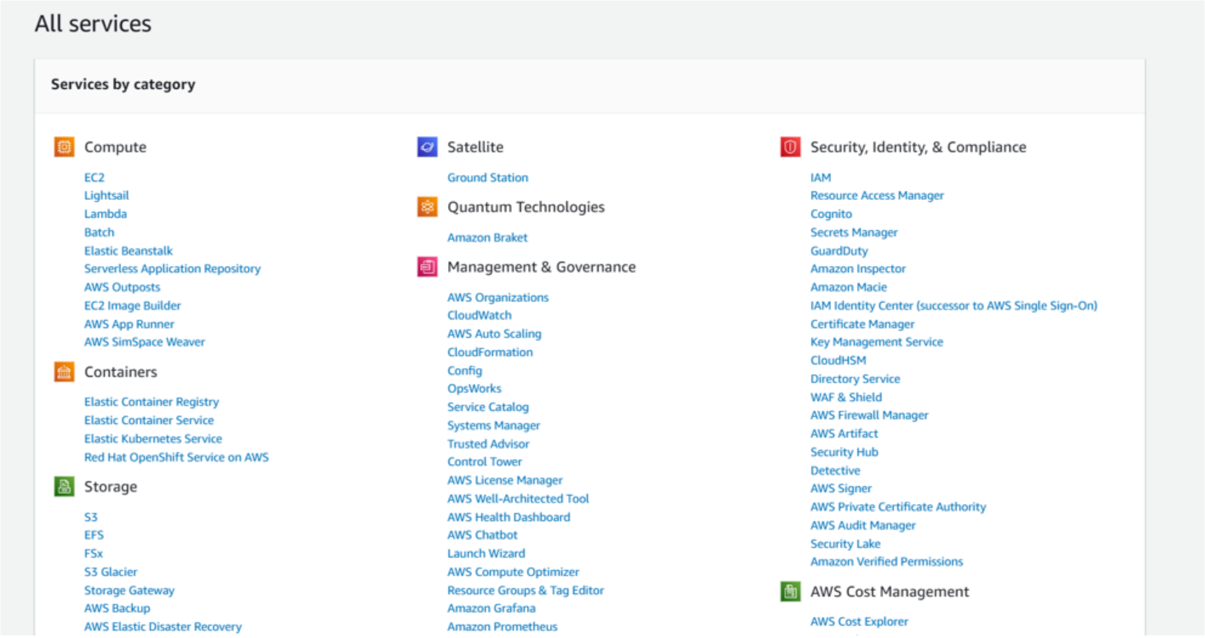
Release Notes
Check out the Release Notes for more information on the latest version.

aws-shell (Developer Preview)

aws-shell is a command-line shell program that provides convenience and productivity features to help both new and advanced users of the AWS Command Line Interface. Key features include the following



```
PS C:\Users\ADMIN> aws configure
AWS Access Key ID [****************XLEB]: AKIAUUZGWJTZYZHXXLEB
AWS Secret Access Key [****************4GaO]: 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
Default region name [ap-south-1]: ap-south-1
Default output format [json]:
PS C:\Users\ADMIN>
```



All services

Services by category

Compute

- EC2
- Lightsail
- Lambda
- Batch
- Elastic Beanstalk
- Serverless Application Repository
- AWS Outposts
- EC2 Image Builder
- AWS App Runner
- AWS SimSpace Weaver

Containers

- Elastic Container Registry
- Elastic Container Service
- Elastic Kubernetes Service
- Red Hat OpenShift Service on AWS

Storage

- S3
- EFS
- FSx
- S3 Glacier
- Storage Gateway
- AWS Backup
- AWS Elastic Disaster Recovery

