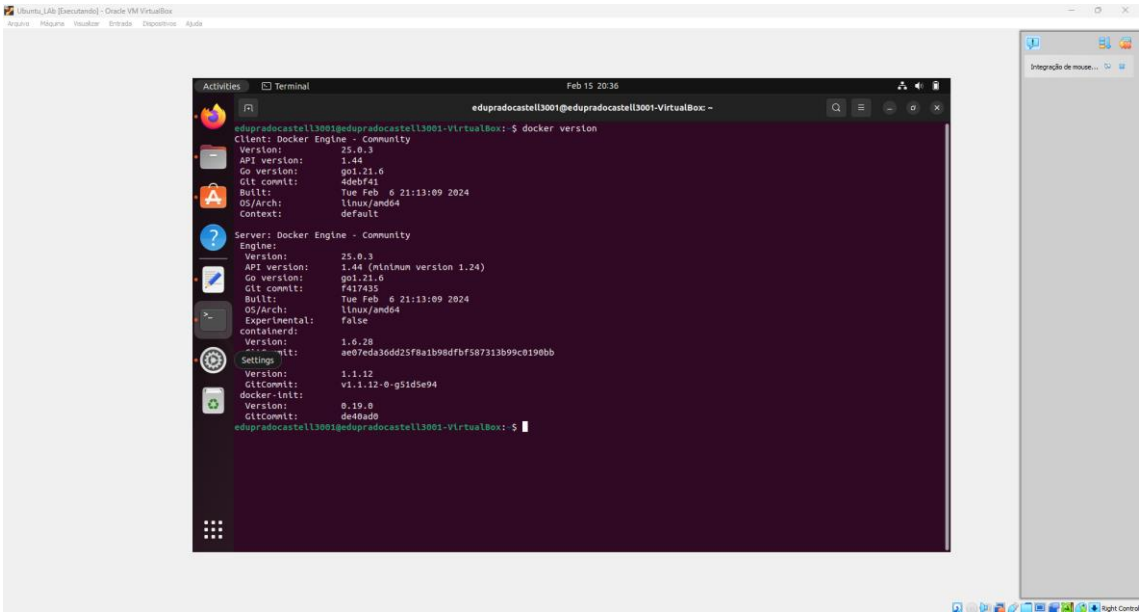
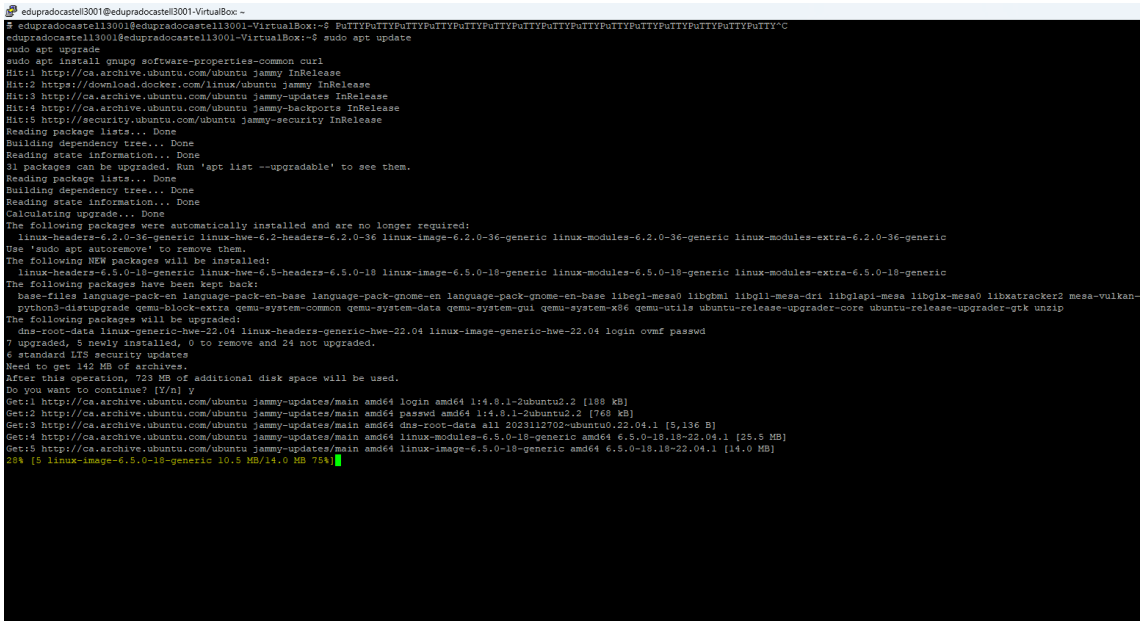
Terraform

Ubuntu 22.04 Run my VirtualBox.

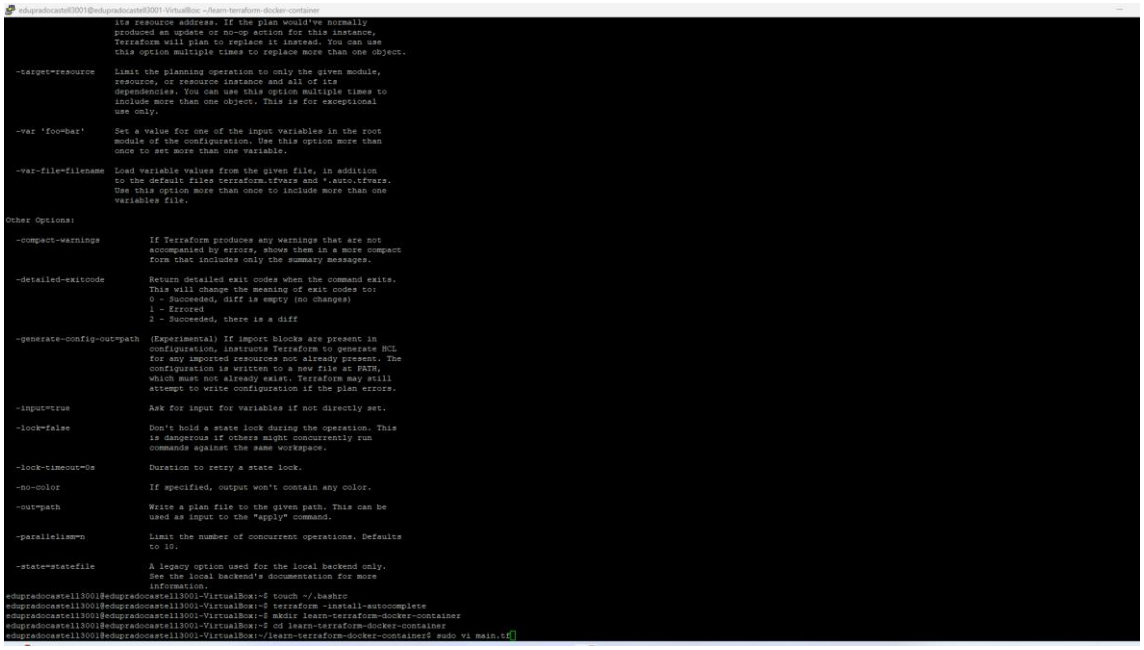


Frist steps for installation Terraform.

