

Lesson 09 Demo 08

Validating, Applying, and Destroying the Terraform file

Objective: To validate, apply, and destroy the Terraform configuration file for efficient and reliable management of your infrastructure lifecycle

Tools required: VS Code and Linux terminal

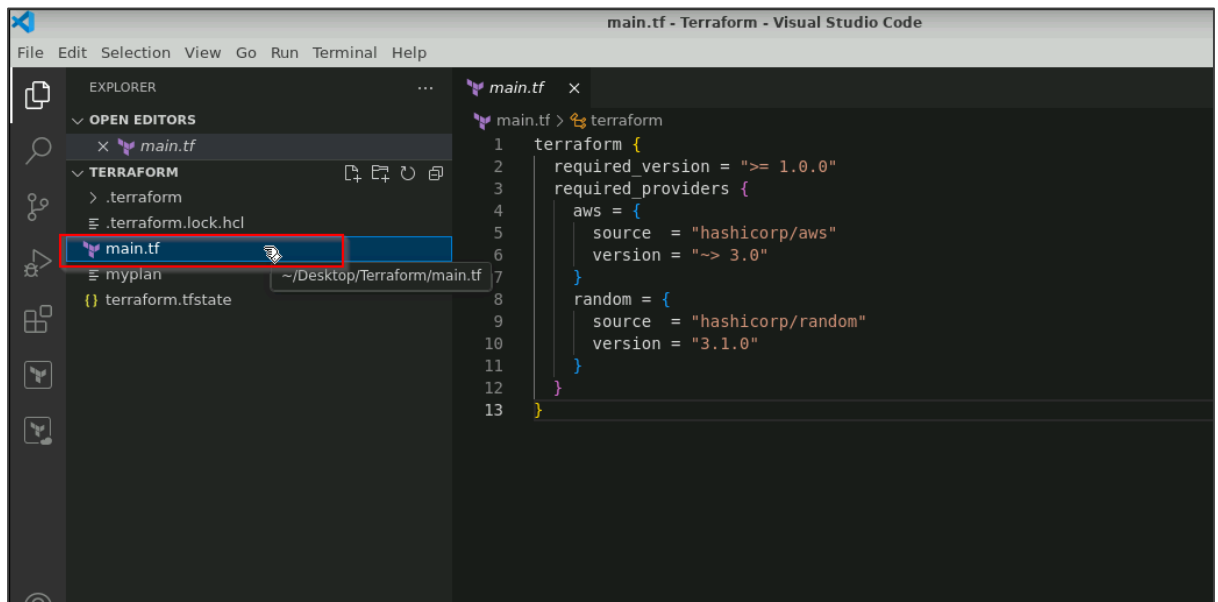
Prerequisites: None

Steps to be followed:

1. Modify the main.tf file with the provider configuration code
2. Validate the Terraform configuration file
3. Apply the validated Terraform configuration file
4. Destroy the applied Terraform configurations

Step 1: Modify the main.tf file with the provider configuration code

1.1 Open the main.tf file



1.2 Modify the **main.tf** file with the code provided below and save the file:

```
terraform {
  required_version = ">= 1.0.0"
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "~> 3.0"
    }
    random = {
      source = "hashicorp/random"
      version = "3.6.2"
    }
    http = {
      source = "hashicorp/http"
      version = "3.4.3"
    }
    local = {
      source = "hashicorp/local"
      version = "2.5.1"
    }
    tls = {
      source = "hashicorp/tls"
      version = "3.1.0"
    }
  }
}

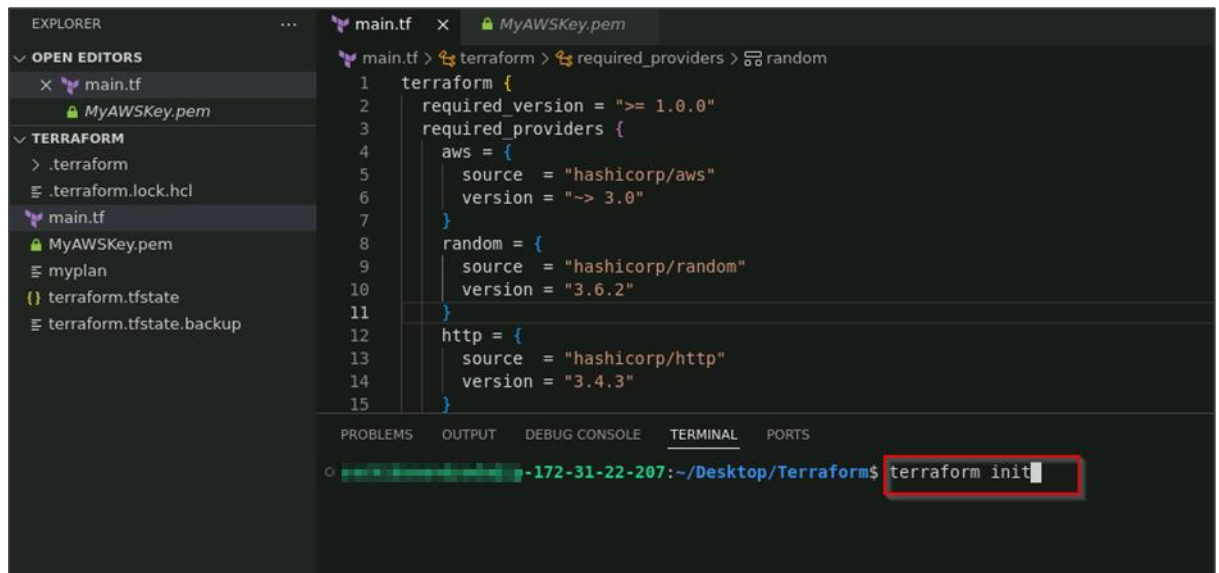
provider "aws" {
  # Replace with your actual AWS credentials
  access_key = "AKIARJTG7GGYJOVDRU3B"
  secret_key = "kq2lAmP5ajai+VEhdHMcic4fXmUMcpQM3avt1wD"
  region     = "us-east-1" # Replace with your desired region
}

resource "tls_private_key" "generated" {
  algorithm = "RSA"
}

resource "local_file" "private_key_pem" {
  content = tls_private_key.generated.private_key_pem
  filename = "MyAWSKey.pem"
}

resource "random_string" "random" {
  length     = 9
  special    = true
  min_numeric = 6
  min_special = 2
  min_upper  = 3
}
```