Satellite

- Ground Station

Quantum Technologies

- Amazon Braket

Management & Governance

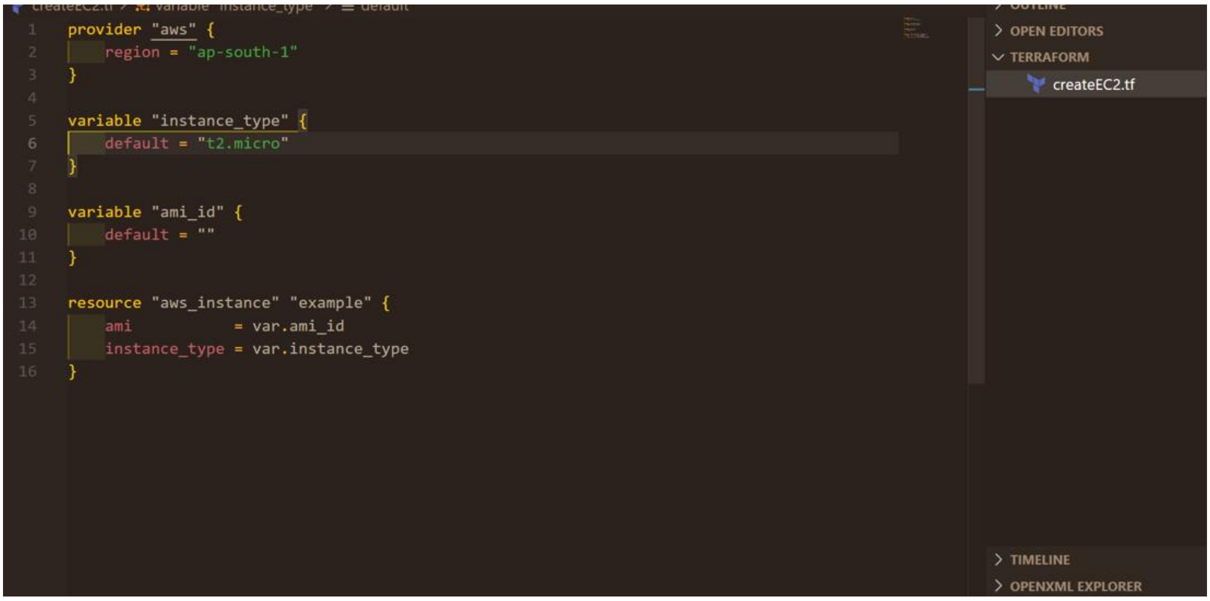
- AWS Organizations
- CloudWatch
- AWS Auto Scaling
- CloudFormation
- Config
- OpsWorks
- Service Catalog
- Systems Manager
- Trusted Advisor
- Control Tower
- AWS License Manager
- AWS Well-Architected Tool
- AWS Health Dashboard
- AWS Chatbot
- Launch Wizard
- AWS Compute Optimizer
- Resource Groups & Tag Editor
- Amazon Grafana
- Amazon Prometheus

Security, Identity, & Compliance

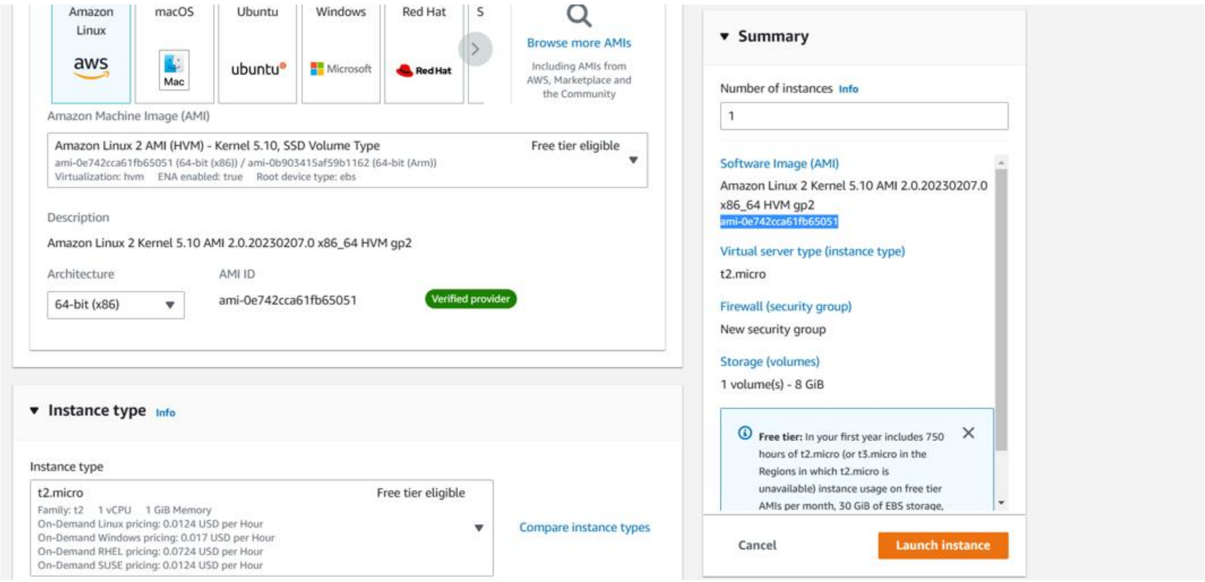
- IAM
- Resource Access Manager
- Cognito
- Secrets Manager
- GuardDuty
- Amazon Inspector
- Amazon Macie
- IAM Identity Center (successor to AWS Single Sign-On)
- Certificate Manager
- Key Management Service
- CloudHSM
- Directory Service
- WAF & Shield
- AWS Firewall Manager
- AWS Artifact
- Security Hub
- Detective
- AWS Signer
- AWS Private Certificate Authority
- AWS Audit Manager
- Security Lake
- Amazon Verified Permissions

