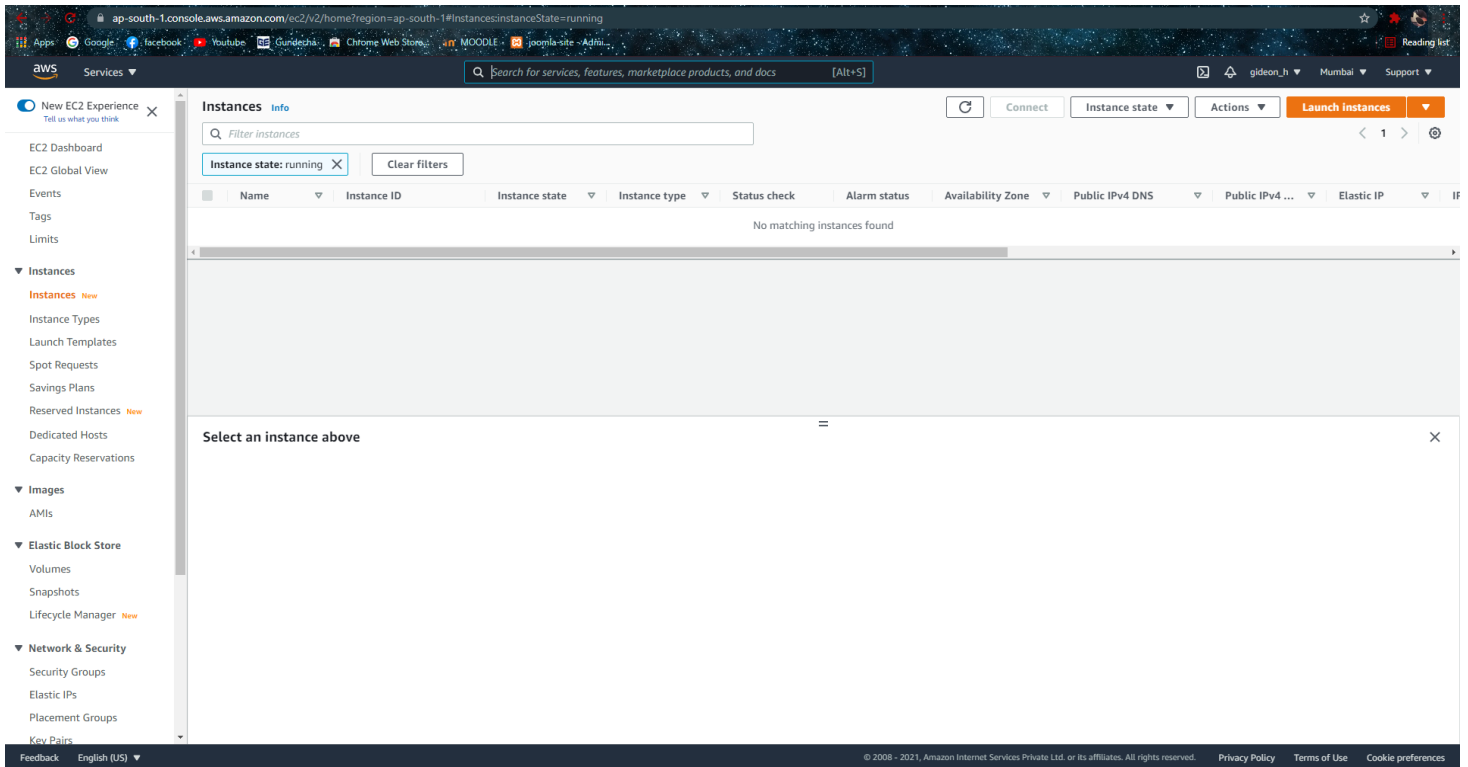


# Advanced DevOps

**AIM To build, apply and destroy AWS using Terraform.**

**Step 1: First we will check that no instance is running on EC2.**



## Step 2: Create an IAM user with Programmatic Password, Administrator access and download access key and secret key from download.csv

### IAM:

The screenshot shows the AWS IAM Management Console dashboard. The left sidebar contains the 'Identity and Access Management (IAM)' menu with options like Dashboard, Access management, Access reports, and Credential report. The main content area displays the 'IAM dashboard' with a sign-in URL, IAM resource counts (Users: 0, Roles: 2), and a 'Best practices' section with links to various AWS security features. The right sidebar provides 'Additional Information' and 'Tools' links. The bottom of the page shows the Windows taskbar and system tray.

### USERS:

The screenshot shows the 'Users' page in the AWS IAM Management Console. The left sidebar is the same as the previous screenshot. The main content area shows 'Users (0)' with an 'Add users' button. Below this is a search bar and a table with columns: User name, Groups, Last activity, MFA, Password age, and Active key age. The table is currently empty, displaying 'No resources to display'. The bottom of the page shows the Windows taskbar and system tray.

## ADD USER:

The screenshot shows the 'Add user' page in the AWS IAM Management Console. The page is divided into two main sections: 'Set user details' and 'Select AWS access type'. In the 'Set user details' section, the 'User name' is set to 'user1'. In the 'Select AWS access type' section, 'Programmatic access' and 'AWS Management Console access' are selected. The 'Console password' is set to 'Custom password' with the value 'abc123!@#'. The 'Require password reset' checkbox is unchecked. At the bottom, there are 'Cancel' and 'Next: Permissions' buttons.

**Add user**

**Set user details**

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name\*

[Add another user](#)

**Select AWS access type**

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

**Access type\***

- ☒ **Programmatic access**  
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.
- ☒ **AWS Management Console access**  
Enables a **password** that allows users to sign-in to the AWS Management Console.

**Console password\***

- ☐ Autogenerated password
- ☒ Custom password  
  
☒ Show password

**Require password reset**

- ☐ User must create a new password at next sign-in  
Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

\* Required

[Cancel](#) [Next: Permissions](#)

## User group with administrator access:

The screenshot shows the 'Create group' page in the AWS IAM Management Console. It displays a list of policies to select for the new group. The 'AdministratorAccess' policy is selected. The page includes a search bar, a table of policies, and 'Cancel' and 'Create group' buttons at the bottom.

