SPCM LAB
LAB EXPERIMENTS

NAME: HIMANSH VATSAL

SAP ID: 500096347

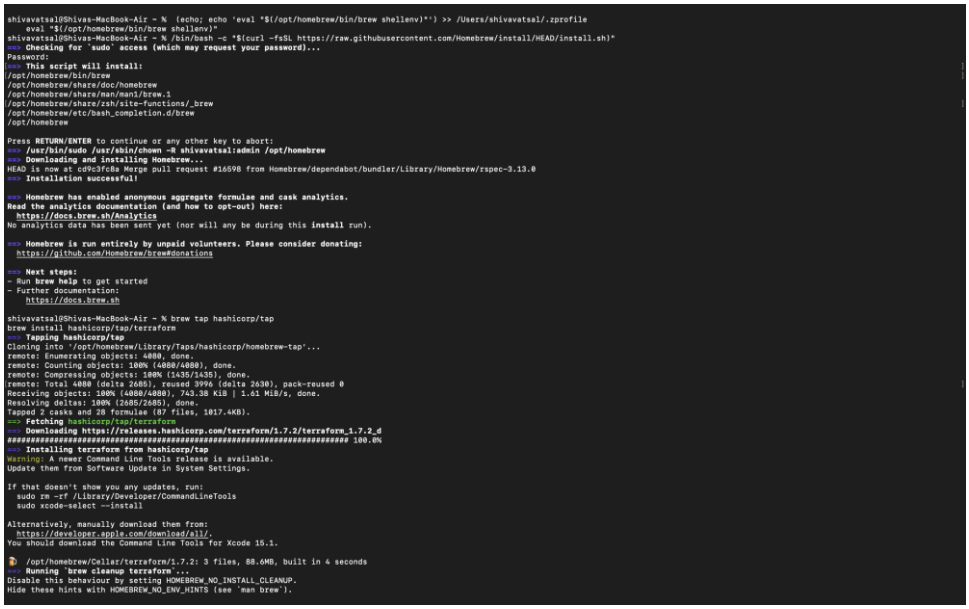
BATCH: B-3(DEVOPS)

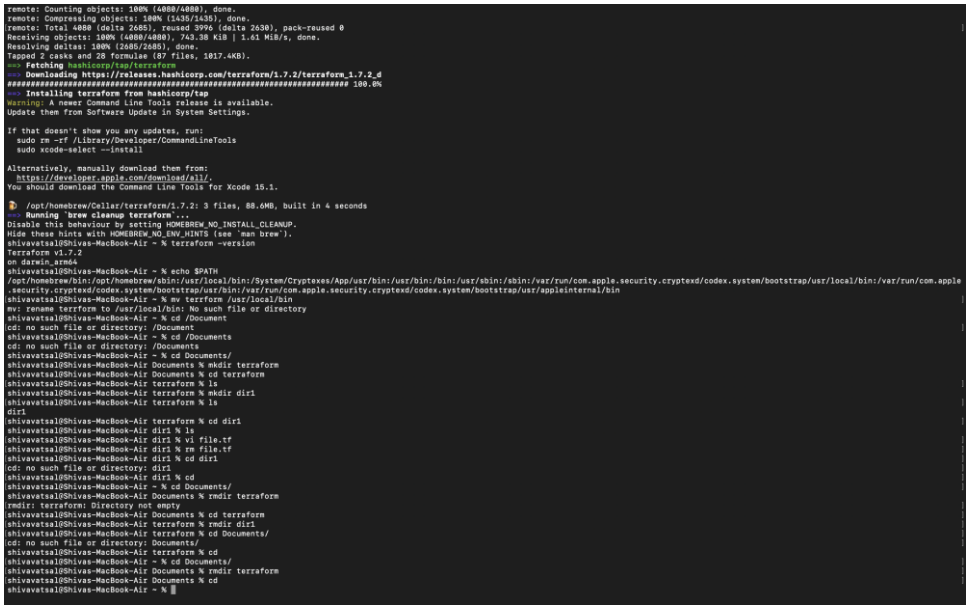
COURSE: SPCM

ROLL NO.: R2142211058

INSTRUCTOR: DR. HITESH SHARMA

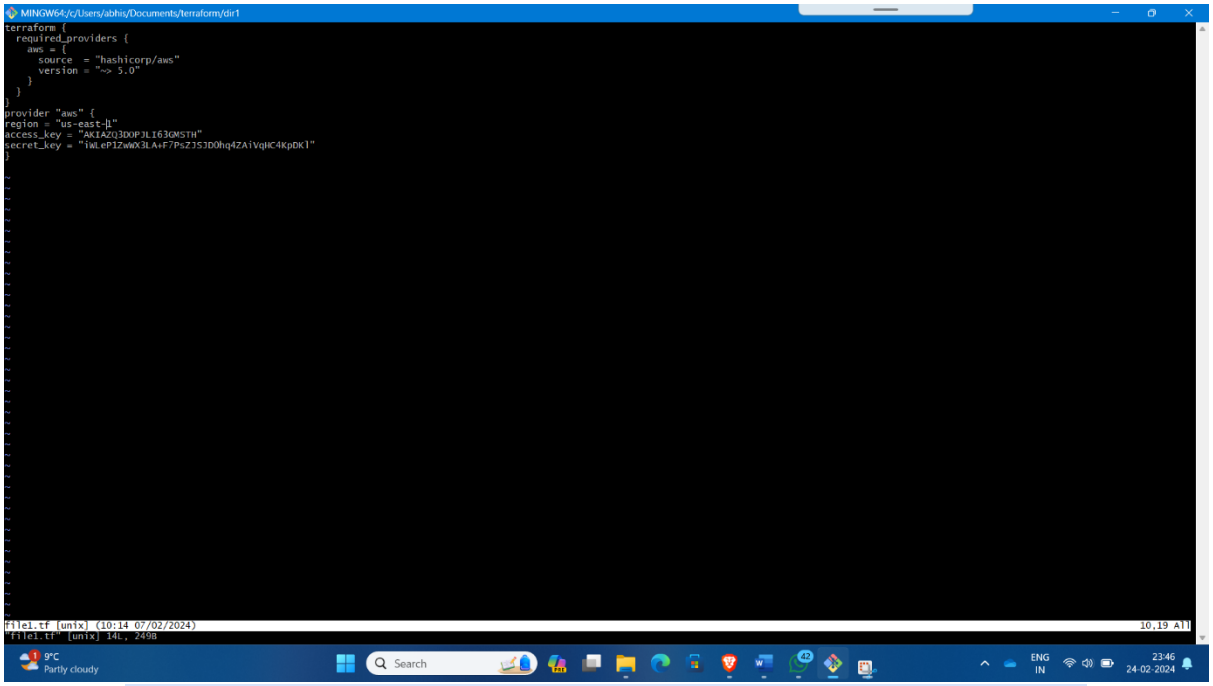
EXPERIMENT-1

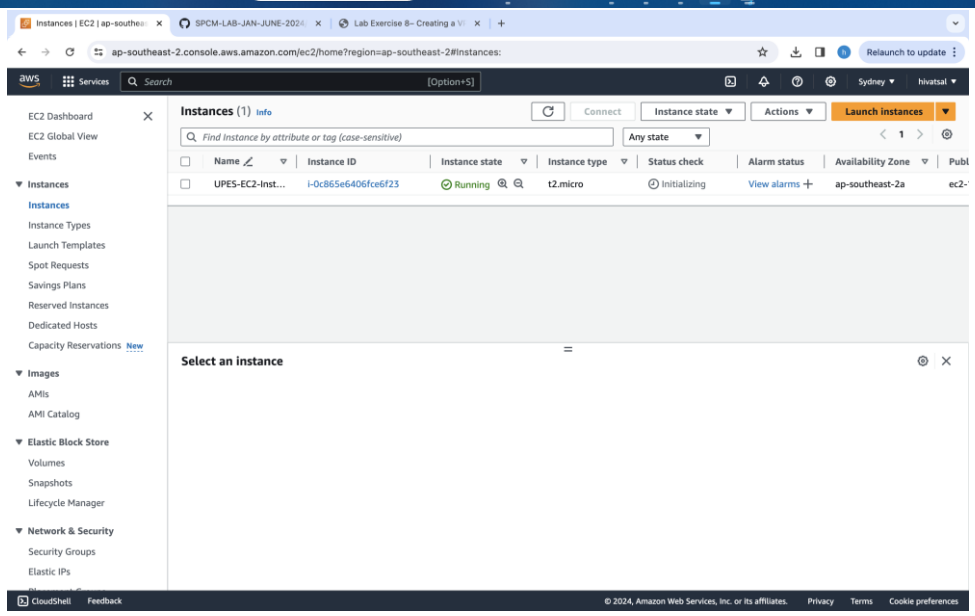




```
Last login: Thu Feb 22 21:31:29 on ttys004
shivavatsal@Shivas-Air ~ % terraform --version
Terraform v1.7.2
on darwin_arm64

Your version of Terraform is out of date! The latest version
is 1.7.4. You can update by downloading from https://www.terraform.io/downloads.html
shivavatsal@Shivas-Air ~ %
```

EXPERIMENT-2

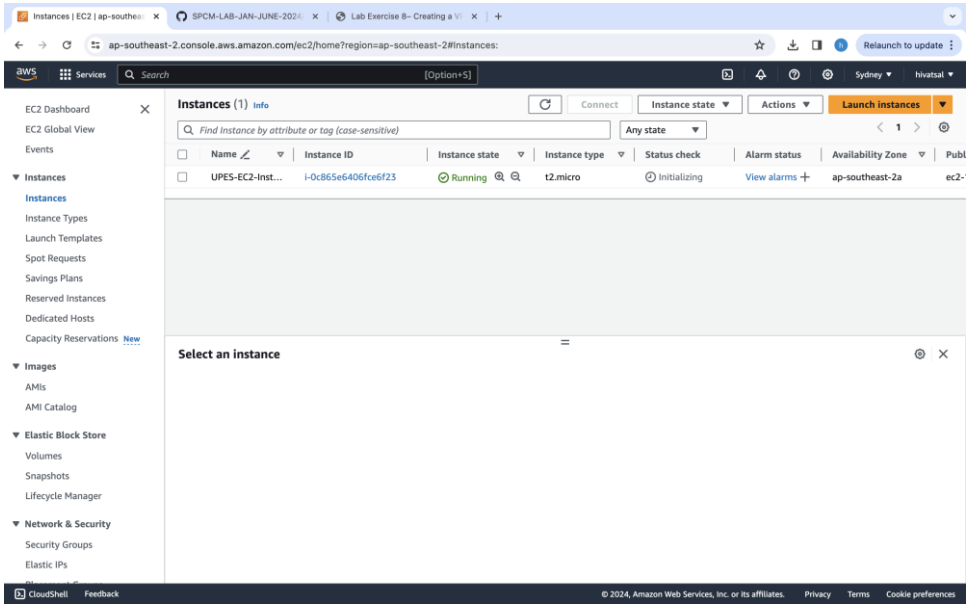

MINGW64:/c/Users/abhis/Documents/terraform/dir1
terraform {
required_providers {
aws = {
source = "hashicorp/aws"
version = "~> 5.0"
}
}
}
provider "aws" {
region = "us-east-1"
access_key = "AKIAZQ3DXPJLI63GM5TH"
secret_key = "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"
}
file1.tf [unix] (10:14 07/02/2024) 10,19 All
"file1.tf" [unix] 14L, 249B
9°C
Partly cloudy
Search
ENG
IN
23:46
24-02-2024


Instances | EC2 | ap-southea
SPCM-LAB-JAN-JUNE-2024
Lab Exercise 8– Creating a V
ap-southeast-2.console.aws.amazon.com/ec2/home?region=ap-southeast-2#Instances:
Relaunch to update
aws
Services
Search
[Option+S]
Sydney
hivatsal
EC2 Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations New
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Instances (1) Info
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
Any state
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
UPES-EC2-Inst...
i-0c865e6406fce6f23
Running
t2.micro
Initializing
View alarms
ap-southeast-2a
Select an instance
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

EXPERIMENT-3

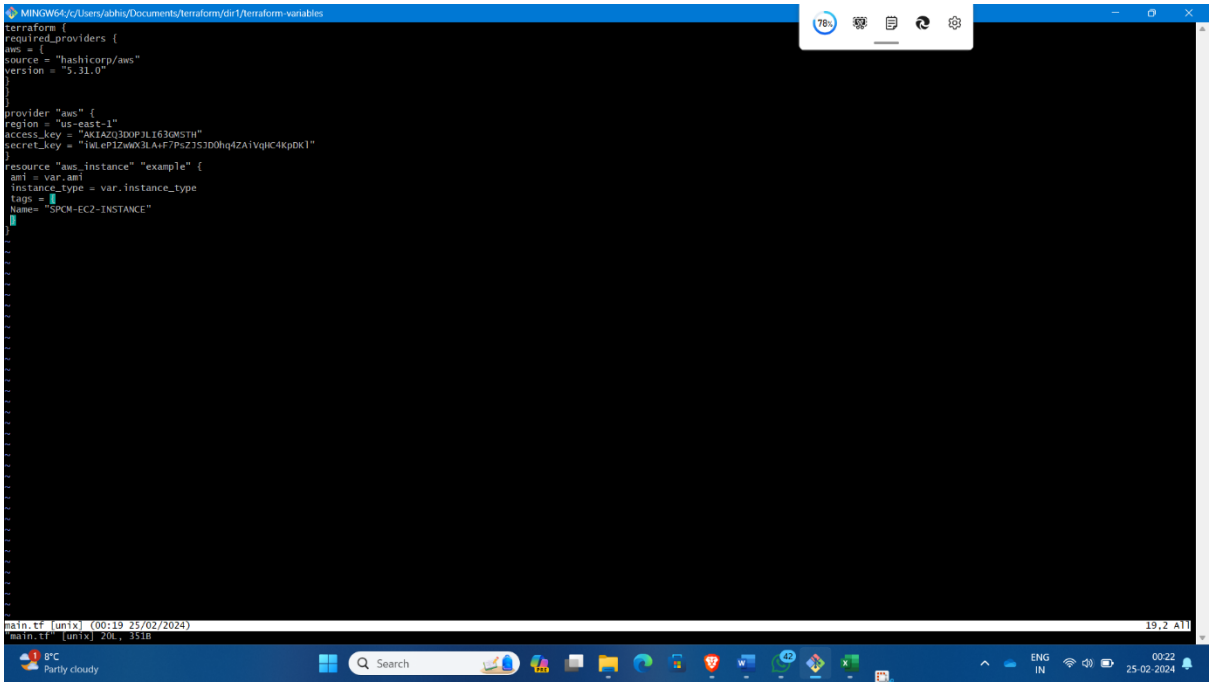
```
Destroy complete! Resources: 0 destroyed.
(shivavatsal@Shivas-MacBook-Air dir1 % ls
main.tf                 terraform-vpc           terraform.tfstate
(shivavatsal@Shivas-MacBook-Air dir1 % mkdir aws-terraform-demo
(shivavatsal@Shivas-MacBook-Air dir1 % cd aws-terraform-demo
(shivavatsal@Shivas-MacBook-Air aws-terraform-demo % vi main.tf
(shivavatsal@Shivas-MacBook-Air aws-terraform-demo % 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!

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.
(shivavatsal@Shivas-MacBook-Air aws-terraform-demo % vi instance.tf
(shivavatsal@Shivas-MacBook-Air aws-terraform-demo % terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.31.0

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.
(shivavatsal@Shivas-MacBook-Air 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.My-instance[0] will be created
  + resource "aws_instance" "My-instance" {
      + ami                                  = "ami-04f5097681773b989"
      + 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)
```