AWS Cost Management

- AWS Cost Explorer



```
provider "aws" {
    region = "ap-south-1"
}

variable "instance_type" {
    default = "t2.micro"
}

variable "ami_id" {
    default = ""
}

resource "aws_instance" "example" {
    ami           = var.ami_id
    instance_type = var.instance_type
}
```



Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type — Free tier eligible
ami-0e742cca61fb65051 (64-bit (x86)) / ami-0b903415af59b1162 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description
Amazon Linux 2 Kernel 5.10 AMI 2.0.20230207.0 x86_64 HVM gp2

Architecture: 64-bit (x86)
AMI ID: ami-0e742cca61fb65051 — Verified provider

Instance type Info

Instance type
t2.micro — Free tier eligible
Family: t2 1 vCPU 1 GiB Memory
On-Demand Linux pricing: 0.0124 USD per Hour
On-Demand Windows pricing: 0.017 USD per Hour
On-Demand RHEL pricing: 0.0724 USD per Hour
On-Demand SUSE pricing: 0.0124 USD per Hour

Summary

Number of instances Info
1

Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI 2.0.20230207.0 x86_64 HVM gp2
ami-0e742cca61fb65051

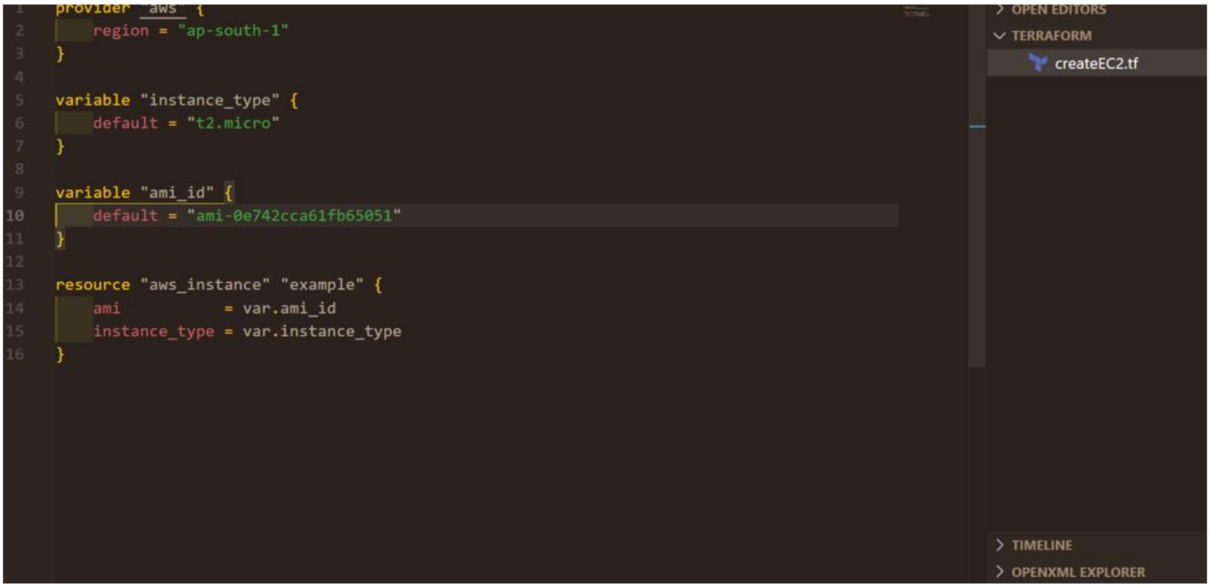
Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

