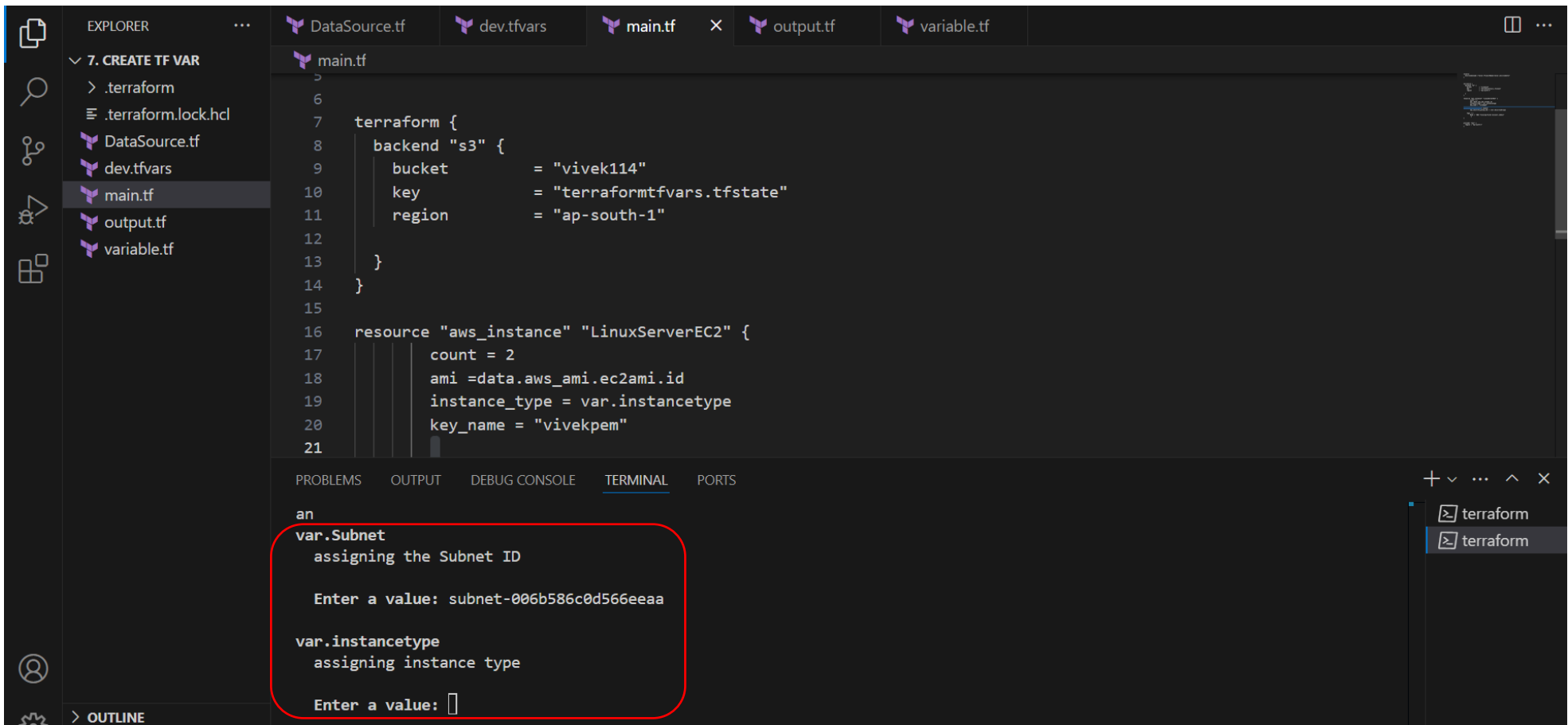
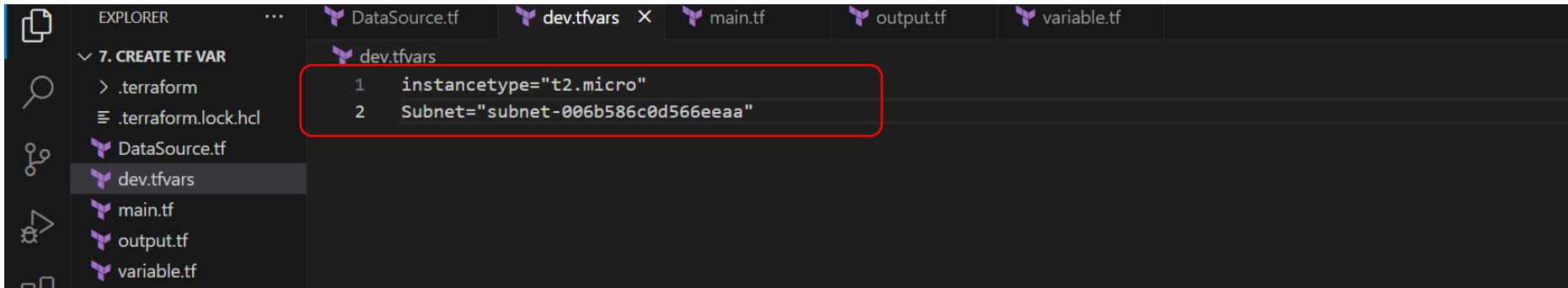