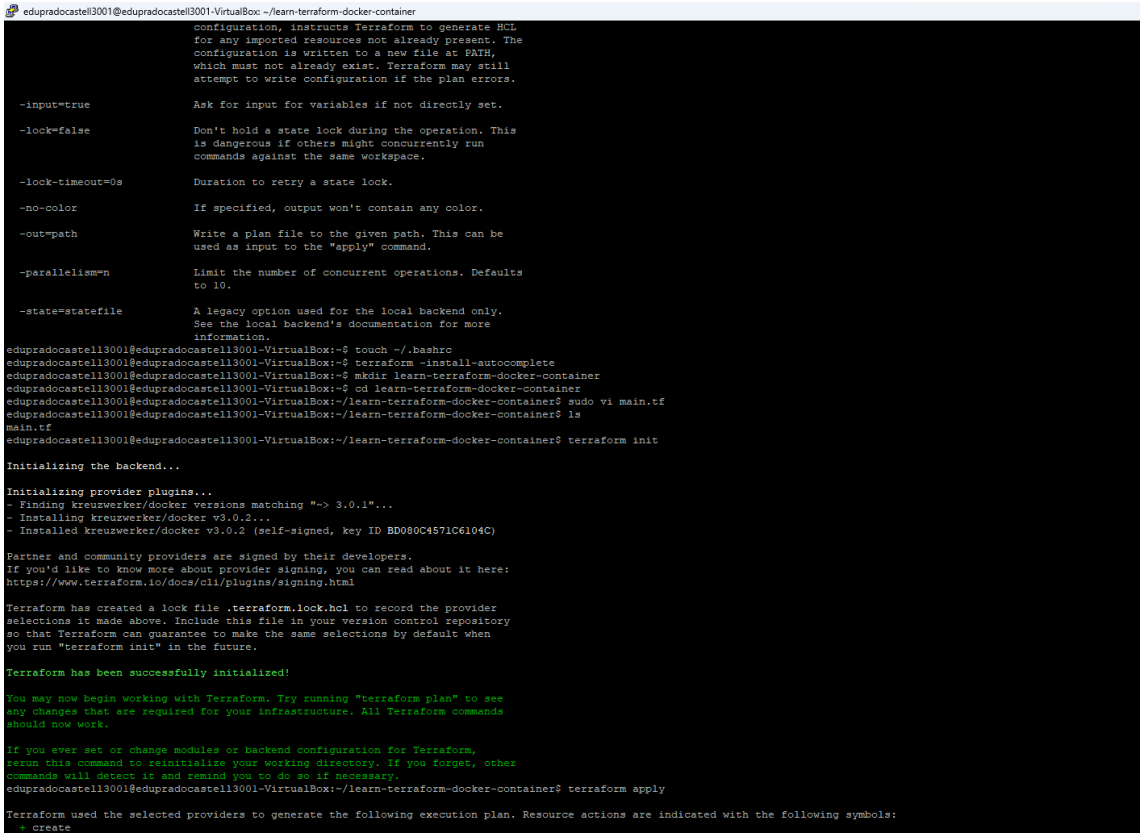
```
edupradocastell3001@edupradocastell3001-VirtualBox: ~
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

edupradocastell3001@edupradocastell3001-VirtualBox:~$
edupradocastell3001@edupradocastell3001-VirtualBox:~$
edupradocastell3001@edupradocastell3001-VirtualBox:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
edupradocastell3001@edupradocastell3001-VirtualBox:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
edupradocastell3001@edupradocastell3001-VirtualBox:~$ sudo apt-get update && sudo apt-get install -y gnupg software-properties-common
Hit:1 https://download.docker.com/linux/ubuntu jammy InRelease
Hit:2 http://ca.archive.ubuntu.com/ubuntu jammy InRelease
Get:3 http://ca.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:5 http://ca.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:6 http://ca.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1,376 kB]
Get:7 http://ca.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [569 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1,161 kB]
Get:9 http://ca.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [273 kB]
Get:10 http://ca.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [1,431 kB]
Get:11 http://ca.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [235 kB]
Get:12 http://ca.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1,049 kB]
Get:13 http://ca.archive.ubuntu.com/ubuntu jammy-updates/universe i386 Packages [689 kB]
Get:14 http://ca.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [237 kB]
Get:15 http://ca.archive.ubuntu.com/ubuntu jammy-updates/multiverse i386 Packages [4,184 B]
Get:16 http://ca.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [42.1 kB]
Get:17 http://security.ubuntu.com/ubuntu jammy-security/main i386 Packages [404 kB]
Get:18 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [213 kB]
Get:19 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1,401 kB]
Get:20 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [231 kB]
Get:21 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [842 kB]
Get:22 http://security.ubuntu.com/ubuntu jammy-security/universe i386 Packages [592 kB]
Get:23 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [161 kB]
Fetched 11.1 MB in 4s (2,950 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gnupg is already the newest version (2.2.27-3ubuntu2.1).
gnupg set to manually installed.
software-properties-common is already the newest version (0.99.22.9).
The following packages were automatically installed and are no longer required:
  linux-headers-6.2.0-36-generic linux-hwe-6.2-headers-6.2.0-36 linux-image-6.2.0-36-generic linux-modules-6.2.0-36-generic linux-modules-extra-6.2.0-36-generic
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 31 not upgraded.
1 not fully installed or removed.
After this operation, 0 B of additional disk space will be used.
Setting up rabbitmq-server (3.9.13-1ubuntu0.22.04.2) ...
edupradocastell3001@edupradocastell3001-VirtualBox:~$
```