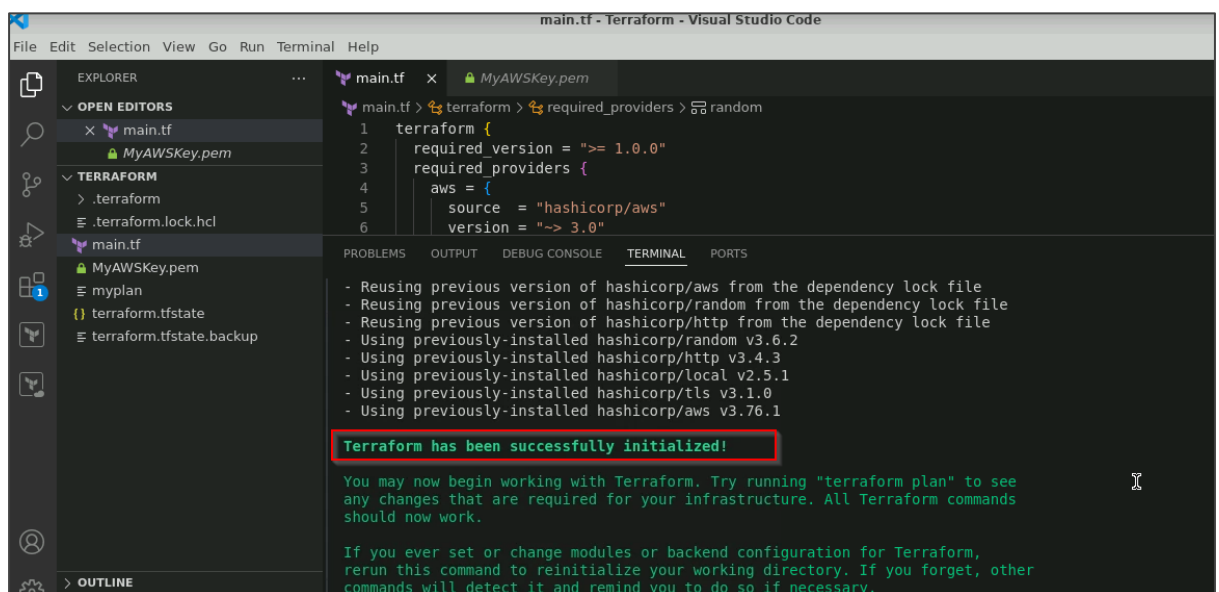
1.3 Run the command given below to initialize the configurations: **terraform init**



The screenshot shows the Visual Studio Code interface. On the left, the Explorer pane shows the file structure with 'main.tf' selected. The main editor displays the content of 'main.tf', which is a Terraform configuration file. The configuration includes a 'required_providers' block with 'aws' and 'random' providers, and an 'http' provider. The terminal at the bottom shows the command 'terraform init' entered at the prompt '-172-31-22-207:~/Desktop/Terraform\$'.

```
1 terraform {
2   required_version = ">= 1.0.0"
3   required_providers {
4     aws = {
5       source = "hashicorp/aws"
6       version = "~> 3.0"
7     }
8     random = {
9       source = "hashicorp/random"
10      version = "3.6.2"
11    }
12   }
13   http = {
14     source = "hashicorp/http"
15     version = "3.4.3"
16   }
17 }
```