TFvar File Execution:

\*Subnet and instance type were not assigned default

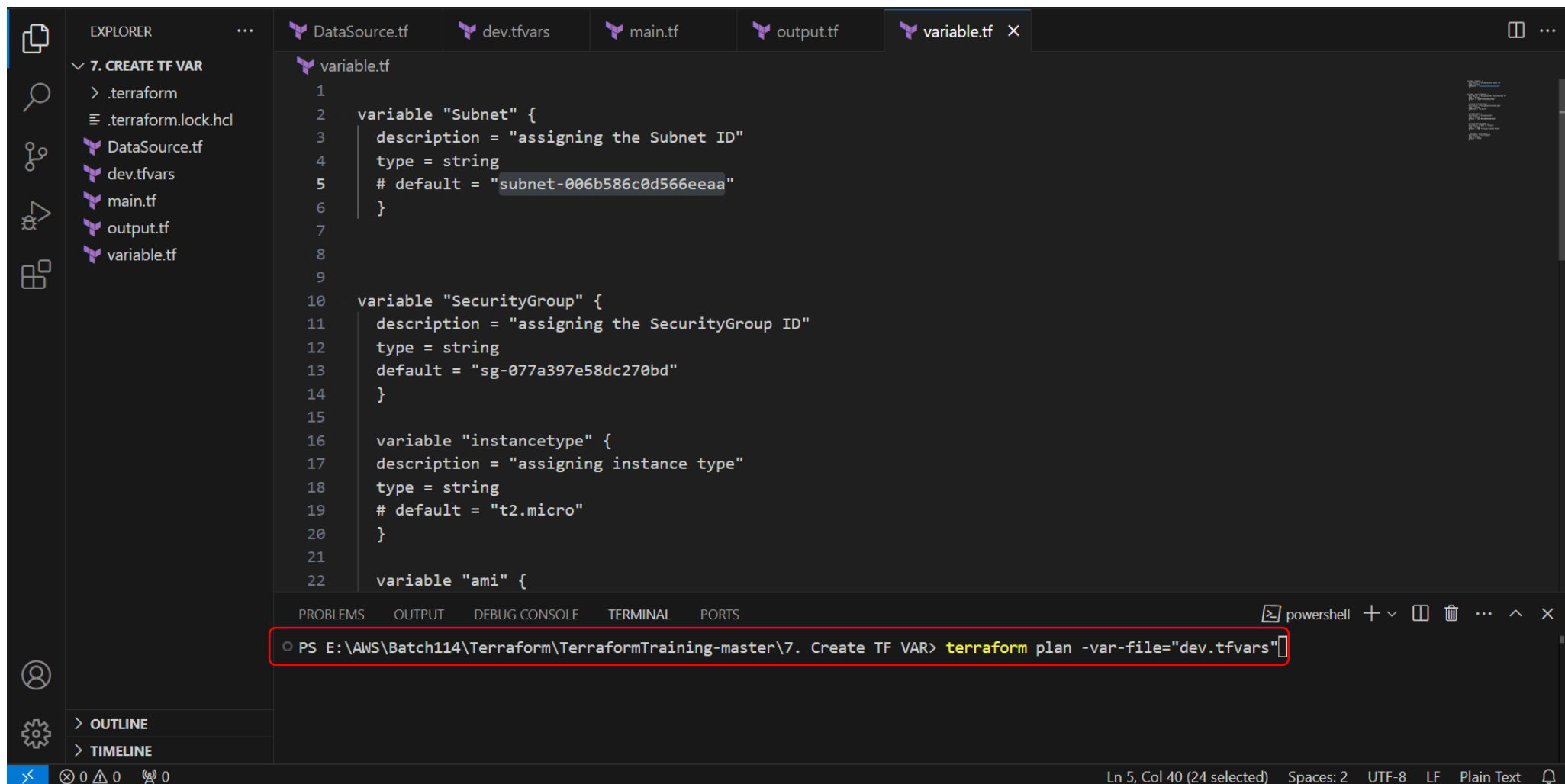
When terraform apply was executed, Subnet ID was asked initially, upon providing Subnet ID, Instance type was asked.

```
1
2 variable "Subnet" {
3   description = "assigning the Subnet ID"
4   type = string
5   # default = "subnet-006b586c0d566eeaa"
6 }
7
8
9
10 variable "SecurityGroup" {
11   description = "assigning the SecurityGroup ID"
12   type = string
13   default = "sg-077a397e58dc270bd"
14 }
15
16 variable "instancetype" {
17   description = "assigning instance type"
18   type = string
19   # default = "t2.micro"
20 }
```



