

## Lesson 09 Demo 07

### Generating Workspaces in Terraform

**Objective:** To effectively utilize Terraform workspaces for environment separation, create, and manage a development workspace, deploy infrastructure within it, and switch between different workspaces.

**Tools required:** VS Code and Linux terminal

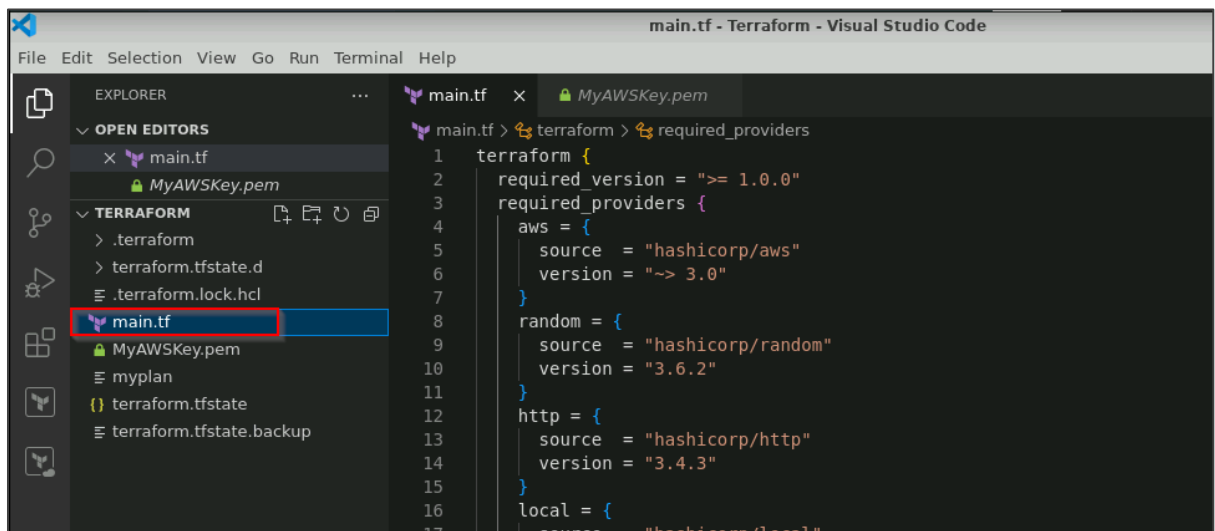
**Prerequisites:** Refer to demo 06

Steps to be followed:

1. Update the main.tf file
2. Check the Terraform workspaces
3. Generate a new Terraform workspace for the development state
4. Modify main.tf file
5. Navigate between workspaces

#### Step 1: Update the main.tf file

##### 1.1 Open the main.tf file



- 1.2 Replace the **main.tf** contents with the code provided below for setting up and configuring your Terraform project to manage infrastructure resources on AWS:

```
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" {  
  region = "us-east-1"  
  default_tags {  
    tags = {  
      Owner    = "Acme"  
      Provisioned = "Terraform"  
    }  
  }  
}
```

```

# access_key = "AKIARJTG7GGYJOVDU3B"
# secret_key = "kq2IAmP5ajai+VEhdHMcic4fXmUMcpQM3avt1wD"
}

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      = 15
  special     = true
  min_numeric = 6
  min_special = 2
  min_upper   = 3
}

```

The screenshot shows a code editor with two tabs: 'main.tf' and 'MyAWSKey.pem'. The 'main.tf' tab is active, displaying a Terraform configuration for required providers. The configuration is as follows:

```

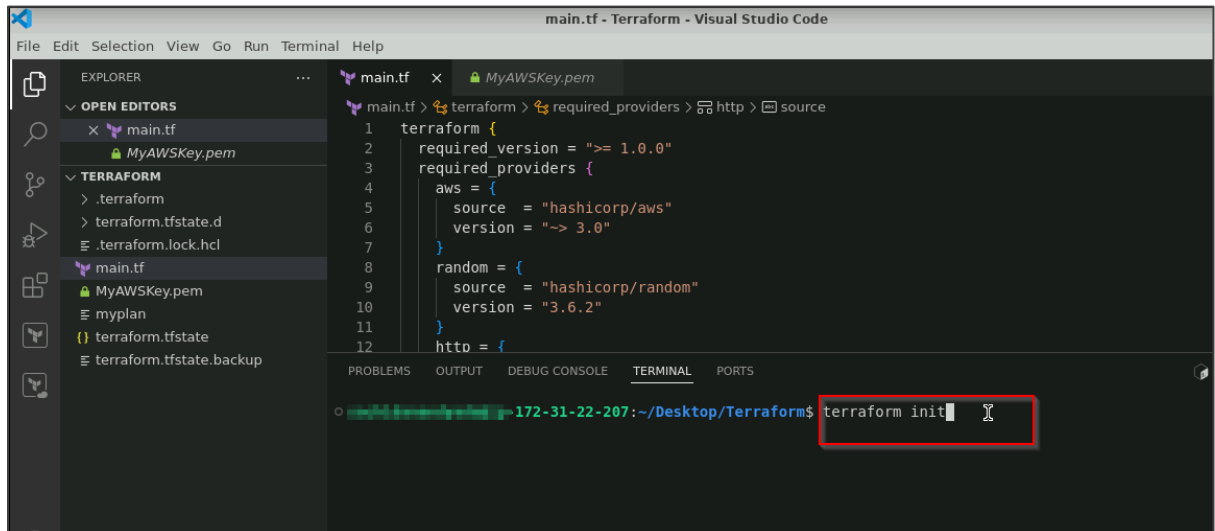
1 terraform {
2   required_version = ">= 1.0.0"
3   required_providers {
4     aws = {
5       source = "hashicorp/aws"
6       version = "~> 3.0"
7     }
8     random = {
9       source = "hashicorp/random"
10      version = "3.6.2"
11    }
12    http = {
13      source = "hashicorp/http"
14      version = "3.4.3"
15    }
16    local = {
17      source = "hashicorp/local"
18      version = "2.5.1"
19    }
20    tls = {
21      source = "hashicorp/tls"