Terminal: -172-31-22-207:~/Desktop/Terraform\$ terraform init



The screenshot shows the same Visual Studio Code interface as before, but the terminal now displays the output of the 'terraform init' command. The output lists the providers being used and their versions, indicating that Terraform has been successfully initialized. The message 'Terraform has been successfully initialized!' is highlighted with a red box.

```
- Reusing previous version of hashicorp/aws from the dependency lock file
- Reusing previous version of hashicorp/random from the dependency lock file
- Reusing previous version of hashicorp/http from the dependency lock file
- Using previously-installed hashicorp/random v3.6.2
- Using previously-installed hashicorp/http v3.4.3
- Using previously-installed hashicorp/local v2.5.1
- Using previously-installed hashicorp/tls v3.1.0
- Using previously-installed hashicorp/aws v3.76.1

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.

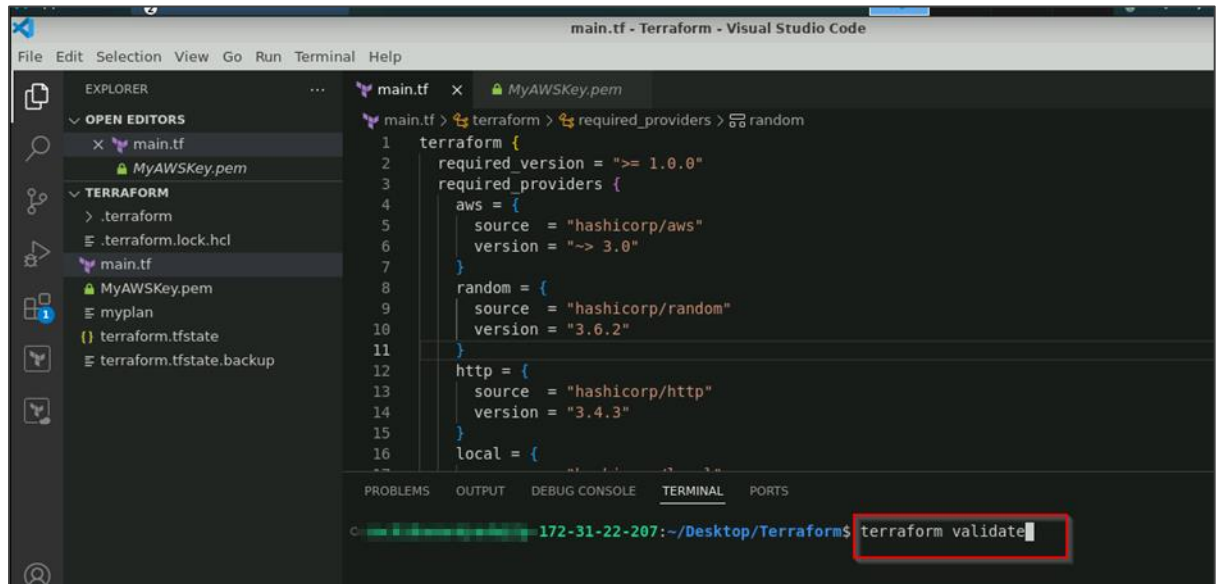
If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

The Terraform file is successfully initialized.

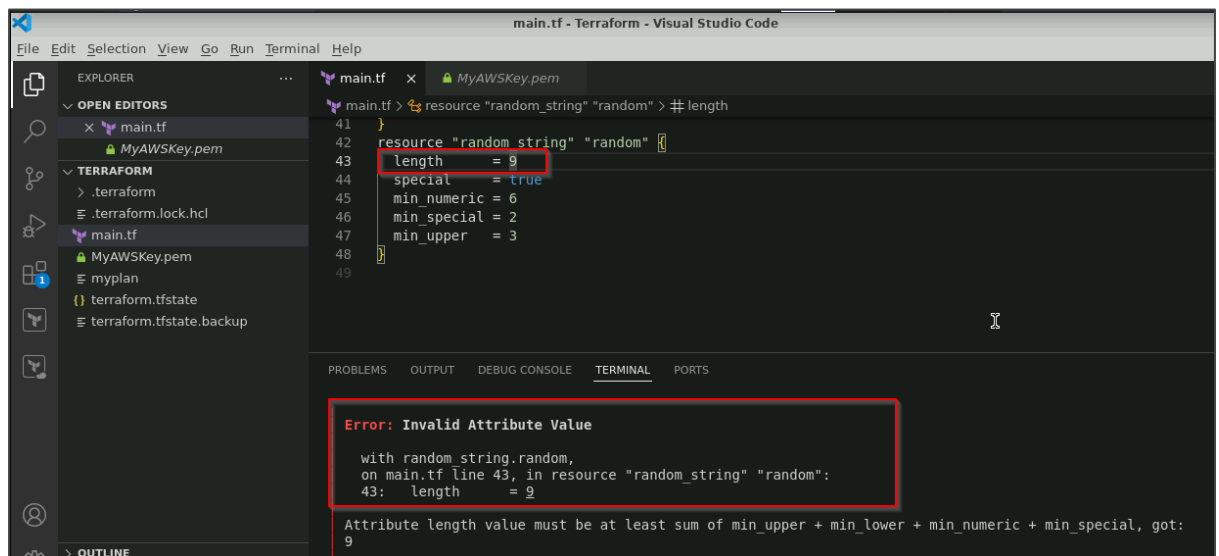
Step 2: Validate the Terraform configuration file

2.1 Open the terminal and execute the command given below to validate the Terraform configuration file:

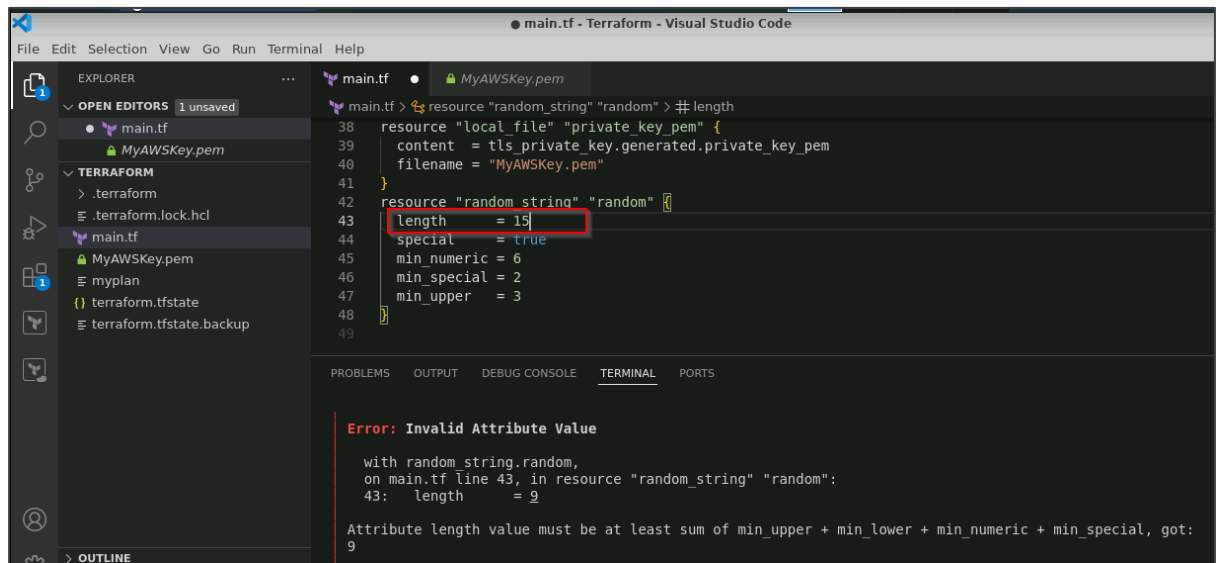
terraform validate



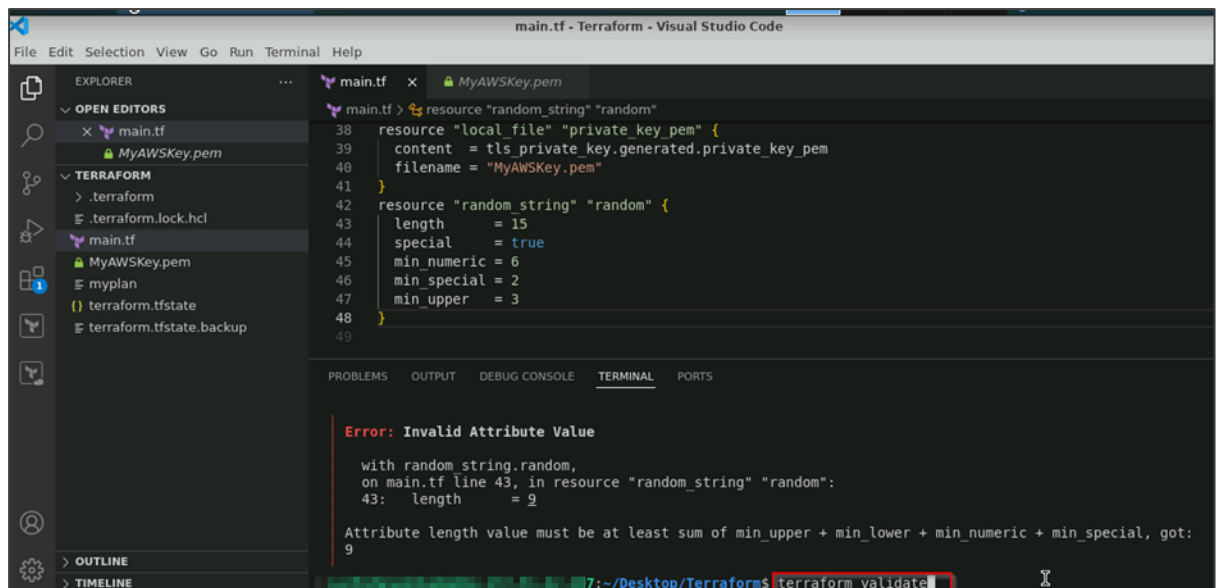
The validate command shows the error in the configuration file and suggests changes.



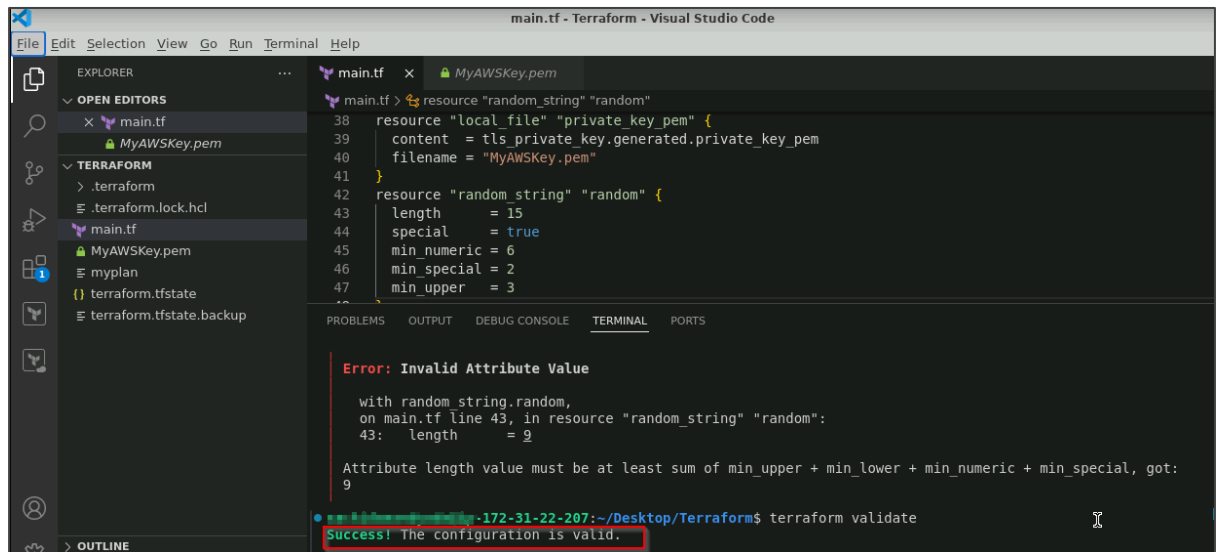
2.2 Make the changes as per the suggestions, replace the value of **length** with **15**, and save the file



2.3 Run the validate command once again to check the validation



The configuration is valid as the output shows:



The screenshot shows the Visual Studio Code interface with a Terraform configuration file named `main.tf` open. The file contains the following HCL code:

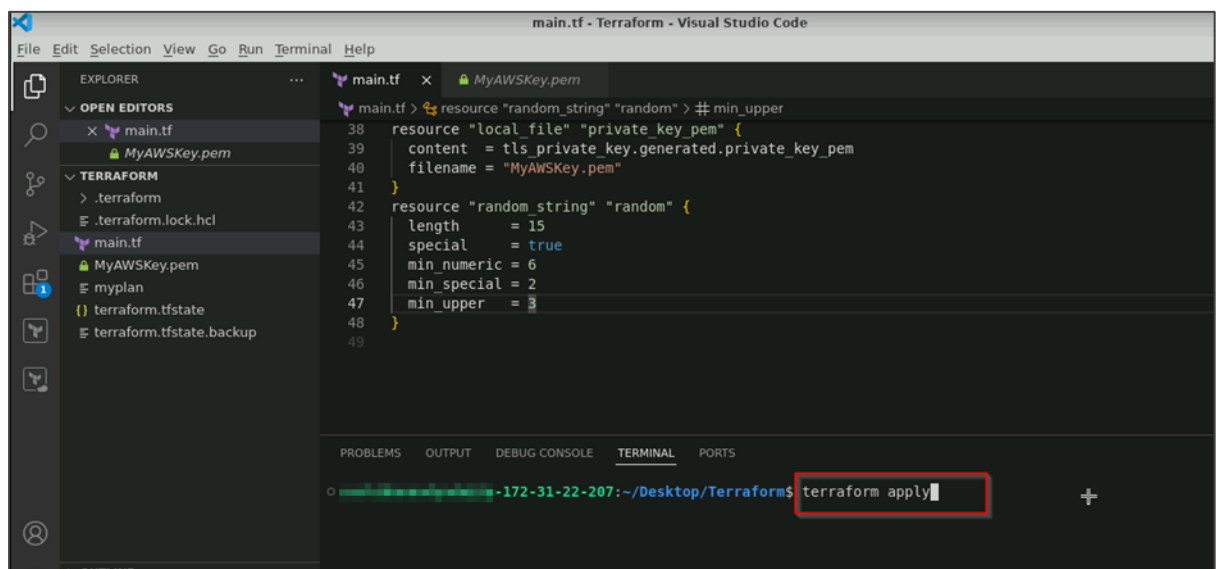
