

School of Computer Science
UNIVERSITY OF PETROLEUM AND ENERGY
STUDIES
DEHRADUN, UTTARAKHAND



**System configuration and
configuration management lab
for
6th Semester**

Submitted to:
Dr. Hitesh Kumar Sharma

Submitted by:
Gaurav Bhandari
Sap:500090993
Btech. CSE DevOps
DevOps-B1

SPCM LAB-1

Objective: Installing terraform on Mac-osX

- First, install the HashiCorp tap, a repository of all our Homebrew packages.

```
gauravbhandari@gauravs-MacBook-Air-2 ~ % brew tap hashicorp/tap
Running 'brew update --auto-update'...
==> Auto-updated Homebrew!
Updated 4 taps (hashicorp/tap, homebrew/services, homebrew/core and homebrew/cask).
==> New Formulae
git-grab          libnsgif          nowplaying-cli    scnlib            tfautomv
hashicorp/tap/tfstacks  libspelling       openjph           senpai            tomplusplus
icloudpd          limesuite         rsync             terrapin-scanner
==> New Casks
bitbox            cleanupbuddy       insomnia          nightshade        ttu-base-suite
bugdom2           domzilla-cafeine   lyricsfinder      theiaide
You have 25 outdated formulae and 1 outdated cask installed.
gauravbhandari@gauravs-MacBook-Air-2 ~ %
```

- Now, install Terraform with hashicorp/tap/terraform.

```
gauravbhandari@gauravs-MacBook-Air-2 ~ % brew install hashicorp/tap/terraform
==> Fetching hashicorp/tap/terraform
==> Downloading https://releases.hashicorp.com/terraform/1.7.1/terraform_1.7.1_darwin_arm64.zip
Already downloaded: /Users/gauravbhandari/Library/Caches/Homebrew/downloads/a9a9a6dfb024c2ab845b9e9eb25cabb393e0a44927b85e4377c470fdbd46bd39--terraform_1.7.1_darwin_arm64.zip
==> Installing terraform from hashicorp/tap
📦 /opt/homebrew/Cellar/terraform/1.7.1: 3 files, 88.6MB, built in 4 seconds
==> Running 'brew cleanup terraform'...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see 'man brew').
Removing: /Users/gauravbhandari/Library/Caches/Homebrew/terraform--1.6.6.zip... (23.7MB)
gauravbhandari@gauravs-MacBook-Air-2 ~ %
```

- To update to the latest version of Terraform, first update Homebrew.

```
gauravbhandari@gauravs-MacBook-Air-2 ~ % brew update
Already up-to-date.
```

- Then, run the upgrade command to download and use the latest Terraform version.

```
gauravbhandari@gauravs-MacBook-Air-2 ~ % brew upgrade hashicorp/tap/terraform
Warning: hashicorp/tap/terraform 1.7.1 already installed
gauravbhandari@gauravs-MacBook-Air-2 ~ %
```

- Check the installation of terraform

```
gauravbhandari@gauravs-MacBook-Air-2 ~ % terraform -v
Terraform v1.6.6
on darwin_arm64
```

SPCM LAB-2

Objective: Terraform AWS Provider and IAM User Setting

- Create a new directory for your Terraform configuration:

```
● gauravbhandari@gauravs-Air-2 Desktop % mkdir System-provisioning
● gauravbhandari@gauravs-Air-2 Desktop % cd System-provisioning
```

- Create a file named main.tf with the following content:

```
● gauravbhandari@gauravs-Air-2 System-provisioning % touch main.tf
```

```
main.tf > ...
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "5.31.0"
6     }
7   }
8 }
9 provider "aws" {
10   region = "ap-south-1"
11   access_key = "your IAM access key"
12   secret_key = "your secret access key"
13 }
```

- Run the following command to initialize your Terraform working directory.

```
● gauravbhandari@gauravs-Air-2 System-provisioning % terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.31.0"...
- Installing hashicorp/aws v5.31.0...
- Installed hashicorp/aws v5.31.0 (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!
```

SPCM LAB-3

Objective: Provisioning an EC2 Instance on AWS

- Create Terraform Configuration File for EC2 instance(instance.tf):

```
instance.tf ×
instance.tf > resource "aws_instance" "UPES1"
1  resource "aws_instance" "UPES1" {
2      ami            = "ami-03f4878755434977f"
3      instance_type = "t2.micro"
4      count          = 1
5      tags = {
6          Name = "My-ec2"
7      }
8  }
```

- Run the command “terraform plan” and review the plan to ensure it aligns with your expectations.

```
gauravbhandari@gauravs-MacBook-Air-2 aws-terraform-demo % 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.UPES1[0] will be created
+ resource "aws_instance" "UPES1" {
+   ami            = "ami-03f4878755434977f"
+   arn            = (known after apply)
+   user_data_replace_on_change = false
+   vpc_security_group_ids      = (known after apply)
}

Plan: 1 to add, 0 to change, 0 to destroy.

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.
```

- Apply the changes to create the AWS resources terraform apply.Type yes when prompted.

```
gauravbhandari@gauravs-MacBook-Air-2 aws-terraform-demo % terraform apply --auto-approve

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.UPES1[0] will be created
+ resource "aws_instance" "UPES1" {
+   ami            = "ami-03f4878755434977f"
+   arn            = (known after apply)
+   user_data_replace_on_change = false
+   vpc_security_group_ids      = (known after apply)
}
```

```
Plan: 1 to add, 0 to change, 0 to destroy.
aws_instance.UPES1[0]: Creating...
aws_instance.UPES1[0]: Still creating... [10s elapsed]
aws_instance.UPES1[0]: Still creating... [20s elapsed]
aws_instance.UPES1[0]: Still creating... [30s elapsed]
aws_instance.UPES1[0]: Creation complete after 34s [id=i-0f3146c2118b61784]
```