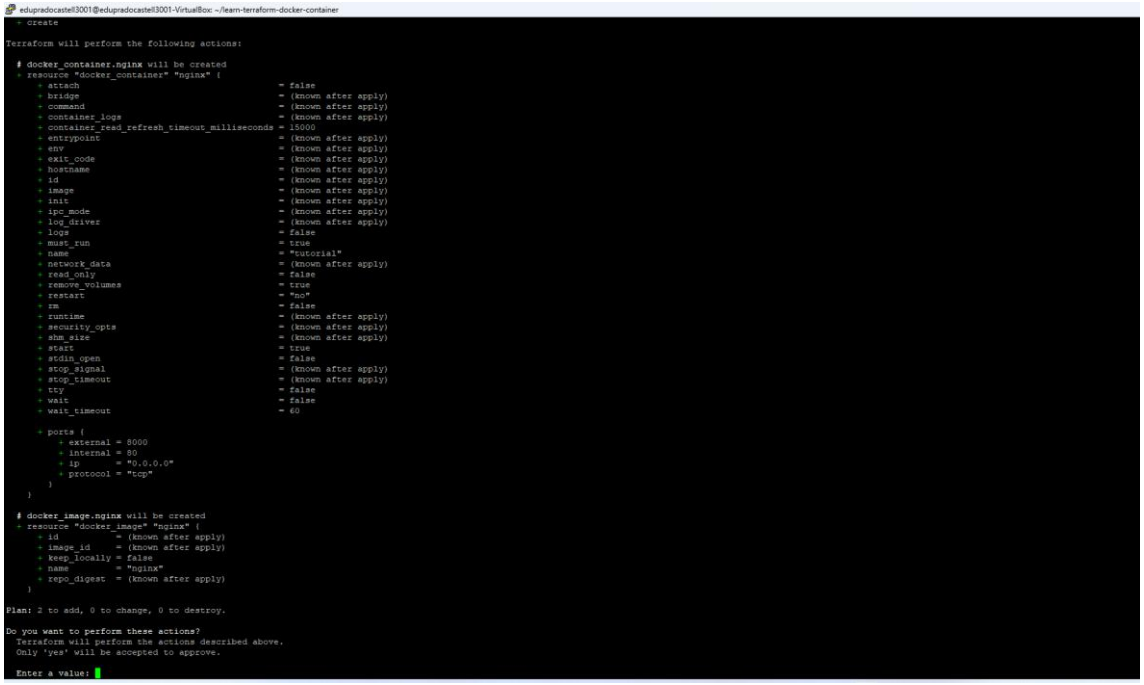
This step Creates a directory and a file main.tf



Run command ->> Terraform init

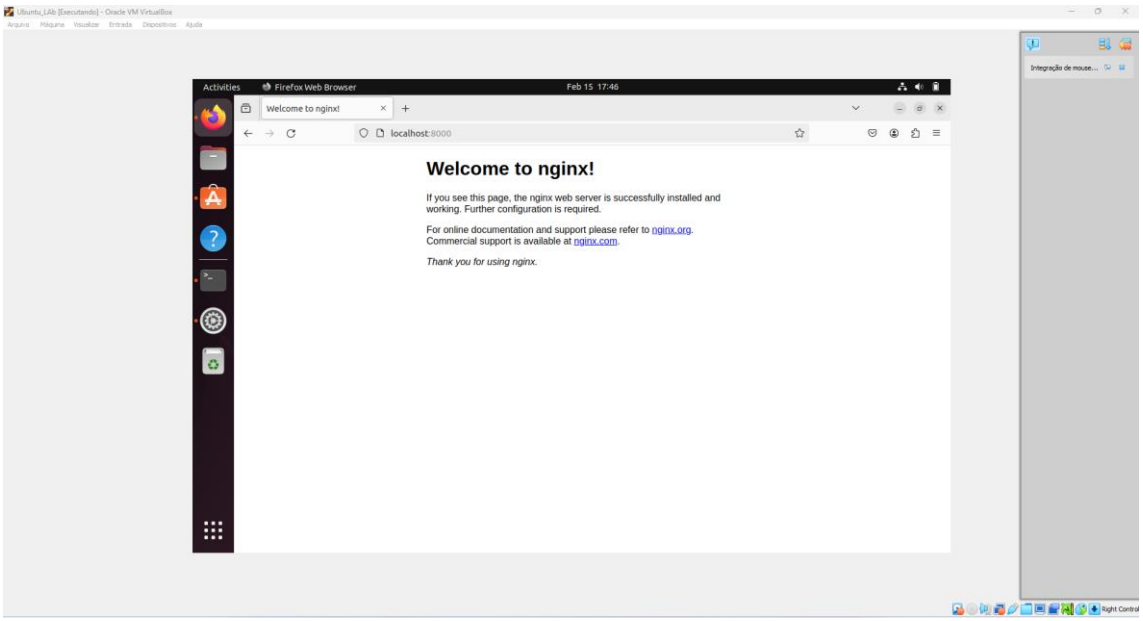


Confirmations YES

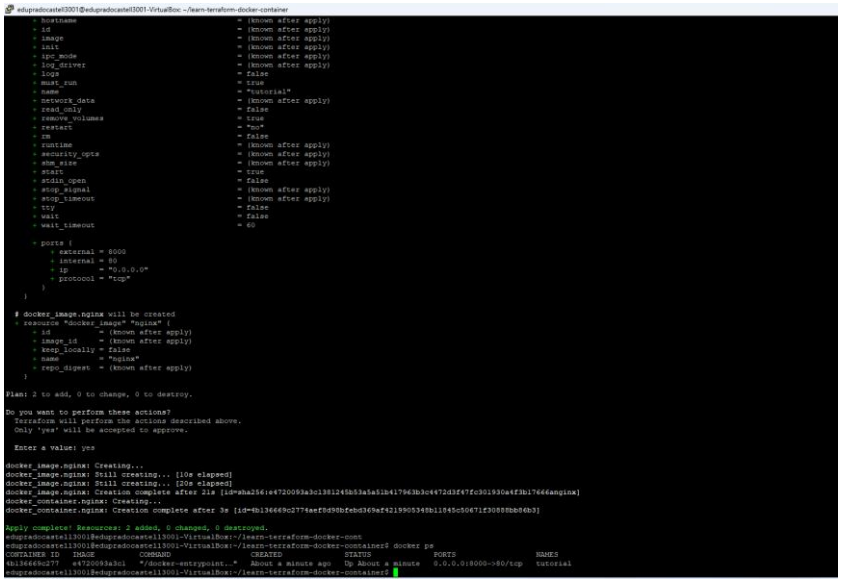


```
edupradocastell3001@edupradocastell3001-VirtualBox: ~/learn-terraform-docker-container

  # docker_container.nginx will be created
  + resource "docker_container" "nginx" {
      + attach                                      = false
      + bridge                                      = (known after apply)
      + command                                     = (known after apply)
      + container_logs                              = (known after apply)
      + container_read_refresh_timeout_milliseconds = 15000
      + entrypoint                                  = (known after apply)
      + env                                         = (known after apply)
      + exit_code                                   = (known after apply)
      + hostname                                    = (known after apply)
      + id                                          = (known after apply)
      + image                                       = (known after apply)
      + init                                        = (known after apply)
      + ipc_mode                                    = (known after apply)
      + log_driver                                  = (known after apply)
      + logs                                        = false
      + must_run                                    = true
      + name                                        = "tutorial"
      + network_data                                = (known after apply)
      + read_only                                   = false
      + remove_volumes                              = true
      + restart                                     = "no"
      + rm                                          = false
      + runtime                                     = (known after apply)
      + security_opts                               = (known after apply)
      + shm_size                                    = (known after apply)
      + start                                       = true
      + stdin_open                                  = false
      + stop_signal                                 = (known after apply)
      + stop_timeout                                = (known after apply)
      + tty                                         = false
      + wait                                        = false
      + wait_timeout                                = 60

      + ports {
          + external = 8000
          + internal = 80
          + ip       = "0.0.0.0"
          + protocol = "tcp"
        }
    }

  # docker_image.nginx will be created
  + resource "docker_image" "nginx" {
      + id           = (known after apply)
      + image_id     = (known after apply)
      + keep_locally = false
      + name         = "nginx"
      + repo_digest  = (known after apply)
    }

Plan: 2 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

docker_image.nginx: Creating...
```