```

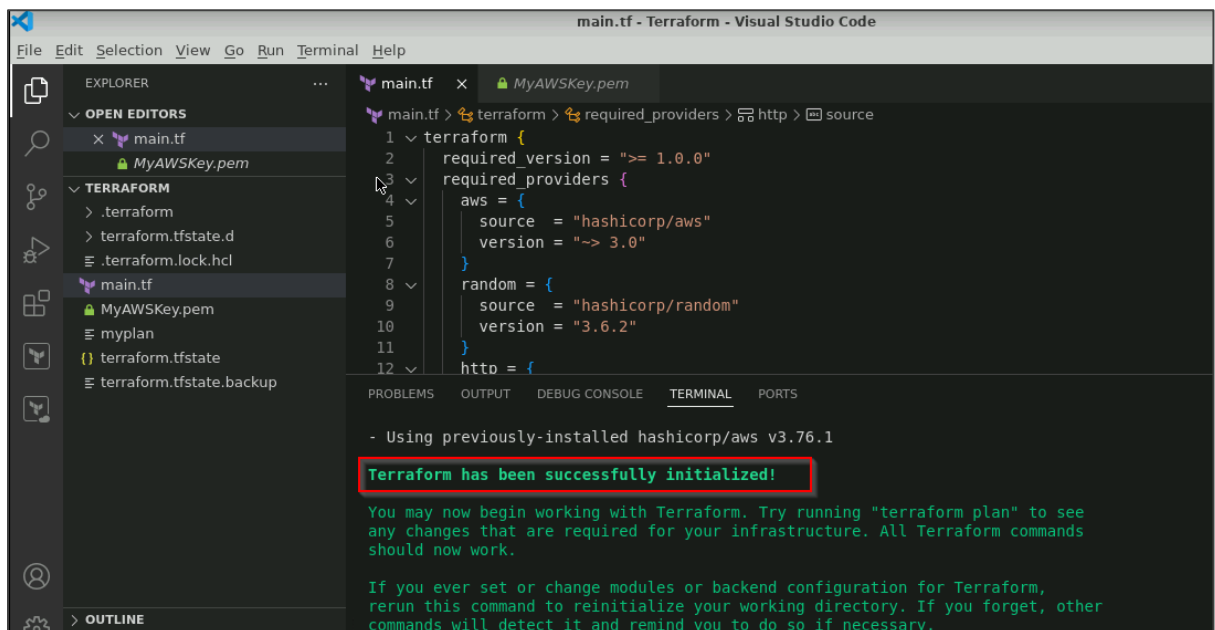
The code is highlighted with a red border. The editor interface includes a top bar with icons for terraform, required\_providers, http, and source. The bottom bar shows tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS, with the TERMINAL tab currently selected.

1.3 Run the **init** command to initialize the Terraform file:

**terraform init**



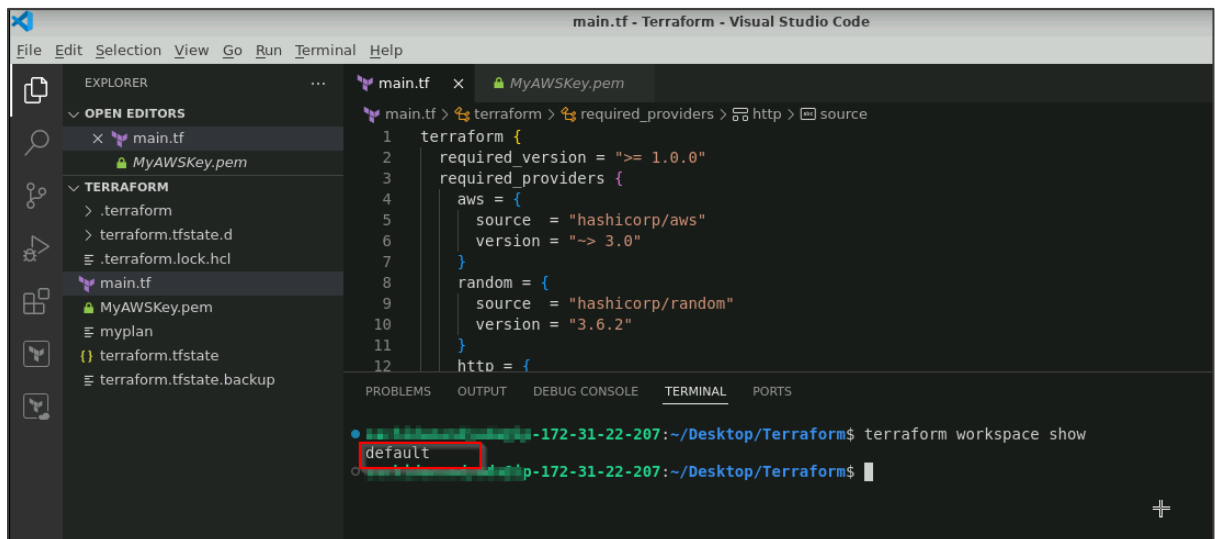
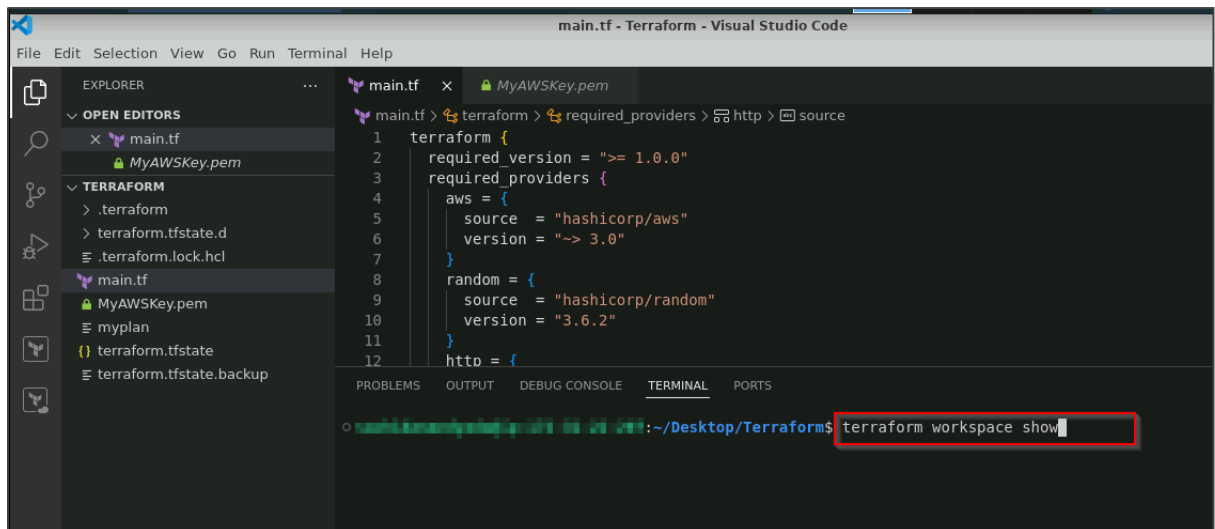
The terraform file is successfully initialized.