[Create policy](#) [Refresh](#)

**Filter policies**

Showing 670 results

	Policy name	Type	Used as	Description
<input checked="" type="checkbox"/>	<a href="#">AdministratorAccess</a>	Job function	Permissions policy (3)	Provides full access to AWS services and resources.
<input type="checkbox"/>	<a href="#">AdministratorAccess-Amplify</a>	AWS managed	None	Grants account administrative permissions while explicitly allowing direct acce...
<input type="checkbox"/>	<a href="#">AdministratorAccess-AWSElastic...</a>	AWS managed	None	Grants account administrative permissions. Explicitly allows developers and a...

[Cancel](#) [Create group](#)

## Next tag: ( no tags added)

### Add user

1 2 3 4 5

### Add tags (optional)

IAM tags are key-value pairs you can add to your user. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this user. [Learn more](#)

Key	Value (optional)	Remove
<input type="text" value="Add new key"/>	<input type="text"/>	

You can add 50 more tags.

Cancel

Previous

Next: Review

## Review:

The screenshot shows the AWS IAM console 'Add user' review page. The browser address bar shows the URL: `console.aws.amazon.com/iam/home#/users$new?step=review&accessKey&login&userNames=user1&passwordType=manual&permissionType=policies&policies=arn:aws:iam::aws:policy%2FAdministratorAccess`. The page has a dark header with the AWS logo and a search bar. The main content area is titled 'Add user' with a progress indicator showing steps 1 through 5, with step 4 (Review) selected. The 'Review' section includes a sub-header 'Review' and a paragraph: 'Review your choices. After you create the user, you can view and download the autogenerated password and access key.' Below this is the 'User details' section, which lists the following information: User name: user1, AWS access type: Programmatic access and AWS Management Console access, Console password type: Custom, Require password reset: No, and Permissions boundary: Permissions boundary is not set. The 'Permissions summary' section states: 'The following policies will be attached to the user shown above.' and shows a table with one policy: Type: Managed policy, Name: AdministratorAccess. The 'Tags' section states: 'No tags were added.' At the bottom of the page, there are three buttons: 'Cancel', 'Previous', and 'Create user'. The footer of the browser window shows the Windows taskbar with various application icons and the system clock indicating 02:16 PM on 10-08-2021.

Feedback English (US) © 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

User added successfully:

IAM Management Console

console.aws.amazon.com/iam/home#/users\$new?step=final&accessKey&login&userNames=user1&passwordType=manual&permissionType=policies&policies=arn:aws:iam::aws:policy%2FAdministratorAccess

Services

Search for services, features, marketplace products, and docs

gideon\_h Global Support

Add user

12346

Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://197069257618.signin.aws.amazon.com/console>

Download .csv

	User	Access key ID	Secret access key	Email login instructions
	user1	AKIAS3YR6HOJO2N4Q7NX	***** Show	Send email

Close

Feedback English (US) © 2006 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

Type here to search

ENG 02:17 PM 10-08-2021

File Home Share View

This PC > D:\

Search D:\

Quick access

Desktop D:\ Documents Pictures AdvDevOps DevOps exp9 IP OneDrive This PC 3D Objects D:\ Desktop Documents

