

Lesson 09 Demo 03

Generating an SSH Key with a Terraform TLS Provider

Objective: To generate an SSH key with the Terraform TLS provider using the latest version and configurations for secure access to your cloud infrastructure

Tools required: VS Code and Linux terminal

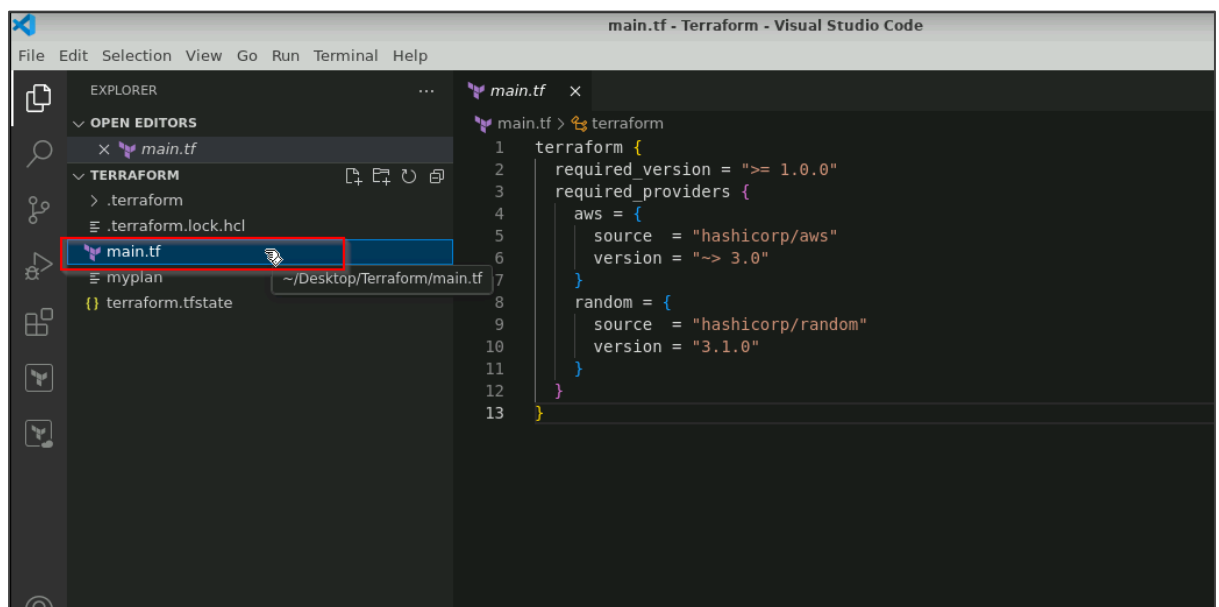
Prerequisites: Refer Demo 02_Running_Multiple_Providers_in_Terraform

Steps to be followed:

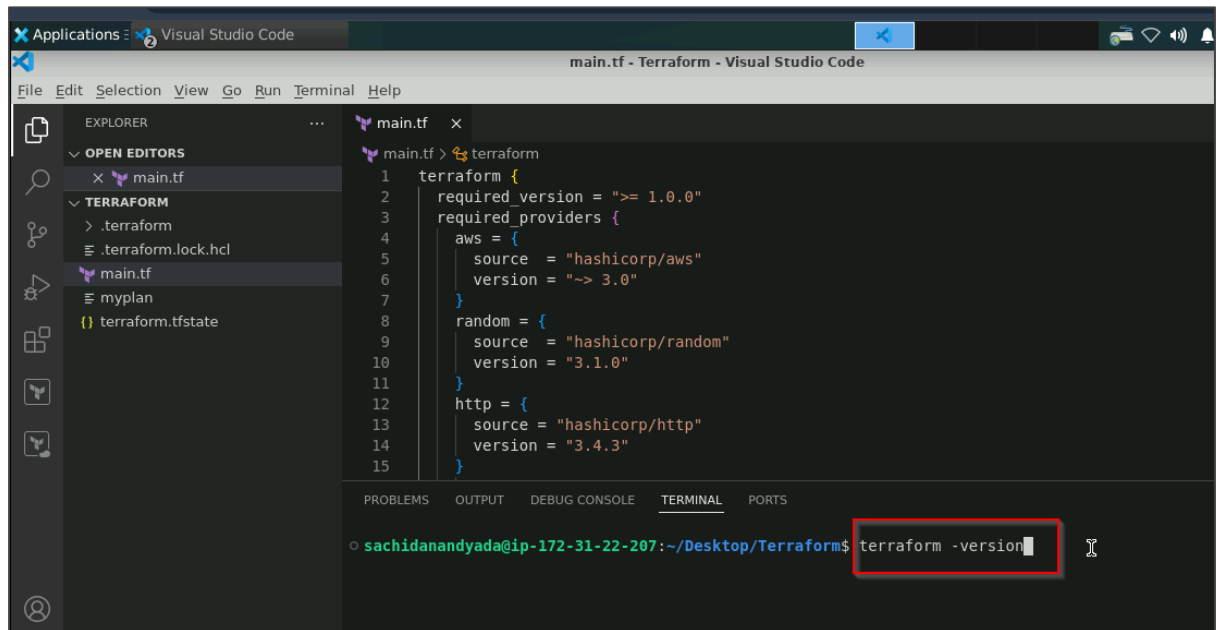
1. Check the version of the installed providers
2. Add the Terraform TLS provider configuration in the main.tf file
3. Initialize the Terraform configuration file
4. Update the main.tf file with a TLS certificate configuration block
5. Execute the apply command and validate the SSH key