Upon executing the command `terraform plan -var-file="dev.tfvars"`

The values for subnet and instance type were taken directly from dev.tfvars and the plan command was executed to show addition of 2 instances as per the code written.



```
variable.tf
1
2  variable "Subnet" {
3      description = "assigning the Subnet ID"
4      type = string
5      # default = "subnet-006b586c0d566eaa"
6  }
7
8
9
10 variable "SecurityGroup" {
11     description = "assigning the SecurityGroup ID"
12     type = string
13     default = "sg-077a397e58dc270bd"
14 }
15
16 variable "instancetype" {
17     description = "assigning instance type"
18     type = string
19     # default = "t2.micro"
20 }
21
22 variable "ami" {
```

```
PS E:\AWS\Batch114\Terraform\TerraformTraining-master\7. Create TF VAR> terraform plan -var-file="dev.tfvars"
```

EXPLORER

7. CREATE TF VAR

> .terraform

terraform.lock.hcl

DataSource.tf

dev.tfvars

main.tf

output.tf

variable.tf

OUTLINE

TIMELINE

DataSource.tf

dev.tfvars

main.tf

output.tf

variable.tf

variable.tf

1

2 variable "Subnet" {

3 description = "assigning the Subnet ID"

4 type = string

5 # default = "subnet-006b586c0d566eaaa"

6 }

7

8

9

10 variable "SecurityGroup" {

11 description = "assigning the SecurityGroup ID"

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

powershell

+

^

□

🗑

⋮

^

×

+ root\_block\_device (known after apply)

}

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

Changes to Outputs:

+ OutputInstance\_public\_ip = [

+ (known after apply),

+ (known after apply),

]

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

PS E:\AWS\Batch114\Terraform\TerraformTraining-master\7. Create TF VAR>

<

0

0

0

0

Ln 5, Col 40 (24 selected)

Spaces: 2

UTF-8

LF

Plain Text

🔔

EXPLORER

7. CREATE TF VAR

.terraform

.terraform.lock.hcl

DataSource.tf

dev.tfvars

main.tf

output.tf

variable.tf

OUTLINE

TIMELINE

DataSource.tf

dev.tfvars

main.tf

output.tf

variable.tf

main.tf

```
16 resource "aws_instance" "LinuxServerEC2" {
17     count = 2
18     ami = data.aws_ami.ec2ami.id
19     instance_type = var.instance_type
20     key_name = "vivekpem"
21
22     subnet_id = var.Subnet
23     vpc_security_group_ids = [var.SecurityGroup]
24
25     tags = {
26         Name = "AWS Training-Vivek ${count.index}"
27     }
28 }
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

powershell

+ instance\_type = "t2.micro"

+ ipv6\_address\_count = (known after apply)

+ ipv6\_addresses = (known after apply)

+ key\_name = "vivekpem"

+ 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)

Ln 21, Col 9 Spaces: 2 UTF-8 LF Plain Text

EXPLORER

7. CREATE TF VAR

.terraform

.terraform.lock.hcl

DataSource.tf

dev.tfvars

main.tf

output.tf

variable.tf

OUTLINE

TIMELINE

DataSource.tf

dev.tfvars

main.tf

output.tf

variable.tf

main.tf