VirtualBox-6.1.18-142142-Win

VBL-CABLE\_Driver\_Pack43

bibleshow2

vMix

Overwatch

openjdk-11+28\_windows-x64\_bin

sonar-scanner-4.6.2.2472-windows

sonarqube-9.0.1.46107

terraform

Proteus\_Professional\_8.6\_SP2

Call of Duty Black Ops Cold War

downloads

admin

media

Git

Pro Tools First

phpMyAdmin-5.0.4-alpha-language

GTA V

Riot Games

sts-4.8.0.RELEASE

bootstrap-4.1.3

Among Us v2020.9.9s

WpSystem

WUDownloadCache

Program Files

WindowsApps

eclipse-jee-2020-06-R-win32-x86\_64

Epic Games

msdownload.tmp

MSIAfterburnerSetup

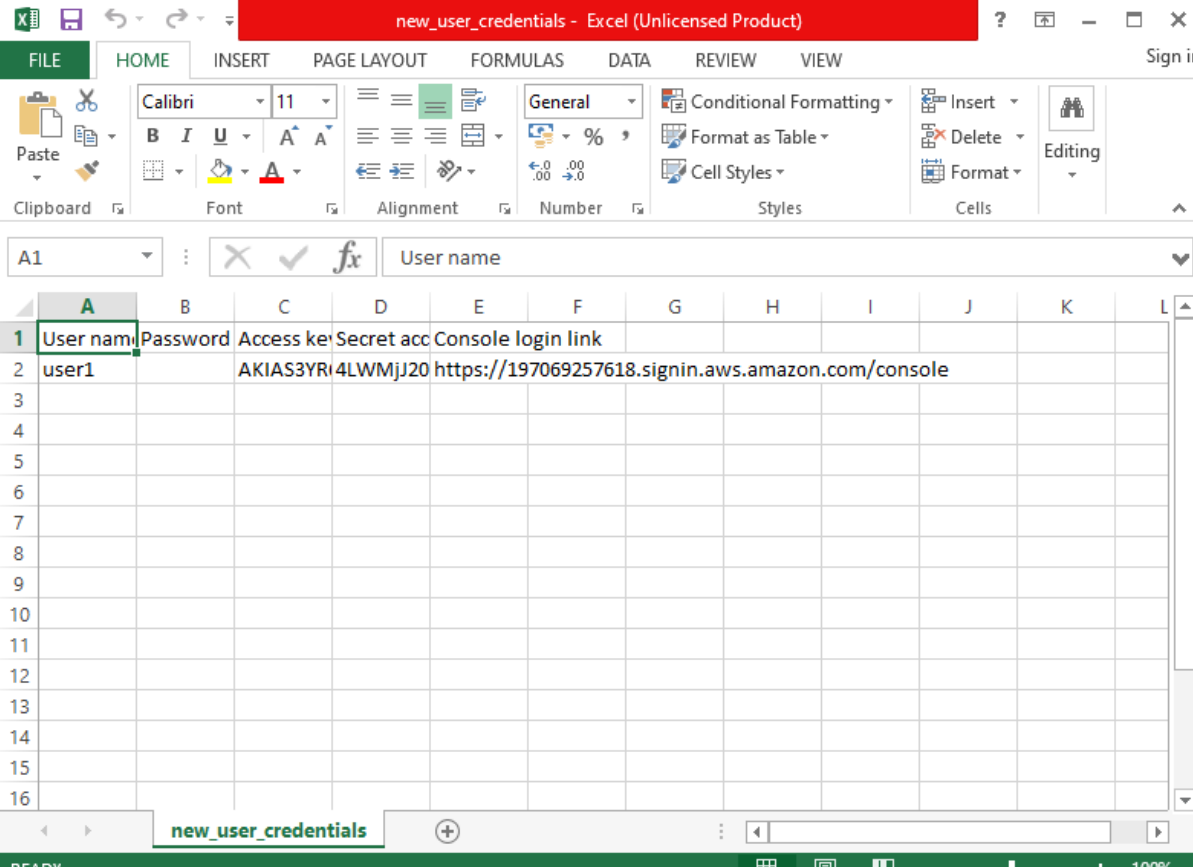
Ubisoft Game Launcher

Steam

new\_user\_credentials

83 items | 1 item selected 192 bytes

### Step 3: Download.csv

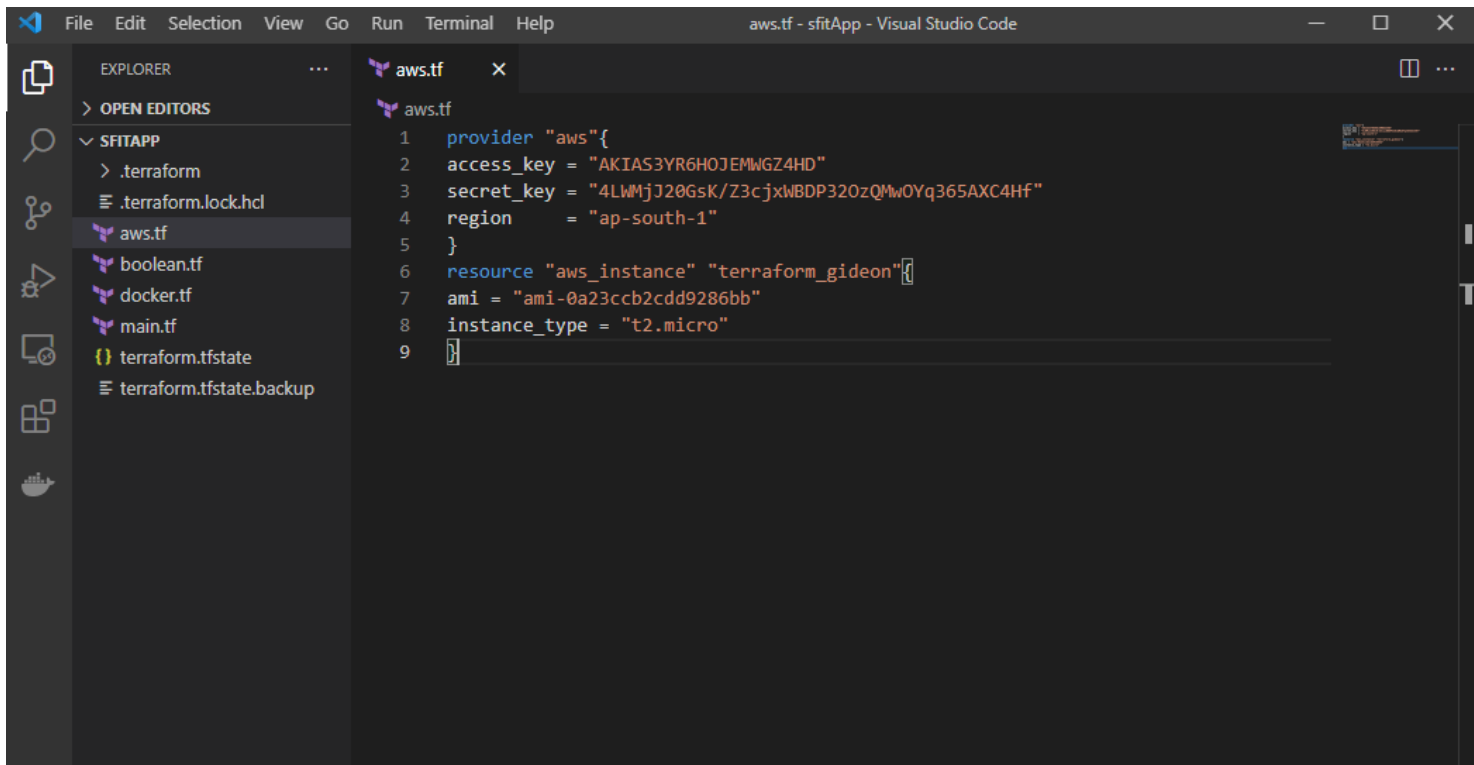


The screenshot shows a Microsoft Excel window titled "new\_user\_credentials - Excel (Unlicensed Product)". The ribbon is set to "HOME". The spreadsheet has the following data:

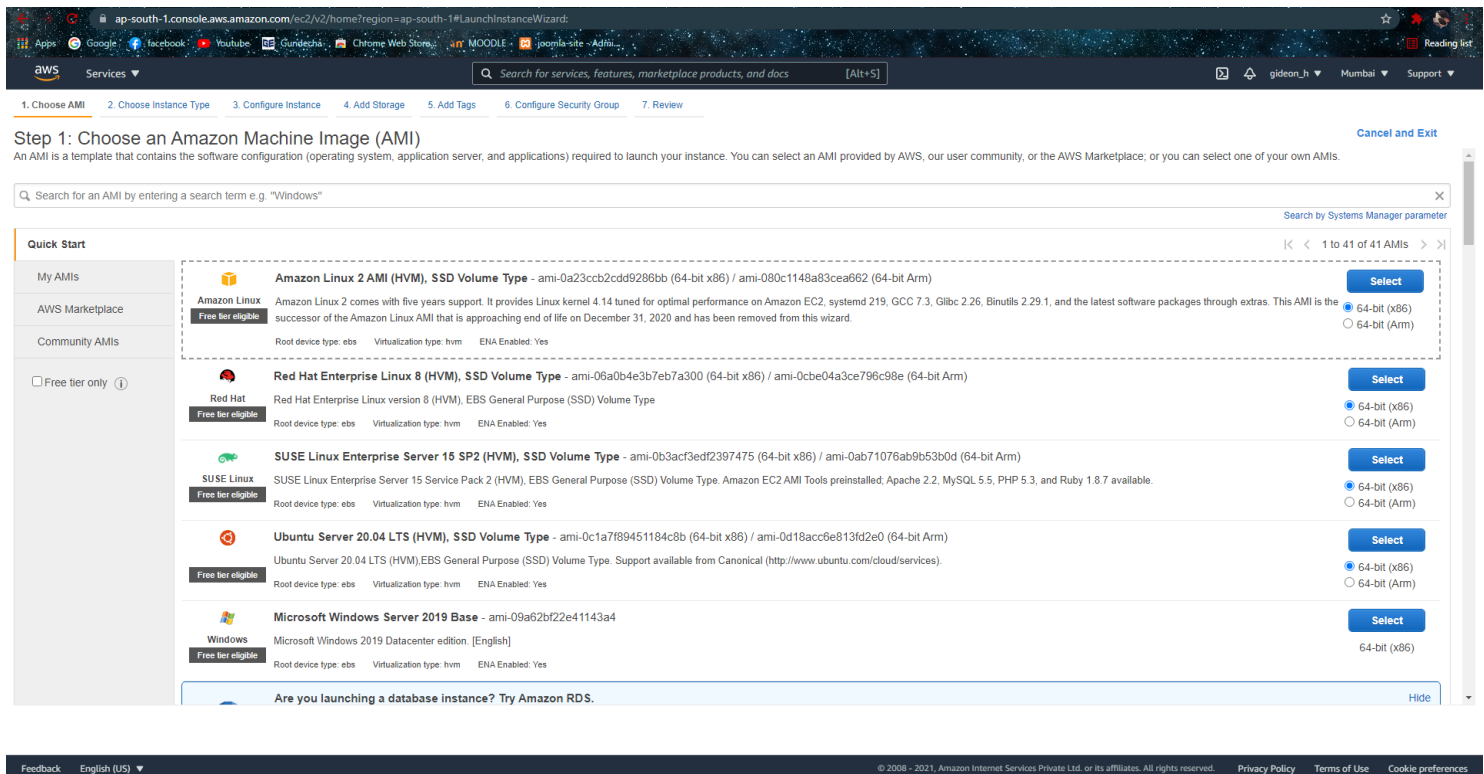
	A	B	C	D	E	F	G	H	I	J	K	L
1	User name	Password	Access key	Secret acc	Console login link							
2	user1		AKIAS3YRI4LWMJJ20		<a href="https://197069257618.signin.aws.amazon.com/console">https://197069257618.signin.aws.amazon.com/console</a>							
3												
4												
5												
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												
16												

The status bar at the bottom shows "READY" and "100%".

## Step 4: Now write a Terraform program



In Launch instance, you will get ami : amazon machine image



For Instance type : t2.micro is freely available

The screenshot shows the AWS Management Console interface for the 'Launch Instance Wizard'. The 'Choose Instance Type' step is active. The 'Filter by' section shows 'All instance families' and 'Current generation'. The 'Currently selected' text indicates 't2.micro (- ECU, 1 vCPUs, 2.5 GHz, ~, 1 GiB memory, EBS only)'. A table lists various instance types, with 't2.micro' highlighted in blue and labeled 'Free tier eligible'.

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GiB)	EBS Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	t2	t2.micro	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.xlarge	4	16	EBS only	-	Moderate	Yes
<input type="checkbox"/>	t2	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
<input type="checkbox"/>	t3	t3.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	t3	t3.micro	2	1	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	t3	t3.small	2	2	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	t3	t3.medium	2	4	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	t3	t3.large	2	8	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	t3	t3.xlarge	4	16	EBS only	Yes	Up to 5 Gigabit	Yes

Buttons at the bottom: Cancel, Previous, Review and Launch, Next: Configure Instance Details.

