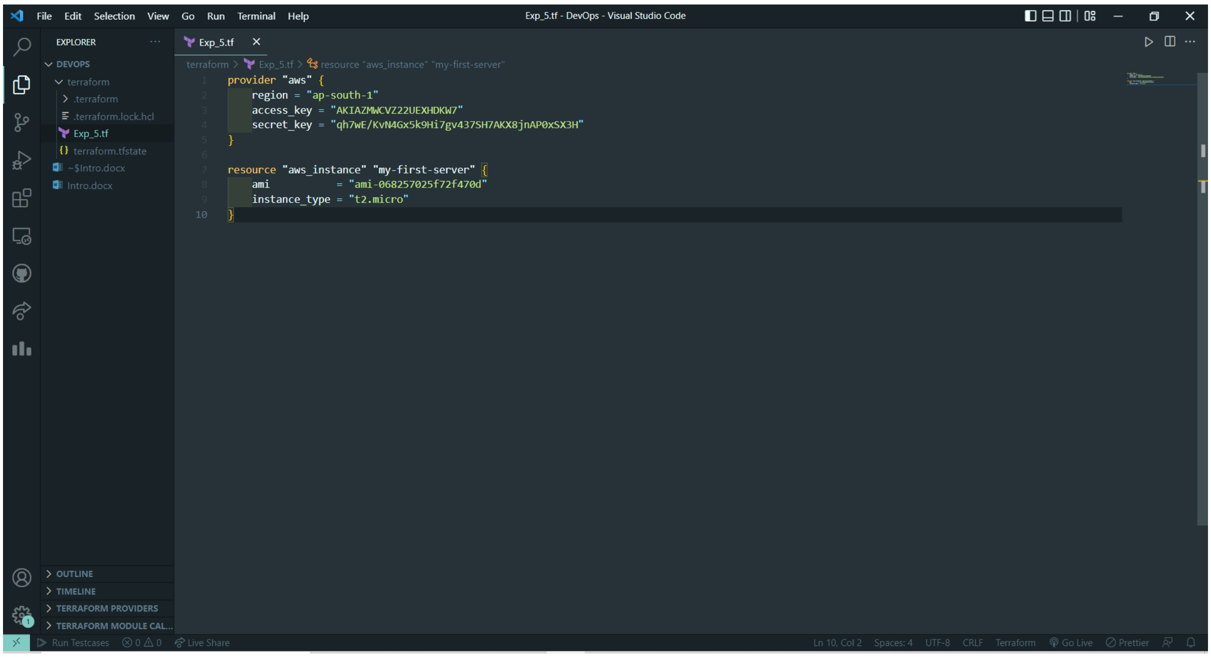
Experiment 5

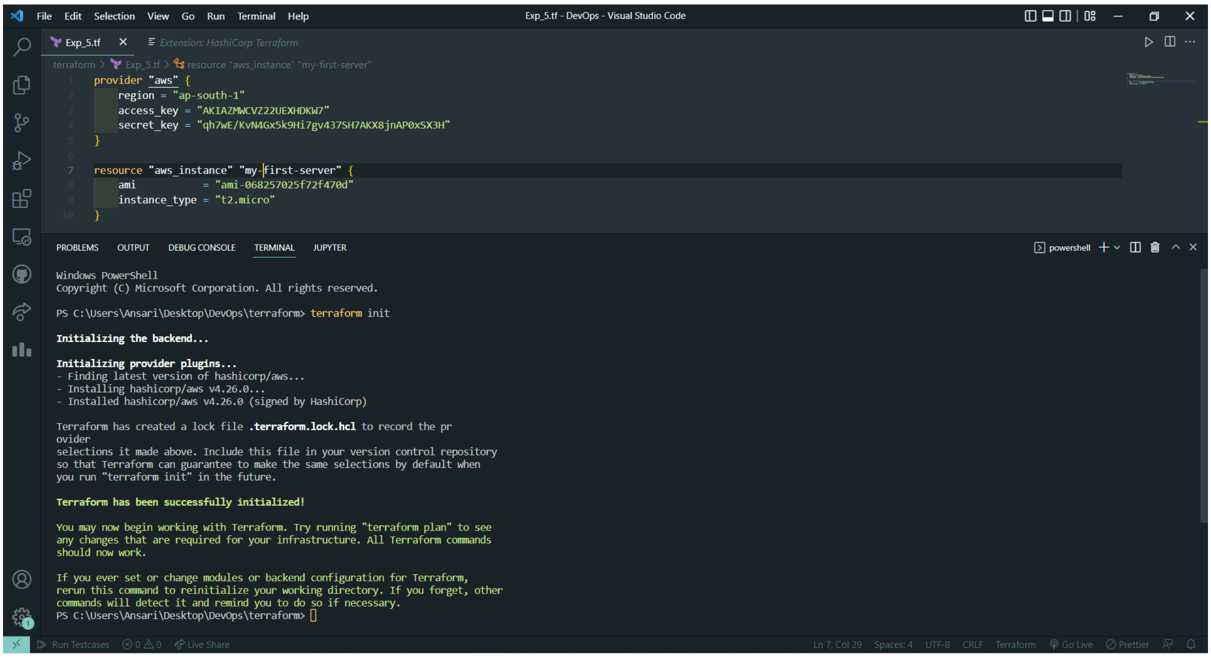
To understand terraform lifecycle, core concepts/terminologies and install it on a Linux Machine