Step 1: Check the version of the installed providers

1.1 Open the **main.tf** file previously created



- 1.2 Open the terminal and execute the following command to check the Terraform version:
- terraform -version**

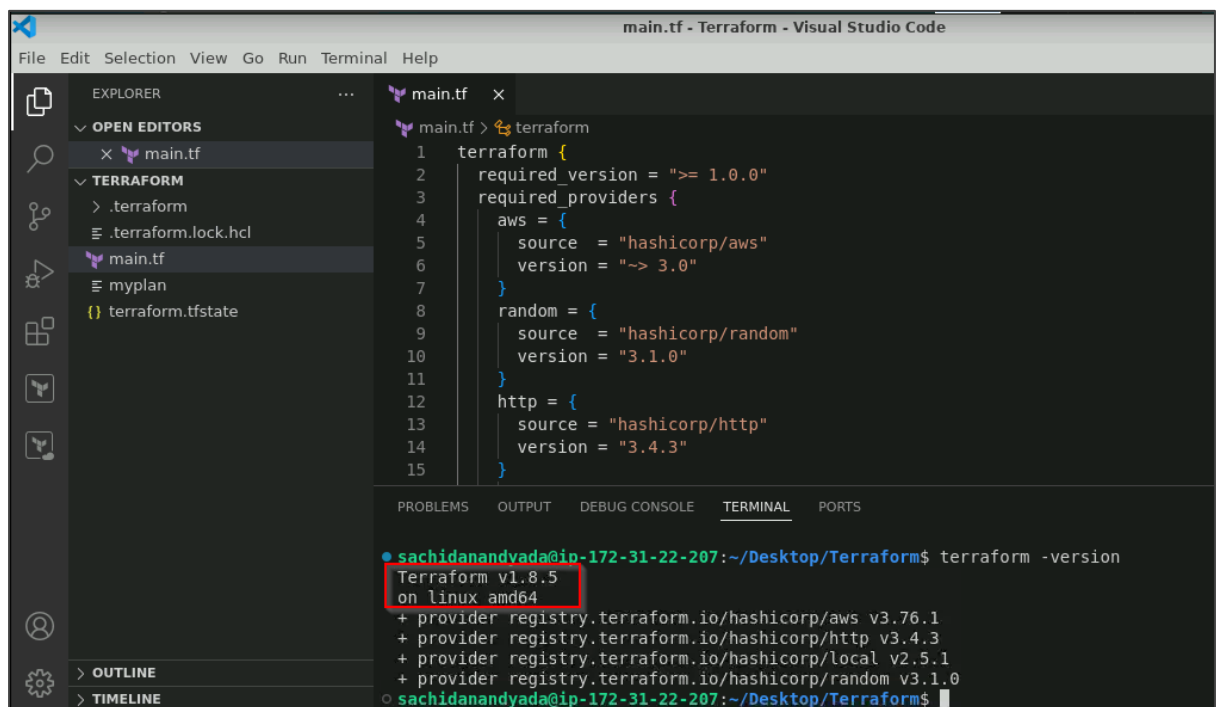


The screenshot shows the Visual Studio Code interface with a Terraform configuration file open. The file content is as follows:

```
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 = "3.1.0"
11    }
12    http = {
13      source = "hashicorp/http"
14      version = "3.4.3"
15    }
16  }
17 }
```

The terminal at the bottom shows the command `terraform -version` being entered, which is highlighted with a red box.