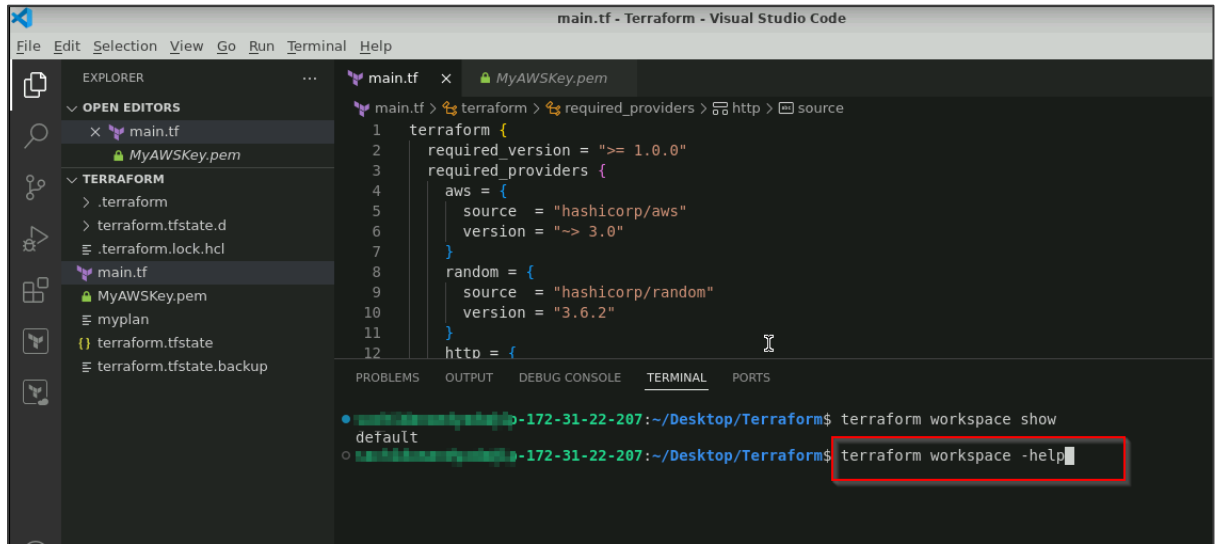
## Step 2: Check the Terraform workspaces

2.1 Open the terminal and run the command given below to check the current workspace:

**terraform workspace show**



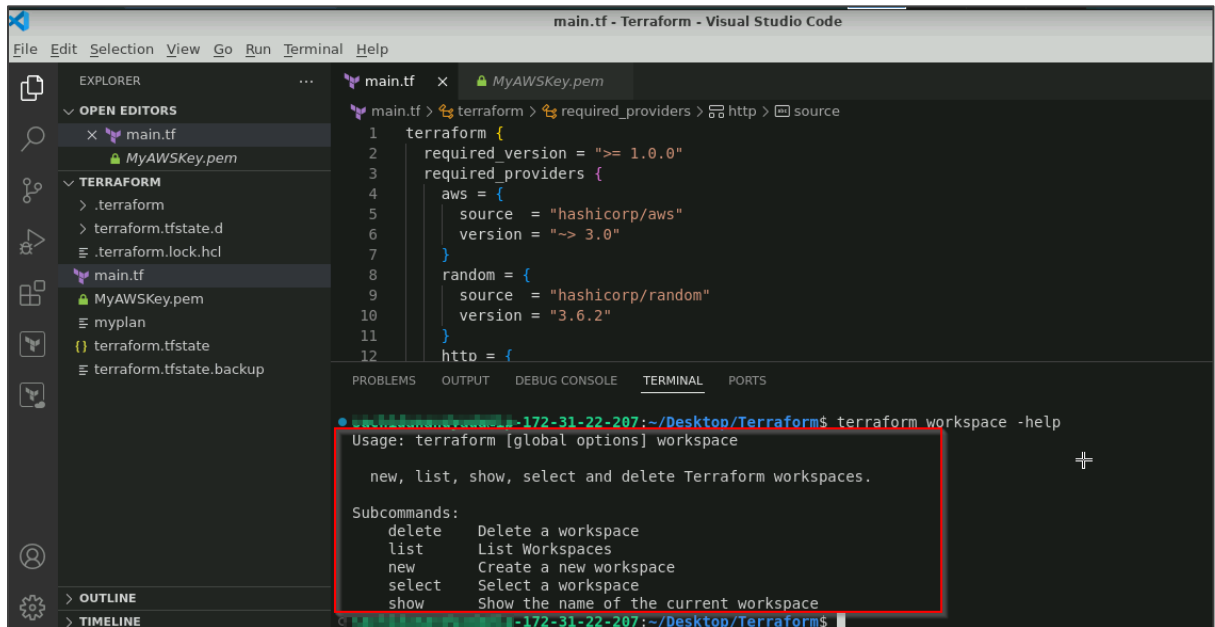
2.2 Run the command given below to display the subcommands available with the **terraform workspace** command:  
**terraform workspace -help**



The screenshot shows the Visual Studio Code interface with a Terraform configuration file open. The terminal window displays the output of the `terraform workspace show` command, which returns `default`. The command `terraform workspace -help` is entered in the terminal and is highlighted with a red box.

```
main.tf - Terraform - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
  OPEN EDITORS
    main.tf
    MyAWSKey.pem
  TERRAFORM
    .terraform
    terraform.tfstate.d
    terraform.lock.hcl
    main.tf
    MyAWSKey.pem
    myplan
    terraform.tfstate
    terraform.tfstate.backup
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 = "3.6.2"
11    }
12    http = {
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
-172-31-22-207:~/Desktop/Terraform$ terraform workspace show
default
-172-31-22-207:~/Desktop/Terraform$ terraform workspace -help
```

The **help** command displays the available subcommands.



