



System Provisioning and Configuration Management



Submitted to-

Dr. Hitesh Kumar Sharma

Submitted by-

Arnav Goel
500108276
(Devops) B-2

Assignment-1

STEP 1: Setup Terraform Project

```
Arnav@Asus-Vivobook MINGW64 ~ (main)
$ cd terraform-lab

Arnav@Asus-Vivobook MINGW64 ~/terraform-lab (main)
$ ls
CICD/ Lab-1-4/ Lab-10/ Lab-11/ Lab-5/ Lab-6/ Lab-7/ Lab-8/ Lab-9/

Arnav@Asus-Vivobook MINGW64 ~/terraform-lab (main)
$ mkdir assignment1

Arnav@Asus-Vivobook MINGW64 ~/terraform-lab (main)
$ cd assignment1

Arnav@Asus-Vivobook MINGW64 ~/terraform-lab/assignment1 (main)
$
```

Create Terraform files

```
Arnav@Asus-Vivobook MINGW64 ~/terraform-lab/assignment1 (main)
$ touch main.tf variables.tf outputs.tf

Arnav@Asus-Vivobook MINGW64 ~/terraform-lab/assignment1 (main)
$ ls
main.tf outputs.tf variables.tf

Arnav@Asus-Vivobook MINGW64 ~/terraform-lab/assignment1 (main)
```

✓ STEP 2: Create IAM User & Get Credentials

1. Go to AWS Console → IAM → Users → Add User

The screenshot shows the 'User details' step of the AWS IAM 'Add User' wizard. It includes fields for 'User name' (set to 'Arnav-1'), a checkbox for 'Provide user access to the AWS Management Console - optional' (unchecked), and a question 'Are you providing console access to a person?'. The 'I want to create an IAM user' option is selected. Below this, there are sections for 'Console password' (radio button for 'Custom password' selected, with a password entered) and 'Access keys' (radio button for 'Programmatic access' selected).

User details

User name
Arnav-1
The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

Provide user access to the AWS Management Console - optional
If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

Are you providing console access to a person?

User type

Specify a user in Identity Center - Recommended
We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.

I want to create an IAM user
We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keypairs, or a backup credential for emergency account access.

Console password

Autogenerated password
You can view the password after you create the user.

Custom password
Enter a custom password for the user.

2. Attach **Administrator access**
3. Save the **Access Key ID** and **Secret Access Key**

[Users](#) > [Arnav-1](#) > [Create access key](#)

Access key created
This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

Step 1
 Access key best practices & alternatives
 Step 2 - optional
 Set description tag
 Retrieve access keys

Retrieve access keys Info

Access key
If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
AKIAR2QJ6FETXGUUY3J	***** Show

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [best practices for managing AWS access keys](#).

[Download .csv file](#) [Done](#)

STEP 3: Get Latest Amazon Linux 2 AMI ID

Recents **Quick Start**

Amazon Linux
macOS
Ubuntu
Windows
Red Hat
SUSE Linux
Debian
 [Browse more AMIs](#)

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type ami-0e35ddab05955cf57 (64-bit (x86)) / ami-0429d68a1cd41ca80 (64-bit (Arm)) Virtualization: hvm ENA enabled: true Root device type: ebs	Free tier eligible
--	------------------------------------

Description
Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture	AMI ID	Publish Date	Username	
64-bit (x86) ▾	ami-0e35ddab05955cf57	2025-03-05	ubuntu	Verified provider

STEP4 : Terraform Code (main.tf)



```
>Welcome   main.tf  1 ×
└ main.tf >  resource "aws_instance" "web"
  1 provider "aws" {
  2   region = "us-east-1"
  3 }
  4
  5 # VPC
  6 resource "aws_vpc" "main" {
  7   cidr_block = "10.0.0.0/16"
  8 }
  9
 10 resource "aws_internet_gateway" "igw" {
 11   vpc_id = aws_vpc.main.id
 12 }
 13
 14 # VPN
 15 resource "aws_vpn_gateway" "vpn_gw" {
 16   vpc_id = aws_vpc.main.id
 17   tags = {
 18     Name = "MyVPN"
 19   }
 20 }
 21
 22 # EC2 Instances
 23 resource "aws_instance" "web" {
 24   count      = 2
 25   ami        = "ami-0e35ddab05955cf57"
 26   instance_type = "t2.micro"
 27
 28   tags = {
 29     Name = "Instance-${count.index + 1}"
 30   }
 31 }
 32
 33 # S3 Bucket
 34 resource "random_id" "bucket_id" {
 35   byte_length = 4
 36 }
 37
```