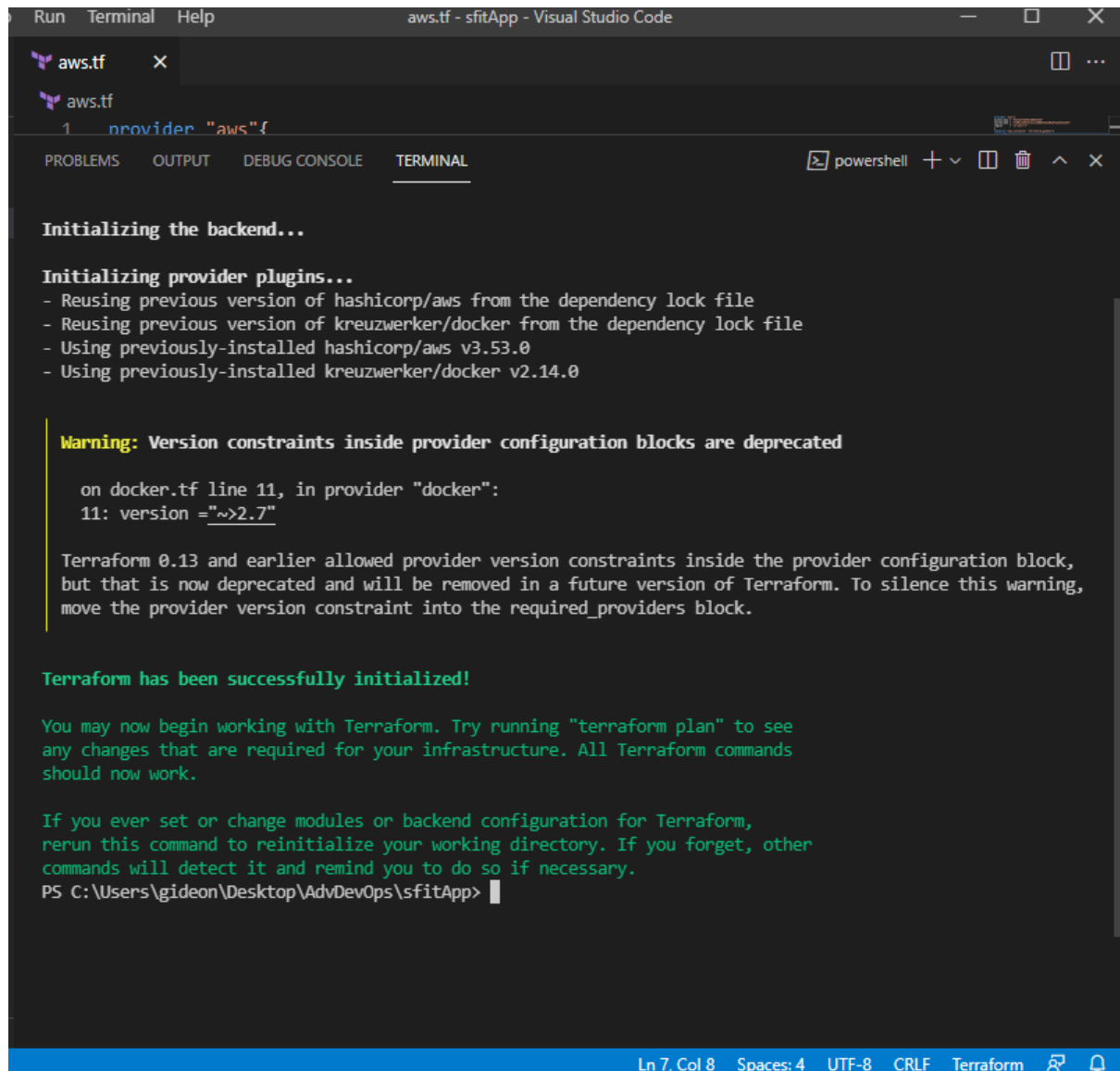
Step 5: Now initialize the terraform ...type c:\SfitApp> terraform init

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

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\gideon\Desktop\AdvDevOps\sfitApp> terraform init
```





The screenshot shows the Visual Studio Code interface with a file named `aws.tf` open. The terminal window is active, displaying the output of the `terraform init` command. The output includes the following text:

```
Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Reusing previous version of kreuzwerker/docker from the dependency lock file
- Using previously-installed hashicorp/aws v3.53.0
- Using previously-installed kreuzwerker/docker v2.14.0

Warning: Version constraints inside provider configuration blocks are deprecated

on docker.tf line 11, in provider "docker":
11: version = "~>2.7"

Terraform 0.13 and earlier allowed provider version constraints inside the provider configuration block,
but that is now deprecated and will be removed in a future version of Terraform. To silence this warning,
move the provider version constraint into the required_providers block.

Terraform has been successfully initialized!

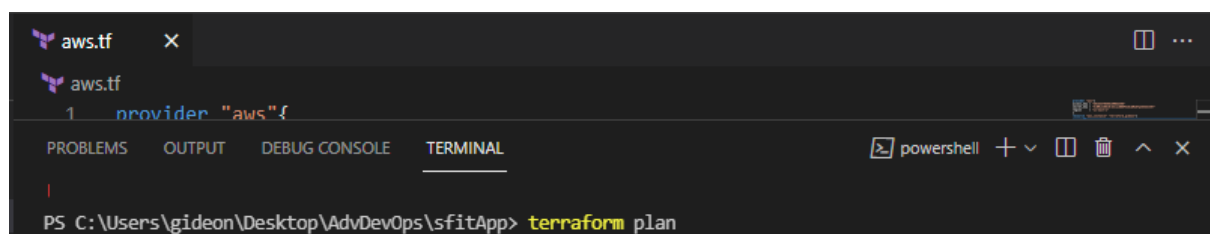
You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