```
16 resource "aws_instance" "LinuxServerEC2" {
17     count = 2
18     ami = data.aws_ami.ec2ami.id
19     instance_type = var.instance_type
20     key_name = "vivekpem"
21
22     subnet_id = var.Subnet
23     vpc_security_group_ids = [var.SecurityGroup]
24
25     tags = {
26         Name = "AWS Training-Vivek ${count.index}"
27     }
28 }
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

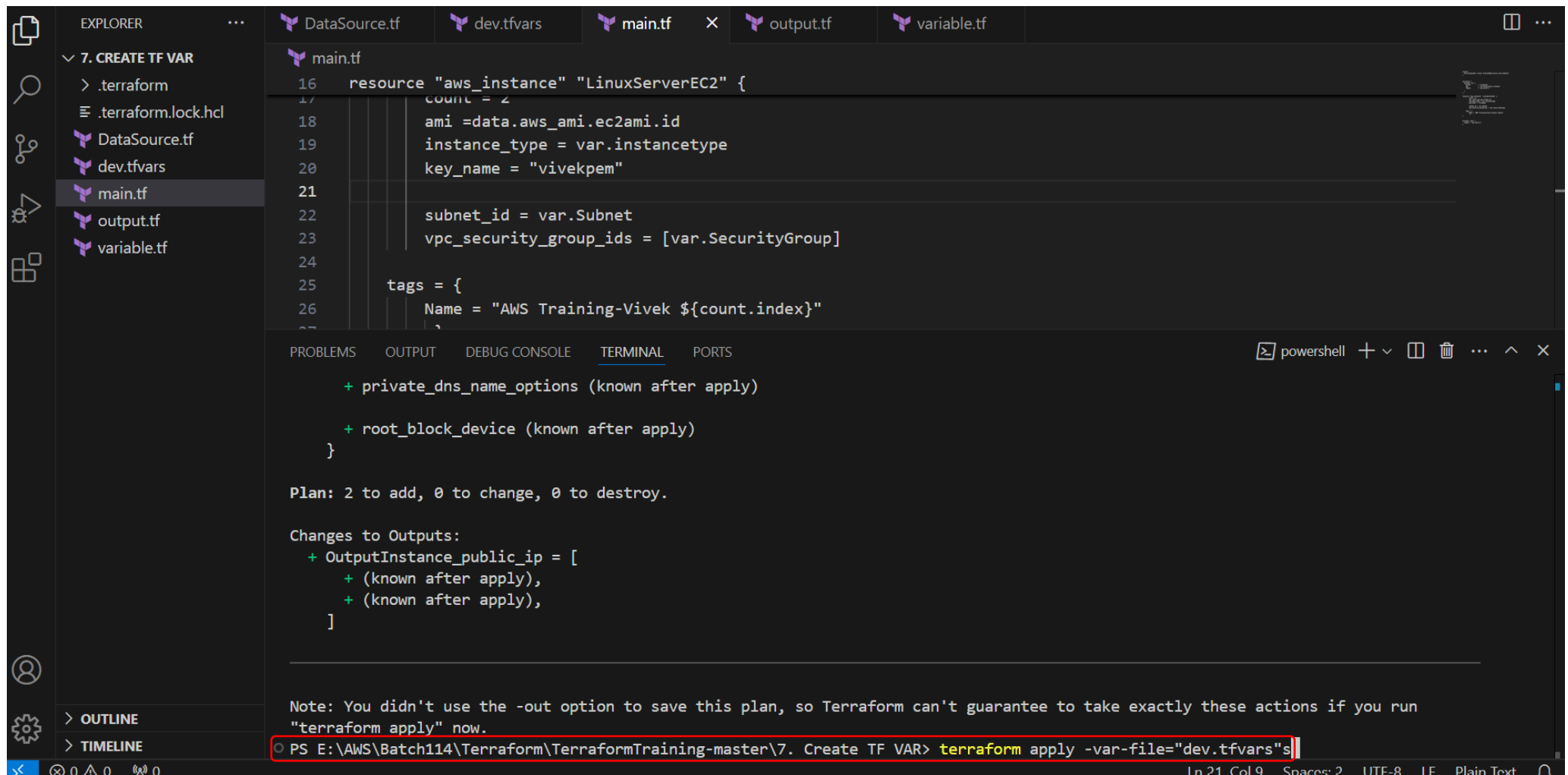
PORTS

powershell

```
+ key_name = "vivekpem"
+ 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 = "subnet-006b586c0d566eeaa"
+ tags = {
    + "Name" = "AWS Training-Vivek 1"
```

Upon executing the command `terraform apply -var-file="dev.tfvars"`

The values for subnet and instance type were taken directly from `dev.tfvars` and the apply command was executed and instances were created in our AWS Counsel and the state file was stored in the S3 bucket as mentioned in the code under the name `terraformtfvars.tfstate` as per written in the code.