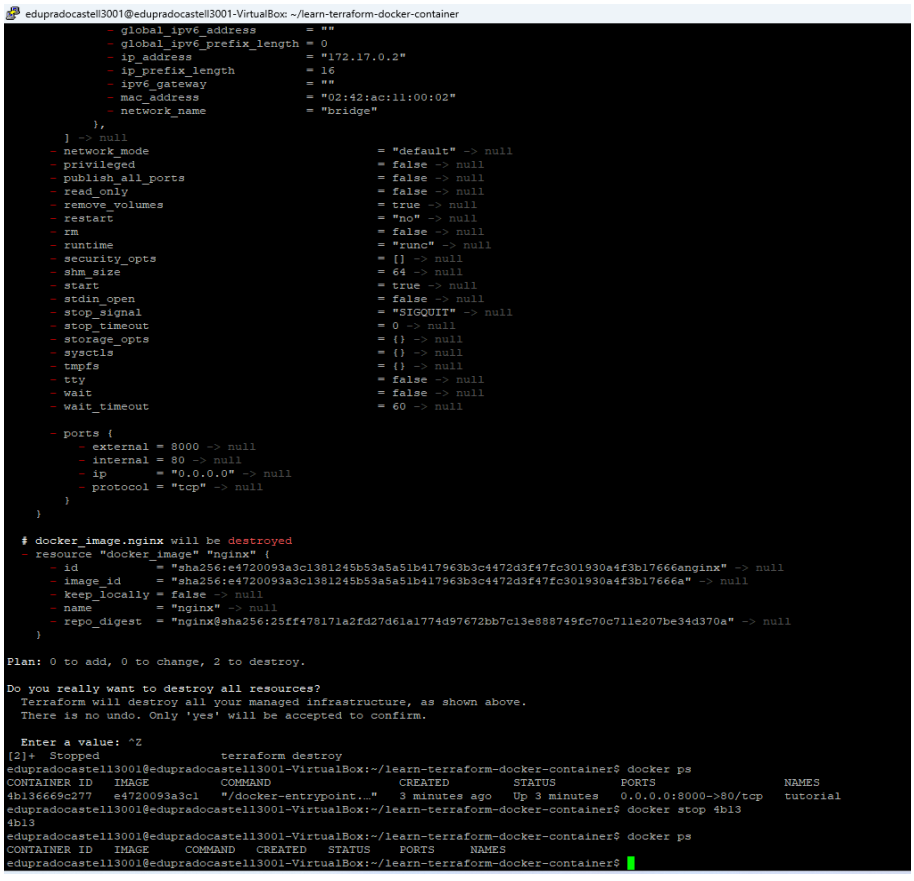
Now Nginx is working in my localhost;

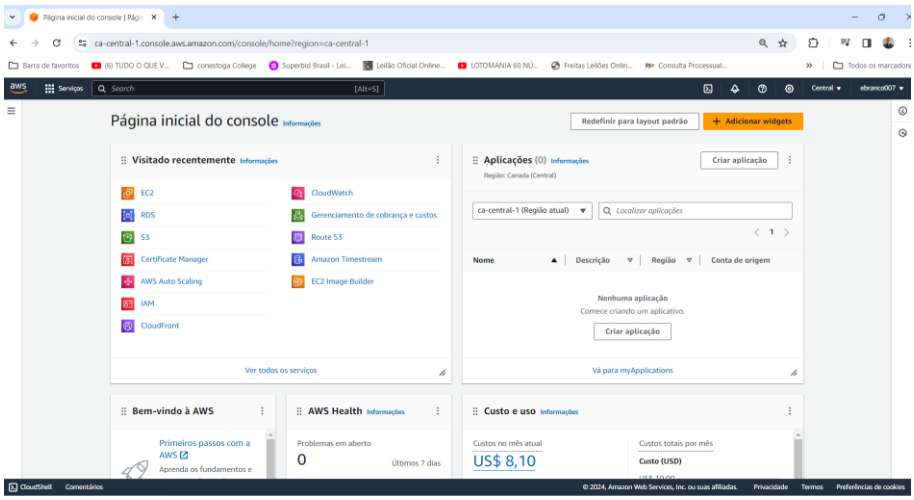


Docker Ps show the container was created.