STEP 5: Run Terraform Commands

```
Arnav@Asus-Vivobook MINGW64 ~/terraform-lab/assignment1 (main)
$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/random...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/random v3.7.1...
- Installed hashicorp/random v3.7.1 (signed by HashiCorp)
- Installing hashicorp/aws v5.94.1...
- Installed hashicorp/aws v5.94.1 (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.
```

Terraform validate

```
Arnav@Asus-Vivobook MINGW64 ~/terraform-lab/assignment1 (main)
$ terraform validate
Success! The configuration is valid.
```

Terraform plan

```
# aws_vpn_gateway.vpn_gw will be created
+ resource "aws_vpn_gateway" "vpn_gw" {
    + amazon_side_asn = (known after apply)
    + arn              = (known after apply)
    + id               = (known after apply)
    + tags             = {
        + "Name" = "MyVPN"
    }
    + tags_all         = {
        + "Name" = "MyVPN"
    }
    + vpc_id          = (known after apply)
}

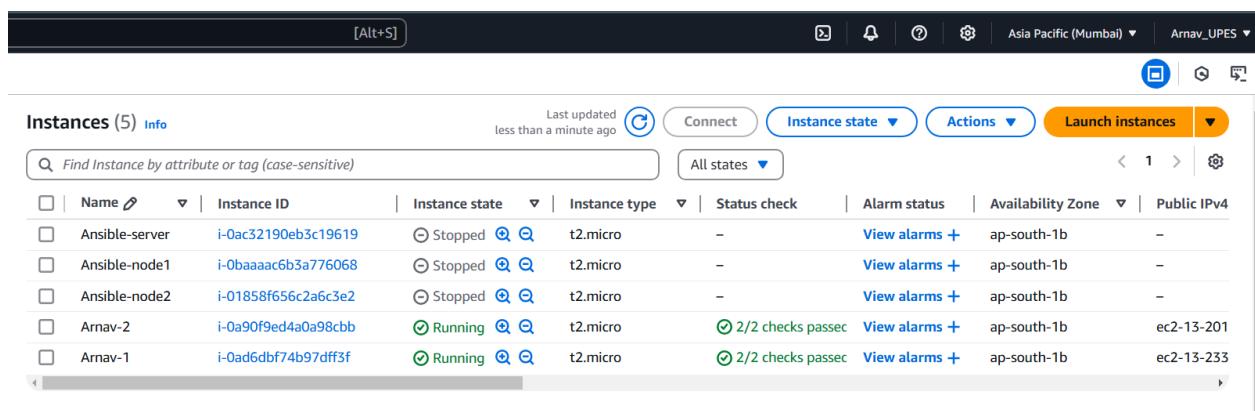
# random_id.bucket_id will be created
+ resource "random_id" "bucket_id" {
    + b64_std          = (known after apply)
    + b64_url          = (known after apply)
    + byte_length      = 4
    + dec              = (known after apply)
    + hex              = (known after apply)
    + id               = (known after apply)
}

Plan: 8 to add, 0 to change, 0 to destroy.
```