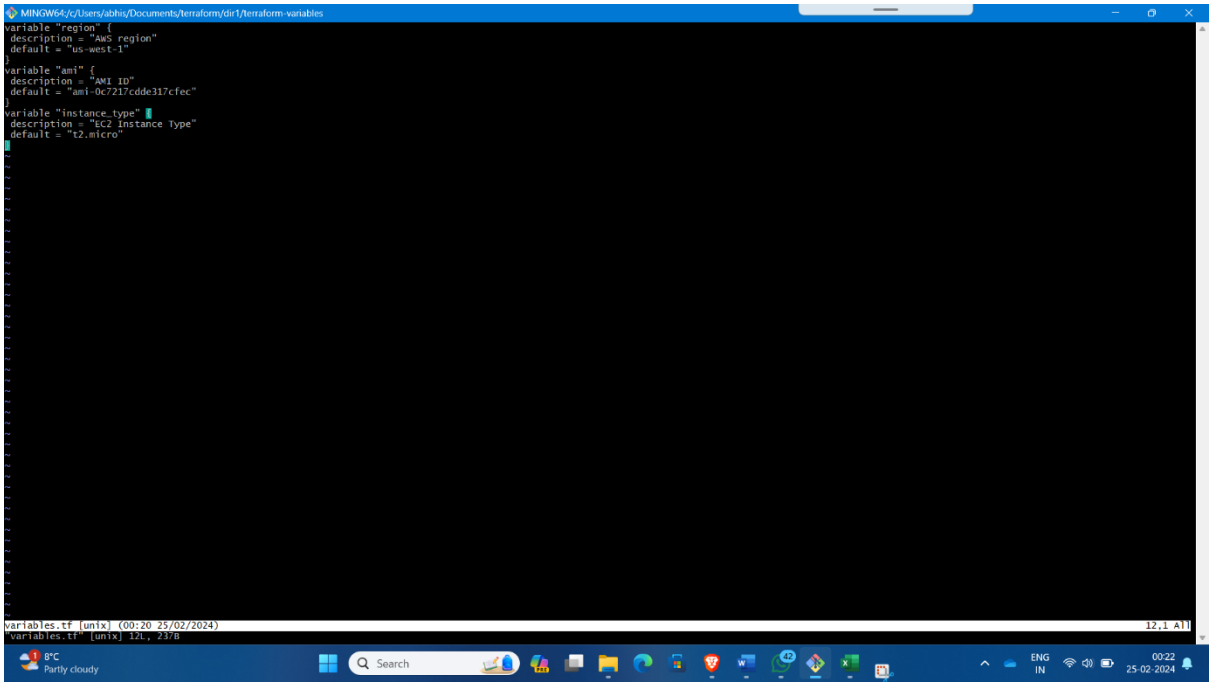
```
shivavatsal@Shivas-MacBook-Air aws-terraform-demo % vi main.tf
shivavatsal@Shivas-MacBook-Air aws-terraform-demo % ls
instance.tf         main.tf             terraform.tfstate
shivavatsal@Shivas-MacBook-Air aws-terraform-demo % terraform terraform plan
Terraform has no command named "terraform".

To see all of Terraform's top-level commands, run:
  terraform -help

shivavatsal@Shivas-MacBook-Air 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.My-instance[0] will be created
  + resource "aws_instance" "My-instance" {
      + ami                                  = "ami-04f5097681773b989"
      + 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_lifecycle                   = (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)
      + placement_partition_number           = (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
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Name" = "UPES-EC2-Instnace"
        }
      + tags_all                             = {
          + "Name" = "UPES-EC2-Instnace"
        }
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
```

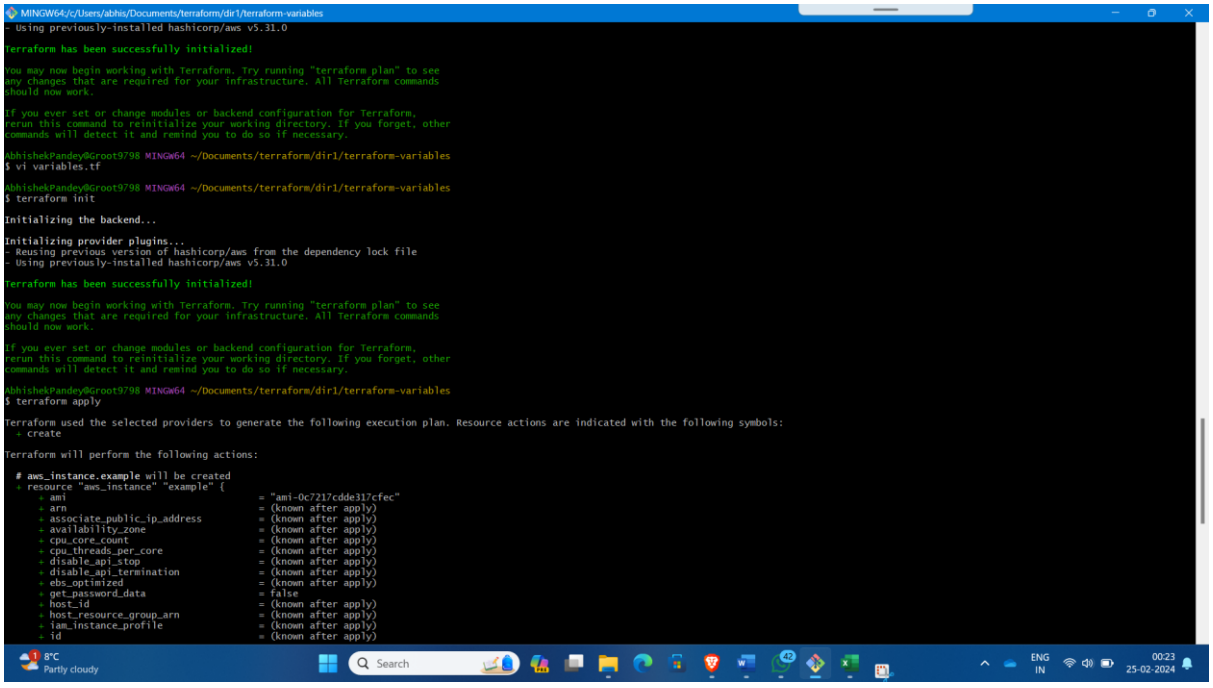


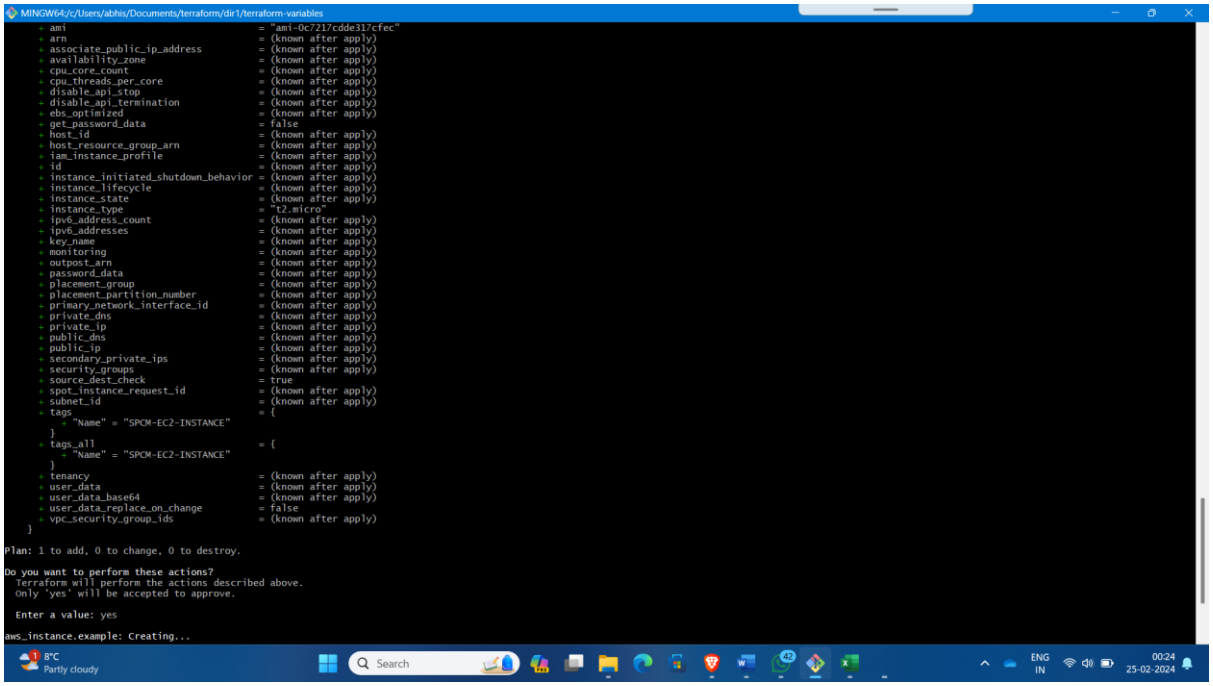
EXPERIMENT-4

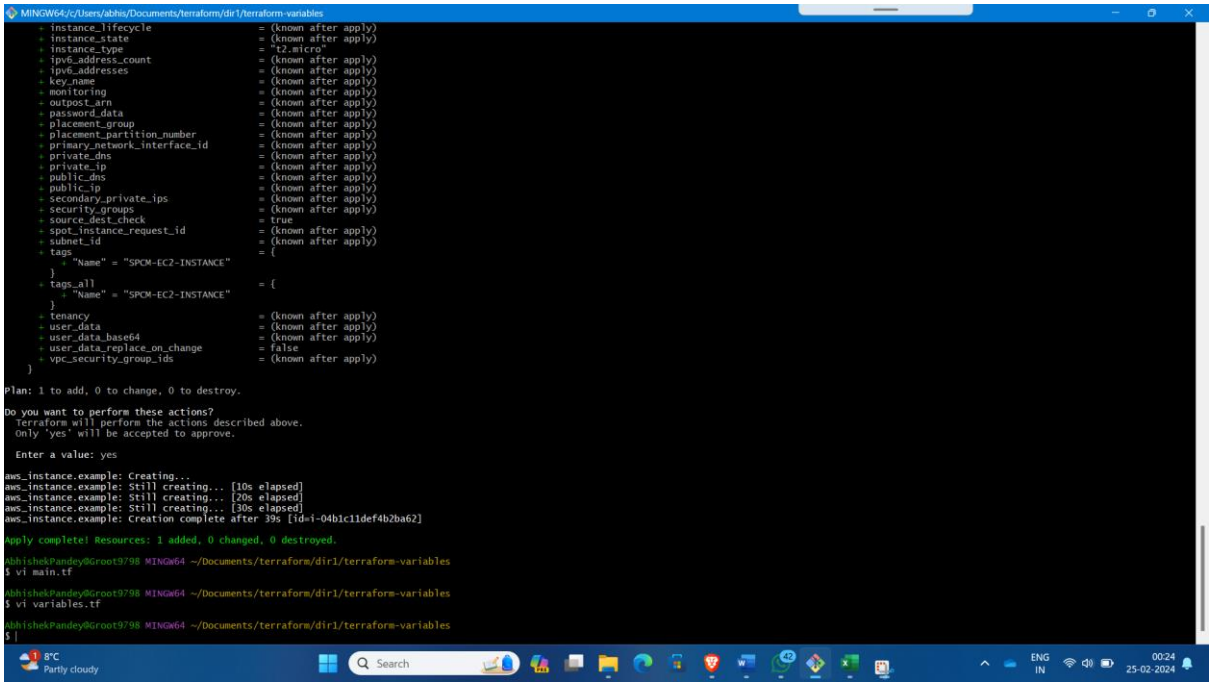
```
AbhishekPandey@Groot9798 MINGW64 ~/Documents/terraform/dir1
$ mkdir terraform-variables

AbhishekPandey@Groot9798 MINGW64 ~/Documents/terraform/dir1
$ cd terraform-variables

AbhishekPandey@Groot9798 MINGW64 ~/Documents/terraform/dir1/terraform-variables
$ vi main.tf

AbhishekPandey@Groot9798 MINGW64 ~/Documents/terraform/dir1/terraform-variables
$ 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!

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.
```