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

- After the terraform apply command completes, log in to your AWS Management Console and navigate to the EC2 dashboard. Verify that the EC2 instance has been created.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
My-ec2	i-0f3146c2118b61784	Running	t2.micro	2/2 checks passed	View alarms	ap-south-1a

- When you are done experimenting, run the following command to destroy the created resources.

```
gauravbhandari@gauravs-MacBook-Air-2 aws-terraform-demo % terraform destroy --auto-approve
aws_instance.UPES1[0]: Refreshing state... [id=i-0f3146c2118b61784]
```

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

- destroy

Terraform will perform the following actions:

```
Plan: 0 to add, 0 to change, 1 to destroy.
aws_instance.UPES1[0]: Destroying... [id=i-0f3146c2118b61784]
aws_instance.UPES1[0]: Still destroying... [id=i-0f3146c2118b61784, 10s elapsed]
aws_instance.UPES1[0]: Still destroying... [id=i-0f3146c2118b61784, 20s elapsed]
aws_instance.UPES1[0]: Still destroying... [id=i-0f3146c2118b61784, 30s elapsed]
aws_instance.UPES1[0]: Destruction complete after 31s
```

Destroy complete! Resources: 1 destroyed.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
My-ec2	i-0f3146c2118b61784	Shutting-down	t2.micro	-	View alarms	ap-south-1a

SPCM LAB-4

Objective: Learn how to define and use variables in Terraform configuration.

- Create a file variable.tf with the following contents.

```
gauravbhandari@gauravs-MacBook-Air-2 aws-terraform-demo % touch variable.tf
```

```
variable.tf > variable "countNumber"
1  variable "ubuntu_ami" {
2    type    = string
3    default = "ami-03f4878755434977f"
4  }
5  variable "instance_type" {
6    type    = string
7    default = "t2.micro"
8  }
9  variable "countNumber" {
10   type    = number
11   default = 1
12 }
```

- Use the variable declared and defined in variable.tf in instance.tf

```
instance.tf > ...
1  resource "aws_instance" "UPES1" {
2    ami            = var.ubuntu_ami
3    instance_type  = var.instance_type
4    count          = var.countNumber
5    tags = {
6      Name = "My-ec2"
7    }
8  }
9
```

- Run the command `terraform plan` and review the plan to see if it meets your expectations

```

gauravbhandari@gauravs-MacBook-Air-2 aws-terraform-demo % 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.UPEs1[0] will be created
+ resource "aws_instance" "UPES1" {
  + ami                        = "ami-03f4878755434977f"
  + arn                       = (known after apply)
  + user_data_replace_on_change = false
  + vpc_security_group_ids     = (known after apply)
}

Plan: 1 to add, 0 to change, 0 to destroy.

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.

```

- Run `terraform apply` and create the resources.

```

gauravbhandari@gauravs-MacBook-Air-2 aws-terraform-demo % terraform apply --auto-approve

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

Terraform will perform the following actions:

# aws_instance.UPEs1[0] will be created
+ resource "aws_instance" "UPES1" {
  + ami                        = "ami-03f4878755434977f"
  + arn                       = (known after apply)
  + user_data_replace_on_change = false
  + vpc_security_group_ids     = (known after apply)
}

Plan: 1 to add, 0 to change, 0 to destroy.
aws_instance.UPEs1[0]: Creating...
aws_instance.UPEs1[0]: Still creating... [10s elapsed]
aws_instance.UPEs1[0]: Still creating... [20s elapsed]
aws_instance.UPEs1[0]: Still creating... [30s elapsed]
aws_instance.UPEs1[0]: Creation complete after 33s [id=i-07be4acb91cff5340]

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

```

[Option+S]								
Mumbai gaurav @ gaurav1954								
Instances (1) Info								
Find Instance by attribute or tag (case-sensitive)								
Running								
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Publ
<input type="checkbox"/>	My-ec2	i-07be4acb91cff5340	Running	t2.micro	Initializing	View alarms +	ap-south-1a	ec2-4

- When you are done experimenting, run the following command to destroy the created resources.

```
gauravbhandari@gauravs-Air-2 aws-terraform-demo % terraform destroy --auto-approve
aws_instance.UPES1[0]: Refreshing state... [id=i-07be4acb91cff5340]

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:
```

```
Plan: 0 to add, 0 to change, 1 to destroy.
aws_instance.UPES1[0]: Destroying... [id=i-07be4acb91cff5340]
aws_instance.UPES1[0]: Still destroying... [id=i-07be4acb91cff5340, 10s elapsed]
aws_instance.UPES1[0]: Still destroying... [id=i-07be4acb91cff5340, 20s elapsed]
aws_instance.UPES1[0]: Still destroying... [id=i-07be4acb91cff5340, 30s elapsed]
aws_instance.UPES1[0]: Destruction complete after 31s

Destroy complete! Resources: 1 destroyed.
```