```
main.tf > resource "random_string" "random"
38 resource "local_file" "private_key_pem" {
39   content = tls_private_key.generated.private_key_pem
40   filename = "MyAWSKey.pem"
41 }
42 resource "random_string" "random" {
43   length     = 15
44   special    = true
45   min_numeric = 6
46   min_special = 2
47   min_upper  = 3
48 }
49
```

The terminal output shows the command `terraform validate` being executed, resulting in the message: `Success! The configuration is valid.`

Step 3: Apply the validated Terraform configuration file

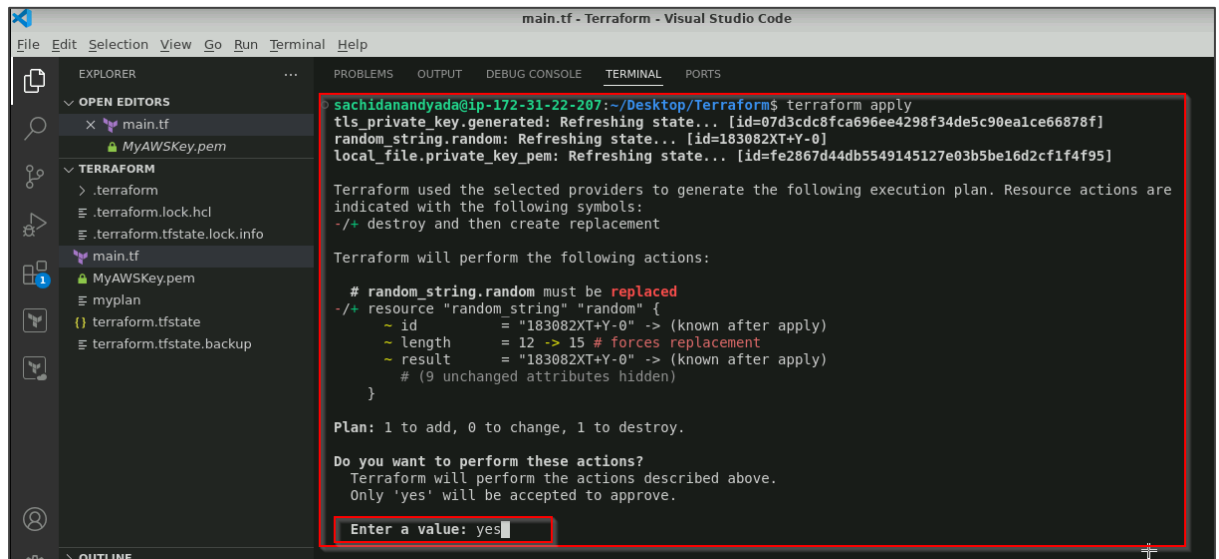
3.1 Open the terminal and enter the command given below to apply the configuration file:

terraform apply



The screenshot shows the Visual Studio Code interface with the same Terraform configuration file `main.tf` open. The terminal shows the command `terraform apply` being entered at the prompt `-172-31-22-207:~/Desktop/Terraform$`.

3.2 Confirm with **yes** to apply the configuration