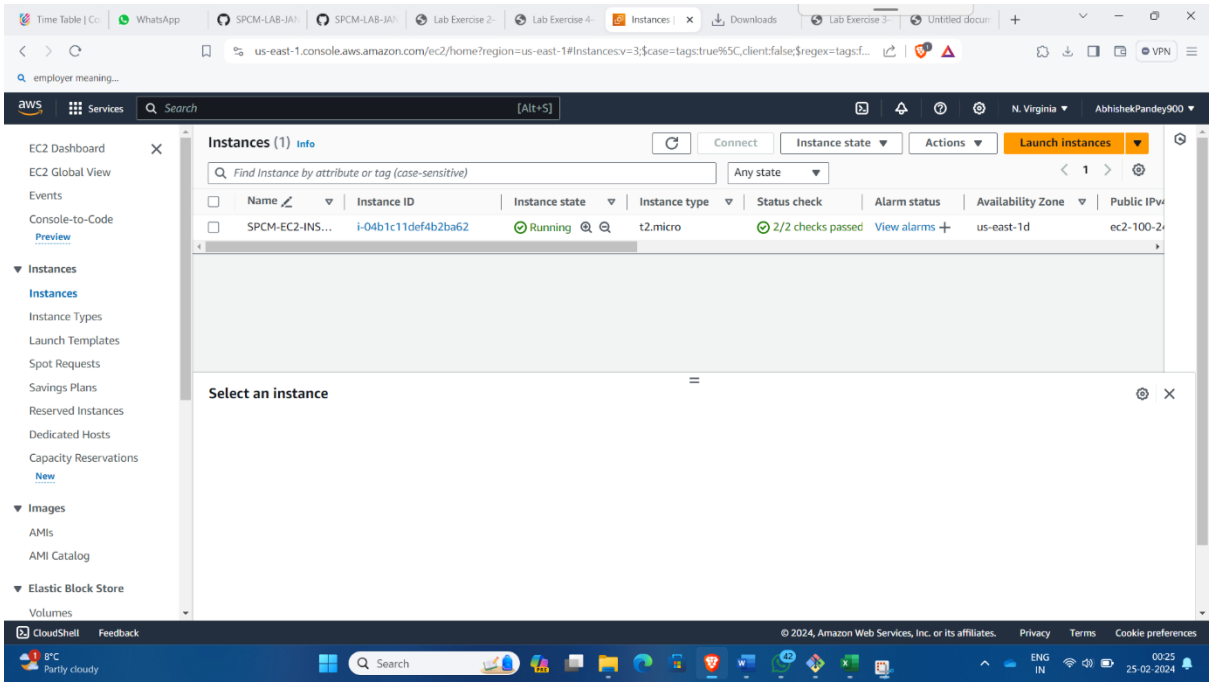


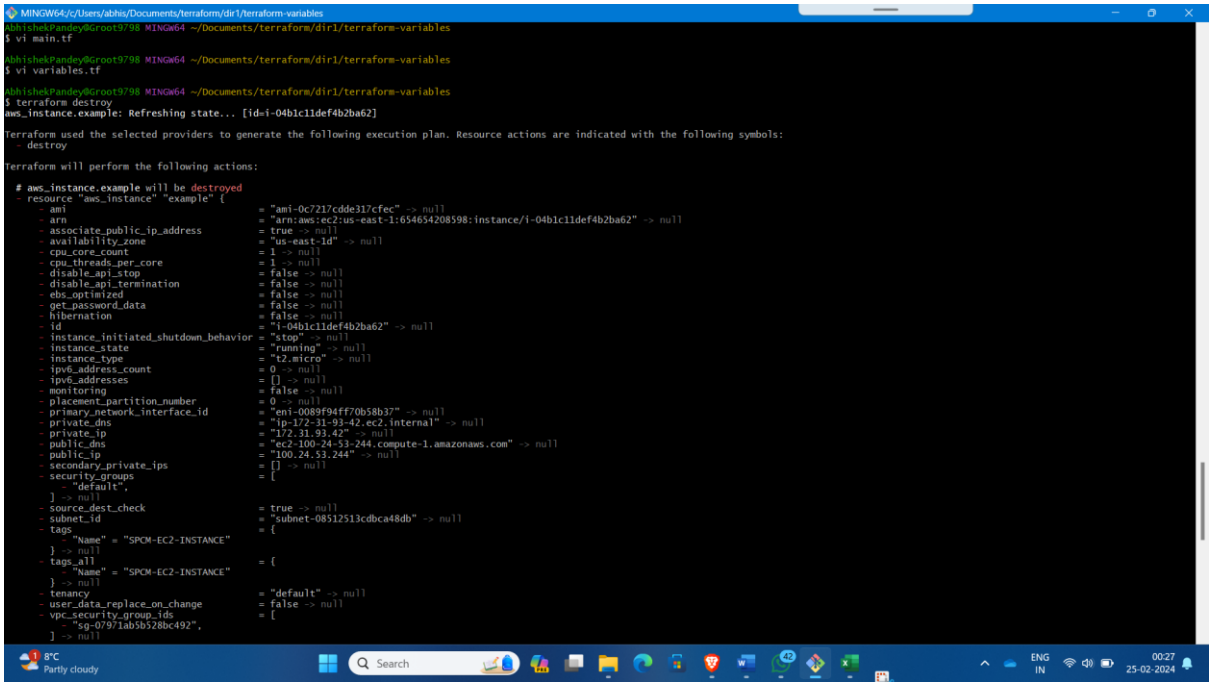

MINGW64:/c/Users/abhis/Documents/terraform/dir1/terraform-variables
variable "region" {
description = "AWS region"
default = "us-west-1"
}
variable "ami" {
description = "AMI ID"
default = "ami-0c7217cdde317cfec"
}
variable "instance_type" {
description = "EC2 Instance Type"
default = "t2.micro"
}
variables.tf [unix] (00:20 25/02/2024)
12,1 All
variables.tf" [unix] 12L, 237B
8°C
Partly cloudy
Search
ENG
IN
00:22
25-02-2024

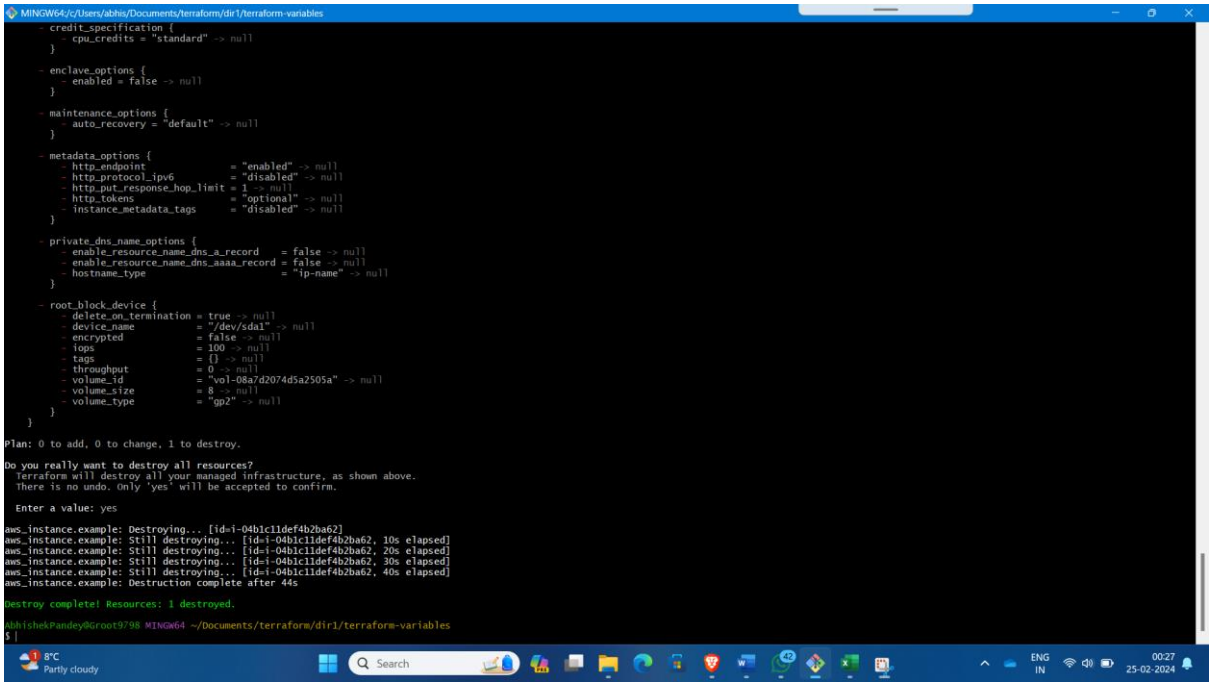

MINGW64:/c/Users/abhis/Documents/terraform/dir1/terraform-variables
- Using previously-installed hashicorp/aws v5.31.0
Terraform has been successfully initialized!
You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All 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.
AbhishekPandey@Groot9798 MINGW64 ~/Documents/terraform/dir1/terraform-variables
$ vi variables.tf
AbhishekPandey@Groot9798 MINGW64 ~/Documents/terraform/dir1/terraform-variables
$ terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.31.0
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.
AbhishekPandey@Groot9798 MINGW64 ~/Documents/terraform/dir1/terraform-variables
$ terraform apply
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-0c7217cdde317cfec"
+ 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)
8°C
Partly cloudy
Search
ENG
IN
00:23
25-02-2024

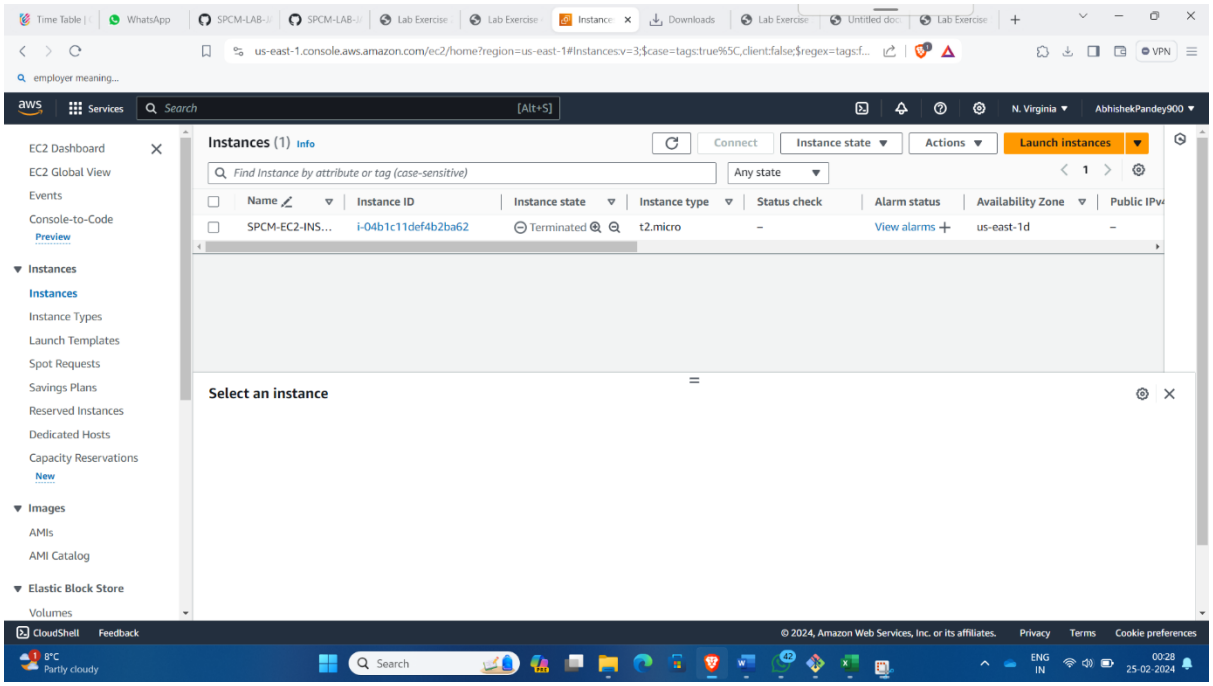

MINGW64:/c/Users/abhis/Documents/terraform/dir1/terraform-variables
+ ami = "ami-0c7217cdde317cfec"
+ arn = (known after apply)
+ associate_public_ip_address = (known after apply)
+ availability_zone = (known after apply)
+ cpu_core_count = (known after apply)
+ cpu_threads_per_core = (known after apply)
+ 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_lifecycle = (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)
+ placement_partition_number = (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
+ spot_instance_request_id = (known after apply)
+ subnet_id = (known after apply)
+ tags = {
+ "Name" = "SPCM-EC2-INSTANCE"
}
+ tags_all = {
+ "Name" = "SPCM-EC2-INSTANCE"
}
+ 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: 1 to add, 0 to change, 0 to destroy.
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.example: Creating...
8°C
Partly cloudy
Search
ENG
IN
00:24
25-02-2024


MINGW64:/c/Users/abhis/Documents/terraform/dir1/terraform-variables
+ instance_lifecycle = (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)
+ placement_partition_number = (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
+ spot_instance_request_id = (known after apply)
+ subnet_id = (known after apply)
+ tags = {
+ "Name" = "SPCM-EC2-INSTANCE"
}
+ tags_all = {
+ "Name" = "SPCM-EC2-INSTANCE"
}
+ 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: 1 to add, 0 to change, 0 to destroy.
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.example: Creating...
aws_instance.example: Still creating... [10s elapsed]
aws_instance.example: Still creating... [20s elapsed]
aws_instance.example: Still creating... [30s elapsed]
aws_instance.example: Creation complete after 39s [id=i-04b1c11def4b2ba62]
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
AbhishekPandey@Groot9798 MINGW64 ~/Documents/terraform/dir1/terraform-variables
$ vi main.tf
AbhishekPandey@Groot9798 MINGW64 ~/Documents/terraform/dir1/terraform-variables
$ vi variables.tf
AbhishekPandey@Groot9798 MINGW64 ~/Documents/terraform/dir1/terraform-variables
$
8°C
Partly cloudy
Search
ENG
IN
00:24
25-02-2024


Instances (1) Info
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
Any state
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
SPCM-EC2-INS...
i-04b1c11def4b2ba62
Running
t2.micro
2/2 checks passed
View alarms
us-east-1d
Select an instance