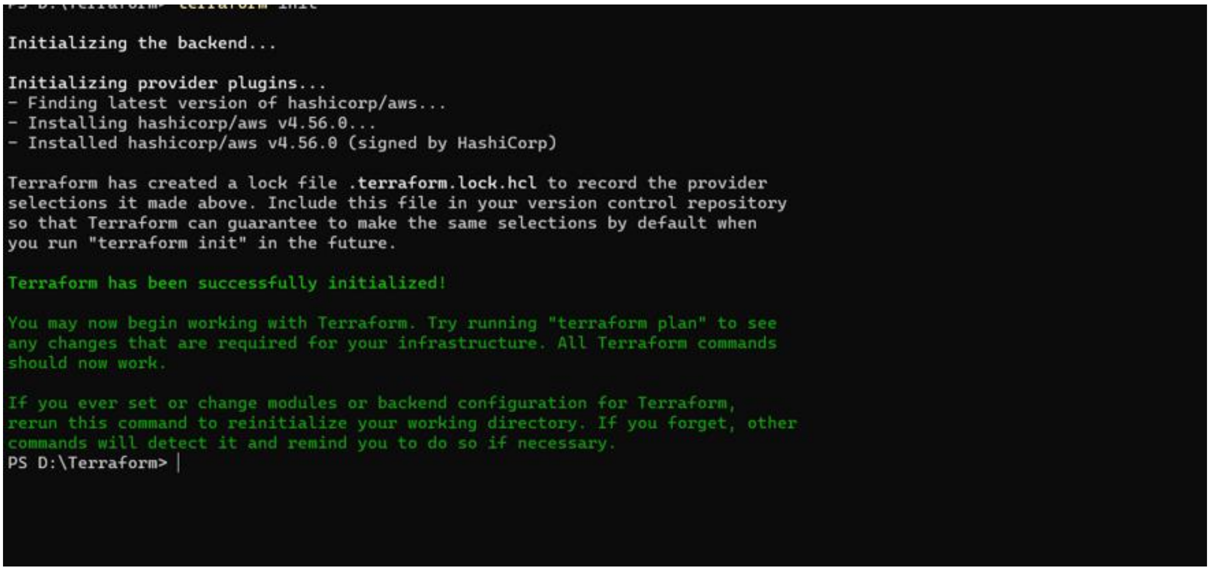
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage,