```
main.tf - Terraform - Visual Studio Code
File Edit Selection View Go Run Terminal Help

EXPLORER
  OPEN EDITORS
    main.tf
    MyAWSKey.pem
  TERRAFORM
    .terraform
    .terraform.lock.hcl
    .terraform.tfstate.lock.info
    main.tf
    MyAWSKey.pem
    myplan
    terraform.tfstate
    terraform.tfstate.backup

sachidanandyada@ip-172-31-22-207:~/Desktop/Terraform$ terraform apply
tls_private_key.generated: Refreshing state... [id=07d3cdc8fca696ee4298f34de5c90ea1ce66878f]
random_string.random: Refreshing state... [id=183082XT+Y-0]
local_file.private_key.pem: Refreshing state... [id=fe2867d44db5549145127e03b5be16d2cf1f4f95]

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

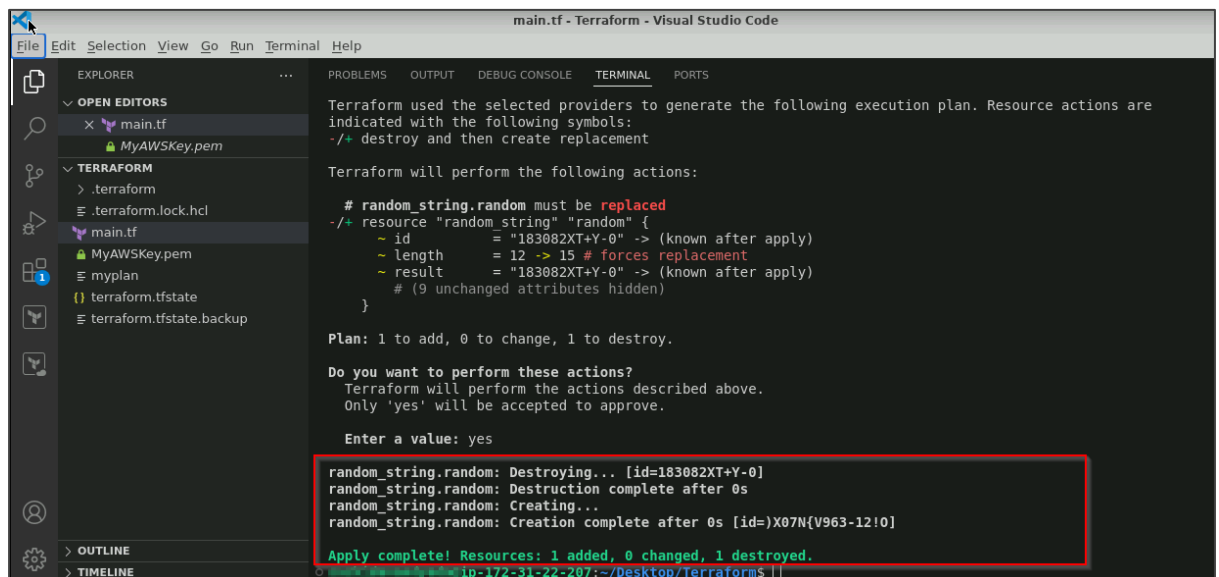
Terraform will perform the following actions:

# random_string.random must be replaced
-/+ resource "random_string" "random" {
  ~ id       = "183082XT+Y-0" -> (known after apply)
  ~ length   = 12 -> 15 # forces replacement
  ~ result    = "183082XT+Y-0" -> (known after apply)
  # (9 unchanged attributes hidden)
}

Plan: 1 to add, 0 to change, 1 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
```



```
main.tf - Terraform - Visual Studio Code
File Edit Selection View Go Run Terminal Help

EXPLORER
  OPEN EDITORS
    main.tf
    MyAWSKey.pem
  TERRAFORM
    .terraform
    .terraform.lock.hcl
    main.tf
    MyAWSKey.pem
    myplan
    terraform.tfstate
    terraform.tfstate.backup

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:

# random_string.random must be replaced
-/+ resource "random_string" "random" {
  ~ id       = "183082XT+Y-0" -> (known after apply)
  ~ length   = 12 -> 15 # forces replacement
  ~ result    = "183082XT+Y-0" -> (known after apply)
  # (9 unchanged attributes hidden)
}

Plan: 1 to add, 0 to change, 1 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

random_string.random: Destroying... [id=183082XT+Y-0]
random_string.random: Destruction complete after 0s
random_string.random: Creating...
random_string.random: Creation complete after 0s [id=X07N{V963-12!0}]

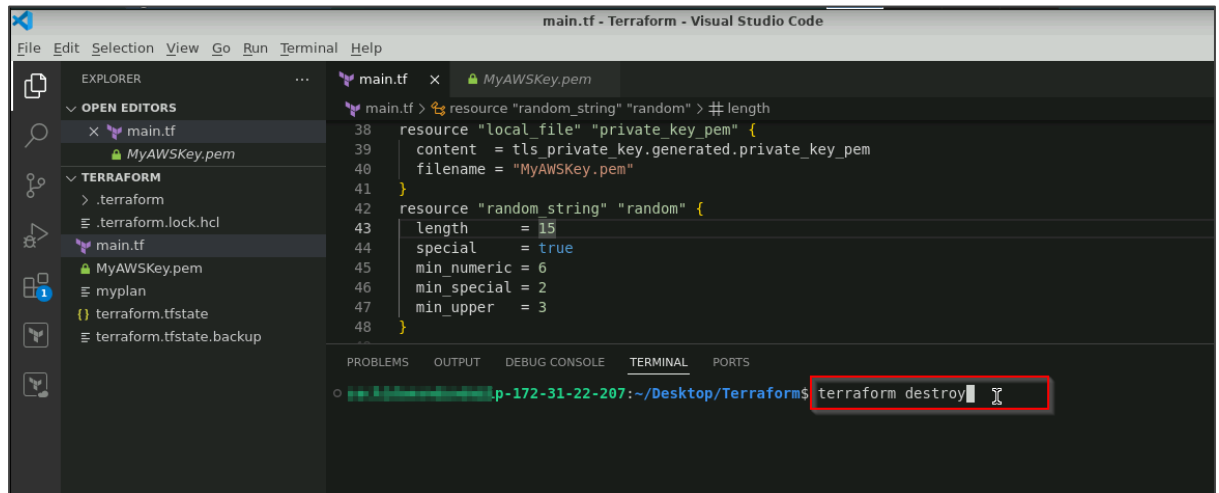
Apply complete! Resources: 1 added, 0 changed, 1 destroyed.