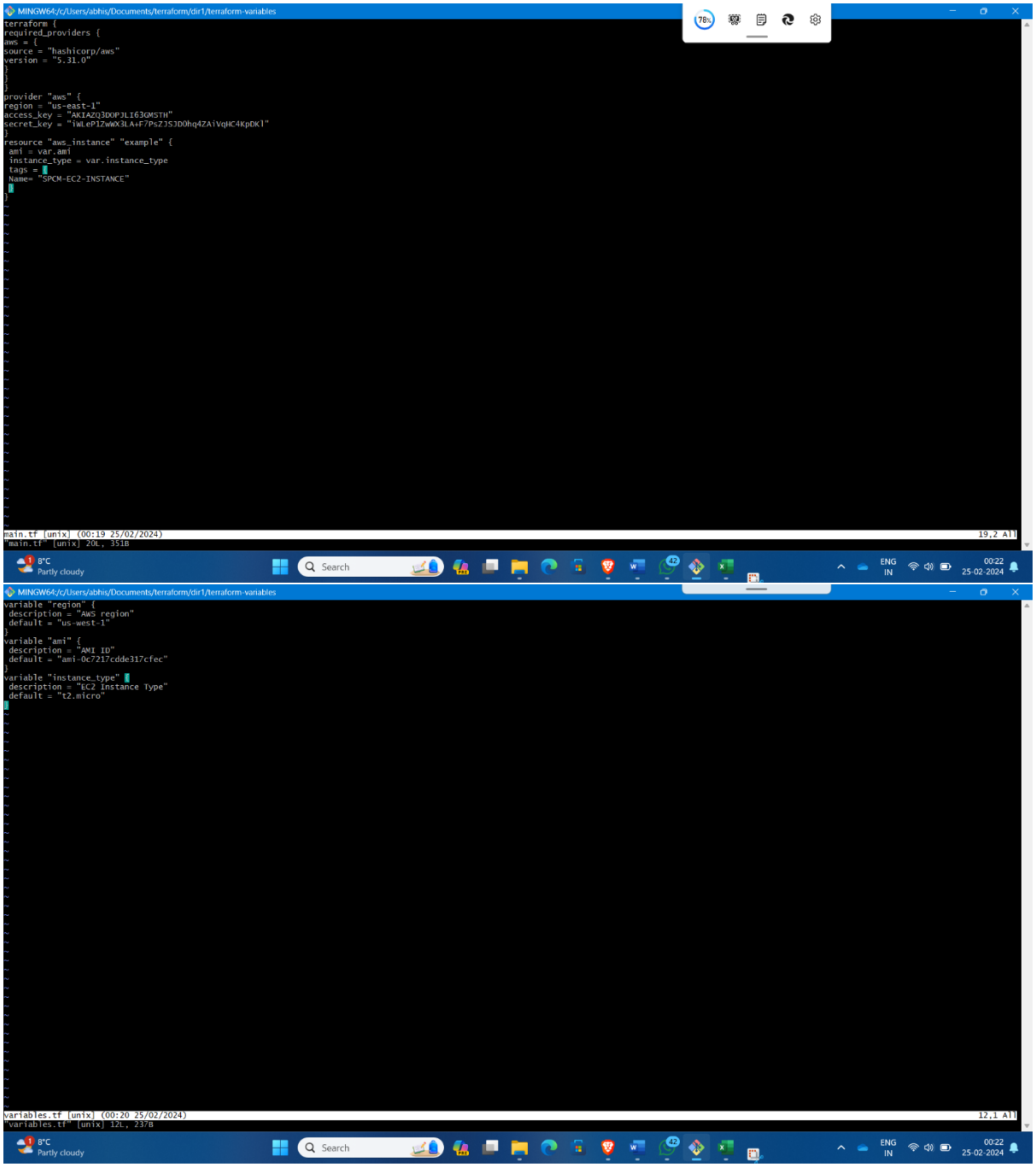

MINGW64:/c/Users/abhis/Documents/terraform/dir1/terraform-variables
$ vi main.tf
$ vi variables.tf
$ terrafom destroy
aws_instance.example: Refreshing state... [id=i-04b1c11def4b2ba62]
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
- destroy
Terraform will perform the following actions:
# aws_instance.example will be destroyed
- resource "aws_instance" "example" {
- ami = "ami-0c7217cdde317cfec" -> null
- arn = "arn:aws:ec2:us-east-1:654654208598:instance/i-04b1c11def4b2ba62" -> null
- associate_public_ip_address = true -> null
- availability_zone = "us-east-1d" -> null
- cpu_core_count = 1 -> null
- cpu_threads_per_core = 1 -> null
- disable_api_stop = false -> null
- disable_api_termination = false -> null
- ebs_optimized = false -> null
- get_password_data = false -> null
- hibernation = false -> null
- id = "i-04b1c11def4b2ba62" -> null
- instance_initiated_shutdown_behavior = "stop" -> null
- instance_state = "running" -> null
- instance_type = "t2.micro" -> null
- ipv6_address_count = 0 -> null
- ipv6_addresses = [] -> null
- monitoring = false -> null
- placement_partition_number = 0 -> null
- primary_network_interface_id = "eni-0089f94ff70b58b37" -> null
- private_dns = "ip-172-31-93-42.ec2.internal" -> null
- private_ip = "172.31.93.42" -> null
- public_dns = "ec2-100-24-53-244.compute-1.amazonaws.com" -> null
- public_ip = "100.24.53.244" -> null
- secondary_private_ips = [] -> null
- security_groups = [
"default",
] -> null
- source_dest_check = true -> null
- subnet_id = "subnet-08512513cdbca48db" -> null
- tags = {
"Name" = "SPCM-EC2-INSTANCE"
} -> null
- tags_all = {
"Name" = "SPCM-EC2-INSTANCE"
} -> null
- tenancy = "default" -> null
- user_data_replace_on_change = false -> null
- vpc_security_group_ids = [
"sg-07971ab5b528bc492",
] -> null


MINGW64:/c/Users/abhis/Documents/terraform/dir1/terraform-variables
- credit_specification {
- cpu_credits = "standard" -> null
}
- enclave_options {
- enabled = false -> null
}
- maintenance_options {
- auto_recovery = "default" -> null
}
- metadata_options {
- http_endpoint = "enabled" -> null
- http_protocol_ipv6 = "disabled" -> null
- http_put_response_hop_limit = 1 -> null
- http_tokens = "optional" -> null
- instance_metadata_tags = "disabled" -> null
}
- private_dns_name_options {
- enable_resource_name_dns_a_record = false -> null
- enable_resource_name_dns_aaaa_record = false -> null
- hostname_type = "ip-name" -> null
}
- root_block_device {
- delete_on_termination = true -> null
- device_name = "/dev/sda1" -> null
- encrypted = false -> null
- iops = 100 -> null
- tags = {} -> null
- throughput = 0 -> null
- volume_id = "vol-08a7d2074d5a2505a" -> null
- volume_size = 8 -> null
- volume_type = "gp2" -> null
}
}
Plan: 0 to add, 0 to change, 1 to destroy.
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
aws_instance.example: Destroying... [id=i-04b1c11def4b2ba62]
aws_instance.example: Still destroying... [id=i-04b1c11def4b2ba62, 10s elapsed]
aws_instance.example: Still destroying... [id=i-04b1c11def4b2ba62, 20s elapsed]
aws_instance.example: Still destroying... [id=i-04b1c11def4b2ba62, 30s elapsed]
aws_instance.example: Still destroying... [id=i-04b1c11def4b2ba62, 40s elapsed]
aws_instance.example: Destruction complete after 44s
Destroy complete! Resources: 1 destroyed.
AbhishekPandey@Groot9798 MINGW64 ~/Documents/terraform/dir1/terraform-variables
$
8°C
Partly cloudy
Search
ENG
IN
00:27
25-02-2024


us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:v=3;$case=tags:true%5C,client:false;$regex=tags:f...
aws
Services
Search
[Alt+S]
N. Virginia
AbhishekPandey900
EC2 Dashboard
EC2 Global View
Events
Console-to-Code
Preview
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
New
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Instances (1) Info
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
Any state
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4
SPCM-EC2-INS...
i-04b1c11def4b2ba62
Terminated
t2.micro
–
View alarms
us-east-1d
–
Select an instance
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
8°C
Partly cloudy
Search
ENG
IN
00:28
25-02-2024

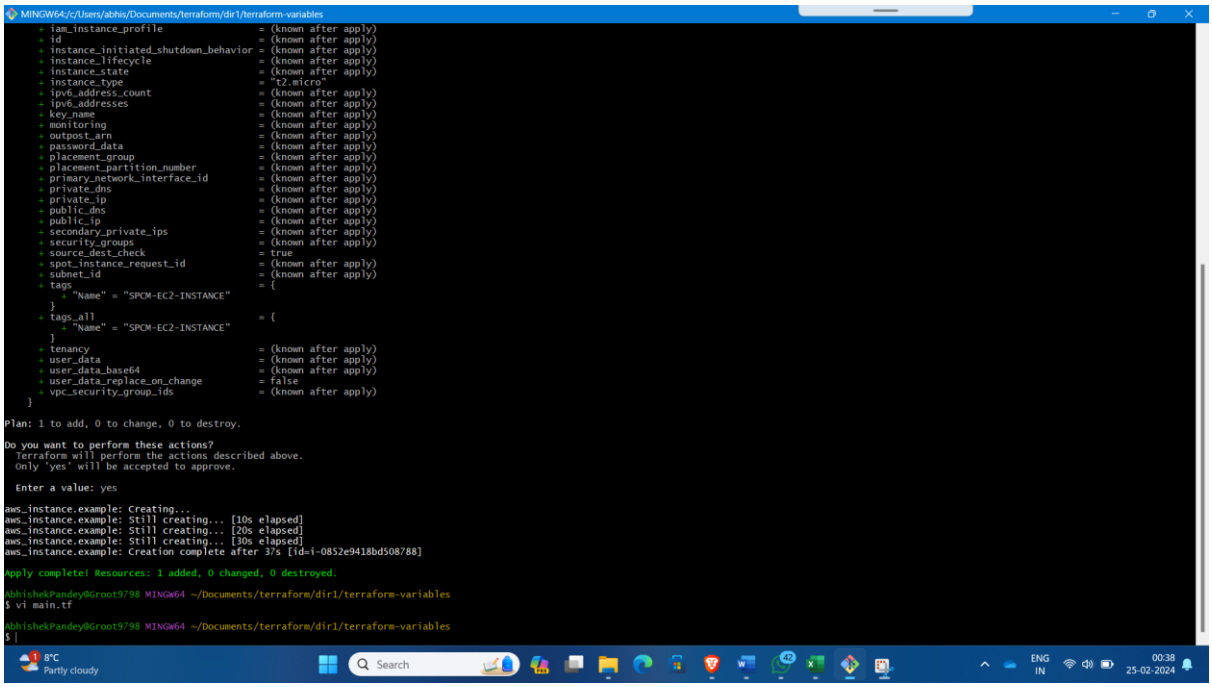
EXPERIMENT-5

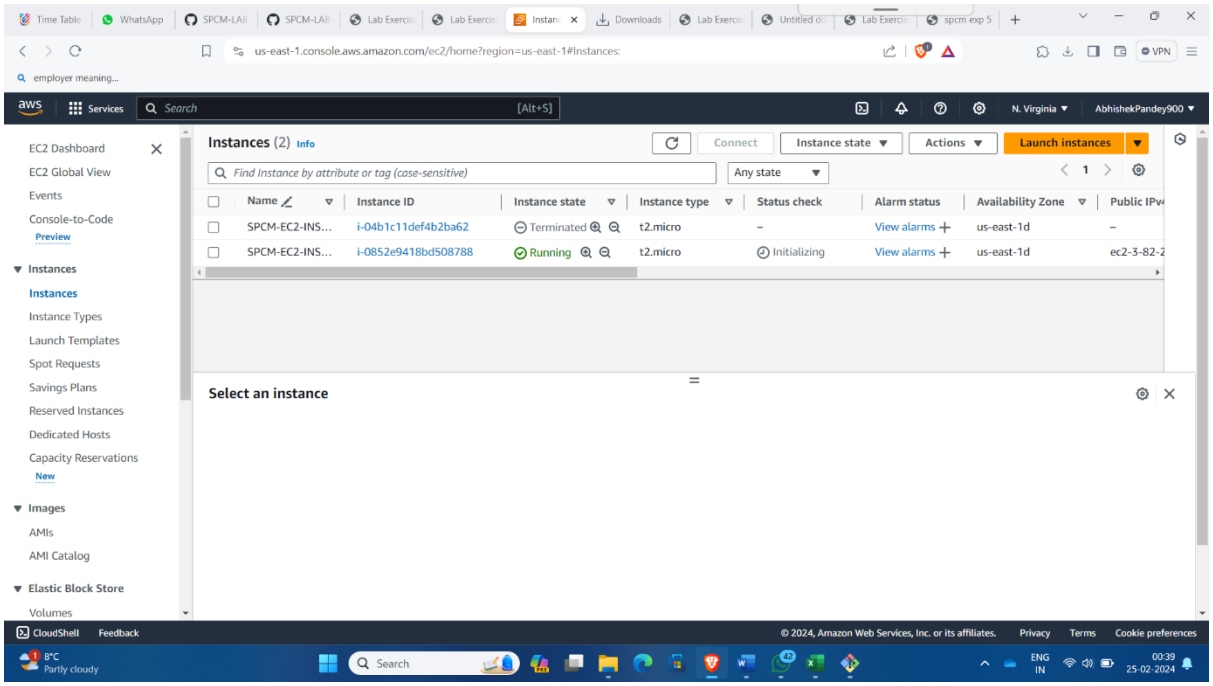
```
AbhishekPandey@Groot9798 MINGW64 ~/Documents/terraform/dir1
$ mkdir terraform-variables

AbhishekPandey@Groot9798 MINGW64 ~/Documents/terraform/dir1
$ cd terraform-variables

AbhishekPandey@Groot9798 MINGW64 ~/Documents/terraform/dir1/terraform-variables
$ vi main.tf

AbhishekPandey@Groot9798 MINGW64 ~/Documents/terraform/dir1/terraform-variables
$ 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!

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.
```

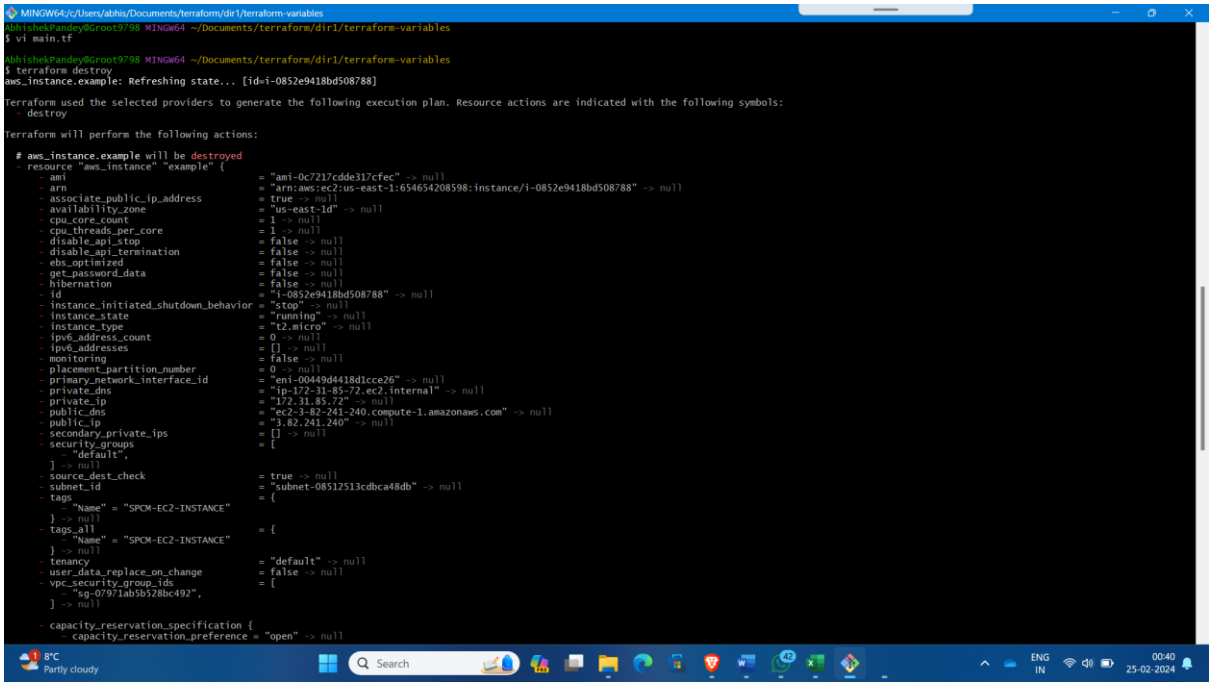

MINGW64:/c/Users/abhis/Documents/terraform/dir1/terraform-variables
terraform {
required_providers {
aws = {
source = "hashicorp/aws"
version = "5.31.0"
}
}
}
provider "aws" {
region = "us-east-1"
access_key = "AKIAZQ3DXPJLI63GMSTH"
secret_key = "iWLeP1ZwWX3LAvF7PsZJS3D0hq4ZAiVqHC4KpDKl"
}
resource "aws_instance" "example" {
ami = var.ami
instance_type = var.instance_type
tags = {
Name= "SPCM-EC2-INSTANCE"
}
}
main.tf [unix] (00:19 25/02/2024) 19,2 All
"main.tf" [unix] 20L, 351B
8°C
Partly cloudy
Search
ENG
IN
00:22
25-02-2024