The screenshot shows the Visual Studio Code interface with the same Terraform configuration file. The terminal window displays the output of the `terraform workspace -help` command, which shows the usage and subcommands for the `terraform workspace` command. The output is highlighted with a red box.

```
main.tf - Terraform - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
  OPEN EDITORS
    main.tf
    MyAWSKey.pem
  TERRAFORM
    .terraform
    terraform.tfstate.d
    terraform.lock.hcl
    main.tf
    MyAWSKey.pem
    myplan
    terraform.tfstate
    terraform.tfstate.backup
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 = "3.6.2"
11    }
12    http = {
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
-172-31-22-207:~/Desktop/Terraform$ terraform workspace -help
Usage: terraform [global options] workspace

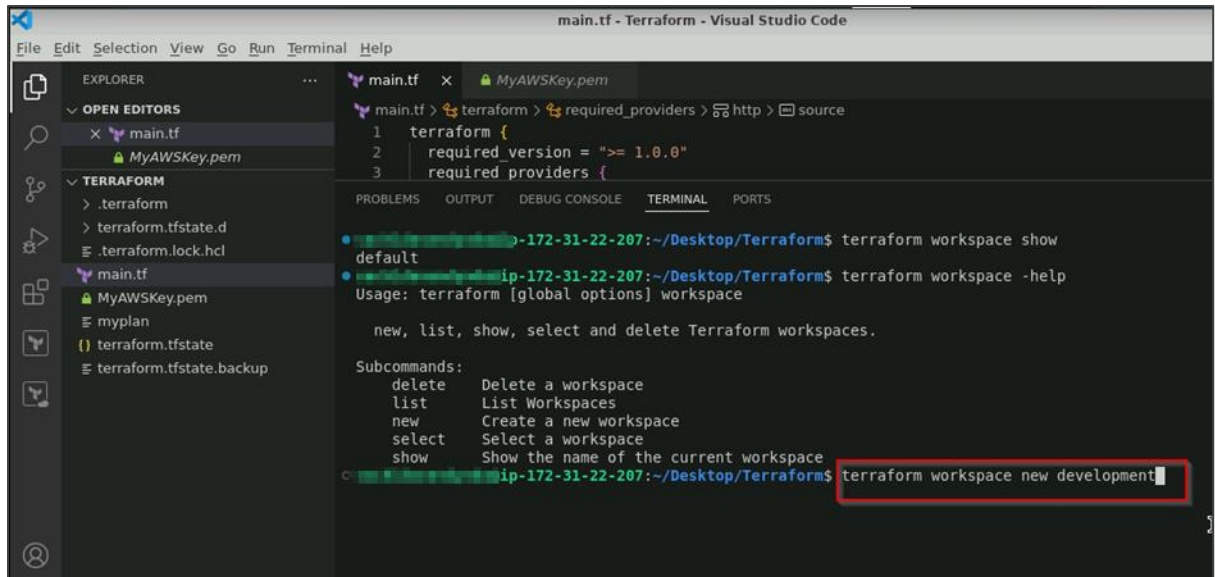
new, list, show, select and delete Terraform workspaces.

Subcommands:
delete  Delete a workspace
list    List Workspaces
new     Create a new workspace
select  Select a workspace
show    Show the name of the current workspace
-172-31-22-207:~/Desktop/Terraform$
```

### Step 3: Generate a new Terraform workspace for the development state

3.1 Open the terminal and run the command given below to generate a new Terraform workspace

**terraform workspace new development**



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

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

main.tf
1 terraform {
2   required_providers {
3     http {
4       source = "hashicorp/http"
5       version = "~>= 1.0.0"
6     }
7   }
8 }

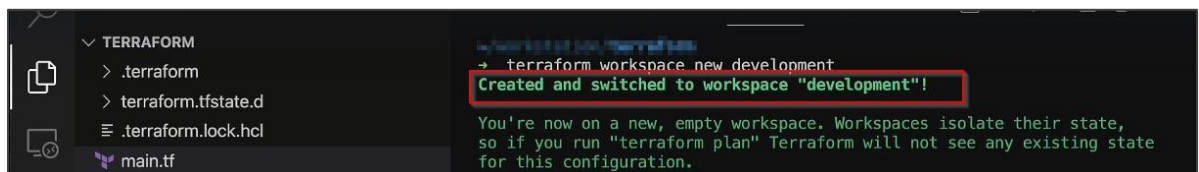
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

ip-172-31-22-207:~/Desktop/Terraform$ terraform workspace show
default
ip-172-31-22-207:~/Desktop/Terraform$ terraform workspace -help
Usage: terraform [global options] workspace

    new, list, show, select and delete Terraform workspaces.

Subcommands:
  delete  Delete a workspace
  list    List Workspaces
  new     Create a new workspace
  select  Select a workspace
  show    Show the name of the current workspace

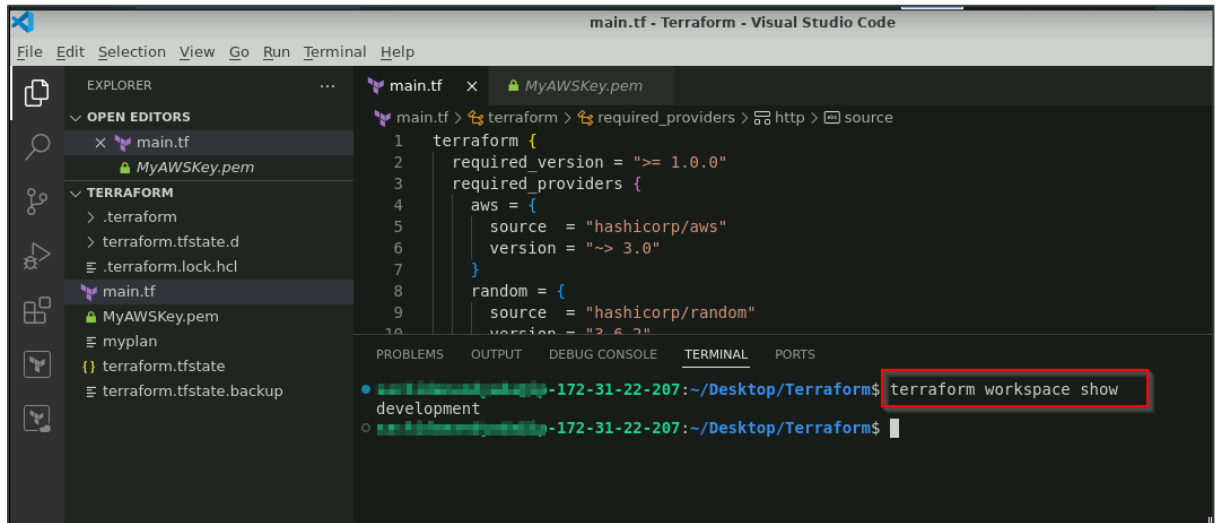
ip-172-31-22-207:~/Desktop/Terraform$ terraform workspace new development
```



```
ip-172-31-22-207:~/Desktop/Terraform$ terraform workspace new development
Created and switched to workspace "development"!