sachidanandyada@ip-172-31-22-207:~/Desktop/Terraform$
```

The apply command is completed successfully.

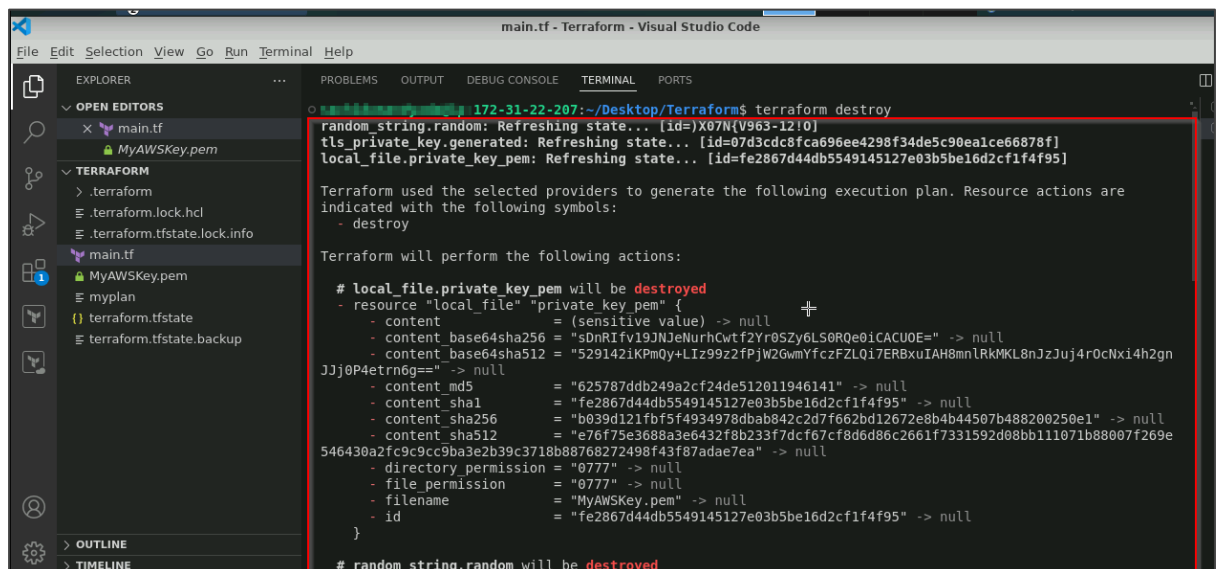
Step 4: Destroy the applied Terraform configurations

- 4.1 Open the terminal and execute the given command to destroy the configuration file:
terraform destroy



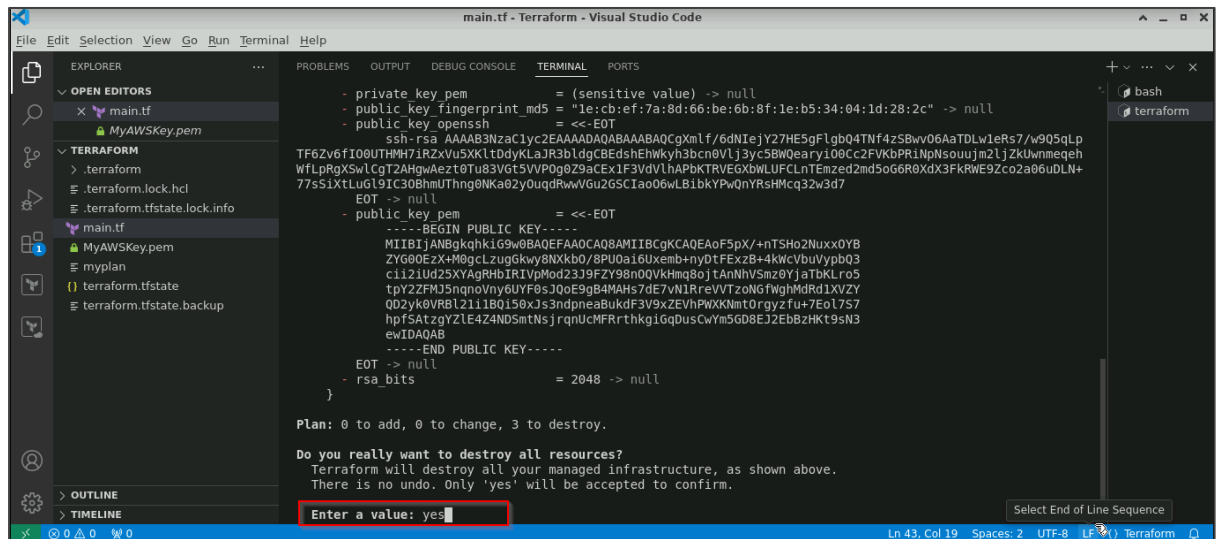
The screenshot shows the Visual Studio Code interface with the Terraform configuration files open. The Explorer pane on the left shows the project structure with files like `main.tf`, `MyAWSKey.pem`, `myplan`, `terraform.tfstate`, and `terraform.tfstate.backup`. The main editor shows the `main.tf` file with Terraform code. The terminal at the bottom shows the command `terraform destroy` being entered at the prompt `p-172-31-22-207:~/Desktop/Terraform$`.

The destroy command will initiate the destruction of the configuration files.



The screenshot shows the output of the `terraform destroy` command in the terminal. The output indicates that Terraform is refreshing the state for the `random_string.random`, `tls_private_key.generated`, and `local_file.private_key.pem` resources. It then shows the execution plan, indicating that the `local_file.private_key.pem` resource will be destroyed. The plan lists the actions for the `local_file.private_key.pem` resource, including setting the `content` to null and removing the `directory_permission`, `file_permission`, `filename`, and `id` attributes. The plan also indicates that the `random_string.random` resource will be destroyed.

4.2 Confirm with **yes** to destroy the configuration files



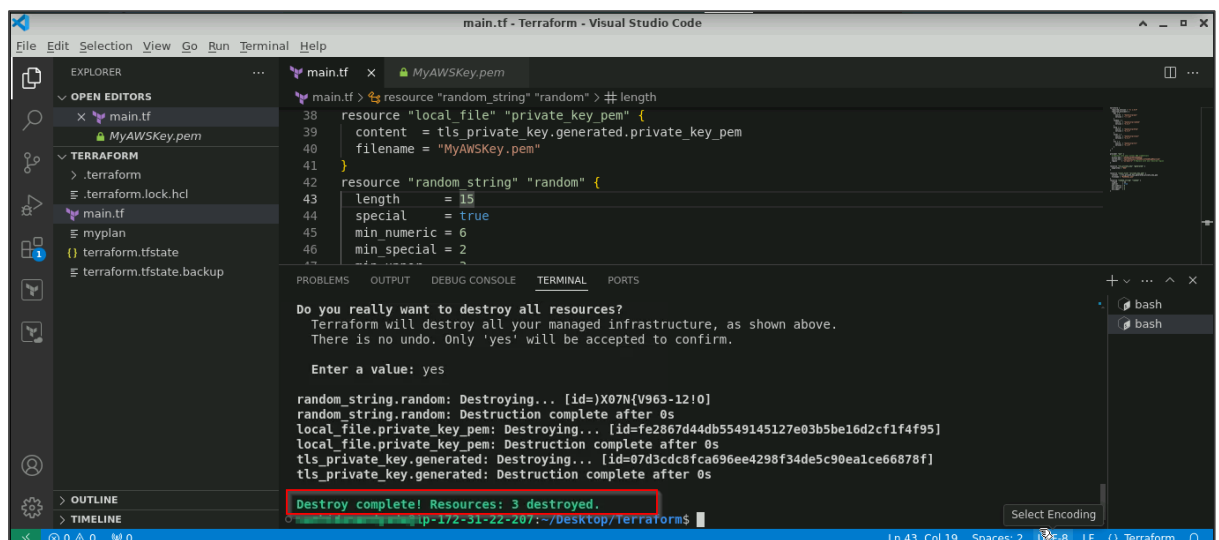
```
main.tf - Terraform - Visual Studio Code