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 C:\Users\gideon\Desktop\AdvDevOps\sfitApp>
```

The status bar at the bottom of the terminal indicates the current line and column: `Ln 7, Col 8`, `Spaces: 4`, `UTF-8`, `CRLF`, and the file type `Terraform`.

Terraform has been initialized successfully.

## Step 6: `c:\sfitApp>terraform plan`



The screenshot shows the Visual Studio Code interface with the `aws.tf` file open. The terminal window is active, and the command `terraform plan` has been entered at the prompt. The output of the command is not yet visible.

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.terraform_gideon will be created
+ resource "aws_instance" "terraform_gideon" {
  + ami                        = "ami-0a23ccb2cdd9286bb"
  + 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_termination    = (known after apply)
  + ebs_optimized              = (known after apply)
  + get_password_data          = false
  + host_id                   = (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)
  + key_name                   = (known after apply)
  + monitoring                 = (known after apply)
  + outpost_arn                = (known after apply)
  + password_data              = (known after apply)
  + placement_group            = (known after apply)
  + primary_network_interface_id = (known after apply)
  + private_dns                = (known after apply)
  + private_ip                 = (known after apply)
  + public_dns                 = (known after apply)
  + public_ip                  = (known after apply)
  + secondary_private_ips      = (known after apply)
  + security_groups            = (known after apply)
  + source_dest_check          = true
  + subnet_id                  = (known after apply)
  + tags_all                   = (known after apply)
```

```
+ capacity_reservation_target {  
  + capacity_reservation_id = (known after apply)  
}  
}  
  
+ ebs_block_device {  
  + delete_on_termination = (known after apply)  
  + device_name           = (known after apply)  
  + encrypted             = (known after apply)  
  + iops                  = (known after apply)  
  + kms_key_id            = (known after apply)  
  + snapshot_id           = (known after apply)  
  + tags                  = (known after apply)  
  + throughput            = (known after apply)  
  + volume_id             = (known after apply)  
  + volume_size           = (known after apply)  
  + volume_type           = (known after apply)  
}  
  
+ enclave_options {  
  + enabled = (known after apply)  
}  
  
+ ephemeral_block_device {  
  + device_name = (known after apply)  
  + no_device   = (known after apply)  
  + virtual_name = (known after apply)  
}  
  
+ metadata_options {  
  + http_endpoint           = (known after apply)  
  + http_put_response_hop_limit = (known after apply)  
  + http_tokens             = (known after apply)  
}  
  
+ network_interface {  
  + delete_on_termination = (known after apply)  
  + device_index          = (known after apply)  
  + network_interface_id  = (known after apply)
```

```
# docker_image.redis will be created
+ resource "docker_image" "redis" {
  + id          = (known after apply)
  + latest      = (known after apply)
  + name        = "redis:latest"
  + output      = (known after apply)
  + repo_digest = (known after apply)
}
```

Plan: 2 to add, 0 to change, 0 to destroy.

**Warning:** Version constraints inside provider configuration blocks are deprecated

on docker.tf line 11, in provider "docker":  
11: version = "~>2.7"

Terraform 0.13 and earlier allowed provider version constraints inside the provider configuration block, but that is now deprecated and will be removed in a future version of Terraform. To silence this warning, move the provider version constraint into the required\_providers block.

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.

PS C:\Users\gideon\Desktop\AdvDevOps\sfitApp>

## Step 7: Check the instance on Ec2 before terraform apply

The screenshot shows the AWS Management Console for the 'ap-south-1' region. The 'Instances' page is active, with the 'Instance state' filter set to 'running'. The console indicates 'No matching instances found'. Below this, the details for instance 'i-003641cff3d9ec84c' are displayed. The 'Instance state' field shows a red 'Terminated' status. The 'Instance summary' section shows the instance ID, name, and type (t2.micro). The 'Public IPv4 address' field is empty. The 'Private IPv4 addresses' field is empty. The 'Public IPv4 DNS' field is empty. The 'VPC ID' field is empty. The 'Subnet ID' field is empty. The 'AWS Compute Optimizer finding' section shows a recommendation to 'Opt-in to AWS Compute Optimizer for recommendations'.

Instance is not yet created.

## Step 8: Terraform apply

```
PS C:\Users\gideon\Desktop\AdvDevOps\sfitApp> terraform apply
```

Ln 4, Col 25 Spaces: 4 UTF-8 CRLF Terraform

```
Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

Enter a value: yes

aws_instance.terraform_gideon: Creating...
aws_instance.terraform_gideon: Still creating... [10s elapsed]
aws_instance.terraform_gideon: Still creating... [20s elapsed]
aws_instance.terraform_gideon: Creation complete after 22s [id=i-06368e930999b7022]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\Users\gideon\Desktop\AdvDevOps\sfitApp>
```

**Step 9: Check terraform created instance on EC2...we have created 3 instances.**

The screenshot displays the AWS Management Console interface. At the top, the navigation bar shows the AWS logo, 'Services', and a search bar. The main content area is titled 'Instances (1/1) Info'. Below this, there's a table of instances. The first instance, 'i-06368e930999b7022', is in the 'Running' state. The instance details panel on the right shows the 'Instance summary' tab, which includes information such as the Instance ID, Public IPv4 address (3.109.217.65), Instance state (Running), Instance type (t2.micro), and VPC ID (vpc-674c270e).

**Instances (1/1) Info**

Filter instances

Instance state: running Clear filters

<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
<input checked="" type="checkbox"/>	-	i-06368e930999b7022	Running	t2.micro	Initializing	No alarms	ap-south-1a	ec2-3-109-217-65.ap-s...	3.109.217.65	-

**Instance: i-06368e930999b7022**

Details Security Networking Storage Status checks Monitoring Tags

**Instance summary Info**

<b>Instance ID</b> i-06368e930999b7022	<b>Public IPv4 address</b> 3.109.217.65   <a href="#">open address</a>	<b>Private IPv4 addresses</b> 172.31.33.12
<b>Instance state</b> Running	<b>Public IPv4 DNS</b> ec2-3-109-217-65.ap-south-1.compute.amazonaws.com   <a href="#">open address</a>	
<b>Instance type</b> t2.micro	<b>Elastic IP addresses</b> -	
<b>VPC ID</b> vpc-674c270e	<b>IAM Role</b> -	

**Step 10: Now destroy the instance from command prompt....c:\SfitApp> terraform destroy**

```
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\Users\gideon\Desktop\AdvDevOps\sfitApp> terraform destroy
docker_image.redis: Refreshing state... [id=sha256:5d89766432d0b0b2ddc60fa3806812d64d7ffa6eb1166c85e3609639bfcfd83fredis:latest]
aws_instance.terraform_gideon: Refreshing state... [id=i-06368e930999b7022]

Note: Objects have changed outside of Terraform

Terraform detected the following changes made outside of Terraform since the last "terraform apply":

# aws_instance.terraform_gideon has been changed
~ resource "aws_instance" "terraform_gideon" {
  id          = "i-06368e930999b7022"
+ tags       = {}
  # (28 unchanged attributes hidden)

  # (5 unchanged blocks hidden)
}

Unless you have made equivalent changes to your configuration, or ignored the relevant attributes using
ignore_changes, the following plan may include actions to undo or respond to these changes.
```

---

```
Terraform used the selected providers to generate the following execution plan. Resource actions are
indicated with the following symbols:
- destroy

Terraform will perform the following actions:

# aws_instance.terraform_gideon will be destroyed
- resource "aws_instance" "terraform_gideon" {
  - ami          = "ami-0a23ccb2cdd9286bb" -> null
  - arn          = "arn:aws:ec2:ap-south-1:197069257618:instance/i-06368e930999b7022"
```

Plan: 0 to add, 0 to change, 2 to destroy.