The latest installed version of the Terraform is available.



The screenshot shows the same Visual Studio Code interface, but now the terminal displays the output of the `terraform -version` command. The output is as follows:

```
• sachidanandyada@ip-172-31-22-207:~/Desktop/Terraform$ terraform -version
Terraform v1.8.5
on linux amd64
+ provider registry.terraform.io/hashicorp/aws v3.76.1
+ provider registry.terraform.io/hashicorp/http v3.4.3
+ provider registry.terraform.io/hashicorp/local v2.5.1
+ provider registry.terraform.io/hashicorp/random v3.1.0
sachidanandyada@ip-172-31-22-207:~/Desktop/Terraform$
```

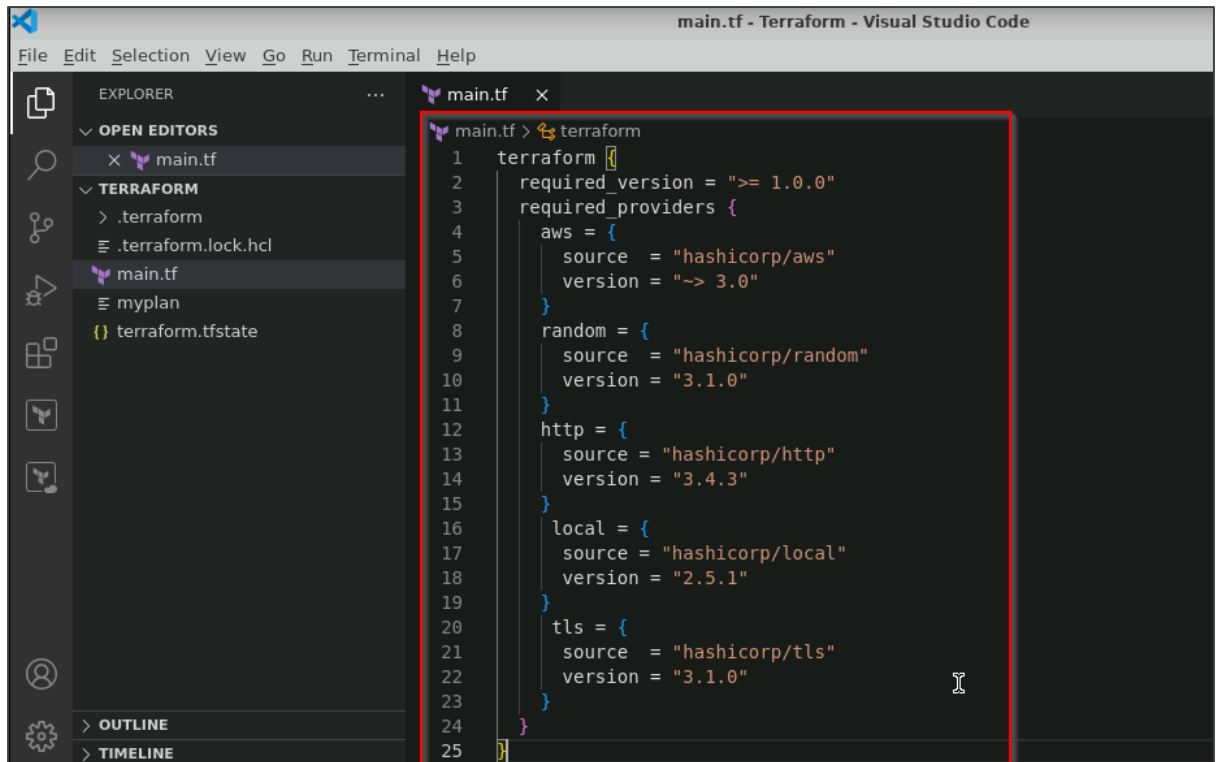
The first two lines of the output, `Terraform v1.8.5` and `on linux amd64`, are highlighted with a red box.