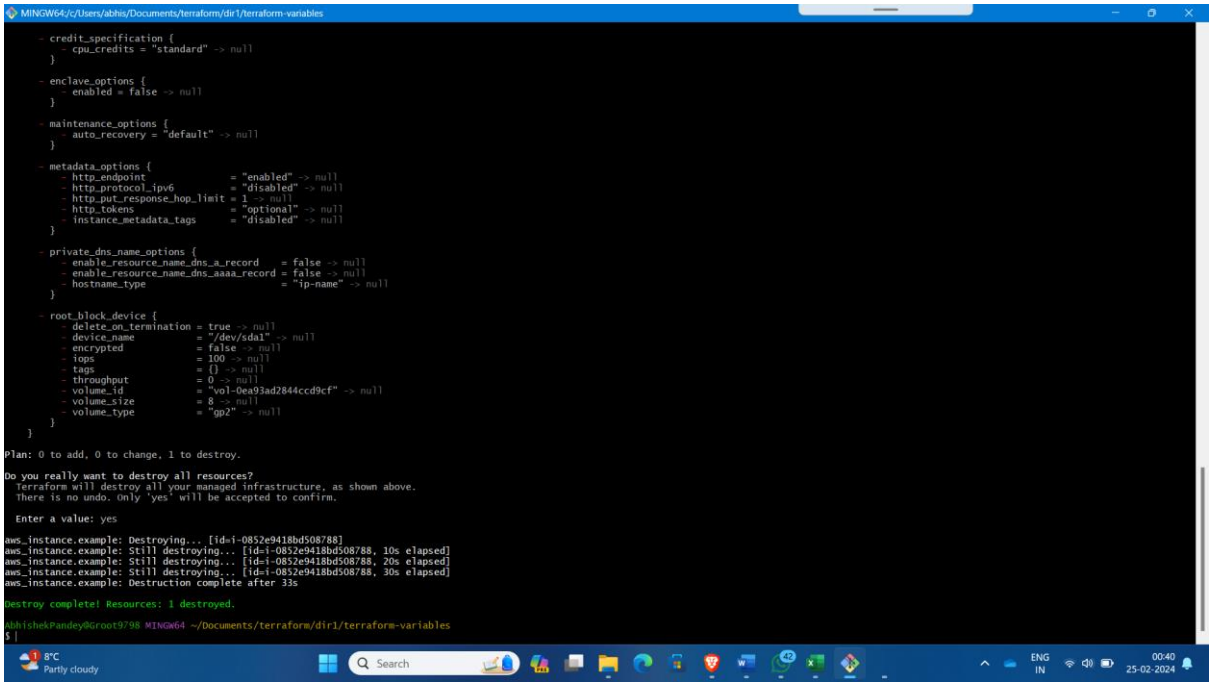
MINGW64:/c/Users/abhis/Documents/terraform/dir1/terraform-variables
variable "region" {
description = "AWS region"
default = "us-west-1"
}
variable "ami" {
description = "AMI ID"
default = "ami-0c7217cdde317cfec"
}
variable "instance_type" {
description = "EC2 Instance Type"
default = "t2.micro"
}
variables.tf [unix] (00:20 25/02/2024) 12,1 All
"variables.tf" [unix] 12L, 237B
8°C
Partly cloudy
Search
ENG
IN
00:22
25-02-2024

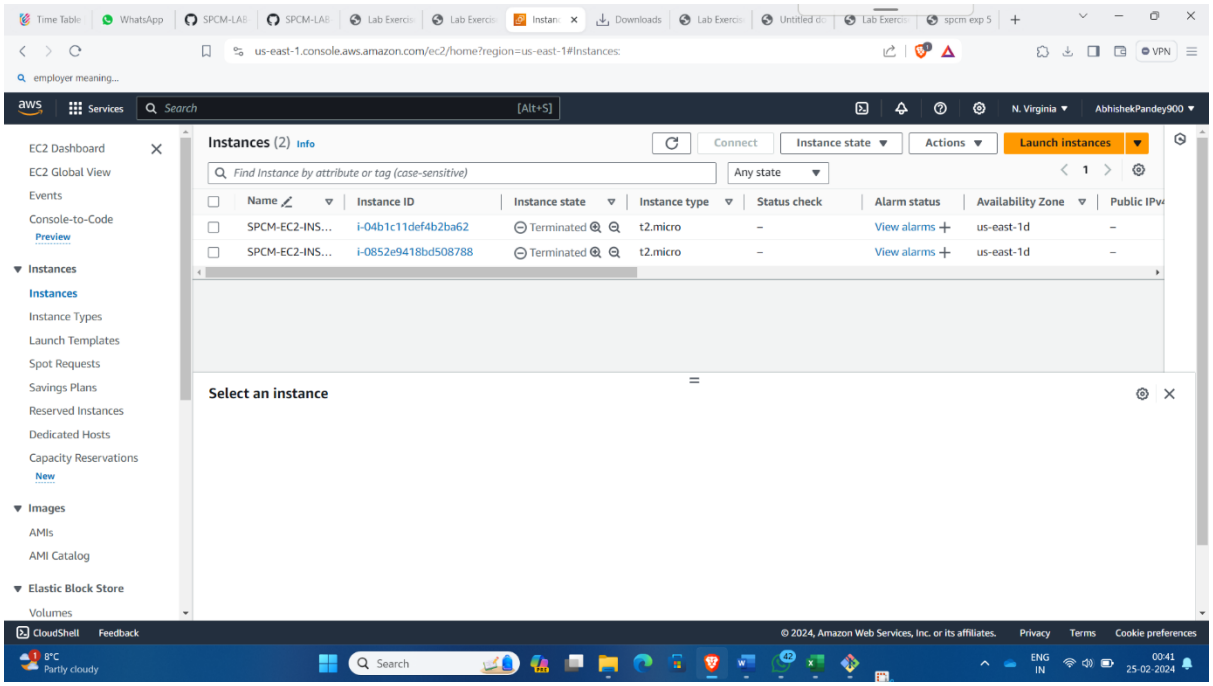
```
AbhishekPandey@Groot9798 MINGW64 ~/Documents/terraform/dir1/terraform-variables
$ terraform apply -var "region=us-east-1' -var "ami=ami-0c7217cdde317cfec" -var
"instance_type="t2.micro"

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-0c7217cdde317cfec"
      + 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_lifecycle                   = (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)
      + placement_partition_number           = (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
```




us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:
Services
Search
[Alt+S]
N. Virginia
AbhishekPandey900
EC2 Dashboard
EC2 Global View
Events
Console-to-Code
Preview
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
New
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Instances (2) Info
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
Any state
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4
SPCM-EC2-INS...
i-04b1c11def4b2ba62
Terminated
t2.micro
–
View alarms
us-east-1d
SPCM-EC2-INS...
i-0852e9418bd508788
Running
t2.micro
Initializing
View alarms
us-east-1d
Select an instance
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
8°C
Partly cloudy
Search
ENG
IN
00:39
25-02-2024


MINGW64:/c/Users/abhis/Documents/terraform/dir1/terraform-variables
AbhishekPandey@Groot9798 MINGW64 ~/Documents/terraform/dir1/terraform-variables
$ vi main.tf
AbhishekPandey@Groot9798 MINGW64 ~/Documents/terraform/dir1/terraform-variables
$ terraform destroy
aws_instance.example: Refreshing state... [id=i-0852e9418bd508788]
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.example will be destroyed
- resource "aws_instance" "example" {
- ami = "ami-0c7217cdde317cfec" -> null
- arn = "arn:aws:ec2:us-east-1:654654208598:instance/i-0852e9418bd508788" -> null
- associate_public_ip_address = true -> null
- availability_zone = "us-east-1d" -> null
- cpu_core_count = 1 -> null
- cpu_threads_per_core = 1 -> null
- disable_api_stop = false -> null
- disable_api_termination = false -> null
- ebs_optimized = false -> null
- get_password_data = false -> null
- hibernation = false -> null
- id = "i-0852e9418bd508788" -> null
- instance_initiated_shutdown_behavior = "stop" -> null
- instance_state = "running" -> null
- instance_type = "t2.micro" -> null
- ipv6_address_count = 0 -> null
- ipv6_addresses = [] -> null
- monitoring = false -> null
- placement_partition_number = 0 -> null
- primary_network_interface_id = "eni-00449d4418d1cce26" -> null
- private_dns = "ip-172-31-85-72.ec2.internal" -> null
- private_ip = "172.31.85.72" -> null
- public_dns = "ec2-3-82-241-240.compute-1.amazonaws.com" -> null
- public_ip = "3.82.241.240" -> null
- secondary_private_ips = [] -> null
- security_groups = [
"default",
] -> null
- source_dest_check = true -> null
- subnet_id = "subnet-08512513cdbca48db" -> null
- tags = {
"Name" = "SPCM-EC2-INSTANCE"
} -> null
- tags_all = {
"Name" = "SPCM-EC2-INSTANCE"
} -> null
- tenancy = "default" -> null
- user_data_replace_on_change = false -> null
- vpc_security_group_ids = [
"sg-07971ab5b528bc492",
] -> null
- capacity_reservation_specification {
capacity_reservation_preference = "open" -> null
8°C
Partly cloudy
Search
ENG
IN
00:40
25-02-2024


MINGW64:/c/Users/abhis/Documents/terraform/dir1/terraform-variables
- credit_specification {
- cpu_credits = "standard" -> null
}
- enclave_options {
- enabled = false -> null
}
- maintenance_options {
- auto_recovery = "default" -> null
}
- metadata_options {
- http_endpoint = "enabled" -> null
- http_protocol_ipv6 = "disabled" -> null
- http_put_response_hop_limit = 1 -> null
- http_tokens = "optional" -> null
- instance_metadata_tags = "disabled" -> null
}
- private_dns_name_options {
- enable_resource_name_dns_a_record = false -> null
- enable_resource_name_dns_aaaa_record = false -> null
- hostname_type = "ip-name" -> null
}
- root_block_device {
- delete_on_termination = true -> null
- device_name = "/dev/sda1" -> null
- encrypted = false -> null
- iops = 100 -> null
- tags = {} -> null
- throughput = 0 -> null
- volume_id = "vol-0ea93ad2844ccd9cf" -> null
- volume_size = 8 -> null
- volume_type = "gp2" -> null
}
}
Plan: 0 to add, 0 to change, 1 to destroy.
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
aws_instance.example: Destroying... [id=i-0852e9418bd508788]
aws_instance.example: Still destroying... [id=i-0852e9418bd508788, 10s elapsed]
aws_instance.example: Still destroying... [id=i-0852e9418bd508788, 20s elapsed]
aws_instance.example: Still destroying... [id=i-0852e9418bd508788, 30s elapsed]
aws_instance.example: Destruction complete after 33s
Destroy complete! Resources: 1 destroyed.
AbhishekPandey@Groot9798 MINGW64 ~/Documents/terraform/dir1/terraform-variables
$
8°C
Partly cloudy
Search
ENG
IN
00:40
25-02-2024


us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:
Services
Search
[Alt+S]
N. Virginia
AbhishekPandey900
EC2 Dashboard
EC2 Global View
Events
Console-to-Code
Preview
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
New
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Instances (2) Info
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
Any state
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4
SPCM-EC2-INS...
i-04b1c11def4b2ba62
Terminated
t2.micro
View alarms
us-east-1d
SPCM-EC2-INS...
i-0852e9418bd508788
Terminated
t2.micro
View alarms
us-east-1d
Select an instance
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
8°C
Partly cloudy
Search
ENG
IN
00:41
25-02-2024

EXPERIMENT-6

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

terraform {
required_providers {
aws = {
source = "hashicorp/aws"
version = "5.31.0"
}
}
}

provider "aws" {
    region = "ap-south-1"
    access_key = "AKIA5FTY77WSIB44R75Q"
    secret_key = "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"
}

```

Terraform-multiple-tfvars > variables.tf > variable "region" > default

```hcl
variable "region" {
    description = "AWS region"
    default = "ap-south-1"
    }


variable "ami" {
     description = "AMI ID"
type = string
default = "ami-03f4878755434977f"
}

variable "instance_type" {
     description = "EC2 Instance Type"
      default   = "t2.micro"
}
```

Terraform-multiple-tfvars > dev.tfvars > region

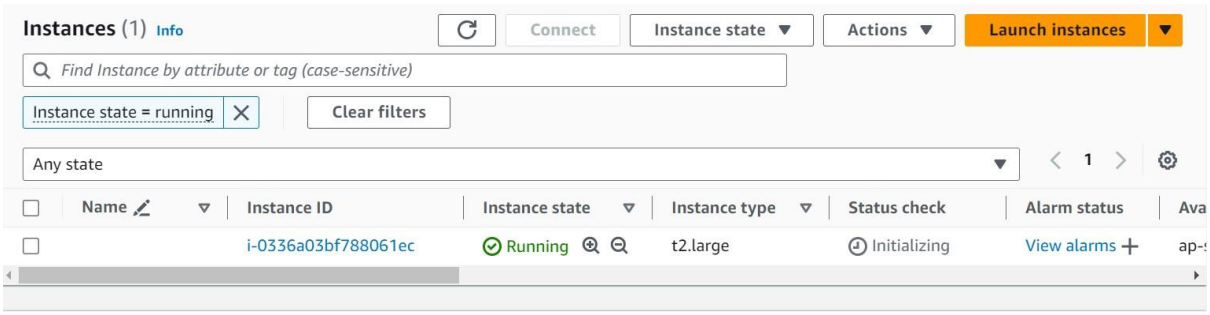
```hcl
region = "ap-south-1"
ami = "ami-0d63de463e6604d0a"
instance_type = "t2.micro"
```

Terraform-multiple-tfvars > ops.tfvars > region

```
region = "ap-south-1"
ami = "ami-03f4878755434977f"
instance_type = "t2.large"
```

```
C:\Desktop\DevOps\Sem6\SMCP\Lab Files\TERRAFORM LAB SCRIPTS\Terraform-multiple-tfvars> 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.example will be created
  + resource "aws_instance" "example" {
      + ami                                  = "ami-0d63de463e6604d0a"
      + 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)
```

```
C:\Desktop\DevOps\Sem6\SMCP\Lab Files\TERRAFORM LAB SCRIPTS\Terraform-multiple-tfvars> terraform apply -var-file=ops.tfvars
aws_instance.example: Refreshing state... [id=i-03753a6e6fb28a490]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the
following symbols:
-/+ destroy and then create replacement