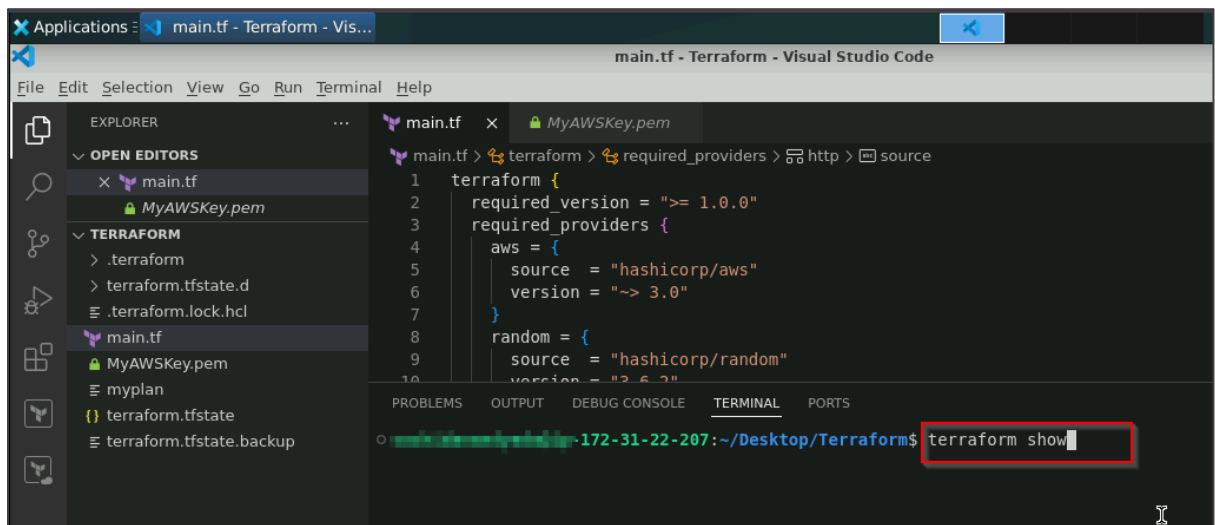
You're now on a new, empty workspace. Workspaces isolate their state,
so if you run "terraform plan" Terraform will not see any existing state
for this configuration.
```

3.2 Now run the command given below to see the current workspace:  
**terraform workspace show**

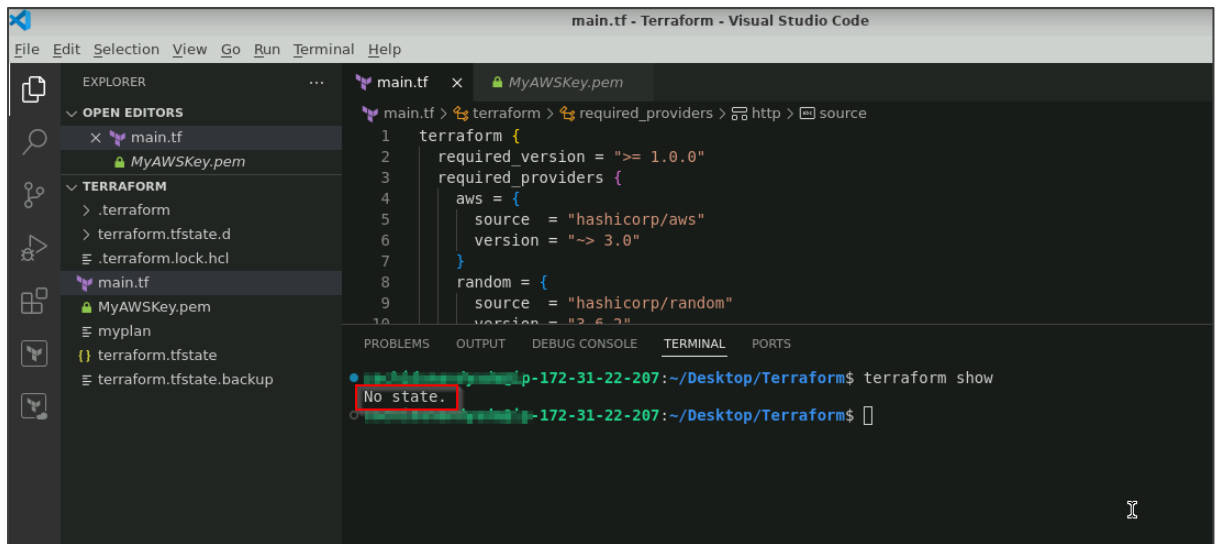


The newly generated workspace is available.

3.3 Now to check the state of the Terraform file run the command given below:  
**terraform show**



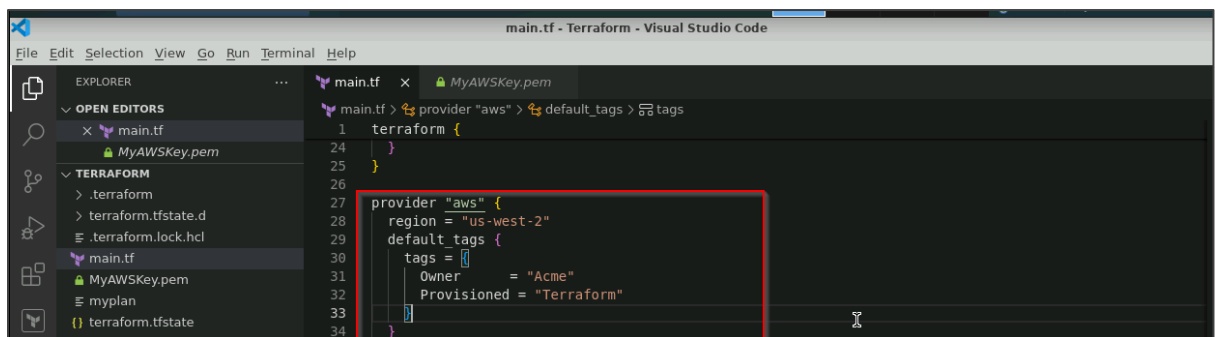




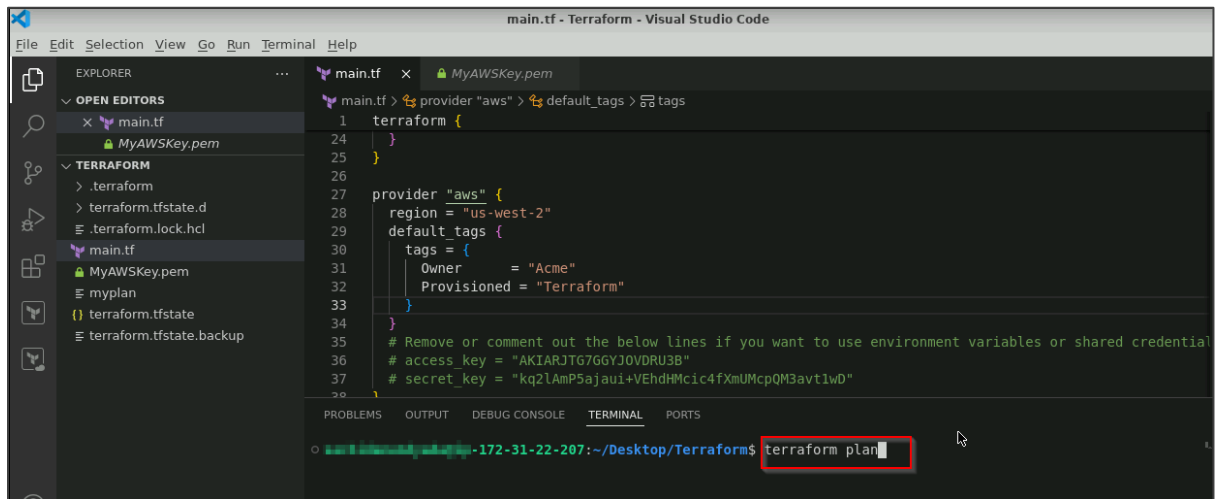
## Step 4: Modify main.tf file

- 4.1 Open the **main.tf** file to make the changes in the **provider block** as per the code given below and save the file:

```
provider "aws" {  
  region = "us-west-2"  
  default_tags {  
    tags = {  
      Owner      = "Acme"  
      Provisioned = "Terraform"  
    }  
  }  
}
```



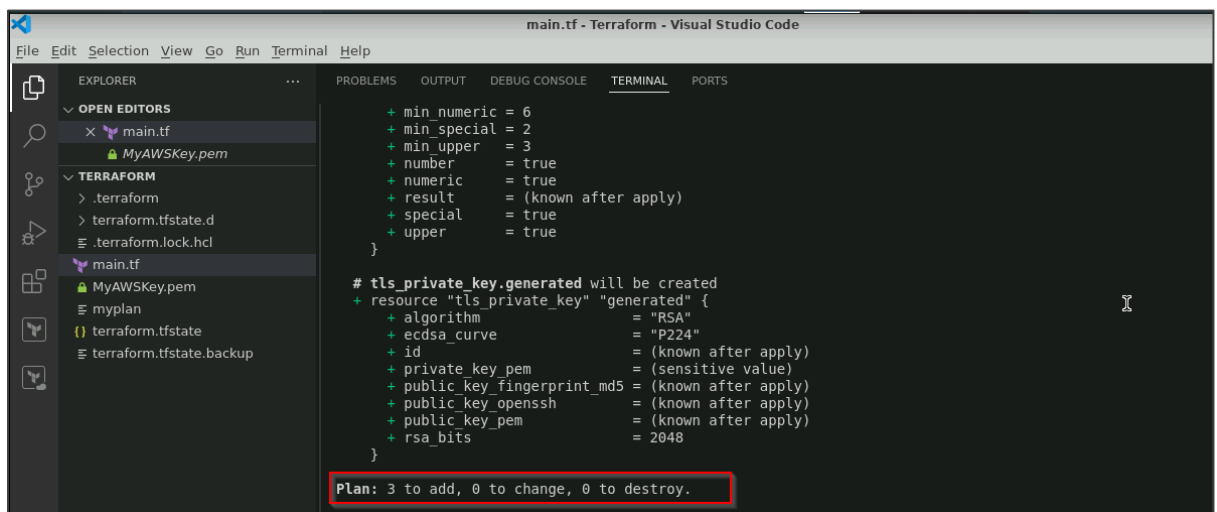
4.2 Run the **Terraform plan** command to see Terraforms run for execution as given below:  
**terraform plan**



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

```
1 terraform {
24 }
25 }
26
27 provider "aws" {
28   region = "us-west-2"
29   default_tags {
30     tags = {
31       Owner      = "Acme"
32       Provisioned = "Terraform"
33     }
34   }
35   # Remove or comment out the below lines if you want to use environment variables or shared credential
36   # access_key = "AKIARJTG7GGYJOVDRU3B"
37   # secret_key = "kq2lAmP5ajai+VEhdHMic4fXmUMcpQM3avt1wD"
38 }
```

The terminal at the bottom shows the command `terraform plan` being entered at the prompt `main.tf - Terraform - Visual Studio Code`.



The screenshot shows the output of the `terraform plan` command in the terminal. The output displays the planned changes for the `tls_private_key.generated` resource:

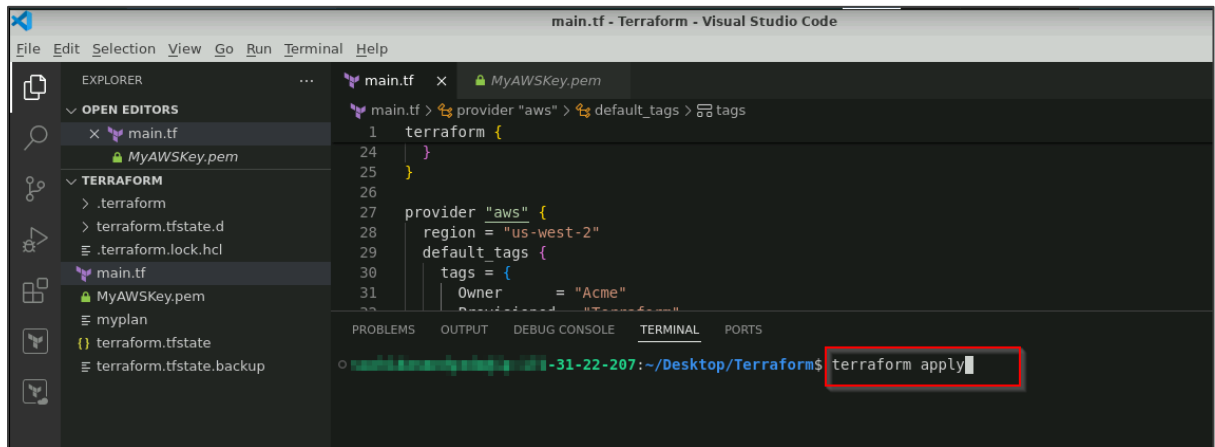
```
+ min_numeric = 6
+ min_special = 2
+ min_upper   = 3
+ number      = true
+ numeric     = true
+ result      = (known after apply)
+ special     = true
+ upper       = true
}