Step 2: Add the Terraform TLS provider configuration in the main.tf file

2.1 Open the **main.tf** file previously created, enter the configuration code given below, and save the file:

```
terraform {
  required_version = ">= 1.0.0"
  required_providers {
    aws = {
      source = "hashicorp/aws"
    }
    http = {
      source = "hashicorp/http"
      version = "2.1.0"
    }
    random = {
      source = "hashicorp/random"
      version = "3.1.0"
    }
    local = {
      source = "hashicorp/local"
      version = "2.1.0"
    }
    tls = {
      source = "hashicorp/tls"
      version = "3.1.0"
    }
  }
}
```

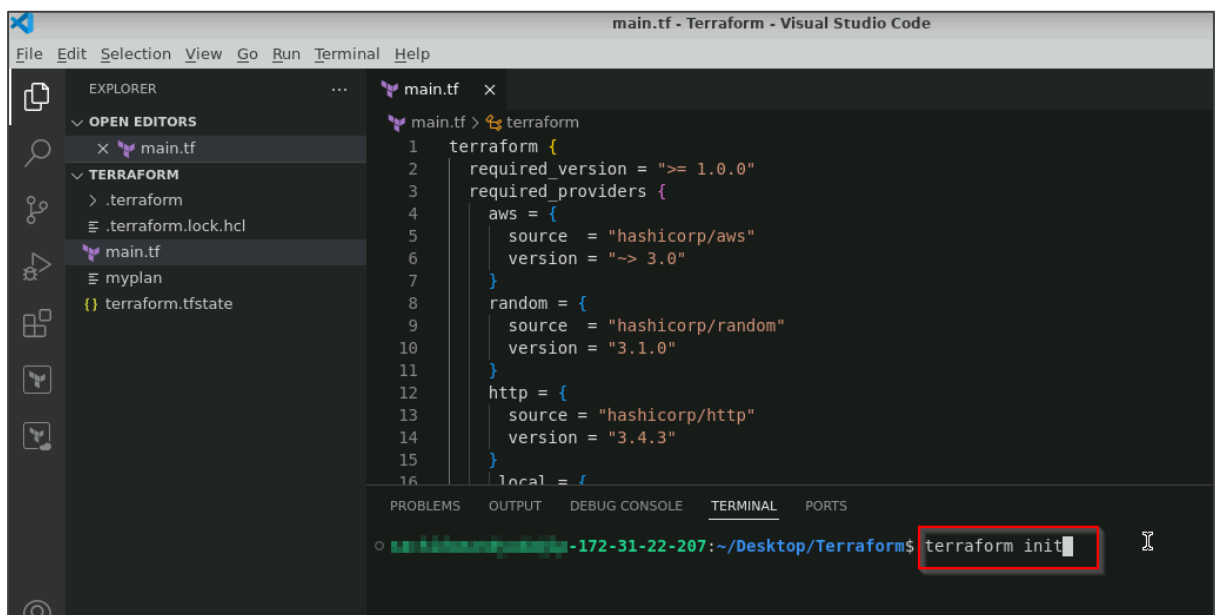
Note: You can get the configuration code of any Terraform provider from the official website of **HashiCorp Terraform** on the **Terraform Registry** page.



Step 3: Initialize the Terraform configuration file

- 3.1 Open the terminal and execute the command given below to initialize the configuration file:

terraform init



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

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

main.tf
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 = "3.1.0"
11    }
12    http = {
13      source = "hashicorp/http"
14      version = "3.4.3"
15    }
16    local = {
17      source = "hashicorp/local"
18      version = "2.5.1"
19    }
20  }
21 }
```

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

Initializing the backend...

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

Terraform has been successfully initialized!
```