- Download and Install Terraform
- After that create a folder and a file with extension .tf
- Mention the provider and resource in the .tf file as shown below
- Before writing the provider get the access key and secret key from aws
- Also you will have to get ami from aws



```
provider "aws" {
    region = "ap-south-1"
    access_key = "AKIAZMWCVZ22UEXHDKW7"
    secret_key = "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"
}

resource "aws_instance" "my-first-server" {
    ami           = "ami-068257025f72f470d"
    instance_type = "t2.micro"
}
```

- After writing the script open the terminal and initialise terraform using the command terraform init

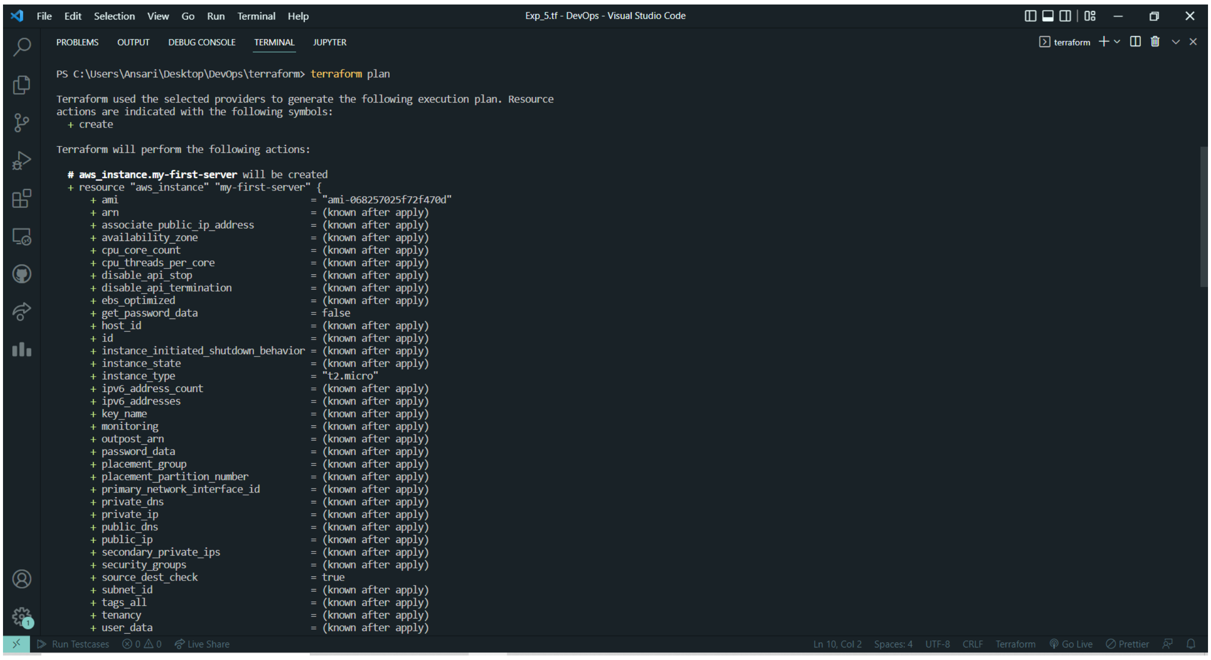


File Edit Selection View Go Run Terminal Help
Exp_5.tf - DevOps - Visual Studio Code
Exp_5.tf
Extension: HashiCorp Terraform
terraform > Exp_5.tf > resource "aws_instance" "my-first-server"
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
powershell
Run Testcases
Live Share
Ln 7, Col 29 Spaces: 4 UTF-8 CRLF Terraform Go Live Prettier