Terraform will perform the following actions:

  # aws_instance.example must be replaced
-/+ resource "aws_instance" "example" {
      ~ ami                                  = "ami-0d63de463e6604d0a" -> "ami-03f4878755434977f" # forces replacement
      ~ arn                                  = "arn:aws:ec2:ap-south-1:905418112420:instance/i-03753a6e6fb28a490" -> (known after apply)
      ~ associate_public_ip_address          = true -> (known after apply)
      ~ availability_zone                    = "ap-south-1b" -> (known after apply)
      ~ cpu_core_count                       = 1 -> (known after apply)
      ~ cpu_threads_per_core                 = 1 -> (known after apply)
      ~ disable_api_stop                     = false -> (known after apply)
      ~ disable_api_termination              = false -> (known after apply)
      ~ ebs_optimized                        = false -> (known after apply)
      - hibernation                          = false -> null
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      ~ id                                   = "i-03753a6e6fb28a490" -> (known after apply)
      ~ instance_initiated_shutdown_behavior = "stop" -> (known after apply)
      + instance_lifecycle                   = (known after apply)
```



```
C:\Desktop\DevOps\Sem6\SMCP\Lab Files\TERRAFORM LAB SCRIPTS\Terraform-multiple-tfvars> terraform destroy -var-file=ops.tfvars
aws_instance.example: Refreshing state... [id=i-0336a03bf788061ec]

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.example will be destroyed
  - resource "aws_instance" "example" {
      - ami                                  = "ami-03f4878755434977f" -> null
      - arn                                  = "arn:aws:ec2:ap-south-1:905418112420:instance/i-0336a03bf788061ec" -> null
      - associate_public_ip_address          = true -> null
      - availability_zone                    = "ap-south-1b" -> null
      - cpu_core_count                       = 2 -> null
      - cpu_threads_per_core                 = 1 -> null
      - disable_api_stop                     = false -> null
      - disable_api_termination              = false -> null
      - ebs_optimized                        = false -> null
      - get_password_data                    = false -> null
      - hibernation                          = false -> null
      - id                                   = "i-0336a03bf788061ec" -> null
```

EXPERIMENT-7

main.tf var.tf

Terraform-iam-users > main.tf > resource "aws_iam_user" "iam_users"

```
terraform {
required_providers {
aws = {
source = "hashicorp/aws"
version = "5.31.0"
}
}
}

provider "aws" {
    region = "us-west-2"
    access_key = "AKIA5FTY77WSIB44R75Q"
    secret_key = "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"
}

resource "aws_iam_user" "iam_users" {
    count = length(var.iam_users)
name = var.iam_users[count.index]


tags = {
Name = "${var.iam_users[count.index]}-user"
}
}

```

main.tf | var.tf

Terraform-iam-users > var.tf > variable "iam_users" > type

```
variable "iam_users" {
    type    = list(string)
default = ["user3", "user4", "user5"]
}

```

```
PS C:\Desktop\DevOps\Sem6\SMCP\Lab Files\TERRAFORM LAB SCRIPTS\Terraform-iam-users> 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!
```

```
PS C:\Desktop\DevOps\Sem6\SMCP\Lab Files\TERRAFORM LAB SCRIPTS\Terraform-iam-users> terraform apply

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

Terraform will perform the following actions:

  # aws_iam_user.iam_users[0] will be created
  + resource "aws_iam_user" "iam_users" {
      + arn           = (known after apply)
      + force_destroy = false
      + id            = (known after apply)
      + name          = "user3"
      + path          = "/"
      + tags          = {
```

IAM > Users

Users (5) Info

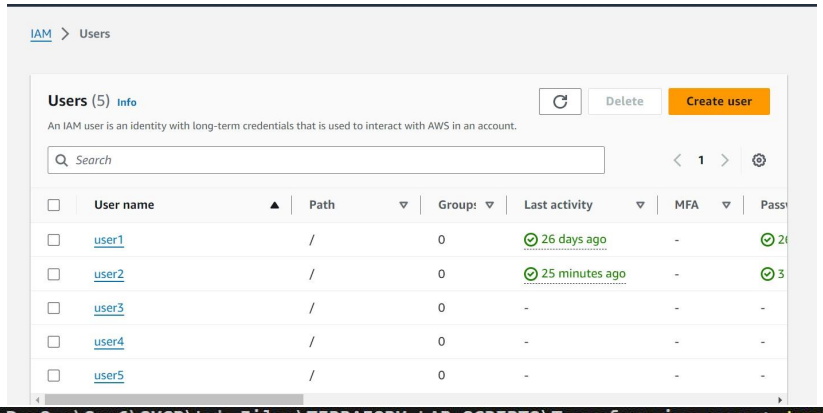
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

Delete | Create user

Search

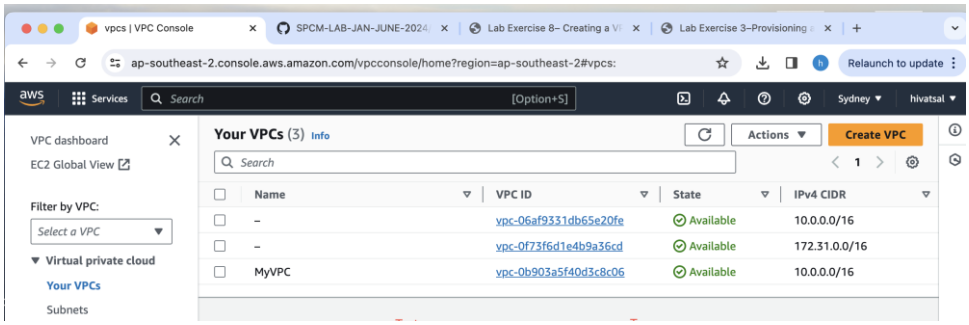
1

User name	Path	Groups	Last activity	MFA	Pass
user1	/	0	26 days ago	-	2(
user2	/	0	25 minutes ago	-	3
user3	/	0	-	-	-
user4	/	0	-	-	-
user5	/	0	-	-	-




```
PS C:\Desktop\DevOps\Sem6\SMCP\Lab Files\TERRAFORM LAB SCRIPTS\Terraform-iam-users> terraform destroy
aws_iam_user.iam_users[2]: Refreshing state... [id=user5]
aws_iam_user.iam_users[1]: Refreshing state... [id=user4]
aws_iam_user.iam_users[0]: Refreshing state... [id=user3]

Terraform used the selected providers to generate the following execution plan. Resource actions are i
  - destroy

Terraform will perform the following actions:

  # aws_iam_user.iam_users[0] will be destroyed
  - resource "aws_iam_user" "iam_users" {
      - arn           = "arn:aws:iam::905418112420:user/user3" -> null
      - force_destroy = false -> null
      - id            = "user3" -> null
      - name          = "user3" -> null
      - path          = "/" -> null
```

EXPERIMENT-8




```
shivavatsal@Shivas-MacBook-Air aws-terraform-demo % cd
shivavatsal@Shivas-MacBook-Air ~ % cd Documents
shivavatsal@Shivas-MacBook-Air Documents % cd terraform
shivavatsal@Shivas-MacBook-Air terraform % cd dir1
shivavatsal@Shivas-MacBook-Air dir1 % cd terraform-vpc
shivavatsal@Shivas-MacBook-Air terraform-vpc % vi main.tf
shivavatsal@Shivas-MacBook-Air terraform-vpc % vi vpc.tf
shivavatsal@Shivas-MacBook-Air terraform-vpc % terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.35.0

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.
shivavatsal@Shivas-MacBook-Air terraform-vpc % terraform apply

  Error: Retrieving AWS account details: validating provider credentials: retrieving caller identity from STS: operation error STS: GetCallerIdentity, https response error StatusCode: 400, RequestID: 4cbe
cb22-fce6-41e1-84b8-6bc264e95100, api error IncompleteSignature: 'IAM' not a valid key=value pair (missing equal-sign) in Authorization header: 'AWS4-HMAC-SHA256 Credential=your IAM access key/20240207/ap
-southeast-2/sts/aws4_request, SignedHeaders=amz-sdk-invocation-id;amz-sdk-request;content-length;content-type;host;x-amz-date, Signature=f5ca3227a2ac40ddc8f88894e84bb028ed6cba891ea105978c2597f3b3f18d03'.

    with provider["registry.terraform.io/hashicorp/aws"],
    on main.tf line 1, in provider "aws":
     1: provider "aws" {

shivavatsal@Shivas-MacBook-Air terraform-vpc % vi main.tf
shivavatsal@Shivas-MacBook-Air terraform-vpc % terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.35.0

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.
shivavatsal@Shivas-MacBook-Air terraform-vpc % terraform apply

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_internet_gateway.gfg-gw will be created
  + resource "aws_internet_gateway" "gfg-gw" {
      + arn      = (known after apply)


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.
shivavatsal@Shivas-MacBook-Air terraform-vpc % terraform apply

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_internet_gateway.gfg-gw will be created
  + resource "aws_internet_gateway" "gfg-gw" {
      + arn      = (known after apply)
      + id       = (known after apply)
      + owner_id = (known after apply)
      + tags     = {
          + "Name" = "gfg-IG"
        }
      + tags_all = {
          + "Name" = "gfg-IG"
        }
      + vpc_id   = (known after apply)
    }

  # aws_route_table.gfg-rt will be created
  + resource "aws_route_table" "gfg-rt" {
      + arn              = (known after apply)
      + id               = (known after apply)
      + owner_id         = (known after apply)
      + propagating_vgws = (known after apply)
      + route            = [
          + {
              + carrier_gateway_id         = ""
              + cidr_block                 = "0.0.0.0/0"
              + core_network_arn           = ""
              + destination_prefix_list_id = ""
              + egress_only_gateway_id     = ""
              + gateway_id                 = (known after apply)
              + ipv6_cidr_block            = ""
              + local_gateway_id           = ""
              + nat_gateway_id             = ""
              + network_interface_id       = ""
              + transit_gateway_id         = ""
              + vpc_endpoint_id            = ""
              + vpc_peering_connection_id  = ""
            },
        ]
      + tags             = {
          + "Name" = "GFG-Route-Table"
        }
      + tags_all         = {
          + "Name" = "GFG-Route-Table"
        }
      + vpc_id           = (known after apply)
    }

  # aws_route_table_association.gfg-rta will be created
  + resource "aws_route_table_association" "gfg-rta" {
      + id             = (known after apply)
      + route_table_id = (known after apply)
```

```
    }

Plan: 9 to add, 0 to change, 0 to destroy.

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_vpc.gfg-vpc: Creating...
aws_vpc.my_vpc: Creating...
aws_vpc.gfg-vpc: Creation complete after 5s [id=vpc-06af9331db65e20fe]
aws_internet_gateway.gfg-gw: Creating...
aws_subnet.gfg-subnet: Creating...
aws_security_group.gfg-sg: Creating...
aws_subnet.gfg-subnet: Creation complete after 3s [id=subnet-08357d5be2c38650d]
aws_internet_gateway.gfg-gw: Creation complete after 3s [id=igw-05b77f965cd6dd967]
aws_route_table.gfg-rt: Creating...
aws_vpc.my_vpc: Still creating... [10s elapsed]
aws_route_table.gfg-rt: Creation complete after 3s [id=rtb-0d6b14180a1488a73]
aws_route_table_association.gfg-rta: Creating...
aws_security_group.gfg-sg: Creation complete after 6s [id=sg-0bbc3c36d60869737]
aws_route_table_association.gfg-rta: Creation complete after 1s [id=rtbassoc-042655538d1f38784]
aws_vpc.my_vpc: Creation complete after 15s [id=vpc-0b903a5f40d3c8c06]
aws_subnet.my_subnet[0]: Creating...
aws_subnet.my_subnet[1]: Creating...
aws_subnet.my_subnet[0]: Still creating... [10s elapsed]
aws_subnet.my_subnet[1]: Still creating... [10s elapsed]
aws_subnet.my_subnet[1]: Creation complete after 12s [id=subnet-01c64566dbeae7f6a]
aws_subnet.my_subnet[0]: Creation complete after 12s [id=subnet-09c60779ab2642307]

Apply complete! Resources: 9 added, 0 changed, 0 destroyed.
(shivavatsal@Shivas-MacBook-Air terraform-vpc % terraform destroy
aws_vpc.my_vpc: Refreshing state... [id=vpc-0b903a5f40d3c8c06]
aws_vpc.gfg-vpc: Refreshing state... [id=vpc-06af9331db65e20fe]
aws_internet_gateway.gfg-gw: Refreshing state... [id=igw-05b77f965cd6dd967]
aws_subnet.my_subnet[1]: Refreshing state... [id=subnet-01c64566dbeae7f6a]
aws_subnet.gfg-subnet: Refreshing state... [id=subnet-08357d5be2c38650d]
aws_subnet.my_subnet[0]: Refreshing state... [id=subnet-09c60779ab2642307]
aws_security_group.gfg-sg: Refreshing state... [id=sg-0bbc3c36d60869737]
aws_route_table.gfg-rt: Refreshing state... [id=rtb-0d6b14180a1488a73]
aws_route_table_association.gfg-rta: Refreshing state... [id=rtbassoc-042655538d1f38784]

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_internet_gateway.gfg-gw will be destroyed
  - resource "aws_internet_gateway" "gfg-gw" {
      - arn      = "arn:aws:ec2:ap-southeast-2:891377313468:internet-gateway/igw-05b77f965cd6dd967" -> null
      - id       = "igw-05b77f965cd6dd967" -> null
      - owner_id = "891377313468" -> null
      - tags     = {
          - "Name" = "gfg-IG"
        } -> null
      - tags_all = {
          - "Name" = "gfg-IG"
        } -> null
      - vpc_id   = "vpc-06af9331db65e20fe" -> null
    }




      - private_dns_hostname_type_on_launch            = "ip-name" -> null
      - tags                                           = {
          - "Name" = "MySubnet-2"
        } -> null
      - tags_all                                       = {
          - "Name" = "MySubnet-2"
        } -> null
      - vpc_id                                         = "vpc-0b903a5f40d3c8c06" -> null
    }

  # aws_vpc.gfg-vpc will be destroyed
  - resource "aws_vpc" "gfg-vpc" {
      - arn                                  = "arn:aws:ec2:ap-southeast-2:891377313468:vpc/vpc-06af9331db65e20fe" -> null
      - assign_generated_ipv6_cidr_block     = false -> null
      - cidr_block                           = "10.0.0.0/16" -> null
      - default_network_acl_id               = "acl-0a5f2721b5e13ac59" -> null
      - default_route_table_id               = "rtb-0949899781 7be74da" -> null
      - default_security_group_id            = "sg-070e408075ef90d74" -> null
      - dhcp_options_id                      = "dopt-0a3cdc788fedf1c98" -> null
      - enable_dns_hostnames                 = false -> null
      - enable_dns_support                   = true -> null
      - enable_network_address_usage_metrics = false -> null
      - id                                   = "vpc-06af9331db65e20fe" -> null
      - instance_tenancy                     = "default" -> null
      - ipv6_netmask_length                  = 0 -> null
      - main_route_table_id                  = "rtb-0949899781 7be74da" -> null
      - owner_id                             = "891377313468" -> null
      - tags                                 = {} -> null
      - tags_all                             = {} -> null
    }

  # aws_vpc.my_vpc will be destroyed
  - resource "aws_vpc" "my_vpc" {
      - arn                                  = "arn:aws:ec2:ap-southeast-2:891377313468:vpc/vpc-0b903a5f40d3c8c06" -> null
      - assign_generated_ipv6_cidr_block     = false -> null
      - cidr_block                           = "10.0.0.0/16" -> null
      - default_network_acl_id               = "acl-0b8d08a2c5e0ba0a1" -> null
      - default_route_table_id               = "rtb-070880172a06ec763" -> null
      - default_security_group_id            = "sg-027998b99873ff626" -> null
      - dhcp_options_id                      = "dopt-0a3cdc788fedf1c98" -> null
      - enable_dns_hostnames                 = true -> null
      - enable_dns_support                   = true -> null
      - enable_network_address_usage_metrics = false -> null
      - id                                   = "vpc-0b903a5f40d3c8c06" -> null
      - instance_tenancy                     = "default" -> null
      - ipv6_netmask_length                  = 0 -> null
      - main_route_table_id                  = "rtb-070880172a06ec763" -> null
      - owner_id                             = "891377313468" -> null
      - tags                                 = {
          - "Name" = "MyVPC"
        } -> null
      - tags_all                             = {
          - "Name" = "MyVPC"
        } -> null
    }

Plan: 0 to add, 0 to change, 9 to destroy.