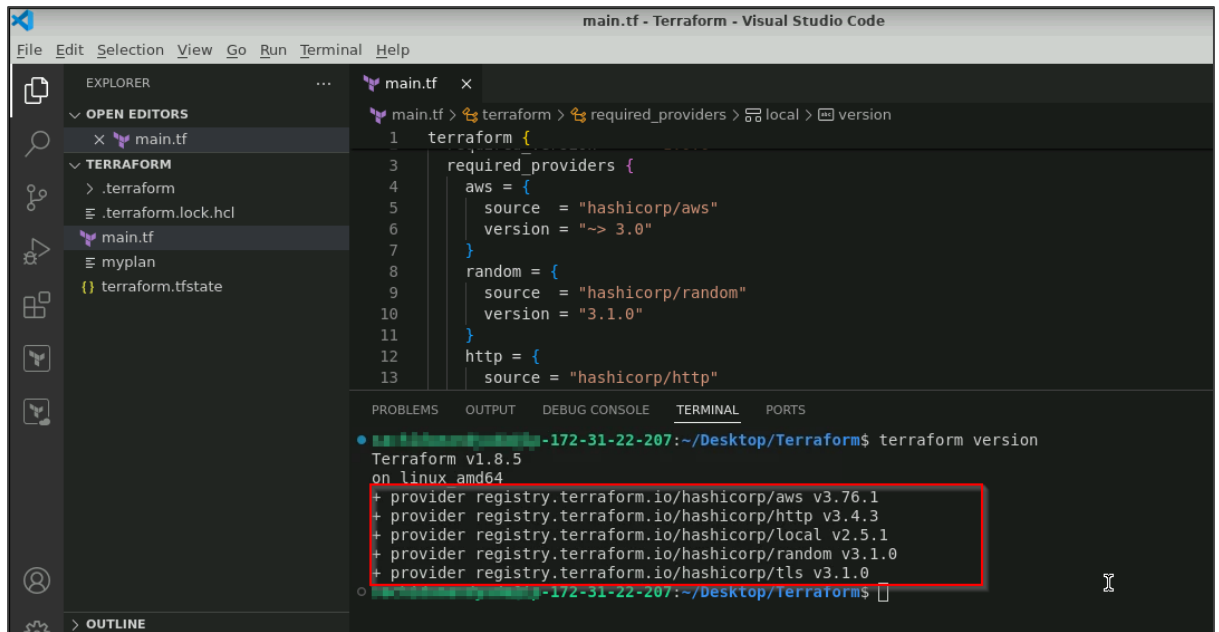
3.2 Open the terminal and enter the given command to check the installed provider:
terraform version

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

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

main.tf
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 = "3.1.0"
11    }
12    http = {
13      source = "hashicorp/http"
14      version = "3.4.3"
15    }
16    local = {
17      source = "hashicorp/local"
18      version = "2.5.1"
19    }
20  }
21 }
```

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



The **tls** provider is successfully installed.

Step 4: Update the main.tf file with a TLS certificate configuration block

4.1 Open the **main.tf** file and add the configuration block code given below to generate a TLS self-signed certificate, and save the file:

```

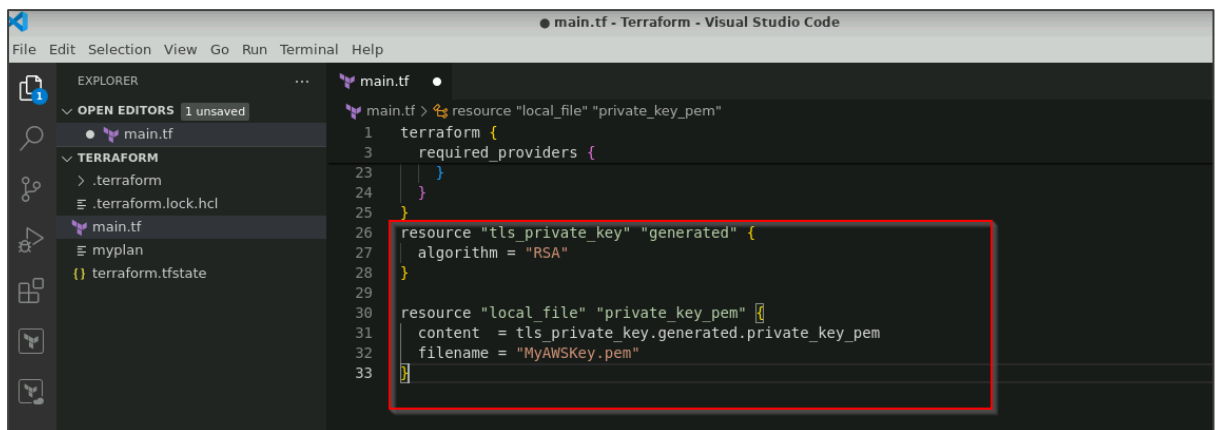
resource "tls_private_key" "generated" {
  algorithm = "RSA"
}

```