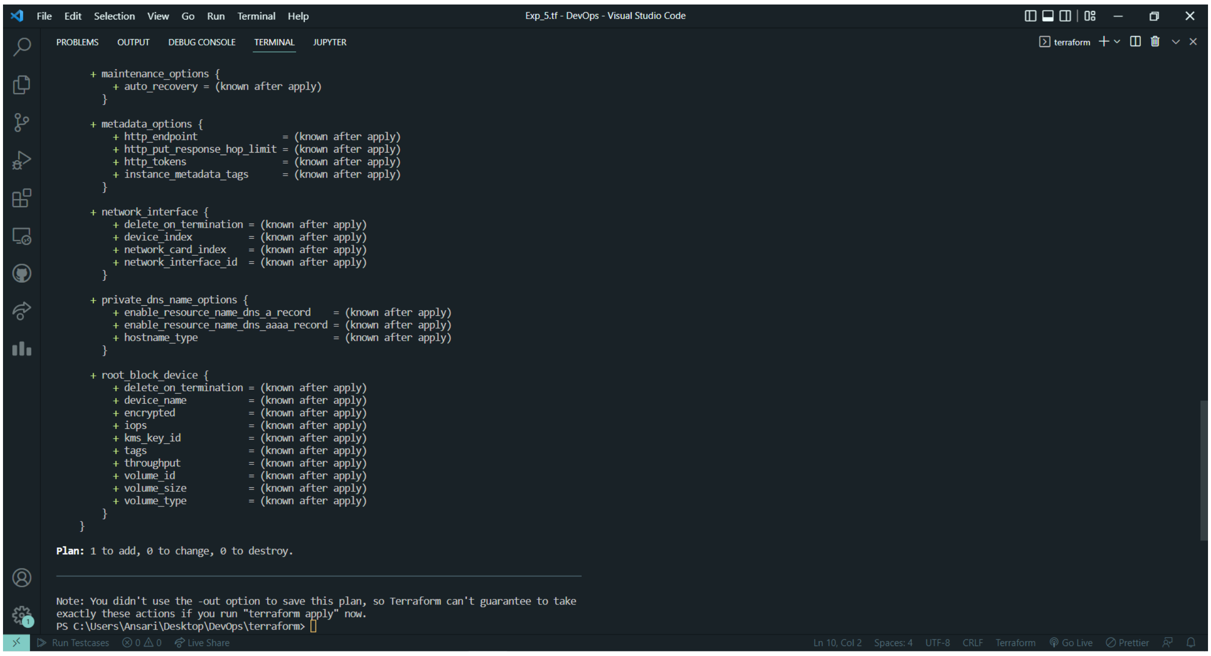
```
provider "aws" {
    region = "ap-south-1"
    access_key = "AKIAZMWCVZ22UEXHDKW7"
    secret_key = "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"
}

resource "aws_instance" "my-first-server" {
    ami           = "ami-068257025f72f470d"
    instance_type = "t2.micro"
}
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\Ansari\Desktop\DevOps\terraform> terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v4.26.0...
- Installed hashicorp/aws v4.26.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the pr
ovider
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.
PS C:\Users\Ansari\Desktop\DevOps\terraform> 
```

- After that write the terraform plan command to see what all instruction will be executed