Copy ami, json

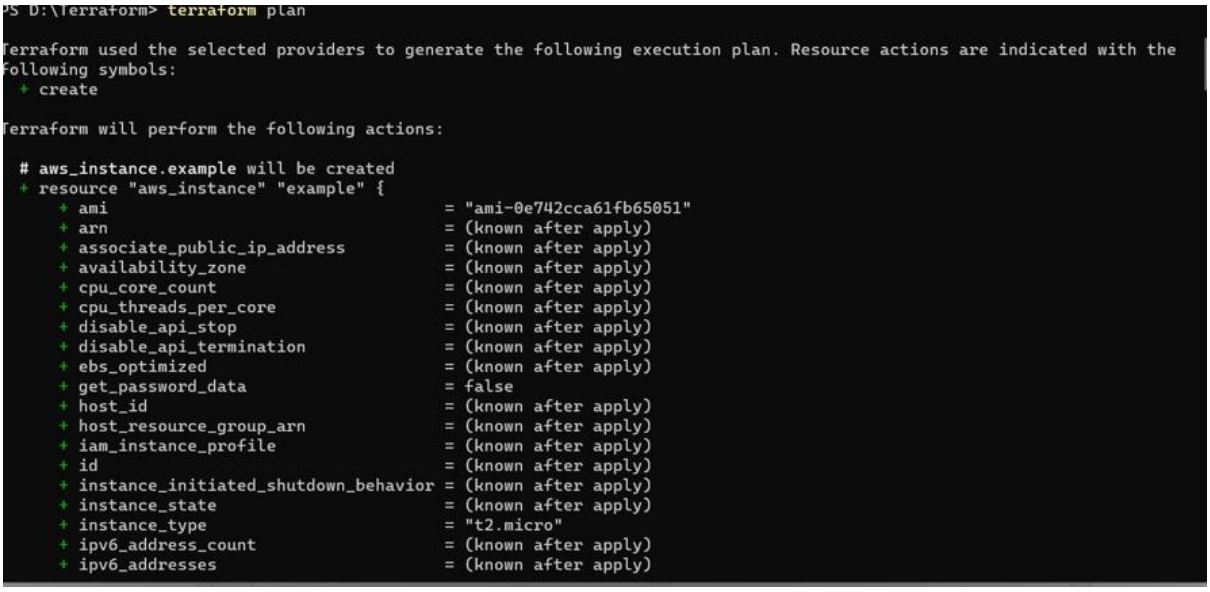


```
provider "aws" {
    region = "ap-south-1"
}

variable "instance_type" {
    default = "t2.micro"
}

variable "ami_id" {
    default = "ami-0e742cca61fb65051"
}

resource "aws_instance" "example" {
    ami           = var.ami_id
    instance_type = var.instance_type
}
```

Terraform script

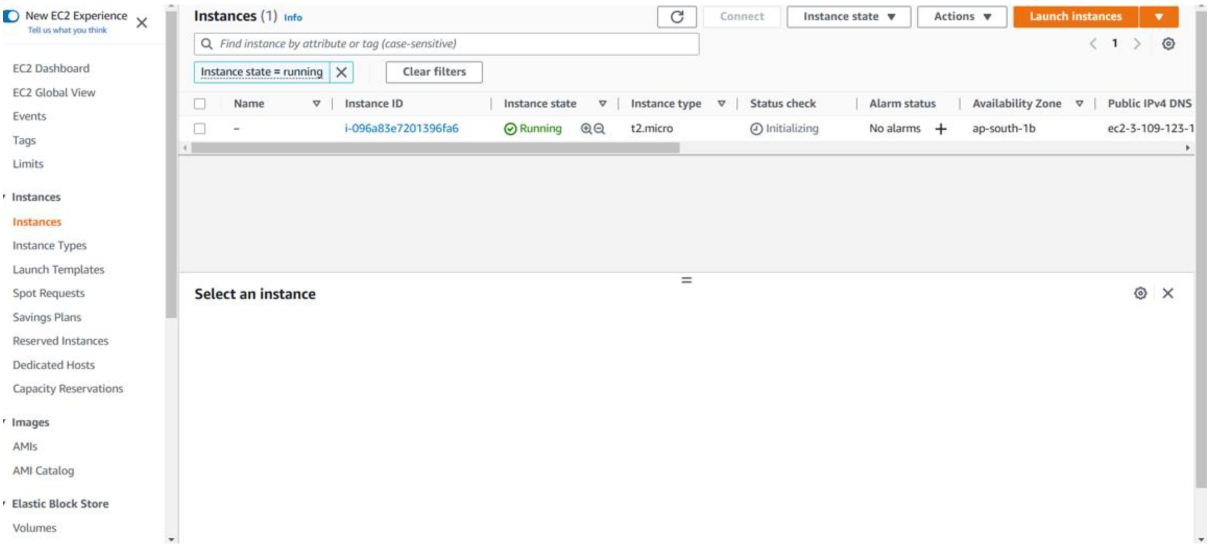


```
PS D:\Terraform> terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v4.56.0...
- Installed hashicorp/aws v4.56.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
PS D:\Terraform> |
```



```
PS D:\Terraform> terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the
following symbols:
  + create

Terraform will perform the following actions:

  # aws_instance.example will be created
  + resource "aws_instance" "example" {
      + ami                                  = "ami-0e742cca61fb65051"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
```