```

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

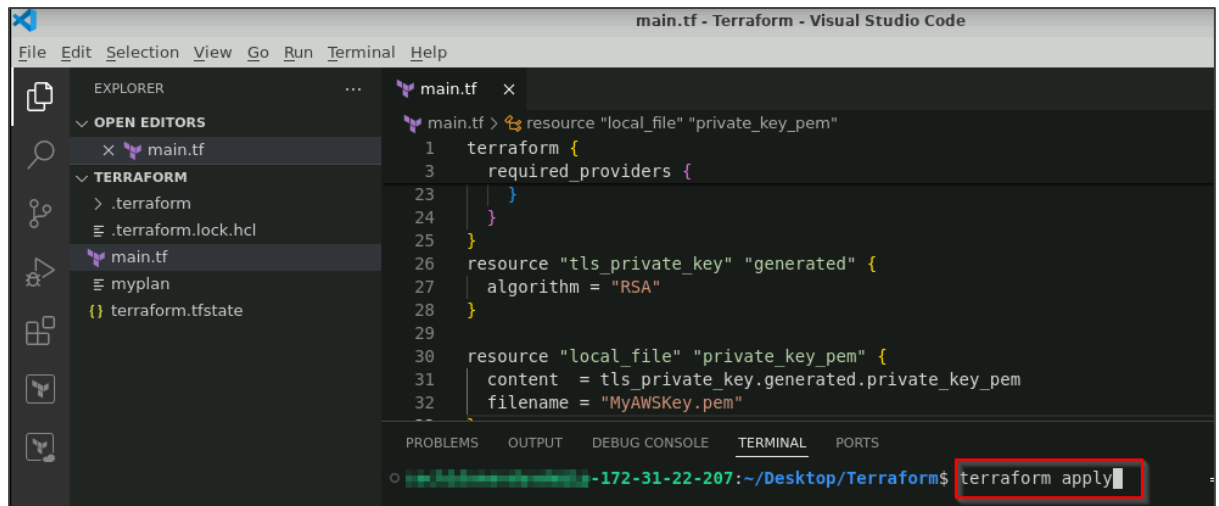
```



Step 5: Execute the apply command and validate the SSH key

5.1 Open the terminal and execute the given command to apply the TLS certificate configurations:

terraform apply

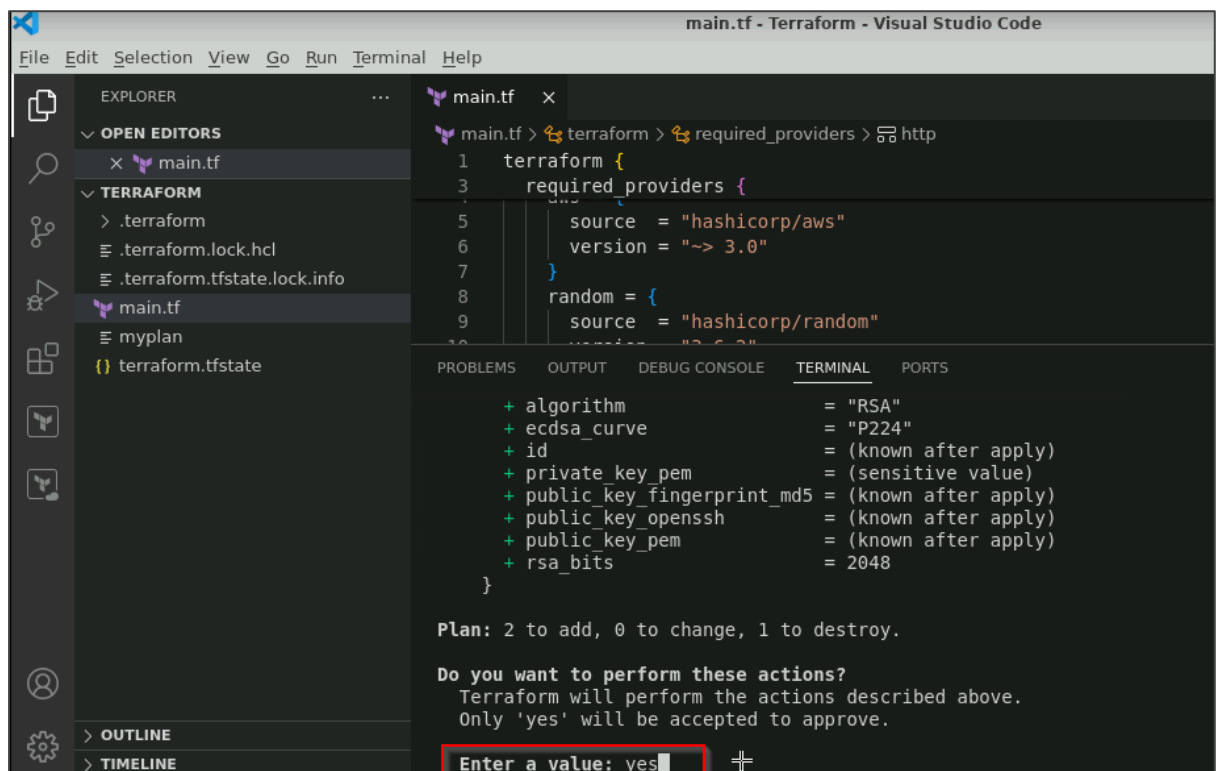


The screenshot shows the Visual Studio Code interface with the Terraform configuration file `main.tf` open in the editor. The file contains the following HCL code:

```
1 terraform {
2   required_providers {
3     http
4   }
5 }
6
7 resource "tls_private_key" "generated" {
8   algorithm = "RSA"
9 }
10
11 resource "local_file" "private_key_pem" {
12   content = tls_private_key.generated.private_key_pem
13   filename = "MyAWSKey.pem"
14 }
```

The terminal at the bottom shows the command `terraform apply` being executed in a shell session. The prompt is `-172-31-22-207:~/Desktop/Terraform$`.

5.2 Enter **yes** to approve the run



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

```
1 terraform {
2   required_providers {
3     http
4   }
5 }
6
7 resource "tls_private_key" "generated" {
8   algorithm = "RSA"
9 }
10
11 resource "local_file" "private_key_pem" {
12   content = tls_private_key.generated.private_key_pem
13   filename = "MyAWSKey.pem"
14 }
```

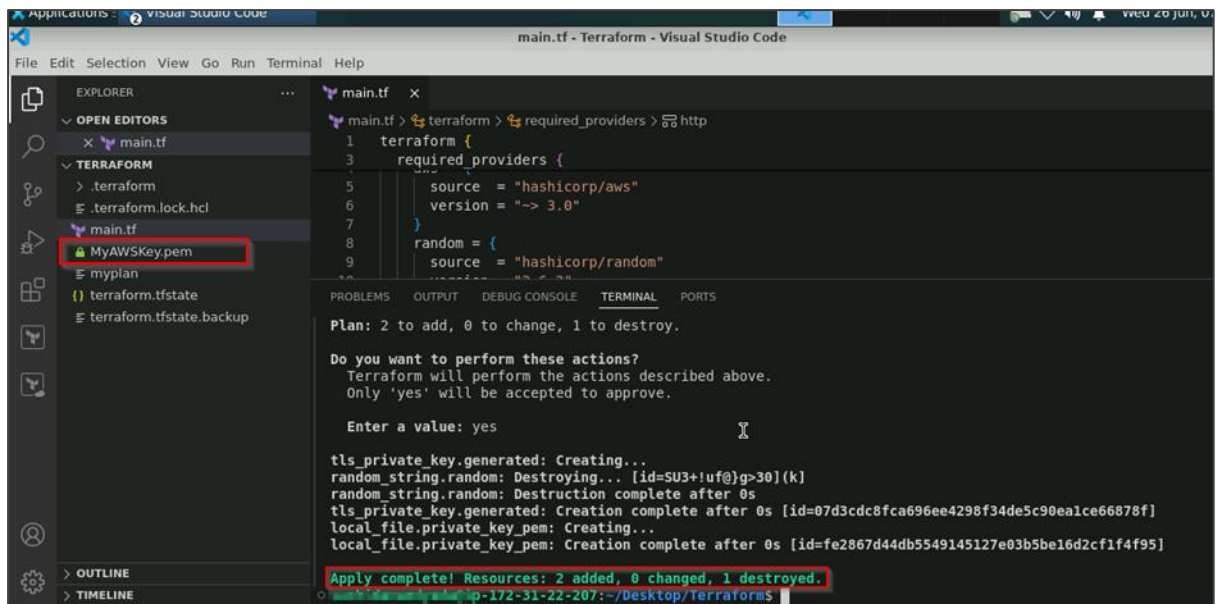
The terminal at the bottom shows the output of the `terraform apply` command. It displays the plan for the resources and the prompt to approve the actions:

```
+ algorithm          = "RSA"
+ ecdsa_curve        = "P224"
+ id                 = (known after apply)
+ private_key_pem    = (sensitive value)
+ public_key_fingerprint_md5 = (known after apply)
+ public_key_openssh = (known after apply)
+ public_key_pem     = (known after apply)
+ rsa_bits           = 2048
}

Plan: 2 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
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "~> 3.0"
6     }
7   }
8   random = {
9     source = "hashicorp/random"
10  }
11 }
```

```
Plan: 2 to add, 0 to change, 1 to destroy.