**Warning:** Version constraints inside provider configuration blocks are deprecated

on docker.tf line 11, in provider "docker":  
11: version = "~>2.7"

Terraform 0.13 and earlier allowed provider version constraints inside the provider configuration block, but that is now deprecated and will be removed in a future version of Terraform. To silence this warning, move the provider version constraint into the required\_providers block.

**Do you really want to destroy all resources?**

Terraform will destroy all your managed infrastructure, as shown above.  
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

docker\_image.redis: Destroying... [id=sha256:5d89766432d0b0b2ddc60fa3806812d64d7ffa6eb1166c85e3609639bfcfd83 fredis:latest]

docker\_image.redis: Destruction complete after 1s

aws\_instance.terraform\_gideon: Destroying... [id=i-06368e930999b7022]

aws\_instance.terraform\_gideon: Still destroying... [id=i-06368e930999b7022, 10s elapsed]

aws\_instance.terraform\_gideon: Still destroying... [id=i-06368e930999b7022, 20s elapsed]

aws\_instance.terraform\_gideon: Destruction complete after 30s

**Destroy complete! Resources: 2 destroyed.**

PS C:\Users\gideon\Desktop\AdvDevOps\sfitApp>

The screenshot shows the AWS Management Console for the 'ap-south-1' region. The 'Instances' page displays a list of EC2 instances. The instance 'i-06368e930999b7022' is highlighted in blue and has a status of 'Terminated'. The instance details panel on the right shows the instance is terminated, with a red 'Terminated' status and a warning icon. The instance summary shows it is a t2.micro instance using the Amazon Linux AMI.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
-	i-06368e930999b7022	Terminated	t2.micro	-	No alarms	ap-south-1a	-	-	-
slave	i-0691b1d8c58cf42e4	Stopped	t2.micro	-	No alarms	ap-south-1b	-	-	-
master	i-08f0157a35e4a4c0a	Stopped	t2.micro	-	No alarms	ap-south-1b	-	-	-
-	i-028e28ef8754f3742	Terminated	t2.micro	-	No alarms	ap-south-1b	-	-	-
-	i-003641cff3d9ec84c	Terminated	t2.micro	-	No alarms	ap-south-1b	-	-	-

**Instance: i-06368e930999b7022**

**Details** | Security | Networking | Storage | Status checks | Monitoring | Tags

**Instance summary** | Info

Instance ID: i-06368e930999b7022

Public IPv4 address: -

Private IPv4 addresses: -

Instance state: **Terminated**

Public IPv4 DNS: -

Instance type: t2.micro

Elastic IP addresses: -

VPC ID: -

AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more

IAM Role: -

Subnet ID: -

**Instance details** | Info

Platform: Amazon Linux (Inferred)

AMI ID: ami-0a23ccb2cdd9286bb

Monitoring: disabled

Platform details

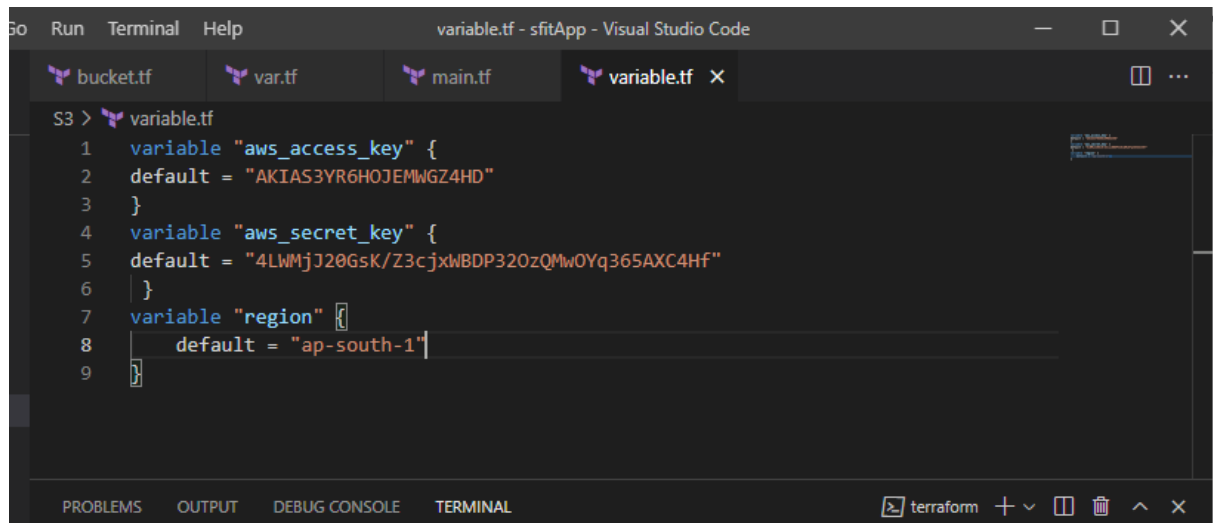
AMI name: -

Termination protection: -

We have created and destroyed EC2 instance from the terraform file.