Terraform apply -auto-approve

```
Plan: 8 to add, 0 to change, 0 to destroy.
random_id.bucket_id: Creating...
random_id.bucket_id: Creation complete after 0s [id=mr391Q]
aws_vpc.main: Creating...
aws_s3_bucket.my_bucket: Creating...
aws_instance.web[0]: Creating...
aws_instance.web[1]: Creating...
aws_vpc.main: Creation complete after 3s [id=vpc-0435c60798d7fa177]
aws_vpn_gateway.vpn_gw: Creating...
aws_internet_gateway.igw: Creating...
aws_s3_bucket.my_bucket: Creation complete after 3s [id=my-terraform-bucket-9abdfdd5]
aws_s3_bucket_acl.my_bucket_acl: Creating...
aws_internet_gateway.igw: Creation complete after 0s [id=igw-0c6220bab815b21a4]
aws_instance.web[0]: Still creating... [10s elapsed]
aws_instance.web[1]: Still creating... [10s elapsed]
aws_vpn_gateway.vpn_gw: Still creating... [10s elapsed]
aws_instance.web[1]: Creation complete after 14s [id=i-0a90f9ed4a0a98cbb]
aws_instance.web[0]: Creation complete after 14s [id=i-0ad6dbf74b97dff3f]
aws_vpn_gateway.vpn_gw: Still creating... [20s elapsed]
aws_vpn_gateway.vpn_gw: Creation complete after 35s [id=vgw-0e2773e855525732f]
```

Check for created resources



Instances (5) Info		Last updated less than a minute ago	Connect	Instance state	Actions	Launch instances		
			All states					
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
<input type="checkbox"/>	Ansible-server	i-0ac32190eb3c19619	Stopped	t2.micro	-	View alarms +	ap-south-1b	-
<input type="checkbox"/>	Ansible-node1	i-0baaaac6b3a776068	Stopped	t2.micro	-	View alarms +	ap-south-1b	-
<input type="checkbox"/>	Ansible-node2	i-01858f656c2a6c3e2	Stopped	t2.micro	-	View alarms +	ap-south-1b	-
<input type="checkbox"/>	Arnav-2	i-0a90f9ed4a0a98cbb	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1b	ec2-13-201
<input type="checkbox"/>	Arnav-1	i-0ad6dbf74b97dff3f	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1b	ec2-13-233

[Alt+S]

Asia Pacific (Mumbai) Arnav_UPES

Account snapshot - updated every 24 hours All AWS Regions

Storage lens provides visibility into storage usage and activity trends. Metrics don't include directory buckets. [Learn more](#)

View Storage Lens dashboard

General purpose buckets Directory buckets

General purpose buckets (1) [Info](#) All AWS Regions

Buckets are containers for data stored in S3.

Find buckets by name

Name AWS Region IAM Access Analyzer Creation date

Name	AWS Region	IAM Access Analyzer	Creation date
my-terraform-bucket-9abdfdd5	Asia Pacific (Mumbai) ap-south-1	View analyzer for ap-south-1	April 11, 2025, 19:44:02 (UTC+05:30)

Copy ARN Empty Delete Create bucket

[Alt+S]

Asia Pacific (Mumbai) Arnav_UPES

Your VPCs (2) [Info](#)

Last updated 1 minute ago Actions Create VPC

Search

Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR
-	vpc-0435c60798d7fa177	Available	Off	10.0.0.0/16	-
-	vpc-0273bb5ce7c45b7bf	Available	Off	172.31.0.0/16	-

Terraform destroy -auto-approve

```
aws_vpn_gateway.vpn_gw: Destroying... [id=vgw-0e2773e855525732f]
aws_internet_gateway.igw: Destroying... [id=igw-0c6220bab815b21a4]
aws_instance.web[1]: Destroying... [id=i-0a90f9ed4a0a98cbb]
aws_instance.web[0]: Destroying... [id=i-0ad6dbf74b97dff3f]
aws_s3_bucket.my_bucket: Destroying... [id=my-terraform-bucket-9abdfdd5]
aws_s3_bucket.my_bucket: Destruction complete after 1s
random_id.bucket_id: Destroying... [id=mr391Q]
random_id.bucket_id: Destruction complete after 0s
aws_internet_gateway.igw: Destruction complete after 1s
aws_vpn_gateway.vpn_gw: Still destroying... [id=vgw-0e2773e855525732f, 10s elapsed]
aws_instance.web[1]: Still destroying... [id=i-0a90f9ed4a0a98cbb, 10s elapsed]
aws_instance.web[0]: Still destroying... [id=i-0ad6dbf74b97dff3f, 10s elapsed]
```