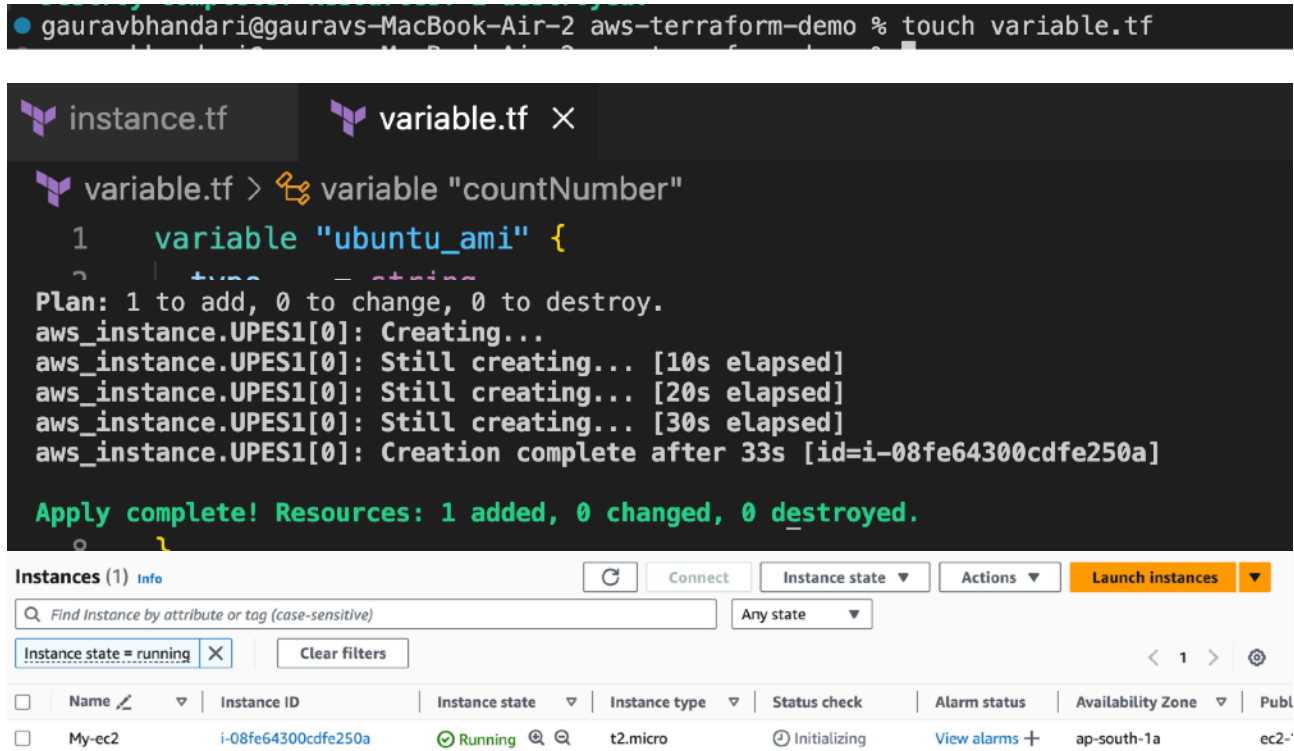
[Option+S]								
Mumbai gaurav @ gaurav1954								
Instances (2) Info								
Find Instance by attribute or tag (case-sensitive) Any state								
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Put
<input type="checkbox"/>	My-ec2	i-07be4acb91cff5340	Terminated	t2.micro	-	View alarms	ap-south-1a	-

SPCM LAB-5

Objective: Learn how to pass values to Terraform variables using command line arguments.

- Create a file variable.tf with the following contents.

```
gauravbhandari@gauravs-MacBook-Air-2 aws-terraform-demo % touch variable.tf
```



The screenshot shows the Terraform CLI interface with two tabs: 'instance.tf' and 'variable.tf'. The 'variable.tf' tab is active, displaying the following content:

```
variable "countNumber"
  type = number
  default = 1

variable "ubuntu_ami" {
  type = string
  default = "ami-08fe64300cdf250a"
}
```

The CLI output shows the plan and the successful creation of the EC2 instance:

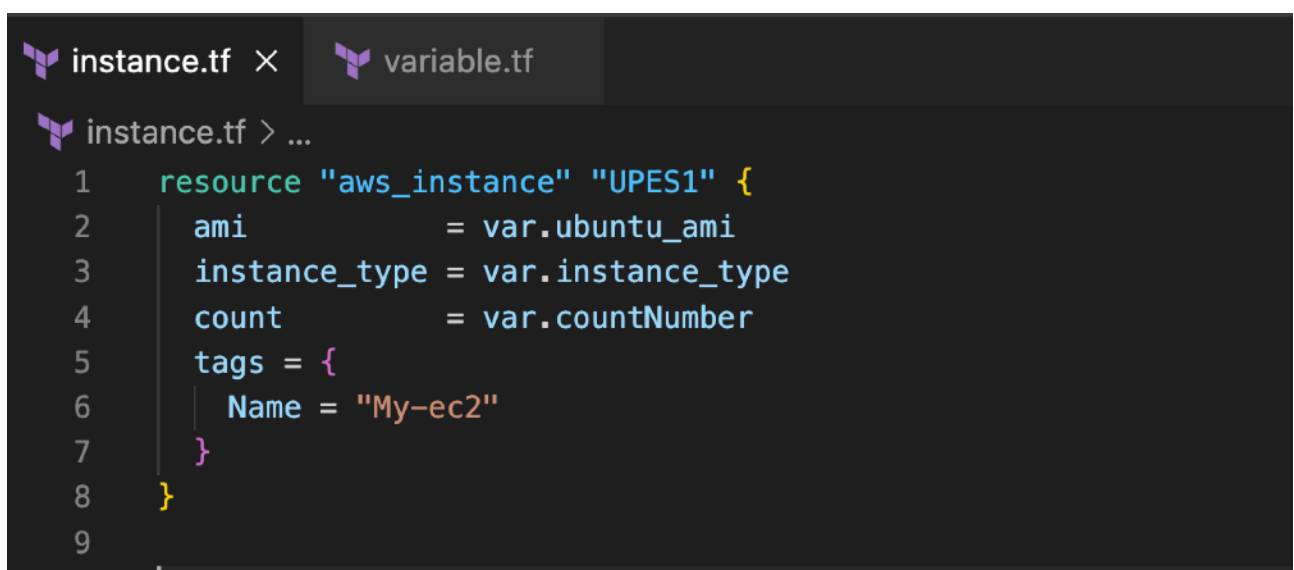
```
Plan: 1 to add, 0 to change, 0 to destroy.
aws_instance.UPE1[0]: Creating...
aws_instance.UPE1[0]: Still creating... [10s elapsed]
aws_instance.UPE1[0]: Still creating... [20s elapsed]
aws_instance.UPE1[0]: Still creating... [30s elapsed]
aws_instance.UPE1[0]: Creation complete after 33s [id=i-08fe64300cdf250a]

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

Below the CLI output, the AWS Management Console 'Instances' page is shown. It displays a table with one instance, 'My-ec2', which is in the 'Running' state. The instance ID is 'i-08fe64300cdf250a' and the instance type is 't2.micro'.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
My-ec2	i-08fe64300cdf250a	Running	t2.micro	Initializing	View alarms	ap-south-1a	ec2-

- Use the variable declared and defined in variable.tf in instance.tf



The screenshot shows the Terraform CLI interface with two tabs: 'instance.tf' and 'variable.tf'. The 'instance.tf' tab is active, displaying the following content:

```
resource "aws_instance" "UPE1" {
  ami           = var.ubuntu_ami
  instance_type = var.instance_type
  count         = var.countNumber
  tags = {
    Name = "My-ec2"
  }
}
```

- Run the command terraform plan and review the plan to see if it meets your expectations

```
gauravbhandari@gauravs-MacBook-Air-2 aws-terraform-demo % terraform plan -var "ubuntu_ami=ami-03f4878755434977f" -var "instance_type=t2.micro" -var "countNumber=1"

+ user_data_replace_on_change = false
+ vpc_security_group_ids      = (known after apply)
}

Plan: 1 to add, 0 to change, 0 to destroy.

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.
```

- Run terraform apply and create the resources.

```
gauravbhandari@gauravs-MacBook-Air-2 aws-terraform-demo % terraform apply -var "ubuntu_ami=ami-03f4878755434977f" -var "instance_type=t2.micro" -var "countNumber=1"
```

- When you are done experimenting, run the following command to destroy the created resources.

```
Plan: 0 to add, 0 to change, 1 to destroy.
aws_instance.UPES1[0]: Destroying... [id=i-08fe64300cdf250a]
aws_instance.UPES1[0]: Still destroying... [id=i-08fe64300cdf250a, 10s elapsed]
aws_instance.UPES1[0]: Still destroying... [id=i-08fe64300cdf250a, 20s elapsed]
aws_instance.UPES1[0]: Still destroying... [id=i-08fe64300cdf250a, 30s elapsed]
aws_instance.UPES1[0]: Destruction complete after 32s

Destroy complete! Resources: 1 destroyed.
```

[Option+S]								
Mumbai gaurav@gaurav1954								
Instances (1) Info								
<input type="text" value="Find Instance by attribute or tag (case-sensitive)"/> <input type="button" value="Any state"/>								
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Publ
<input type="checkbox"/>	My-ec2	i-08fe64300cdf250a	Shutting-d...	t2.micro	-	View alarms +	ap-south-1a	ec2-

SPCM LAB-6

Objective: Learn how to use multiple tfvars files in Terraform for different environments.

- Create multiple tfvar files
- Create a dev.tfvar file

```
instance.tf  variable.tf  dev.tfvars ×
dev.tfvars > ...
1  ubuntu_ami    = "ami-03f4878755434977f"
2  countNumber   = 2
3  instance_type = "t3.micro"
4
```

- Create a prod.tfvar file

```
prod.tfvars ●
prod.tfvars > # countNumber
1  ubuntu_ami    = "ami-03f4878755434977f"
2  countNumber   = 1
3  instance_type = "t2.micro"
4
```

- Run the terraform apply -var-file=dev.tfvars commands to initialize and apply the configuration for the dev environment:

```
gauravbhandari@gauravs-Air-2 aws-terraform-demo % terraform apply -var-file=dev.tfvars

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.UPES1[0] will be created
+ resource "aws_instance" "UPES1" {
+   ami           = "ami-03f4878755434977f"
```

```
Only 'yes' will be accepted to approve.