The screenshot shows the Visual Studio Code interface with a Terraform project. The Explorer pane on the left shows the file structure under '7. CREATE TF VAR', including `.terraform`, `.terraform.lock.hcl`, `DataSource.tf`, `dev.tfvars`, `main.tf`, `output.tf`, and `variable.tf`. The `main.tf` file is open in the editor, showing a Terraform resource definition for `aws_instance` named `LinuxServerEC2`. The resource is configured with `count = 2`, `ami = data.aws_ami.ec2ami.id`, `instance_type = varinstancetype`, `key_name = "vivekpem"`, `subnet_id = var.Subnet`, `vpc_security_group_ids = [var.SecurityGroup]`, and a `tags` block with `Name = "AWS Training-Vivek ${count.index}"`. The terminal pane at the bottom shows the output of the `terraform apply` command, including the plan for adding 2 instances and the changes to outputs. The command prompt shows the user is in the directory `E:\AWS\Batch114\Terraform\TerraformTraining-master\7. Create TF VAR` and has executed `terraform apply -var-file="dev.tfvars"`.

```
16 resource "aws_instance" "LinuxServerEC2" {
17     count = 2
18     ami = data.aws_ami.ec2ami.id
19     instance_type = varinstancetype
20     key_name = "vivekpem"
21
22     subnet_id = var.Subnet
23     vpc_security_group_ids = [var.SecurityGroup]
24
25     tags = {
26         Name = "AWS Training-Vivek ${count.index}"
27     }
28 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

+ private\_dns\_name\_options (known after apply)

+ root\_block\_device (known after apply)

}

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

Changes to Outputs:

+ OutputInstance\_public\_ip = [

+ (known after apply),

+ (known after apply),

]

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

PS E:\AWS\Batch114\Terraform\TerraformTraining-master\7. Create TF VAR> terraform apply -var-file="dev.tfvars"





EC2 Dashboard

EC2 Global View

Events

Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity

Reservations

Images

Instances (2) Info

Last updated less than a minute ago

Connect

Instance state

Actions

Launch instances

Find Instance by attribute or tag (case-sensitive)

All states

Instance state = running

Clear filters

< 1 >

	Name	Instance ID	Instance state	Instance type	Status check	Alarm s
<input type="checkbox"/>	AWS Training-Vivek 1	i-06e832a0bae01e79f	Running	t2.micro	2/2 checks passec	View ala