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:
```

Destroy complete! Resources: 9 destroyed.

shivavatsal@Shivas-MacBook-Air terraform-vpc %

EXPERIMENT-9

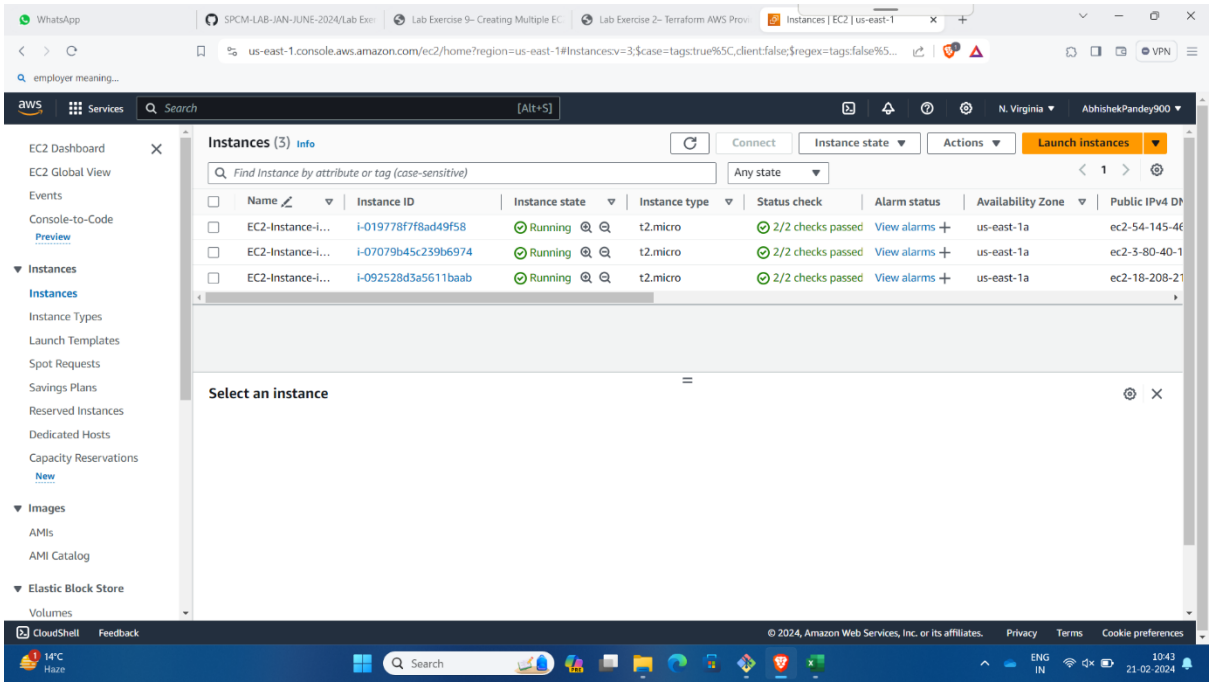
```
shivavatsal@Shivas-Air terraform-rds % cd terraform ec2-for-each
cd: no such file or directory: /Users/shivavatsal/Documents/ec2-for-each/terraform-rds
shivavatsal@Shivas-Air terraform-rds % cd --
shivavatsal@Shivas-Air ~ % cd Documents
shivavatsal@Shivas-Air Documents % cd terraform
shivavatsal@Shivas-Air terraform % cd terraform ec2-for-each
cd: no such file or directory: /Users/shivavatsal/Documents/ec2-for-each
shivavatsal@Shivas-Air terraform % cd terraform-ec2-for-each
cd: no such file or directory: terraform-ec2-for-each
shivavatsal@Shivas-Air terraform % cd dir1
shivavatsal@Shivas-Air dir1 % cd terraform-ec2-for-each
shivavatsal@Shivas-Air terraform-ec2-for-each % vi main.tf
shivavatsal@Shivas-Air terraform-ec2-for-each % vi main.tf
shivavatsal@Shivas-Air terraform-ec2-for-each % terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.31.0

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.
shivavatsal@Shivas-Air terraform-ec2-for-each % terraform apply

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" {
      + ami                                  = "ami-07e1aeb90edb268a3"
      + 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
```

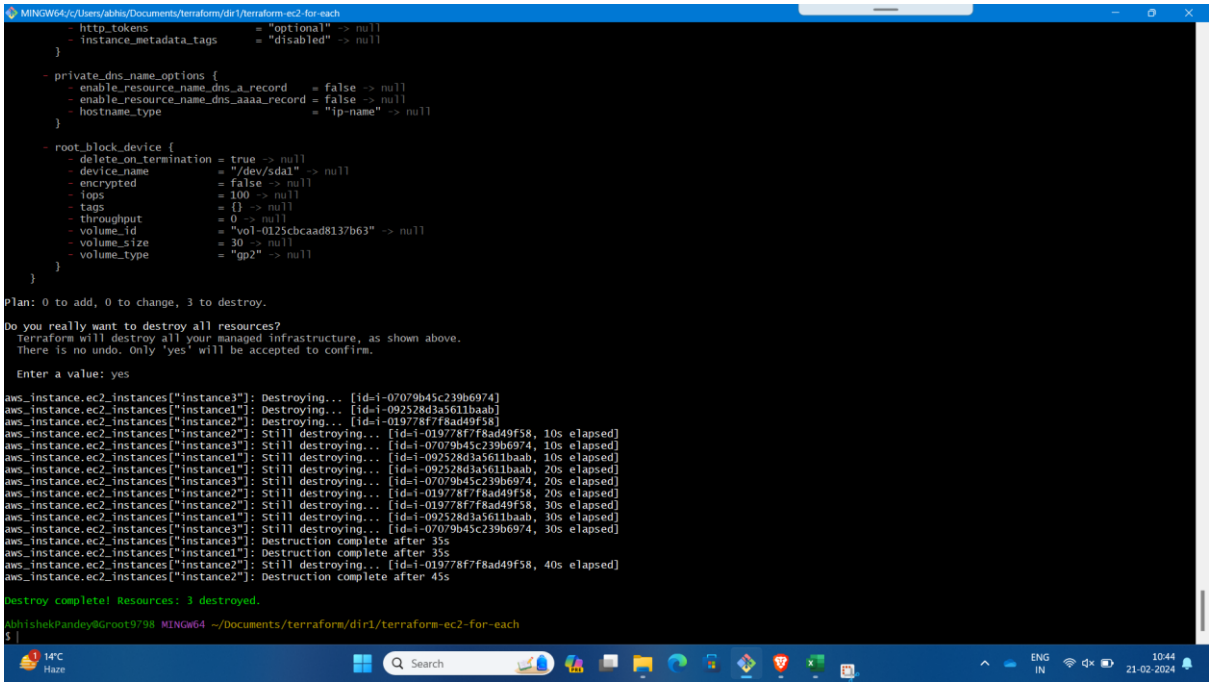
```
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Name" = "EC2-Instance-instance1"
        }
      + tags_all                             = {
          + "Name" = "EC2-Instance-instance1"
        }
      + 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)
    }

  # aws_instance.ec2_instances["instance2"] will be created
  + resource "aws_instance" "ec2_instances" {
      + ami                                  = "ami-04f5097681773b989"
      + 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_lifecycle                   = (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)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
```

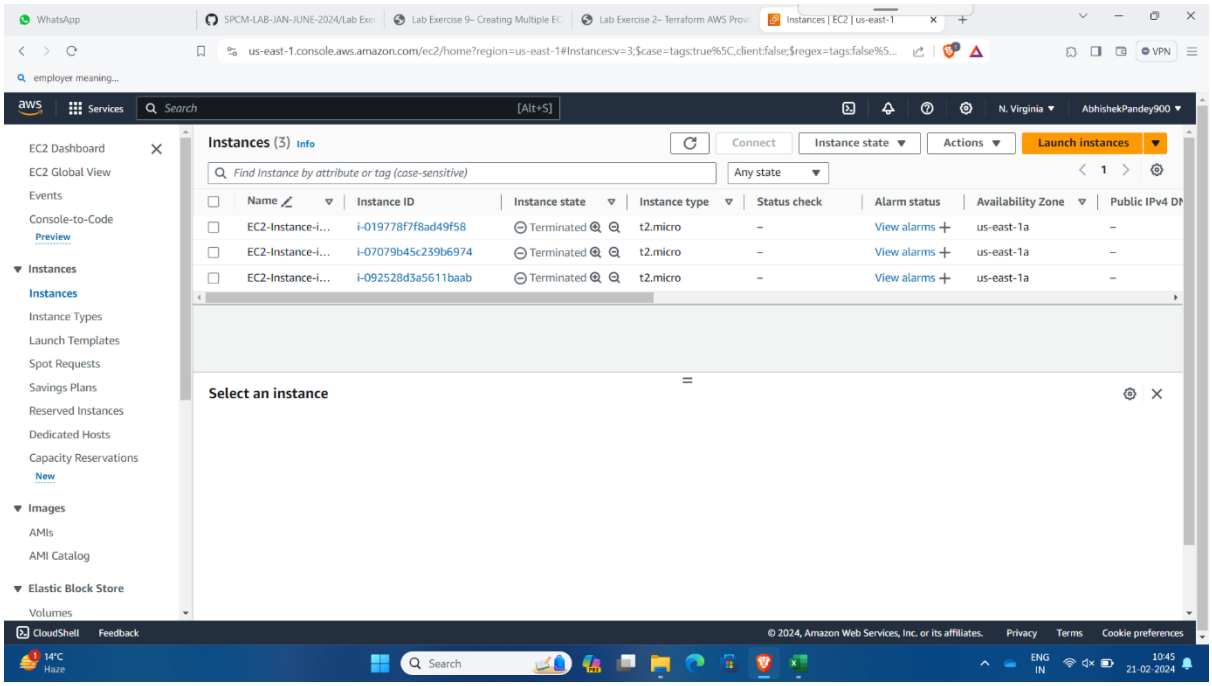
```
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)
    }

  # aws_instance.ec2_instances["instance3"] will be created
  + resource "aws_instance" "ec2_instances" {
      + ami                                  = "ami-0d6857b844e855670"
      + 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_lifecycle                   = (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)
      + placement_partition_number           = (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
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Name" = "EC2-Instance-instance3"
        }
      + tags_all                             = {
```

```
      + instance_lifecycle                   = (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)
      + placement_partition_number           = (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
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Name" = "EC2-Instance-instance3"
        }
      + tags_all                             = {
          + "Name" = "EC2-Instance-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.

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.ec2_instances["instance3"]: Creating...
aws_instance.ec2_instances["instance2"]: Creating...
aws_instance.ec2_instances["instance1"]: Creating...
aws_instance.ec2_instances["instance1"]: Still creating... [10s elapsed]
aws_instance.ec2_instances["instance1"]: Still creating... [20s elapsed]
aws_instance.ec2_instances["instance1"]: Still creating... [30s elapsed]
aws_instance.ec2_instances["instance1"]: Creation complete after 35s [id=i-0e92cb42914fe6aa3]
```