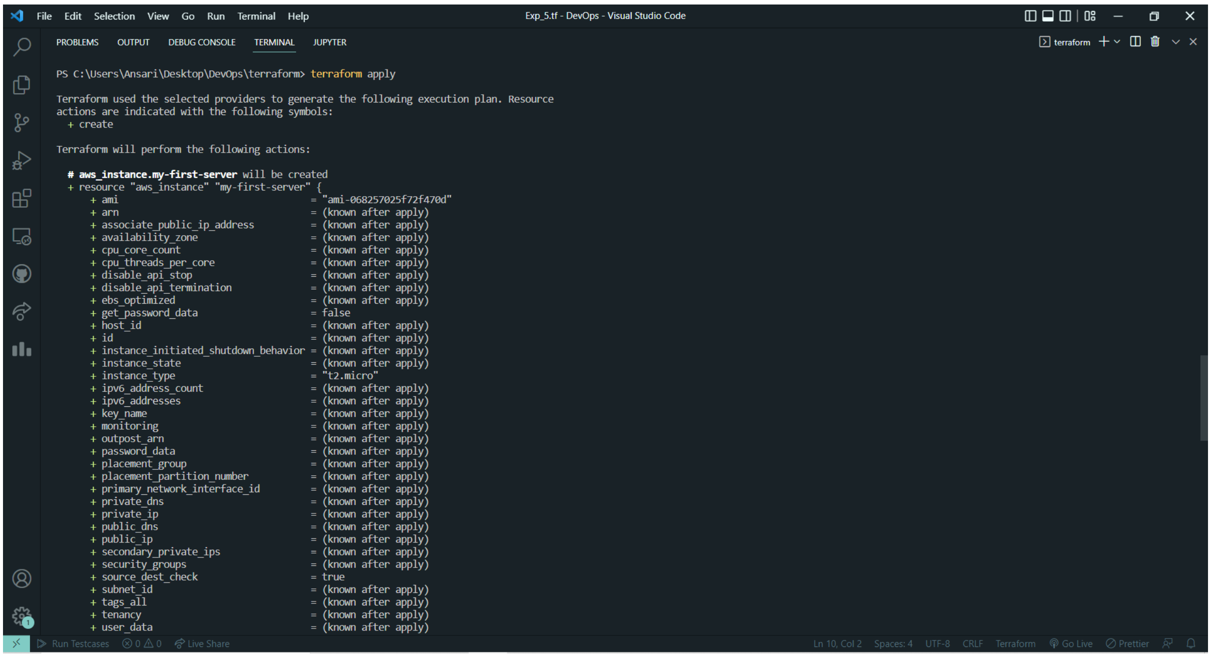

File Edit Selection View Go Run Terminal Help
Exp_5.tf - DevOps - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
terraform
Run Testcases
Live Share
Ln 10, Col 2 Spaces: 4 UTF-8 CRLF Terraform Go Live Prettier

```
PS C:\Users\Ansari\Desktop\DevOps\terraform> 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-first-server will be created
  + resource "aws_instance" "my-first-server" {
      + ami                                  = "ami-068257025f72f470d"
      + 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)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (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
      + subnet_id                            = (known after apply)
      + tags_all                             = (known after apply)
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
```