# tls_private_key.generated will be created
+ resource "tls_private_key" "generated" {
+   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
}
```

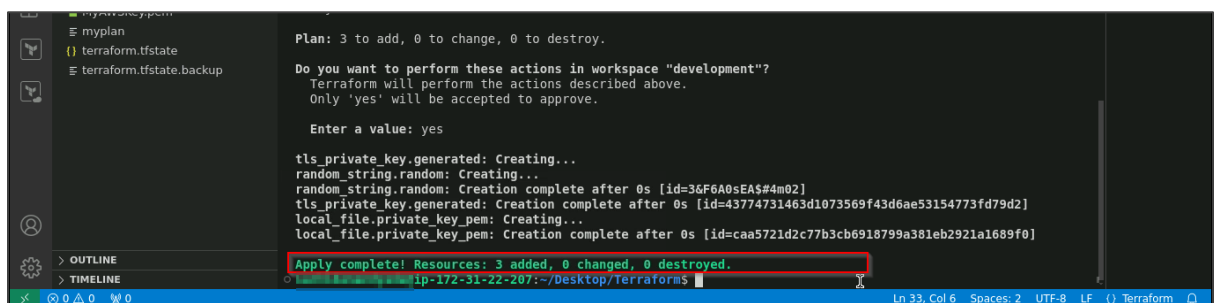
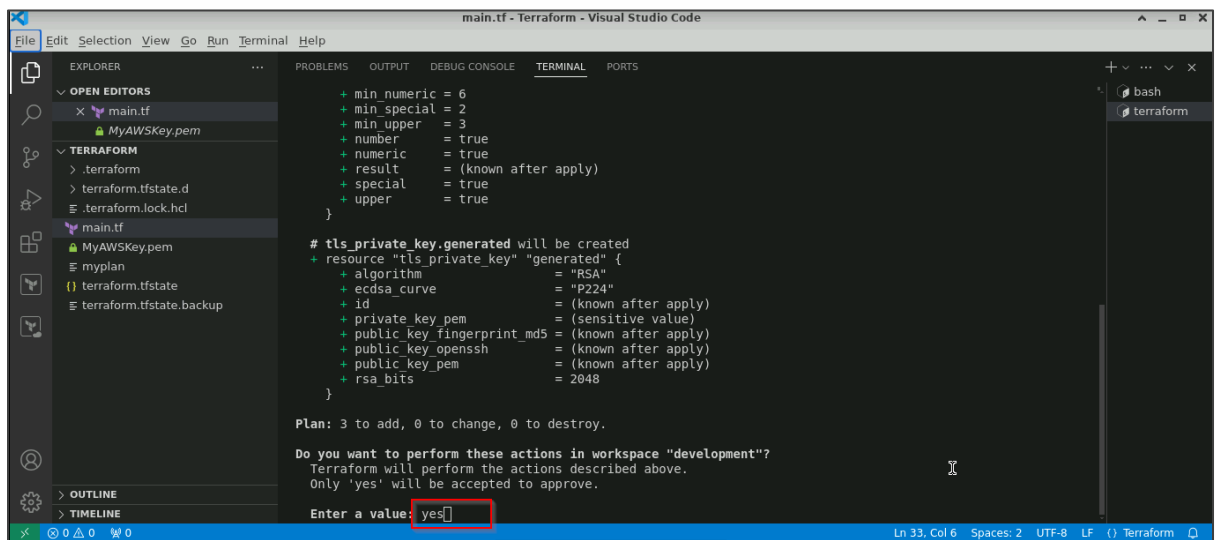
The terminal output concludes with the summary: `Plan: 3 to add, 0 to change, 0 to destroy.`

The **terraform plan** command runs successfully.

4.3 Open the terminal and run the command given below to apply the modifications:



Confirm as **yes** to proceed further.

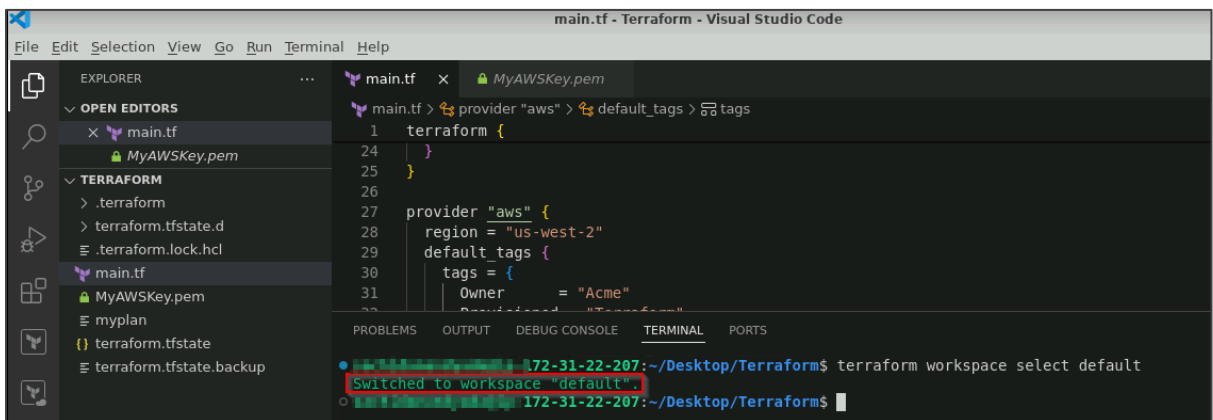
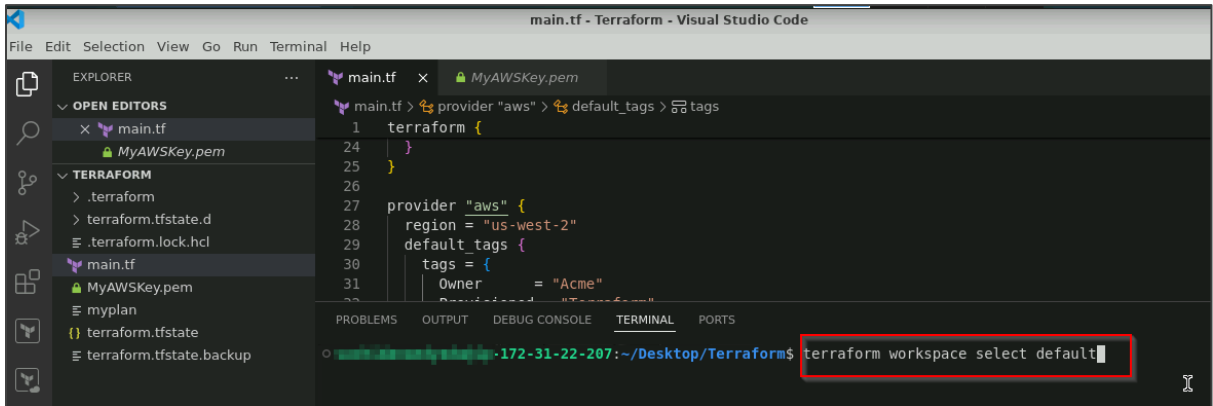


The apply command is completed successfully.

## Step 5: Navigate between workspaces

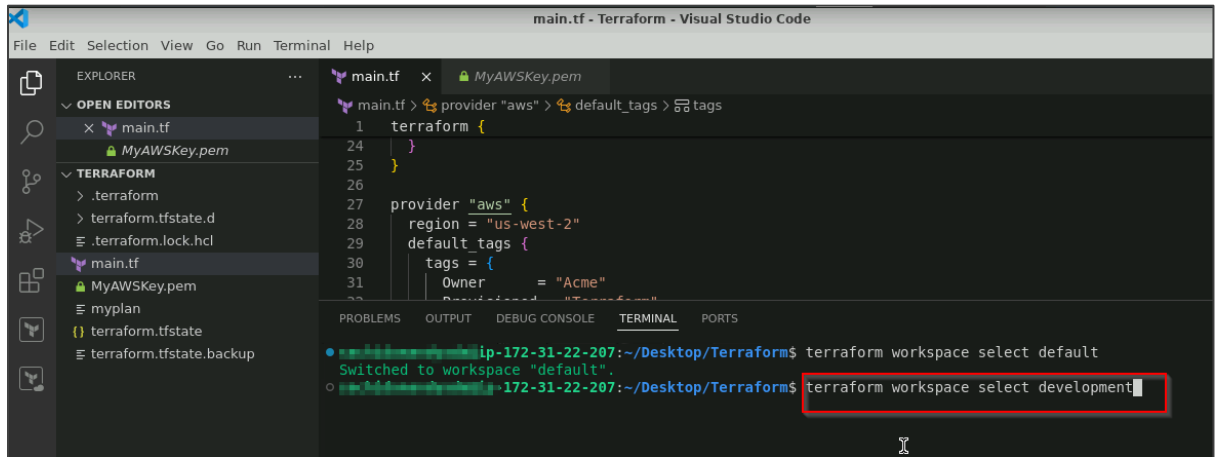
5.1 Run the following command to navigate back to the default workspace:

**terraform workspace select default**



5.2 Run the following command to navigate back to the development workspace:

### terraform workspace select development



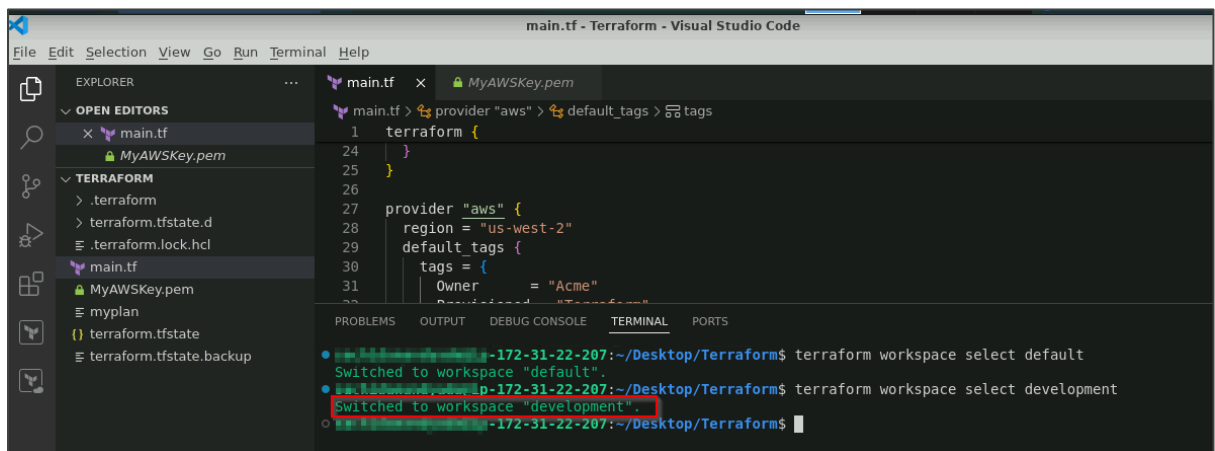
The screenshot shows the Visual Studio Code interface with the 'main.tf' file open in the editor. The file contains Terraform configuration for the AWS provider, including region 'us-west-2' and default tags for 'Owner' and 'Acme'. The Explorer sidebar on the left shows the project structure with files like 'main.tf', 'MyAWSKey.pem', and 'terraform.tfstate.d'. The Terminal panel at the bottom shows the command prompt with the command 'terraform workspace select development' entered and highlighted by a red box.

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

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

main.tf
1 terraform {
24   }
25 }
26
27 provider "aws" {
28   region = "us-west-2"
29   default_tags {
30     tags = {