```
terraform {
required_providers {
aws = {
source = "hashicorp/aws"
version = "5.31.0"
}
}
}
provider "aws"{
region = "ap-southeast-2"
access_key = "AKIA47CR2MZ6OV4Y53G3"
secret_key = "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"
}
variable "instances"{
 description = "Map of EC2 instances with settings"
 default = {
 "instance1" = {
ami = "ami-07e1aeb90edb268a3"
 instance_type = "t2.micro"
 },
 "instance2" ={
 ami = "ami-04f5097681773b989"
 instance_type = "t2.micro "
 },
 "instance3" ={
 ami = "ami-0d6857b844e855670"
 instance_type = "t2.micro "
 }
 }
}
resource "aws_instance" "ec2_instances"{
 for_each = var.instances
 ami = var.instances[each.key].ami
 instance_type = var.instances[each.key].instance_type
 tags ={
 Name = "EC2-Instance-${each.key}"
 }
}
~
~
~
~
~
~
~
~
~
~
"main.tf" 38L, 763B
```


Instances (3) Info
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
Any state
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
EC2-Instance-i... i-019778f7f8ad49f58 Running t2.micro 2/2 checks passed View alarms us-east-1a
EC2-Instance-i... i-07079b45c239b6974 Running t2.micro 2/2 checks passed View alarms us-east-1a
EC2-Instance-i... i-092528d3a5611baab Running t2.micro 2/2 checks passed View alarms us-east-1a
Select an instance


MINGW64:/c/Users/abhis/Documents/terraform/dir1/terraform-ec2-for-each
Plan: 0 to add, 0 to change, 3 to destroy.
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
aws_instance.ec2_instances["instance3"]: Destroying... [id=i-07079b45c239b6974]
aws_instance.ec2_instances["instance1"]: Destroying... [id=i-092528d3a5611baab]
aws_instance.ec2_instances["instance2"]: Destroying... [id=i-019778f7f8ad49f58]
aws_instance.ec2_instances["instance2"]: Still destroying... [id=i-019778f7f8ad49f58, 10s elapsed]
aws_instance.ec2_instances["instance3"]: Still destroying... [id=i-07079b45c239b6974, 10s elapsed]
aws_instance.ec2_instances["instance1"]: Still destroying... [id=i-092528d3a5611baab, 10s elapsed]
aws_instance.ec2_instances["instance1"]: Still destroying... [id=i-092528d3a5611baab, 20s elapsed]
aws_instance.ec2_instances["instance3"]: Still destroying... [id=i-07079b45c239b6974, 20s elapsed]
aws_instance.ec2_instances["instance2"]: Still destroying... [id=i-019778f7f8ad49f58, 20s elapsed]
aws_instance.ec2_instances["instance2"]: Still destroying... [id=i-019778f7f8ad49f58, 30s elapsed]
aws_instance.ec2_instances["instance1"]: Still destroying... [id=i-092528d3a5611baab, 30s elapsed]
aws_instance.ec2_instances["instance3"]: Still destroying... [id=i-07079b45c239b6974, 30s elapsed]
aws_instance.ec2_instances["instance3"]: Destruction complete after 35s
aws_instance.ec2_instances["instance1"]: Destruction complete after 35s
aws_instance.ec2_instances["instance2"]: Still destroying... [id=i-019778f7f8ad49f58, 40s elapsed]
aws_instance.ec2_instances["instance2"]: Destruction complete after 45s
Destroy complete! Resources: 3 destroyed.
AbhishekPandey@Groot9798 MINGW64 ~/Documents/terraform/dir1/terraform-ec2-for-each

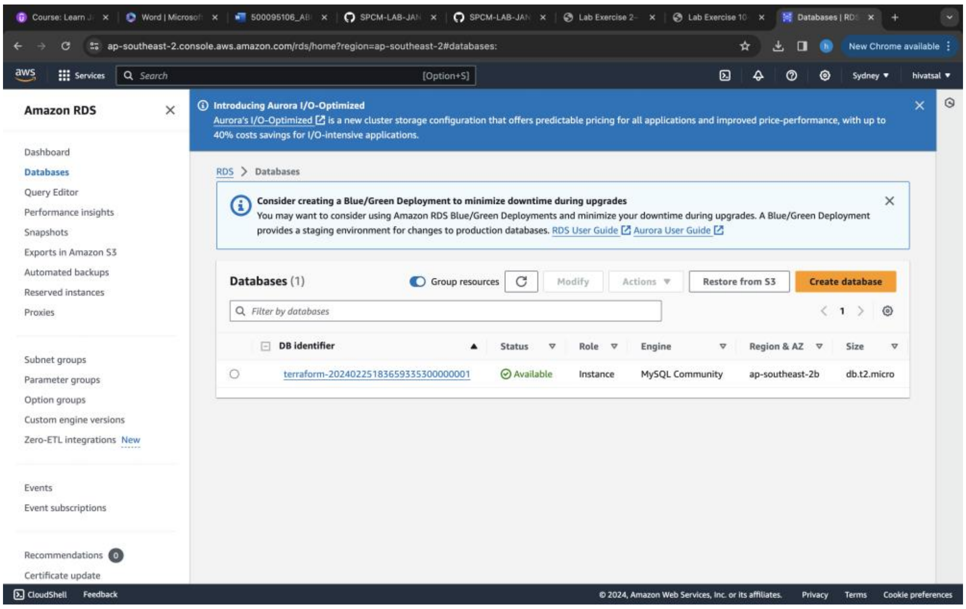


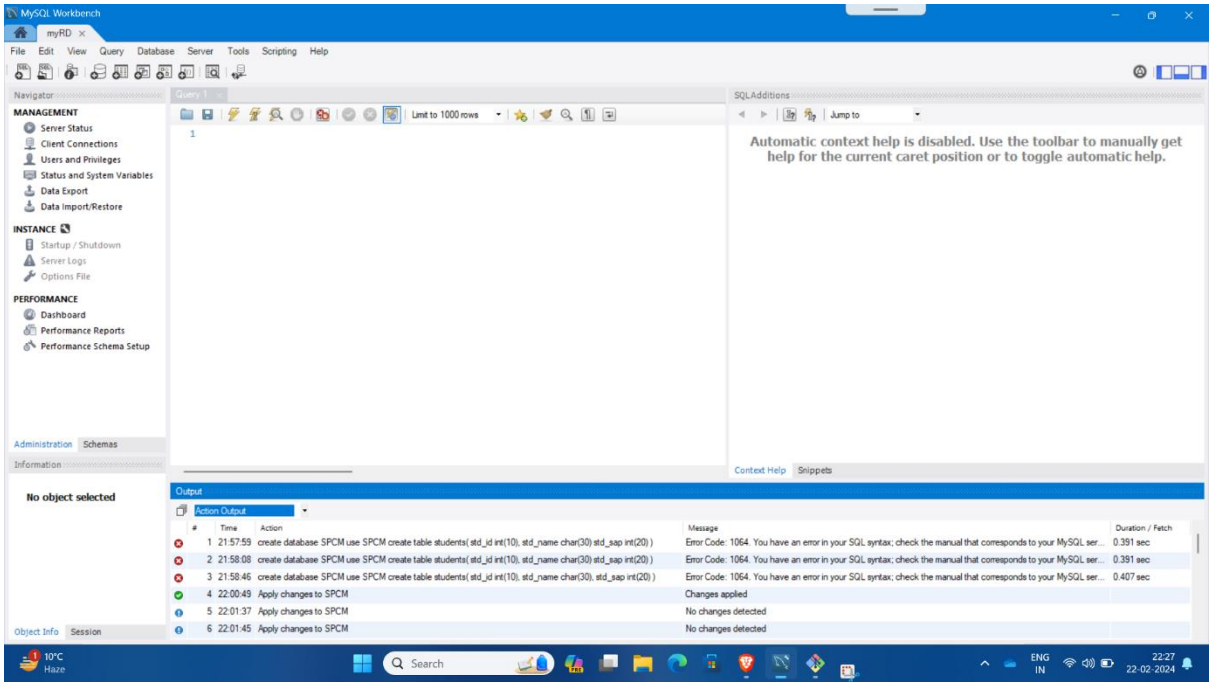
EXPERIMENT-10

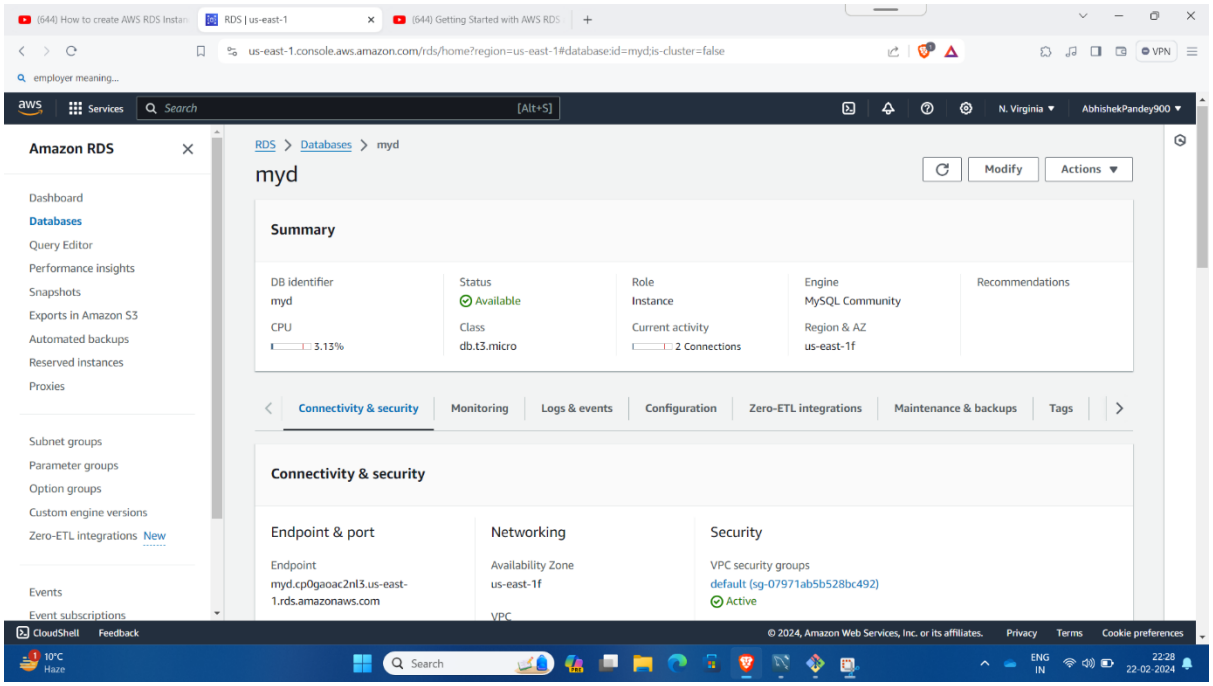
```
shivavatsal@Shivas-Air ~ % cd Documents
shivavatsal@Shivas-Air Documents % cd terraform
shivavatsal@Shivas-Air terraform % mkdir terraform-rds
shivavatsal@Shivas-Air terraform % cd terraform-rds
shivavatsal@Shivas-Air terraform-rds % vi main
shivavatsal@Shivas-Air terraform-rds % terraform init
Terraform initialized in an empty directory!

The directory has no Terraform configuration files. You may begin working
with Terraform immediately by creating Terraform configuration files.
shivavatsal@Shivas-Air terraform-rds % vi main
shivavatsal@Shivas-Air terraform-rds % vi main.tf
shivavatsal@Shivas-Air terraform-rds % 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!

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.
shivavatsal@Shivas-Air terraform-rds % terraform apply

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
      + apply_immediately                     = false
      + arn                                   = (known after apply)
```

```
terraform {
required_providers {
aws = {
source = "hashicorp/aws"
version = "5.31.0"
}
}
}
provider "aws" {
region = "ap-southeast-2"
access_key = "AKIA47CR2M26OV4Y53G3"
secret_key = "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"
}
resource "aws_db_instance" "My-RDS" {
 allocated_storage = 10
 db_name = "upesdb"
 engine = "mysql"
engine_version = "5.7"
 instance_class = "db.t2.micro"
 username = "shivaa"
 password = "shivaa111"
 parameter_group_name = "default.mysql5.7"
 skip_final_snapshot = true
 }

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"main.tf" 25L, 503B
```

```
      + replicas                              = (known after apply)
      + resource_id                           = (known after apply)
      + skip_final_snapshot                   = true
      + snapshot_identifier                   = (known after apply)
      + status                                = (known after apply)
      + storage_throughput                    = (known after apply)
      + storage_type                          = (known after apply)
      + tags_all                              = (known after apply)
      + timezone                              = (known after apply)
      + username                              = "shivaa"
      + vpc_security_group_ids                = (known after apply)
    }

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

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_db_instance.My-RDS: Creating...
aws_db_instance.My-RDS: Still creating... [10s elapsed]
aws_db_instance.My-RDS: Still creating... [20s elapsed]
aws_db_instance.My-RDS: Still creating... [30s elapsed]
aws_db_instance.My-RDS: Still creating... [40s elapsed]
aws_db_instance.My-RDS: Still creating... [50s elapsed]
aws_db_instance.My-RDS: Still creating... [1m0s elapsed]
aws_db_instance.My-RDS: Still creating... [1m10s elapsed]
aws_db_instance.My-RDS: Still creating... [1m20s elapsed]
aws_db_instance.My-RDS: Still creating... [1m30s elapsed]
aws_db_instance.My-RDS: Still creating... [1m40s elapsed]
aws_db_instance.My-RDS: Still creating... [1m50s elapsed]
aws_db_instance.My-RDS: Still creating... [2m0s elapsed]
aws_db_instance.My-RDS: Still creating... [2m10s elapsed]
aws_db_instance.My-RDS: Still creating... [2m20s elapsed]
aws_db_instance.My-RDS: Still creating... [2m30s elapsed]
aws_db_instance.My-RDS: Still creating... [2m40s elapsed]
aws_db_instance.My-RDS: Still creating... [2m50s elapsed]
aws_db_instance.My-RDS: Still creating... [3m0s elapsed]
aws_db_instance.My-RDS: Still creating... [3m10s elapsed]
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: Creation complete after 3m59s [id=db-4KCRF5BWNPXNOUEGGIOQKBLZYU]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
shivavatsal@Shivas-Air terraform-rds %
```




MySQL Workbench
myRD
File Edit View Query Database Server Tools Scripting Help
Navigator
MANAGEMENT
Server Status
Client Connections
Users and Privileges
Status and System Variables
Data Export
Data Import/Restore
INSTANCE
Startup / Shutdown
Server Logs
Options File
PERFORMANCE
Dashboard
Performance Reports
Performance Schema Setup
Administration Schemas
Information
No object selected
Object Info Session
Query 1
Limit to 1000 rows
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Context Help Snippets
Output
Action Output
Time Action Message Duration / Fetch
1 21:57:59 create database SPCM use SPCM create table students(std_id int(10), std_name char(30) std_sap int(20)) Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL ser... 0.391 sec
2 21:58:08 create database SPCM use SPCM create table students(std_id int(10), std_name char(30) std_sap int(20)) Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL ser... 0.391 sec
3 21:58:46 create database SPCM use SPCM create table students(std_id int(10), std_name char(30), std_sap int(20)) Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL ser... 0.407 sec
4 22:00:49 Apply changes to SPCM Changes applied
5 22:01:37 Apply changes to SPCM No changes detected
6 22:01:45 Apply changes to SPCM No changes detected
10°C Haze
Search
ENG IN
22:27 22-02-2024


(644) How to create AWS RDS Instan
RDS | us-east-1
(644) Getting Started with AWS RDS
us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#database:id=myd;is-cluster=false
VPN
employer meaning...
aws Services Search [Alt+S]
N. Virginia
AbhishekPandey900
Amazon RDS
Dashboard
Databases
Query Editor
Performance insights
Snapshots
Exports in Amazon S3
Automated backups
Reserved instances
Proxies
Subnet groups
Parameter groups
Option groups
Custom engine versions
Zero-ETL integrations New
Events
Event subscriptions
RDS > Databases > myd
myd
Modify
Actions
Summary
DB identifier
myd
Status
Available
Role
Instance
Engine
MySQL Community
Recommendations
CPU
3.13%
Class
db.t3.micro
Current activity
2 Connections
Region & AZ
us-east-1f
Connectivity & security
Monitoring
Logs & events
Configuration
Zero-ETL integrations
Maintenance & backups
Tags
Connectivity & security
Endpoint & port
Endpoint
myd.cp0gaoac2nl3.us-east-1.rds.amazonaws.com
Networking
Availability Zone
us-east-1f
VPC
Security
VPC security groups
default (sg-07971ab5b528bc492)
Active
CloudShell Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy Terms Cookie preferences
10°C Haze
Search
ENG IN
22:28 22-02-2024