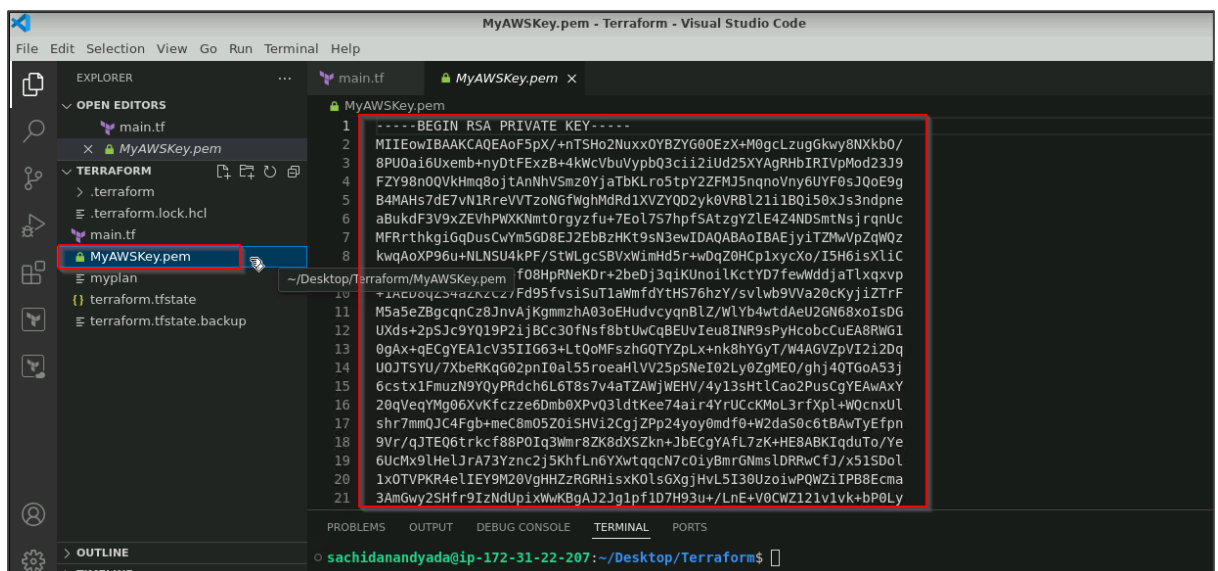
Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

tls_private_key.generated: Creating...
random_string.random: Destroying... [id=SU3+!uf@g>30](k)
random_string.random: Destruction complete after 0s
tls_private_key.generated: Creation complete after 0s [id=07d3cdc8fca696ee4298f34de5c90ealce66878f]
local_file.private_key.pem: Creating...
local_file.private_key.pem: Creation complete after 0s [id=fe2867d44db5549145127e03b5be16d2cf1f4f95]

Apply complete! Resources: 2 added, 0 changed, 1 destroyed.
```

5.3 Navigate to the **MyAWSKey.pem** to check the created SSH key



```
MyAWSKey.pem
1 -----BEGIN RSA PRIVATE KEY-----
2 MIIIEowIBAAKCAQEAoF5pX/+nTSho2Nuxx0YBZYG00EzX+M0gcLzugGkwy8NXkb0/
3 8PU0ai6Uxemb+nyDtFEzB+4kVcVbuVypbQ3cii2iUd25XYAgRHbIRIVpMod23J9
4 FZY98n00VkhHmq8ojtAnNhVSmz0YjaTbKLro5tpY2ZFMJ5nqnoVny6UYF0sJ0oE9g
5 B4MAHs7dE7W1RreVVTzoNGfWghMdRd1XVZYQD2yk0VRB12ii1B0i50xJs3ndpne
6 aBukdF3V9xZEvhPWxKNmt0rgyzfu+7Eol7S/hpfSATzgYzLE4Z4NDsmtNs jrqnUc
7 MFRrthkg1GqDusCwYm5G08EJ2EbBzHkt9sN3ewIDAQABAEjy1T2MwVpZqWQz
8 kwqAoxP96u+NLNSU4kPF/StWlgcSBVxWimHd5r+wDqZ0HCp1xycXo/I5H6isXliC
9 f08HprNeKDr+2beDj3qIKUnoilKctYD7fewWddjaTLxqxp
10 f08HprNeKDr+2beDj3qIKUnoilKctYD7fewWddjaTLxqxp
11 -----END RSA PRIVATE KEY-----
```

The SSH key with Terraform TLS provider is created in this file.

By following these steps, you have successfully generated the SSH key with the Terraform TLS provider using the latest version and configurations for secure access to your cloud infrastructure.