- private key pem = (sensitive value) -> null
- public key fingerprint_md5 = "1e:cb:ef:7a:8d:66:be:6b:8f:1e:b5:34:04:1d:28:2c" -> null
- public key openssh = <<- EOT
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQACgXmLf/6dNIejY27HE5gF1gb04TNf4zSBwv06AaTDLw1eRs7/w905qLp
TF6Zv6fI08UTHMH7iRZxVu5XK1tdDyKLaJR3blgdCBEdshEhmkYh3bcn0Vlj3yc5BW0eary108Cc2FVKbPRiNpNsouujm2ljZkUwmeqeh
WfLpRgXSwLcgT2AHgwAeZt0T83Vgt5VVP0g029acEx1F3VdVlhaPbKTRVEGxbWLUfCLnTEmded2md5oG6R0Xdx3FkRWE9Zco2a06uDLN+
77s5iXtLuG19IC30BhmUThng8Nka02yDuqDRwVGu2G5Ciao06wLBibYPwQnYRsHMcq32w3d7
EOT -> null
- public key pem = <<- EOT
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQFAAQCAQ8AMIIBCgKCAQEAoF5pX/+nTSho2Nuxx0YB
ZYG00EzX+M0gCLzugGkwy8NXkb0/8PU0ai6Uxemb+nyDTFExzB+4KwCvbuVypbQ3
ci12iUd25XYAgRHbIRIVpMod23J9FZY98n0QVkhmq8ojtAnNhVSmz0YjaTbKLro5
tpY2ZFMJ5nqnoVny6UYF0sJ0oE9gB4MAHS7dE7vN1RreVVTzoNGfwghMdRd1XVZY
QD2yk0VRB12111BQ150xJs3ndpneaBukdF3V9x2EVhPwXKNmtOrgyzfu+7eol757
hpfSAtzgyZLE424ND5mtNsjrqnuCmFRrthkg1GqDuscWym5G08EJ2EbBzHKT9sN3
cwIDAQAB
-----END PUBLIC KEY-----
EOT -> null
}
rsa_bits = 2048 -> null

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
```



```
main.tf - Terraform - Visual Studio Code

main.tf x MyAWSKey.pem
main.tf x resource "random_string" "random" { length
38 resource "local_file" "private_key_pem" {
39 content = tls_private_key.generated.private_key_pem
40 filename = "MyAWSKey.pem"
41 }
42 resource "random_string" "random" {
43 length = 15
44 special = true
45 min_numeric = 6
46 min_special = 2
47 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

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

random_string.random: Destroying... [id=X07N{V963-1210}
random_string.random: Destruction complete after 0s
local_file.private_key_pem: Destroying... [id=fe2867d44db5549145127e03b5be16d2cf1f4f95]
local_file.private_key_pem: Destruction complete after 0s
tls_private_key.generated: Destroying... [id=07d3cdc8fca696ee4298f34de5c90ealce66878f]
tls_private_key.generated: Destruction complete after 0s

Destroy complete! Resources: 3 destroyed.
```

The Terraform configuration files are successfully destroyed.

By following these steps, you have successfully validated, applied, and destroyed the Terraform configuration file for efficient and reliable management of your infrastructure lifecycle.