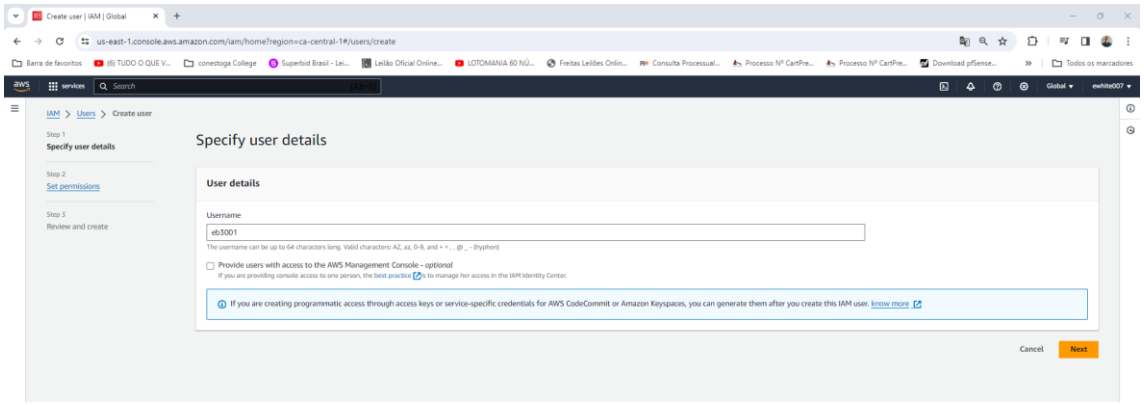


Stop container

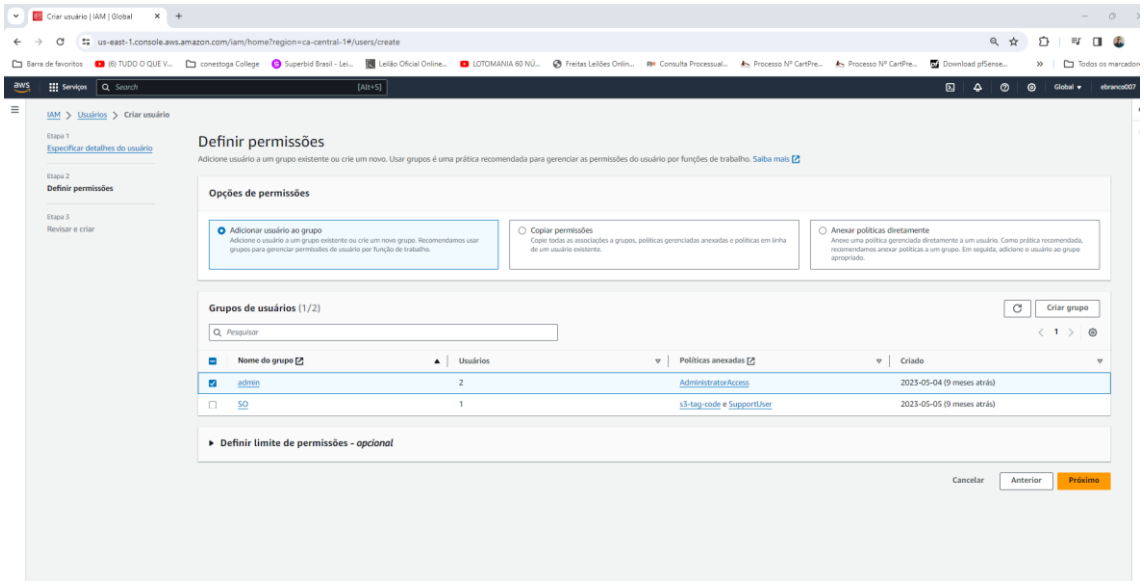




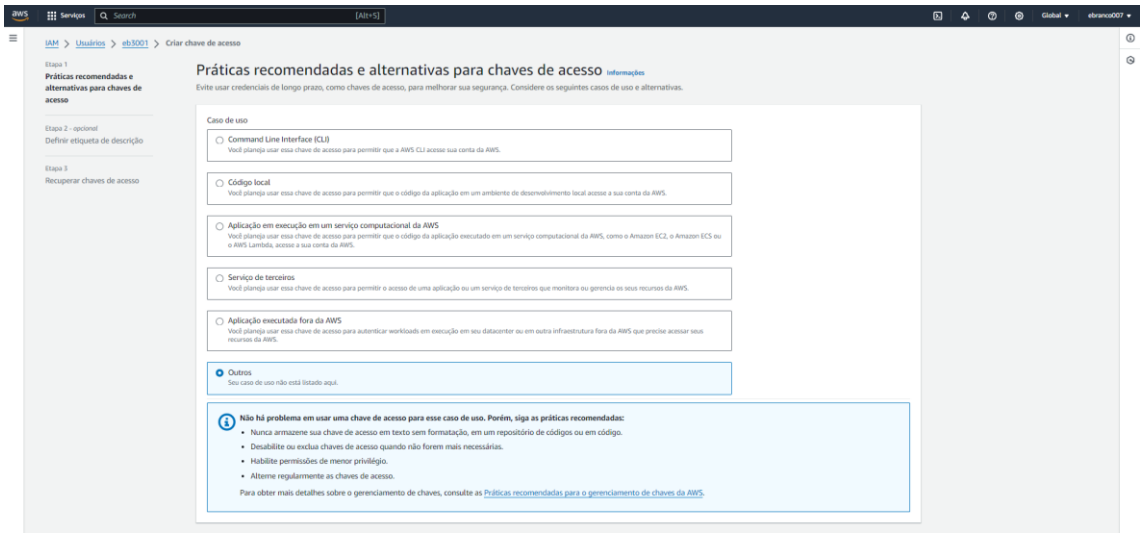
Create User EB3001 (it is my initials and last digit number of Conestoga)



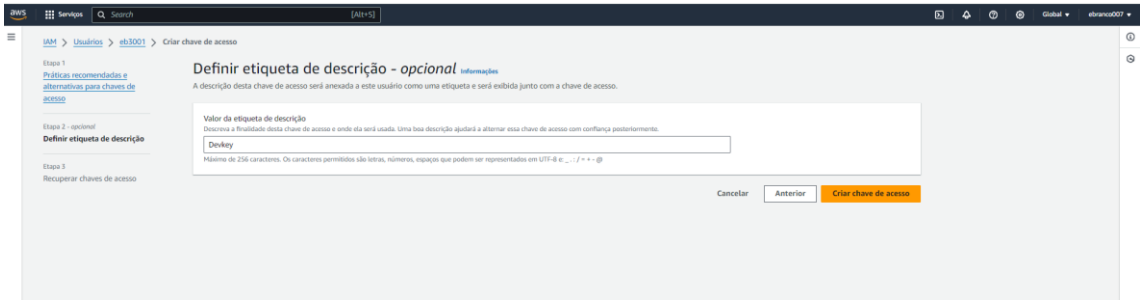
Give permissions



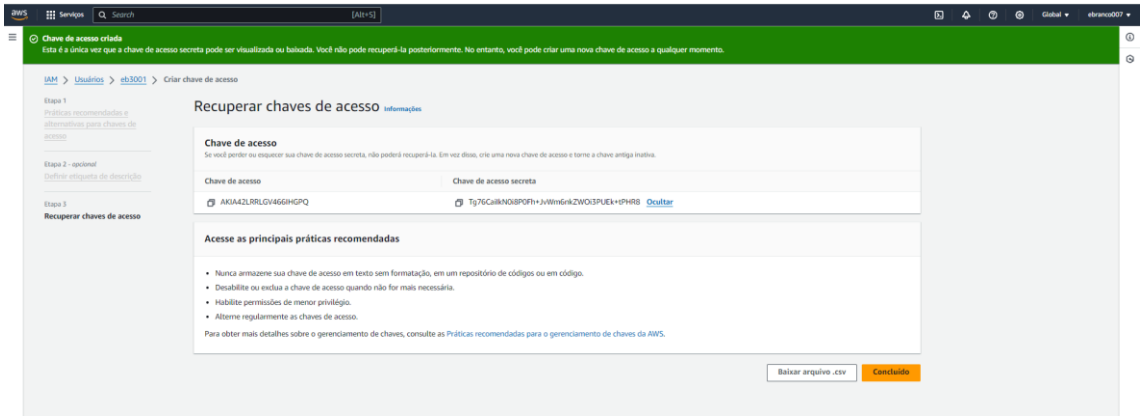
Another method for auth.