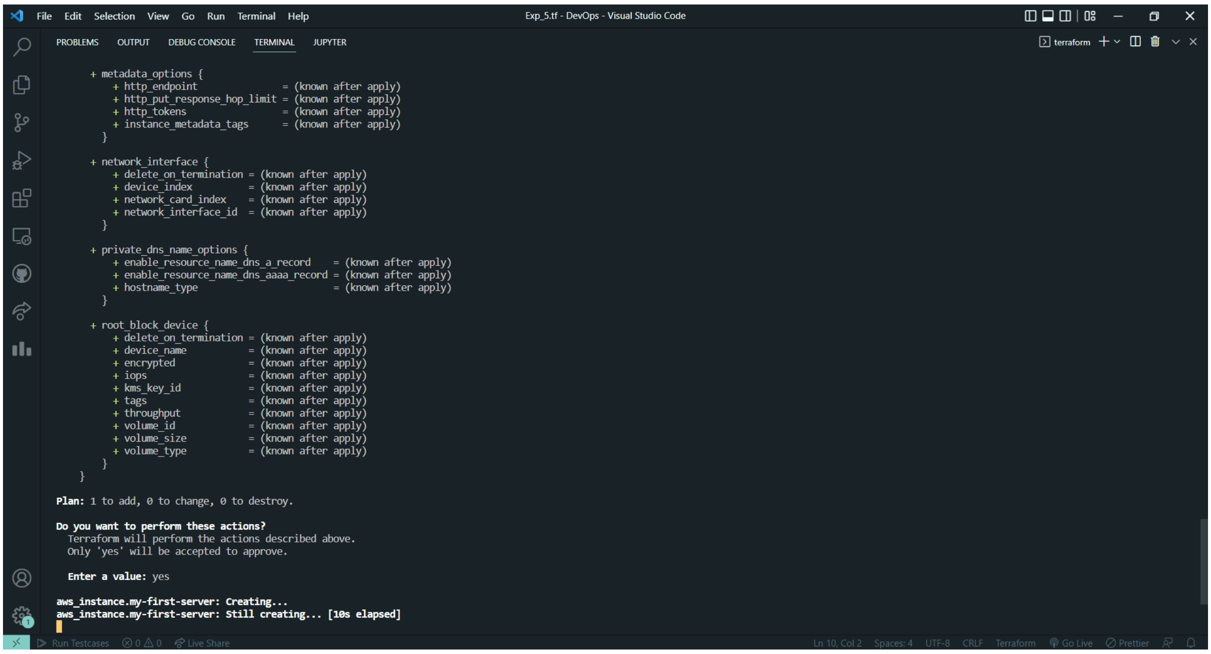

File Edit Selection View Go Run Terminal Help
Exp_5.tf - DevOps - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
terraform
Run Testcases
Live Share
Ln 10, Col 2 Spaces: 4 UTF-8 CRLF Terraform Go Live Prettier

```
      + maintenance_options {
          + auto_recovery = (known after apply)
        }

      + metadata_options {
          + http_endpoint               = (known after apply)
          + http_put_response_hop_limit = (known after apply)
          + http_tokens                 = (known after apply)
          + instance_metadata_tags      = (known after apply)
        }

      + network_interface {
          + delete_on_termination = (known after apply)
          + device_index          = (known after apply)
          + network_card_index    = (known after apply)
          + network_interface_id  = (known after apply)
        }

      + private_dns_name_options {
          + enable_resource_name_dns_a_record    = (known after apply)
          + enable_resource_name_dns_aaaa_record = (known after apply)
          + hostname_type                        = (known after apply)
        }

      + root_block_device {
          + delete_on_termination = (known after apply)
          + device_name           = (known after apply)
          + encrypted             = (known after apply)
          + iops                  = (known after apply)
          + kms_key_id            = (known after apply)
          + tags                  = (known after apply)
          + throughput            = (known after apply)
          + volume_id             = (known after apply)
          + volume_size           = (known after apply)
          + volume_type           = (known after apply)
        }
    }

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

─────────────────────────────────────────────────────────────────────────────

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 C:\Users\Ansari\Desktop\DevOps\terraform> 
```

- After that write terraform apply command then a instance will be created named acc to mentioned in resource



```
PS C:\Users\Ansari\Desktop\DevOps\terraform> 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.my-first-server will be created
  + resource "aws_instance" "my-first-server" {
      + ami                                  = "ami-068257025f72f470d"
      + 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)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (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
      + subnet_id                            = (known after apply)
      + tags_all                             = (known after apply)
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
```



```
      + metadata_options {
          + http_endpoint               = (known after apply)
          + http_put_response_hop_limit = (known after apply)
          + http_tokens                 = (known after apply)
          + instance_metadata_tags      = (known after apply)
        }

      + network_interface {
          + delete_on_termination = (known after apply)
          + device_index          = (known after apply)
          + network_card_index    = (known after apply)
          + network_interface_id  = (known after apply)
        }

      + private_dns_name_options {
          + enable_resource_name_dns_a_record    = (known after apply)
          + enable_resource_name_dns_aaaa_record = (known after apply)
          + hostname_type                        = (known after apply)
        }

      + root_block_device {
          + delete_on_termination = (known after apply)
          + device_name           = (known after apply)
          + encrypted             = (known after apply)
          + iops                  = (known after apply)
          + kms_key_id            = (known after apply)
          + tags                  = (known after apply)
          + throughput            = (known after apply)
          + volume_id             = (known after apply)
          + volume_size           = (known after apply)
          + volume_type           = (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.my-first-server: Creating...
aws_instance.my-first-server: Still creating... [10s elapsed]
```