31       Owner = "Acme"
32     }
33   }
34 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
ip-172-31-22-207:~/Desktop/Terraform$ terraform workspace select default
Switched to workspace "default".
ip-172-31-22-207:~/Desktop/Terraform$ terraform workspace select development
```



The screenshot shows the Visual Studio Code interface with the 'main.tf' file open in the editor. The file contains Terraform configuration for the AWS provider, including region 'us-west-2' and default tags for 'Owner' and 'Acme'. The Explorer sidebar on the left shows the project structure with files like 'main.tf', 'MyAWSKey.pem', and 'terraform.tfstate.d'. The Terminal panel at the bottom shows the command prompt with the command 'terraform workspace select development' entered and highlighted by a red box. The output of the command is 'Switched to workspace "development"', which is also highlighted by a red box.

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

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

main.tf
1 terraform {
24   }
25 }
26
27 provider "aws" {
28   region = "us-west-2"
29   default_tags {
30     tags = {
31       Owner = "Acme"
32     }
33   }
34 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
ip-172-31-22-207:~/Desktop/Terraform$ terraform workspace select default
Switched to workspace "default".
ip-172-31-22-207:~/Desktop/Terraform$ terraform workspace select development
Switched to workspace "development".
ip-172-31-22-207:~/Desktop/Terraform$
```

By following the above steps, you have successfully utilized Terraform workspaces for environment separation, created and managed a development workspace, deployed infrastructure within it, and seamlessly switched between different workspaces.