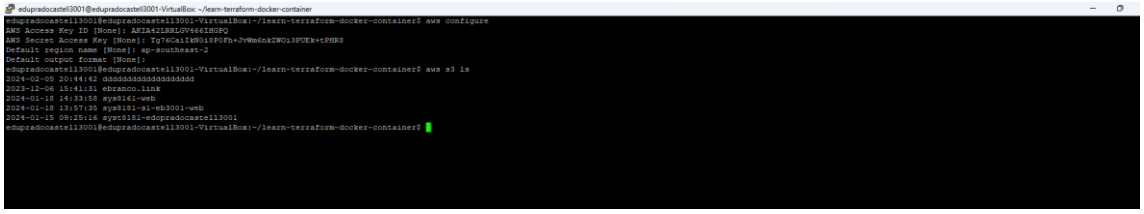
Create key



ACESSkey and Secrect key



Login in aws configure and just LS in my buckets



Create Directory, and file main.tf;

```
edupradocastell3001@edupradocastell3001-VirtualBox: ~/learn-terraform-docker-container/learn-terraform-aws-instance
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container$ aws configure
AWS Access Key ID [None]: AKIA42LRRLGV466IHGPQ
AWS Secret Access Key [None]: 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
Default region name [None]: ap-southeast-2
Default output format [None]:
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container$ aws s3 ls
2024-02-05 20:44:42 ddddddddddddddddddd
2023-12-06 15:41:31 ebranco.link
2024-01-18 14:33:58 sys8161-web
2024-01-18 13:57:35 sys8181-s1-eb3001-web
2024-01-15 09:25:16 syst8181-edopradocastell3001
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container$ export AWS_ACCESS_KEY_ID=
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container$ export AWS_SECRET_ACCESS_KEY=
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container$ mkdir learn-terraform-aws-instance
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container$ cd learn-terraform-aws-instance
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ touch main.tf
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ ls
main.tf
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ vi main.tf
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ ls
main.tf
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ vi main.tf

[5]+  Stopped                 vi main.tf
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ vi main.tf

[6]+  Stopped                 vi main.tf
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$
```

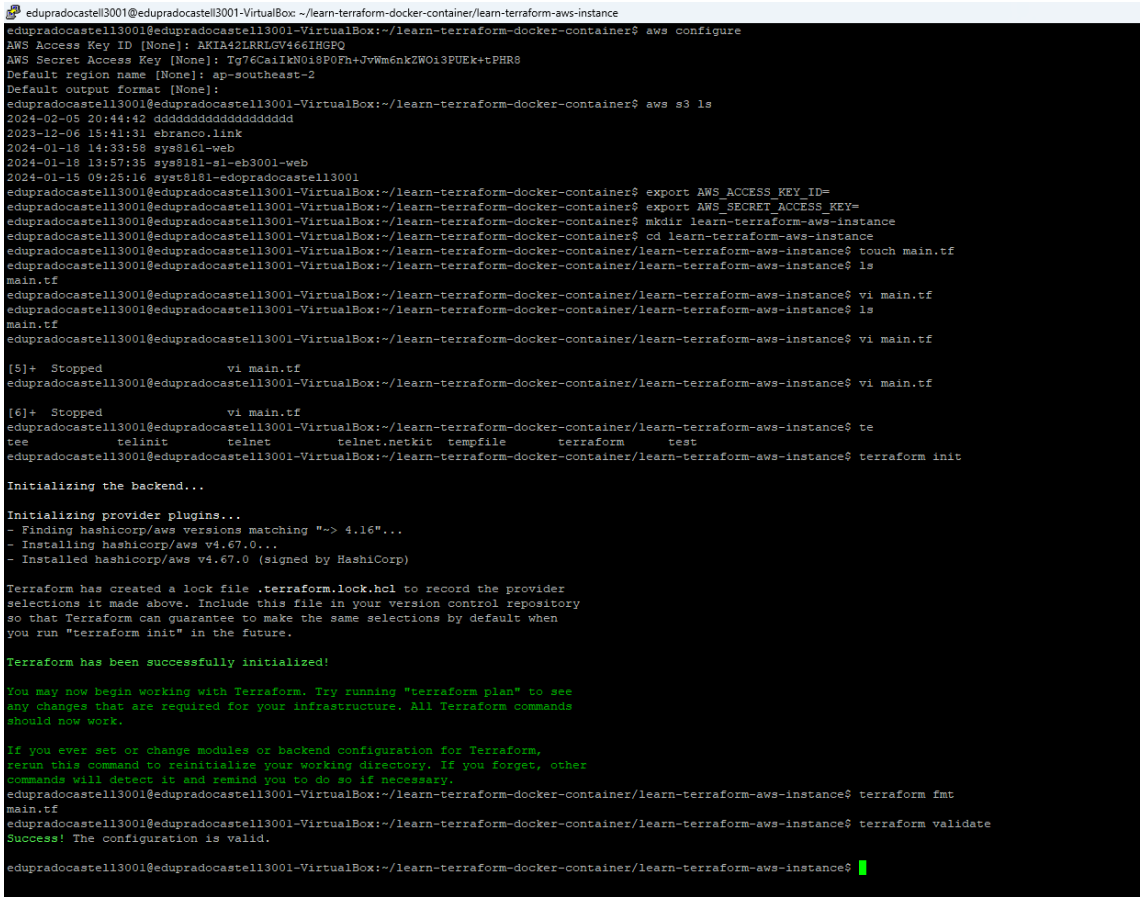
Just copy and past.

```
edupradocastell3001@edupradocastell3001-VirtualBox: ~/learn-terraform-docker-container/learn-terraform-aws-instance
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 4.16"
    }
  }

  required_version = ">= 1.2.0"
}

provider "aws" {
  region  = "us-west-2"
}

resource "aws_instance" "app_server" {
  ami           = "ami-830c94e3"
  instance_type = "t2.micro"

  tags = {
    Name = "ExampleAppServerInstance"
  }
}
~
~
~
~
~
```