Aws instance:



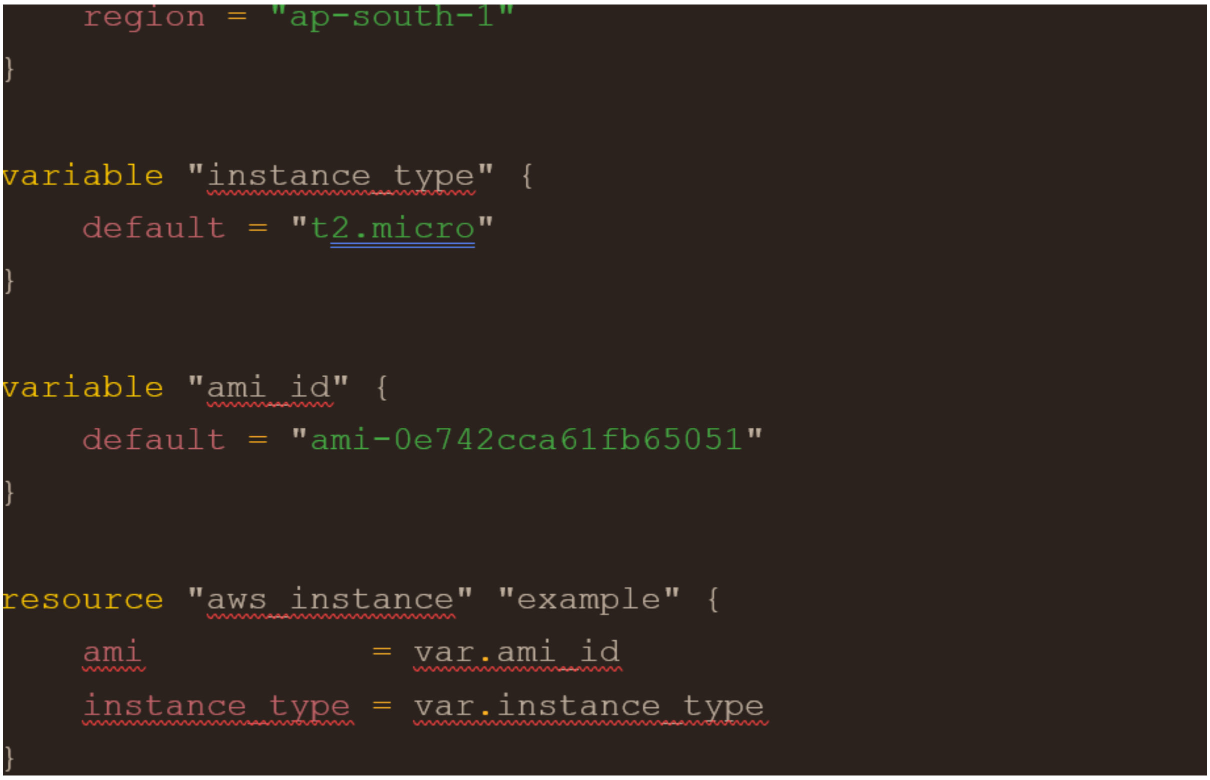
Instances (1) Info

Find instance by attribute or tag (case-sensitive)

Instance state = running

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
-	i-096a83e7201396fa6	Running	t2.micro	Initializing	No alarms	ap-south-1b	ec2-3-109-123-1

Terraform json code:



```
    region = "ap-south-1"
}

variable "instance_type" {
    default = "t2.micro"
}

variable "ami_id" {
    default = "ami-0e742cca61fb65051"
}

resource "aws_instance" "example" {
    ami           = var.ami_id
    instance_type = var.instance_type
}
```