Enter a value: yes

aws_instance.UPES1[0]: Creating...
aws_instance.UPES1[1]: Creating...
aws_instance.UPES1[0]: Still creating... [10s elapsed]
aws_instance.UPES1[1]: Still creating... [10s elapsed]
aws_instance.UPES1[1]: Creation complete after 12s [id=i-04fed2da1d1561dc0]
aws_instance.UPES1[0]: Creation complete after 12s [id=i-03d35a2e431a382b2]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
```

Instances (2) Info								
Find Instance by attribute or tag (case-sensitive)				Any state				
Instance state = running X				Clear filters				
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
<input type="checkbox"/>	My-ec2	i-04fed2da1d1561dc0	Running	t3.micro	Initializing	View alarms +	ap-south-1a	ec2-...
<input type="checkbox"/>	My-ec2	i-03d35a2e431a382b2	Running	t3.micro	Initializing	View alarms +	ap-south-1a	ec2-...

- Run the following Terraform commands to initialize and apply the configuration for the prod environment:

```
gauravbhandari@gauravs-Air-2 aws-terraform-demo % terraform apply -var-file=prod.tfvars
aws_instance.UPEs1[0]: Refreshing state... [id=i-03d35a2e431a382b2]
aws_instance.UPEs1[1]: Refreshing state... [id=i-04fed2da1d1561dc0]

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

Terraform will perform the following actions:
```

```
aws_instance.UPEs1[0]: Still modifying... [id=i-03d35a2e431a382b2, 50s elapsed]
aws_instance.UPEs1[0]: Still modifying... [id=i-03d35a2e431a382b2, 1m0s elapsed]
aws_instance.UPEs1[0]: Still modifying... [id=i-03d35a2e431a382b2, 1m10s elapsed]
aws_instance.UPEs1[0]: Still modifying... [id=i-03d35a2e431a382b2, 1m20s elapsed]
aws_instance.UPEs1[0]: Still modifying... [id=i-03d35a2e431a382b2, 1m30s elapsed]
aws_instance.UPEs1[0]: Still modifying... [id=i-03d35a2e431a382b2, 1m40s elapsed]
aws_instance.UPEs1[0]: Modifications complete after 1m42s [id=i-03d35a2e431a382b2]

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

Instances (2) Info								
Find Instance by attribute or tag (case-sensitive)				Any state				
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
<input type="checkbox"/>	My-ec2	i-04fed2da1d1561dc0	Terminated	t3.micro	-	View alarms +	ap-south-1a	-
<input type="checkbox"/>	My-ec2	i-03d35a2e431a382b2	Running	t2.micro	Initializing	View alarms +	ap-south-1a	ec2-...

- After successful experimentation clean up the environment using terraform destroy -var-file=prod.tfvars

```
gauravbhandari@gauravs-Air-2 aws-terraform-demo % terraform destroy -var-file=prod.tfvars --auto-approve
aws_instance.UPEs1[0]: Refreshing state... [id=i-03d35a2e431a382b2]

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.UPEs1[0] will be destroyed
- resource "aws_instance" "UPEs1" {
  ami           = "ami-03f4878755434077f"
  instance_type = "t2.micro"
  ...
}
```

```
Plan: 0 to add, 0 to change, 1 to destroy.
aws_instance.UPEs1[0]: Destroying... [id=i-03d35a2e431a382b2]
aws_instance.UPEs1[0]: Still destroying... [id=i-03d35a2e431a382b2, 10s elapsed]
aws_instance.UPEs1[0]: Still destroying... [id=i-03d35a2e431a382b2, 20s elapsed]
aws_instance.UPEs1[0]: Still destroying... [id=i-03d35a2e431a382b2, 30s elapsed]
aws_instance.UPEs1[0]: Destruction complete after 30s

Destroy complete! Resources: 1 destroyed.
```

Instances (2) Info								
Find Instance by attribute or tag (case-sensitive)				Any state				
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
<input type="checkbox"/>	My-ec2	i-04fed2da1d1561dc0	Terminated	t3.micro	-	View alarms +	ap-south-1a	-
<input type="checkbox"/>	My-ec2	i-03d35a2e431a382b2	Shutting-d...	t2.micro	-	View alarms +	ap-south-1a	ec2-...

SPCM LAB-7

Objective: Learn how to use Terraform to create multiple IAM users with unique settings.

- Create two files, main.tf and run the commands terraform init

```
main.tf > ...
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "5.31.0"
6     }
7   }
8 }
9 provider "aws" {
10   region = "ap-south-1"
11   access_key = "your IAM access key"
12   secret_key = "your secret access key"
13 }
```

```
gauravbhandari@gauravs-Air-2 System-provisioning % terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.31.0"...
- Installing hashicorp/aws v5.31.0...
- Installed hashicorp/aws v5.31.0 (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!
```

- Create another file named i_am_users.tf

```
i_am_users.tf ×
i_am_users.tf > ...
1 resource "aws_iam_user" "my_iam_user" {
2   count = length(var.iam_users)
3   name = var.iam_users[count.index]
4   tags = {
5     Name = "${var.iam_users[count.index]}-upes"
6   }
7 }
8 variable "iam_users" {
9   type = list(string)
10  default = ["user1", "user2", "user3"]
11 }
```

- Run the command terraform apply -auto-approve

```
gauravbhandari@gauravs-Air-2 aws-terraform-demo % terraform apply -auto-approve

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

```
Plan: 3 to add, 0 to change, 0 to destroy.
aws_iam_user.my_iam_user[2]: Creating...
aws_iam_user.my_iam_user[1]: Creating...
aws_iam_user.my_iam_user[0]: Creating...
aws_iam_user.my_iam_user[2]: Creation complete after 2s [id=user3]
aws_iam_user.my_iam_user[1]: Creation complete after 2s [id=user2]
aws_iam_user.my_iam_user[0]: Creation complete after 3s [id=user1]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
```

- Check the aws console and verify the creation of required resources.

Users (4) Info Refresh Delete Create user

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

Search < 1 > Settings

<input type="checkbox"/>	User name	Path	Group	Last activity	MFA	Password age	Con
<input type="checkbox"/>	gaurav	/	0		-	✓ 27 days	Feb
<input type="checkbox"/>	user1	/	0		-	-	-
<input type="checkbox"/>	user2	/	0		-	-	-
<input type="checkbox"/>	user3	/	0		-	-	-

- After successful experimentation run terraform apply -auto-approve to destroy the resources.