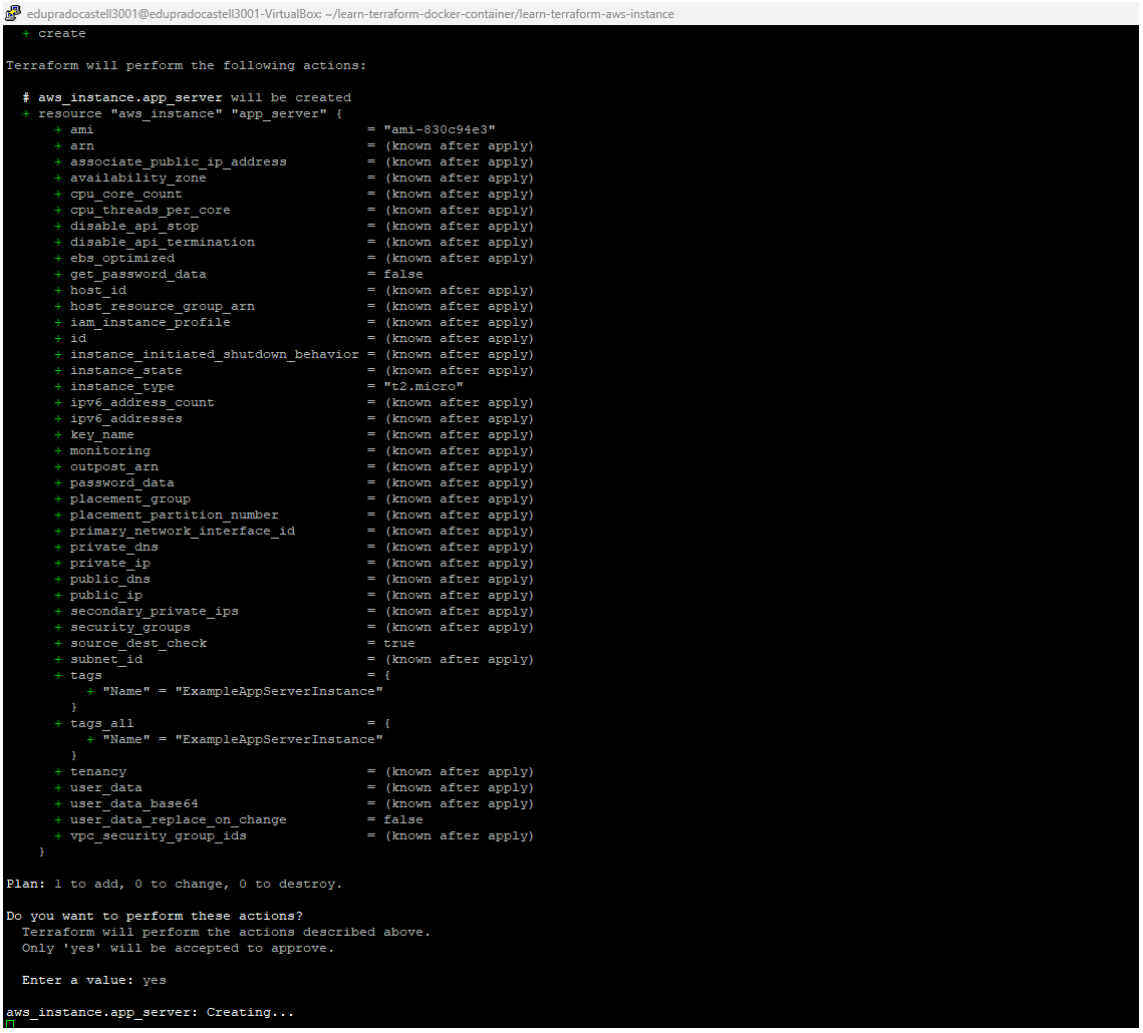
Terraform init

```
edupradocastell3001@edupradocastell3001-VirtualBox: ~/learn-terraform-docker-container/learn-terraform-aws-instance
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container$ aws configure
AWS Access Key ID [None]: AKIA42LRRLGV466IHGPQ
AWS Secret Access Key [None]: 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
Default region name [None]: ap-southeast-2
Default output format [None]:
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container$ aws s3 ls
2024-02-05 20:44:42 ddddddddddddddddddd
2023-12-06 15:41:31 ebranco.link
2024-01-18 14:33:58 sys8161-web
2024-01-18 13:57:35 sys8181-s1-eb3001-web
2024-01-15 09:25:16 syst8181-edopradocastell3001
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container$ export AWS_ACCESS_KEY_ID=
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container$ export AWS_SECRET_ACCESS_KEY=
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container$ mkdir learn-terraform-aws-instance
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container$ cd learn-terraform-aws-instance
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ touch main.tf
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ ls
main.tf
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ vi main.tf
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ ls
main.tf
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ vi main.tf

[5]+  Stopped                 vi main.tf
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ vi main.tf

[6]+  Stopped                 vi main.tf
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ te
tee            telinit        telnet         telnet.netkit  tempfile       terraform      test
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "~> 4.16"...
- Installing hashicorp/aws v4.67.0...
- Installed hashicorp/aws v4.67.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
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.
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$
```

Validating..


edupradocastell3001@edupradocastell3001-VirtualBox: ~/learn-terraform-docker-container/learn-terraform-aws-instance
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container$ aws configure
AWS Access Key ID [None]: AKIA42LRRLGV466IHGPQ
AWS Secret Access Key [None]: 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
Default region name [None]: ap-southeast-2
Default output format [None]:
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container$ aws s3 ls
2024-02-05 20:44:42 ddddddddddddddddddd
2023-12-06 15:41:31 ebranco.link
2024-01-18 14:33:58 sys8161-web
2024-01-18 13:57:35 sys8181-s1-eb3001-web
2024-01-15 09:25:16 syst8181-edopradocastell3001
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container$ export AWS_ACCESS_KEY_ID=
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container$ export AWS_SECRET_ACCESS_KEY=
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container$ mkdir learn-terraform-aws-instance
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container$ cd learn-terraform-aws-instance
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ touch main.tf
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ ls
main.tf
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ vi main.tf
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ ls
main.tf
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ vi main.tf
[5]+ Stopped vi main.tf
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ vi main.tf
[6]+ Stopped vi main.tf
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ te
tee telinit telnet telnet.netkit tempfile terraform test
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding hashicorp/aws versions matching "~> 4.16"...
- Installing hashicorp/aws v4.67.0...
- Installed hashicorp/aws v4.67.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
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.
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ terraform fmt
main.tf
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ terraform validate
Success! The configuration is valid.
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$