<input type="checkbox"/>	AWS Training-Vivek 0	i-093c0a718f94280c4	Running	t2.micro	Initializing	View ala

Select an instance

Instance summary for i-06e832a0bae01e79f (AWS Training-Vivek 1) Info




Connect

Instance state ▼

Actions ▼

Updated 1 minute ago

Instance ID

  
i-06e832a0bae01e79f (AWS Training-Vivek 1)

IPv6 address

–


Hostname type

IP name: ip-192-168-0-4.ap-south-1.compute.internal

Answer private resource DNS name

–


Public IPv4 address

 65.1.114.44 | [open address](#) 

Instance state

 Running

Private IP DNS name (IPv4 only)

  
ip-192-168-0-4.ap-south-1.compute.internal

Instance type

t2.micro

Private IPv4 addresses

 192.168.0.4

Public IPv4 DNS

–

Elastic IP addresses

–



EC2

EC2 Dashboard

EC2 Global View

Events

## ▼ Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity

Reservations [New](#)

## ▼ Images



Hostname type

IP name: ip-192-168-0-4.ap-south-1.compute.internal

Answer private resource DNS name

-

Auto-assigned IP address

65.1.114.44 [Public IP]

IAM Role

-

IMDSv2

Optional

EC2 recommends setting IMDSv2 to required

[Learn more](#)

Private IP DNS name (IPv4 only)



ip-192-168-0-4.ap-south-1.compute.internal

Instance type

t2.micro

VPC ID

vpc-0343bfe016dcf2454 (MyVPC) [Learn more](#)

Subnet ID

subnet-006b586c0d566eaa (Subnet1-MyVPC) [Learn more](#)

Instance ARN



arn:aws:ec2:ap-south-1:245712304097:instance/i-06a832a0b3c01a70f

Elastic IP addresses

-

AWS Compute Optimizer finding

Opt-in to AWS Compute Optimizer for recommendations.

[Learn more](#)

Auto Scaling Group name

-



EC2

EC2 Dashboard

EC2 Global View

Events

▼ Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity

Reservations [New](#)

▼ Images

[EC2](#) > [Instances](#) > i-093c0a718f94280c4

**Instance summary for i-093c0a718f94280c4 (AWS Training-Vivek 0)** [Info](#)



Connect

Instance state ▼

Actions ▼

Updated less than a minute ago

Instance ID



i-093c0a718f94280c4 (AWS Training-Vivek 0)

IPv6 address

–

Hostname type

IP name: ip-192-168-0-23.ap-south-1.compute.internal

Answer private resource DNS name

–

Public IPv4 address



13.201.29.26 | [open address](#)

Instance state

✔ Running

Private IP DNS name (IPv4 only)



ip-192-168-0-23.ap-south-1.compute.internal

Instance type

t2.micro