```
gauravbhandari@gauravs-Air-2 aws-terraform-demo % terraform destroy -auto-approve
aws_iam_user.my_iam_user[0]: Refreshing state... [id=user1]
aws_iam_user.my_iam_user[2]: Refreshing state... [id=user3]
aws_iam_user.my_iam_user[1]: Refreshing state... [id=user2]
```

```
Plan: 0 to add, 0 to change, 3 to destroy.
aws_iam_user.my_iam_user[1]: Destroying... [id=user2]
aws_iam_user.my_iam_user[2]: Destroying... [id=user3]
aws_iam_user.my_iam_user[0]: Destroying... [id=user1]
aws_iam_user.my_iam_user[1]: Destruction complete after 2s
aws_iam_user.my_iam_user[2]: Destruction complete after 2s
aws_iam_user.my_iam_user[0]: Destruction complete after 2s

Destroy complete! Resources: 3 destroyed.
```

SPCM LAB-8

Objective: Learn how to use Terraform to create a basic Virtual Private Cloud (VPC) in AWS.

- Create a VPC tf file and run terraform apply -auto-approve

```
vpc.tf > resource "aws_subnet" "sub1" > tags > Name
1  resource "aws_vpc" "my_vpc" {
2      cidr_block = "10.0.0.0/16"
3
4      tags = {
5          Name = "myvpc"
6      }
7  }
8
9  resource "aws_subnet" "sub1" {
10     count      = 2
11     vpc_id     = aws_vpc.my_vpc.id
12     cidr_block = "10.0.${count.index}.0/24"
13     tags = {
14         Name = "My-subnet-${count.index + 1}"
15     }
16 }
```

```
gauravbhandari@gauravs-Air-2 aws-terraform-demo % terraform apply -auto-approve
```

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

- + create

```
Plan: 3 to add, 0 to change, 0 to destroy.
aws_vpc.my_vpc: Creating...
aws_vpc.my_vpc: Creation complete after 2s [id=vpc-0d78d16f0eeb94678]
aws_subnet.sub1[0]: Creating...
aws_subnet.sub1[1]: Creating...
aws_subnet.sub1[1]: Creation complete after 0s [id=subnet-040b83f3ed0058874]
aws_subnet.sub1[0]: Creation complete after 0s [id=subnet-0940302ced58ed075]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
```

- Check the aws console and verify the creation of required resources.

Your VPCs (2) Info								Actions	Create VPC
<input type="text" value="Search"/>									
<input type="checkbox"/>	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR				
<input type="checkbox"/>	myvpc	vpc-001be7140d7ccb30e	Available	10.0.0.0/16	-				
<input type="checkbox"/>	-	vpc-0ee329ac4f6dfbab7	Available	172.31.0.0/16	-				

- After experimentation run terraform destroy -auto-approve

```
gauravbhandari@gauravs-Air-2 aws-terraform-demo % terraform destroy -auto-approve
aws_vpc.my_vpc: Refreshing state... [id=vpc-001be7140d7ccb30e]
aws_subnet.sub1[1]: Refreshing state... [id=subnet-05543105e8dbf47c2]
aws_subnet.sub1[0]: Refreshing state... [id=subnet-0105bdbdabb3dd032]
```

```
Plan: 0 to add, 0 to change, 3 to destroy.
aws_subnet.sub1[0]: Destroying... [id=subnet-0105bdbdabb3dd032]
aws_subnet.sub1[1]: Destroying... [id=subnet-05543105e8dbf47c2]
aws_subnet.sub1[0]: Destruction complete after 0s
aws_subnet.sub1[1]: Destruction complete after 0s
aws_vpc.my_vpc: Destroying... [id=vpc-001be7140d7ccb30e]
aws_vpc.my_vpc: Destruction complete after 1s
```

Destroy complete! Resources: 3 destroyed.

SPCM LAB-9

Objective: Creating multiple EC2 instances with for_each in terraform.

- Create a instance.tf file with the following contents

```
main.tf  instance.tf x
9th > instance.tf > resource "aws_instance" "ec2-instances" > tags > Name
1  variable "instances" {
2      description = "map of ec2 with settings"
3      default = {
4          "instance1" = {
5              ami         = "ami-03f4878755434977f"
6              instance_type = "t2.micro"
7          }
8          "instance2" = {
9              ami         = "ami-03f4878755434977f"
10             instance_type = "t2.micro"
11         }
12         "instance3" = {
13             ami         = "ami-03f4878755434977f"
14             instance_type = "t2.micro"
15         }
16     }
17 }
18 resource "aws_instance" "ec2-instances" {
19     for_each      = var.instances
20     ami           = var.instances[each.key].ami
21     instance_type = var.instances[each.key].instance_type
22     tags = {
23         Name = "My-Ec2-${each.key}"
24     }
25 }
```

- Run the command terraform plan to check that the configuration aligns with your requirements.