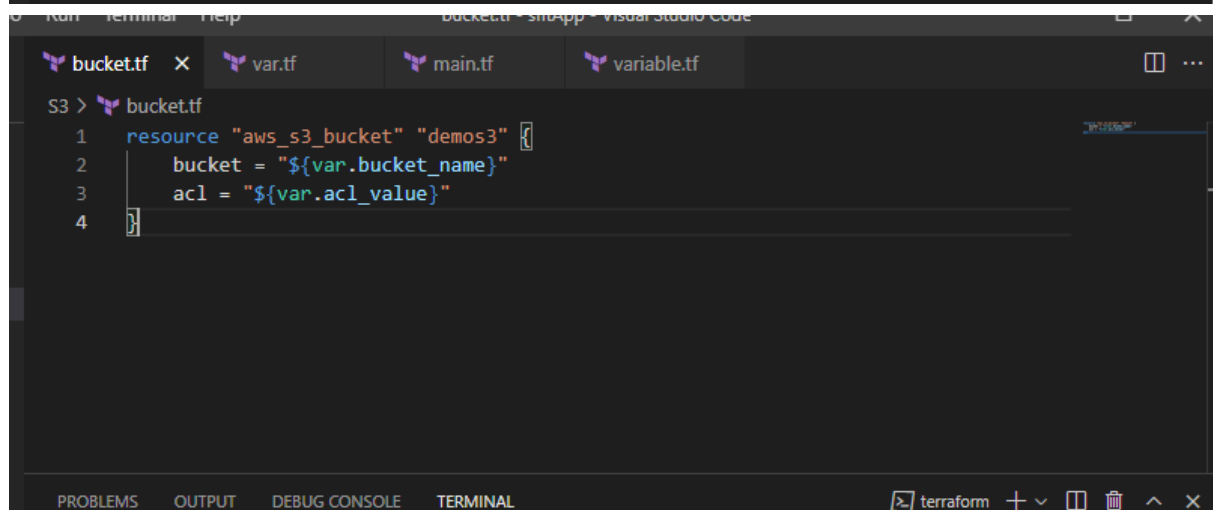
→ the above highlighted instance is terminated as seen in the ec2 instances list

## Code:



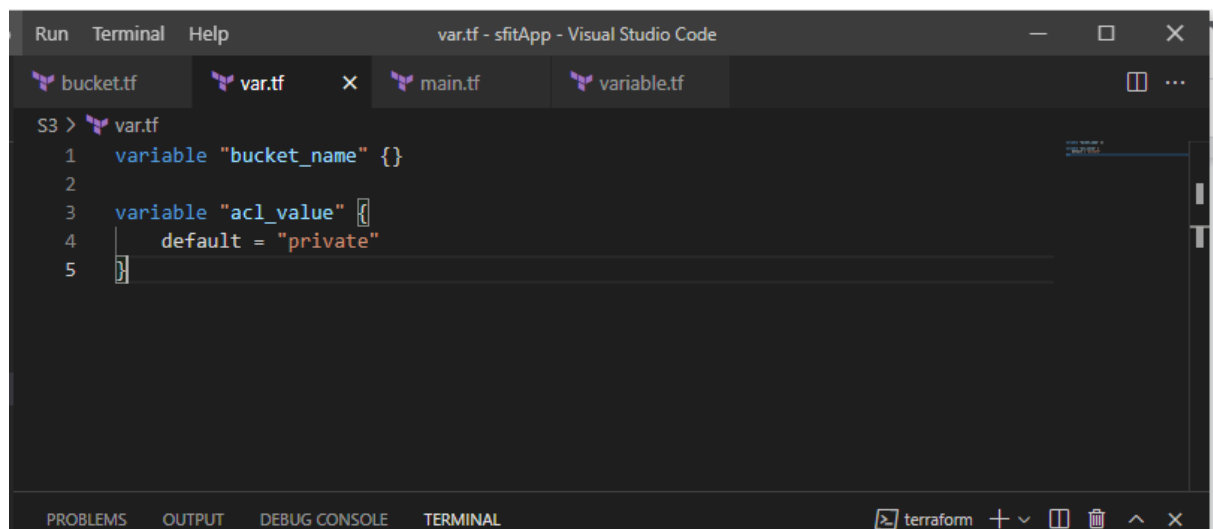
This screenshot shows the Visual Studio Code editor with the 'variable.tf' file open. The file contains Terraform variable definitions for AWS access and secret keys, and the region. The interface includes a top menu bar with 'Run', 'Terminal', and 'Help', and a bottom status bar with 'terraform' and various icons.

```
S3 > variable.tf
1 variable "aws_access_key" {
2   default = "AKIAS3YR6HOJEMWGZ4HD"
3 }
4 variable "aws_secret_key" {
5   default = "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"
6 }
7 variable "region" {}
8   default = "ap-south-1"
9 }
```



This screenshot shows the Visual Studio Code editor with the 'bucket.tf' file open. The file contains a Terraform resource definition for an AWS S3 bucket named 'demos3'. The interface is similar to the previous screenshot, showing the same menu and status bars.

```
S3 > bucket.tf
1 resource "aws_s3_bucket" "demos3" {}
2   bucket = "${var.bucket_name}"
3   acl = "${var.acl_value}"
4 }
```



This screenshot shows the Visual Studio Code editor with the 'var.tf' file open. The file contains Terraform variable definitions for 'bucket\_name' and 'acl\_value'. The interface is consistent with the previous screenshots, showing the same menu and status bars.

```
S3 > var.tf
1 variable "bucket_name" {}
2
3 variable "acl_value" {}
4   default = "private"
5 }
```



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\gideon\Desktop\AdvDevOps\sfitApp\S3> terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v3.60.0

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 C:\Users\gideon\Desktop\AdvDevOps\sfitApp\S3> terraform apply
var.bucket_name
Enter a value: demos3

provider.aws.region
The region where AWS operations will take place. Examples
are us-east-1, us-west-2, etc.

Enter a value: ap-south-1

module.s3.aws_s3_bucket.demos3: Creating...
module.s3.aws_s3_bucket.demos3: Creation complete after 5s [id=terraform-demo-bucket-376]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

```

### References:

<https://www.bacancytechnology.com/blog/aws-s3-bucket-using-terraform>

<https://k21academy.com/terraform-iac/terraform-interview-questions/>

---