Private IPv4 addresses



192.168.0.23

Public IPv4 DNS

–

Elastic IP addresses

–

aws

Services

Search [Alt+S]

EC2

EC2 Dashboard

EC2 Global View

Events

Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity

Reservations [New](#)

Images

IPv6 address

—

Hostname type

IP name: ip-192-168-0-23.ap-south-1.compute.internal

Answer private resource DNS name

—

Auto-assigned IP address

13.201.29.26 [Public IP]

IAM Role

—

Instance state

Running

Private IP DNS name (IPv4 only)

ip-192-168-0-23.ap-south-1.compute.internal

Instance type

t2.micro

VPC ID

vpc-0343bfe016dcf2454 (MyVPC)

Subnet ID

subnet-006b586c0d566eeaa (Subnet1-MyVPC)

Public IPv4 DNS

—

Elastic IP addresses

—

AWS Compute Optimizer finding

Opt-in to AWS Compute Optimizer for recommendations.

[Learn more](#)

Auto Scaling Group name

—

**Amazon S3**

**Buckets**

- Access Grants
- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- IAM Access Analyzer for S3

Block Public Access settings for this account

**Storage Lens**

- Dashboards

**Objects (4)** Info

Copy S3 URI Copy URL Download Open Delete Actions

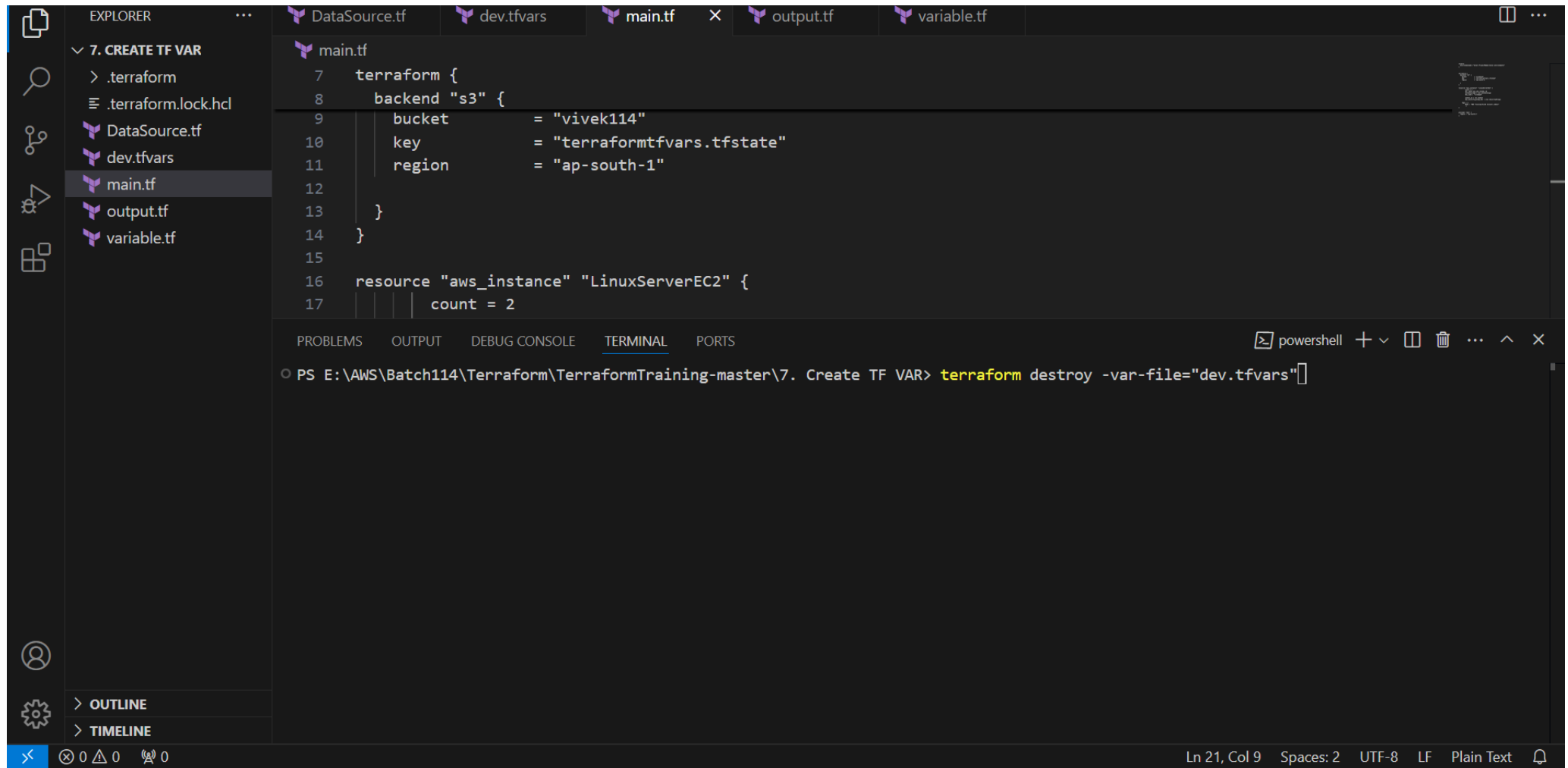
Create folder Upload

Find objects by prefix Show versions

	Name	Type	Last modified (UTC+05:30)	Size	Storage class
<input type="checkbox"/>	terraformcount.tfstate	tfstate	September 19, 2024, 23:50:23 (UTC+05:30)	180.0 B	Standard
<input type="checkbox"/>	terraformtfvars.tfstate	tfstate	September 20, 2024, 01:01:53 (UTC+05:30)	12.4 KB	Standard

Upon executing the command `terraform destroy -var-file="dev.tfvars"`

The instances created were destroyed, if only terraform destroy command was given in Terraform, the Subnet ID and instance type were asked as before for plan and apply.



The screenshot shows the Visual Studio Code interface with a Terraform project. The Explorer panel on the left shows a directory structure for '7. CREATE TF VAR' containing files like .terraform, .terraform.lock.hcl, DataSource.tf, dev.tfvars, main.tf, output.tf, and variable.tf. The main editor displays the content of main.tf, which defines a terraform backend 's3' and an aws\_instance resource 'LinuxServerEC2' with a count of 2. The terminal window at the bottom shows a PowerShell prompt where the command 'terraform destroy -var-file="dev.tfvars"' has been entered.