```
gauravbhandari@gauravs-Air-2 9th % 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.ec2-instances["instance1"] will be created
+ resource "aws_instance" "ec2-instances" {
```

```
    + "Name" = "My-Ec2-instance3"
  }
+ tenancy           = (known after apply)
+ user_data         = (known after apply)
+ user_data_base64  = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids = (known after apply)
}
```

Plan: 3 to add, 0 to change, 0 to destroy.

- Run the command terraform apply to create the intended resources.

```

actions if you run terraform apply now
gauravbhandari@gauravs-Air-2 9th % terraform apply -auto-approve

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:

Plan: 3 to add, 0 to change, 0 to destroy.
aws_instance.ec2-instances["instance3"]: Creating...
aws_instance.ec2-instances["instance2"]: Creating...
aws_instance.ec2-instances["instance1"]: Creating...
aws_instance.ec2-instances["instance3"]: Still creating... [10s elapsed]
aws_instance.ec2-instances["instance1"]: Still creating... [10s elapsed]
aws_instance.ec2-instances["instance2"]: Still creating... [10s elapsed]
aws_instance.ec2-instances["instance1"]: Still creating... [20s elapsed]
aws_instance.ec2-instances["instance3"]: Still creating... [20s elapsed]
aws_instance.ec2-instances["instance2"]: Still creating... [20s elapsed]
aws_instance.ec2-instances["instance1"]: Still creating... [30s elapsed]
aws_instance.ec2-instances["instance2"]: Still creating... [30s elapsed]
aws_instance.ec2-instances["instance3"]: Still creating... [30s elapsed]
aws_instance.ec2-instances["instance2"]: Creation complete after 32s [id=i-05cb0021d91a7d2ba]
aws_instance.ec2-instances["instance3"]: Creation complete after 32s [id=i-04580b6ca87309bee]
aws_instance.ec2-instances["instance1"]: Creation complete after 32s [id=i-03c57e65f0dd6d4ad]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

```

- Verify the same from your AWS console.

Instances (8) Info

Find Instance by attribute or tag (case-sensitive)

Any state

< 1 >

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm
<input type="checkbox"/>	promo	i-02228e104fafe704a	Stopped	t2.micro	–	View al
<input type="checkbox"/>	grafana	i-0b7379f749a6f3b53	Stopped	t2.micro	–	View al
<input type="checkbox"/>	ansible	i-079ddb535f62a560	Stopped	t2.micro	–	View al
<input type="checkbox"/>	node1	i-062789a6c96af8ebf	Stopped	t2.micro	–	View al
<input type="checkbox"/>	My-Ec2-instan...	i-03c57e65f0dd6d4ad	Running	t2.micro	Initializing	View al
<input type="checkbox"/>	My-Ec2-instan...	i-05cb0021d91a7d2ba	Running	t2.micro	Initializing	View al
<input type="checkbox"/>	My-Ec2-instan...	i-04580b6ca87309bee	Running	t2.micro	Initializing	View al

- After successful experimentation run terraform destroy to destroy the resources.

```

gauravbhandari@gauravs-Air-2 9th % terraform destroy
aws_instance.ec2-instances["instance3"]: Refreshing state... [id=i-04580b6ca87309bee]
aws_instance.ec2-instances["instance1"]: Refreshing state... [id=i-03c57e65f0dd6d4ad]
aws_instance.ec2-instances["instance2"]: Refreshing state... [id=i-05cb0021d91a7d2ba]

aws_instance.ec2-instances["instance1"]: Destroying... [id=i-03c57e65f0dd6d4ad]
aws_instance.ec2-instances["instance2"]: Destroying... [id=i-05cb0021d91a7d2ba]
aws_instance.ec2-instances["instance3"]: Destroying... [id=i-04580b6ca87309bee]
aws_instance.ec2-instances["instance2"]: Still destroying... [id=i-05cb0021d91a7d2ba, 10s elapsed]
aws_instance.ec2-instances["instance1"]: Still destroying... [id=i-03c57e65f0dd6d4ad, 10s elapsed]
aws_instance.ec2-instances["instance3"]: Still destroying... [id=i-04580b6ca87309bee, 10s elapsed]
aws_instance.ec2-instances["instance2"]: Still destroying... [id=i-05cb0021d91a7d2ba, 20s elapsed]
aws_instance.ec2-instances["instance3"]: Still destroying... [id=i-04580b6ca87309bee, 20s elapsed]
aws_instance.ec2-instances["instance1"]: Still destroying... [id=i-03c57e65f0dd6d4ad, 20s elapsed]
aws_instance.ec2-instances["instance3"]: Still destroying... [id=i-04580b6ca87309bee, 30s elapsed]
aws_instance.ec2-instances["instance1"]: Still destroying... [id=i-03c57e65f0dd6d4ad, 30s elapsed]
aws_instance.ec2-instances["instance2"]: Still destroying... [id=i-05cb0021d91a7d2ba, 30s elapsed]
aws_instance.ec2-instances["instance1"]: Destruction complete after 30s
aws_instance.ec2-instances["instance2"]: Destruction complete after 30s
aws_instance.ec2-instances["instance3"]: Destruction complete after 30s

Destroy complete! Resources: 3 destroyed.

```

SPCM LAB-10

Objective: Creating and AWS RDS instance using terraform

- Create a file rds.tf with the following contents

```
main.tf  rds.tf  ×
10th > rds.tf > resource "aws_db_instance" "my_RDS" > password
1  resource "aws_db_instance" "my_RDS" {
2      allocated_storage      = 10
3      db_name                 = "upes_db"
4      engine                 = "mysql"
5      engine_version         = "5.7"
6      instance_class         = "db.t2.micro"
7      username               = "admin"
8      password               = "admin1234"
9      skip_final_snapshot    = true
10     parameter_group_name    = "default.mysql5.7"
11
12 }
```

- Run terraform plan to check if the configurations align with your requirements

```
gauravbhandari@gauravs-Air-2 10th % 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_db_instance.my_RDS will be created
+ resource "aws_db_instance" "my_RDS" {
+ address                               = (known after apply)
+ allocated_storage                     = 10
```

Plan: 1 to add, 0 to change, 0 to destroy.

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.

- Run terraform apply to create your resource.