```
7 terraform {
8   backend "s3" {
9     bucket      = "vivek114"
10    key          = "terraformtfvars.tfstate"
11    region       = "ap-south-1"
12  }
13 }
14 }
15
16 resource "aws_instance" "LinuxServerEC2" {
17   count = 2
```

```
PS E:\AWS\Batch114\Terraform\TerraformTraining-master\7. Create TF VAR> terraform destroy -var-file="dev.tfvars"
```

EXPLORER

7. CREATE TF VAR

- .terraform
- .terraform.lock.hcl
- DataSource.tf
- dev.tfvars
- main.tf
- output.tf
- variable.tf

main.tf

```
7 terraform {
8   backend "s3" {
9     bucket      = "vivek114"
10    key          = "terraformtfvars.tfstate"
11    region       = "ap-south-1"
12  }
13 }
14 }
15
16 resource "aws_instance" "LinuxServerEC2" {
17   count = 2
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

powershell

```
aws_instance.LinuxServerEC2[1]: Destroying... [id=i-06e832a0bae01e79f]
aws_instance.LinuxServerEC2[0]: Destroying... [id=i-093c0a718f94280c4]
aws_instance.LinuxServerEC2[0]: Still destroying... [id=i-093c0a718f94280c4, 10s elapsed]
aws_instance.LinuxServerEC2[1]: Still destroying... [id=i-06e832a0bae01e79f, 10s elapsed]
aws_instance.LinuxServerEC2[1]: Still destroying... [id=i-06e832a0bae01e79f, 20s elapsed]
aws_instance.LinuxServerEC2[0]: Still destroying... [id=i-093c0a718f94280c4, 20s elapsed]
aws_instance.LinuxServerEC2[1]: Still destroying... [id=i-06e832a0bae01e79f, 30s elapsed]
aws_instance.LinuxServerEC2[0]: Still destroying... [id=i-093c0a718f94280c4, 30s elapsed]
aws_instance.LinuxServerEC2[0]: Still destroying... [id=i-093c0a718f94280c4, 40s elapsed]
aws_instance.LinuxServerEC2[1]: Still destroying... [id=i-06e832a0bae01e79f, 40s elapsed]
aws_instance.LinuxServerEC2[1]: Still destroying... [id=i-06e832a0bae01e79f, 50s elapsed]
aws_instance.LinuxServerEC2[0]: Still destroying... [id=i-093c0a718f94280c4, 50s elapsed]
aws_instance.LinuxServerEC2[0]: Destruction complete after 51s
aws_instance.LinuxServerEC2[1]: Destruction complete after 51s

Destroy complete! Resources: 2 destroyed.
```

PS E:\AWS\Batch114\Terraform\TerraformTraining-master\7. Create TF VAR>

Ln 21, Col 9 Spaces: 2 UTF-8 LF Plain Text



EC2 Dashboard

EC2 Global View

Events

▼ Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances


Dedicated Hosts

Capacity

Reservations [New](#)

▼ Images

Instances (4) [Info](#)

Last updated less than a minute ago 


Connect










Instance state ▾

Actions ▾

Launch instances ▾

All states ▾

< 1 > 

<input type="checkbox"/>	Name 	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm s
<input type="checkbox"/>	linux	<a href="#">i-0cab4259817bf4526</a>	⊖ Stopped  	t2.micro	–	<a href="#">View ala</a>
<input type="checkbox"/>	MyInstance	<a href="#">i-0df0b8943dbdfe388</a>	⊖ Stopped  	t2.micro	–	<a href="#">View ala</a>
<input type="checkbox"/>	AWS Training-Vivek 1	<a href="#">i-06e832a0bae01e79f</a>	⊖ Terminated  	t2.micro	–	<a href="#">View ala</a>
<input type="checkbox"/>	AWS Training-Vivek 0	<a href="#">i-093c0a718f94280c4</a>	⊖ Terminated  	t2.micro	–	<a href="#">View ala</a>

Select an instance