```
gauravbhandari@gauravs-Air-2 10th % terraform apply -auto-approve

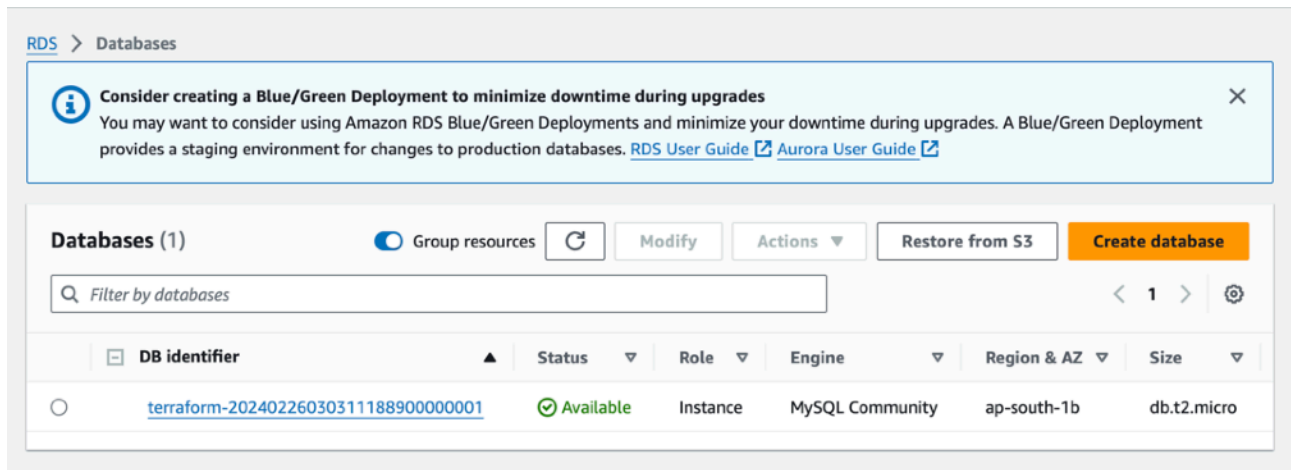
Terraform used the selected providers to generate the following execution plan. Resource actions are
indicated with the following symbols:
+ create

Terraform will perform the following actions:
```

```
aws_db_instance.my_RDS: Still creating... [3m20s elapsed]
aws_db_instance.my_RDS: Still creating... [3m30s elapsed]
aws_db_instance.my_RDS: Still creating... [3m40s elapsed]
aws_db_instance.my_RDS: Still creating... [3m50s elapsed]
aws_db_instance.my_RDS: Still creating... [4m0s elapsed]
aws_db_instance.my_RDS: Still creating... [4m10s elapsed]
aws_db_instance.my_RDS: Creation complete after 4m16s [id=db-D2Y5MANAJ4DGMU3QIQ07C5KWSQ]
```

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

- Verify resource creation from AWS console.



- After successful experimentation, run sql destroy to clean up the resources.

```
gauravbhandari@gauravs-Air-2 10th % terraform destroy -auto-approve
aws_db_instance.my_RDS: Refreshing state... [id=db-D2Y5MANAJ4DGMU3QIQ07C5KWSQ]

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_db_instance.my_RDS will be destroyed
- resource "aws_db_instance" "my_RDS" {
  - address = "terraform-20240226030311188900000001.c5ouuqgashuw.ap-south-1.
rds.amazonaws.com" -> null
```

```
aws_db_instance.my_RDS: Still destroying... [id=db-D2Y5MANAJ4DGMU3QIQ07C5KWSQ, 3m10s elapsed]
aws_db_instance.my_RDS: Still destroying... [id=db-D2Y5MANAJ4DGMU3QIQ07C5KWSQ, 3m20s elapsed]
aws_db_instance.my_RDS: Still destroying... [id=db-D2Y5MANAJ4DGMU3QIQ07C5KWSQ, 3m30s elapsed]
aws_db_instance.my_RDS: Still destroying... [id=db-D2Y5MANAJ4DGMU3QIQ07C5KWSQ, 3m40s elapsed]
aws_db_instance.my_RDS: Still destroying... [id=db-D2Y5MANAJ4DGMU3QIQ07C5KWSQ, 3m50s elapsed]
aws_db_instance.my_RDS: Still destroying... [id=db-D2Y5MANAJ4DGMU3QIQ07C5KWSQ, 4m0s elapsed]
aws_db_instance.my_RDS: Still destroying... [id=db-D2Y5MANAJ4DGMU3QIQ07C5KWSQ, 4m10s elapsed]
aws_db_instance.my_RDS: Destruction complete after 4m14s

Destroy complete! Resources: 1 destroyed.
```