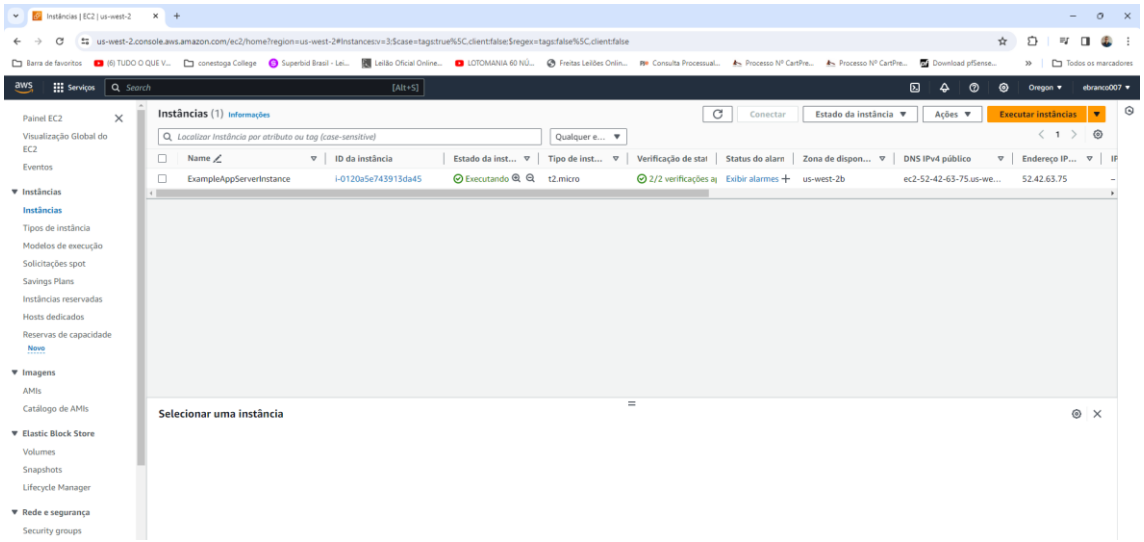
Run and confirmations



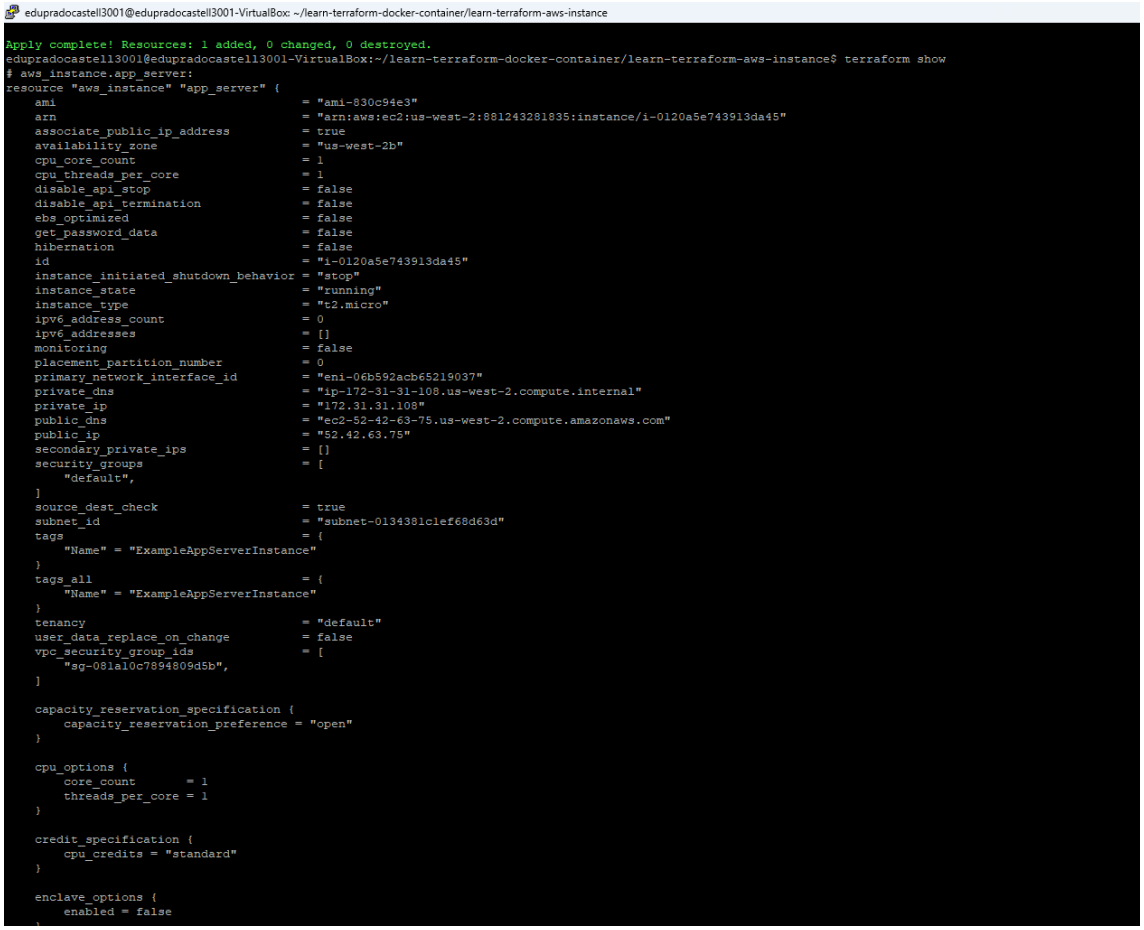
Create EC2 instance by terraform

```
edupradocastell3001@edupradocastell3001-VirtualBox: ~/learn-terraform-docker-container/learn-terraform-aws-instance
      + 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)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (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                                 = {
          + "Name" = "ExampleAppServerInstance"
        }
      + tags_all                             = {
          + "Name" = "ExampleAppServerInstance"
        }
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (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.app_server: Creating...
aws_instance.app_server: Still creating... [10s elapsed]
aws_instance.app_server: Still creating... [20s elapsed]
aws_instance.app_server: Still creating... [30s elapsed]
aws_instance.app_server: Still creating... [40s elapsed]
aws_instance.app_server: Still creating... [50s elapsed]
aws_instance.app_server: Creation complete after 54s [id=i-0120a5e743913da45]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$
```

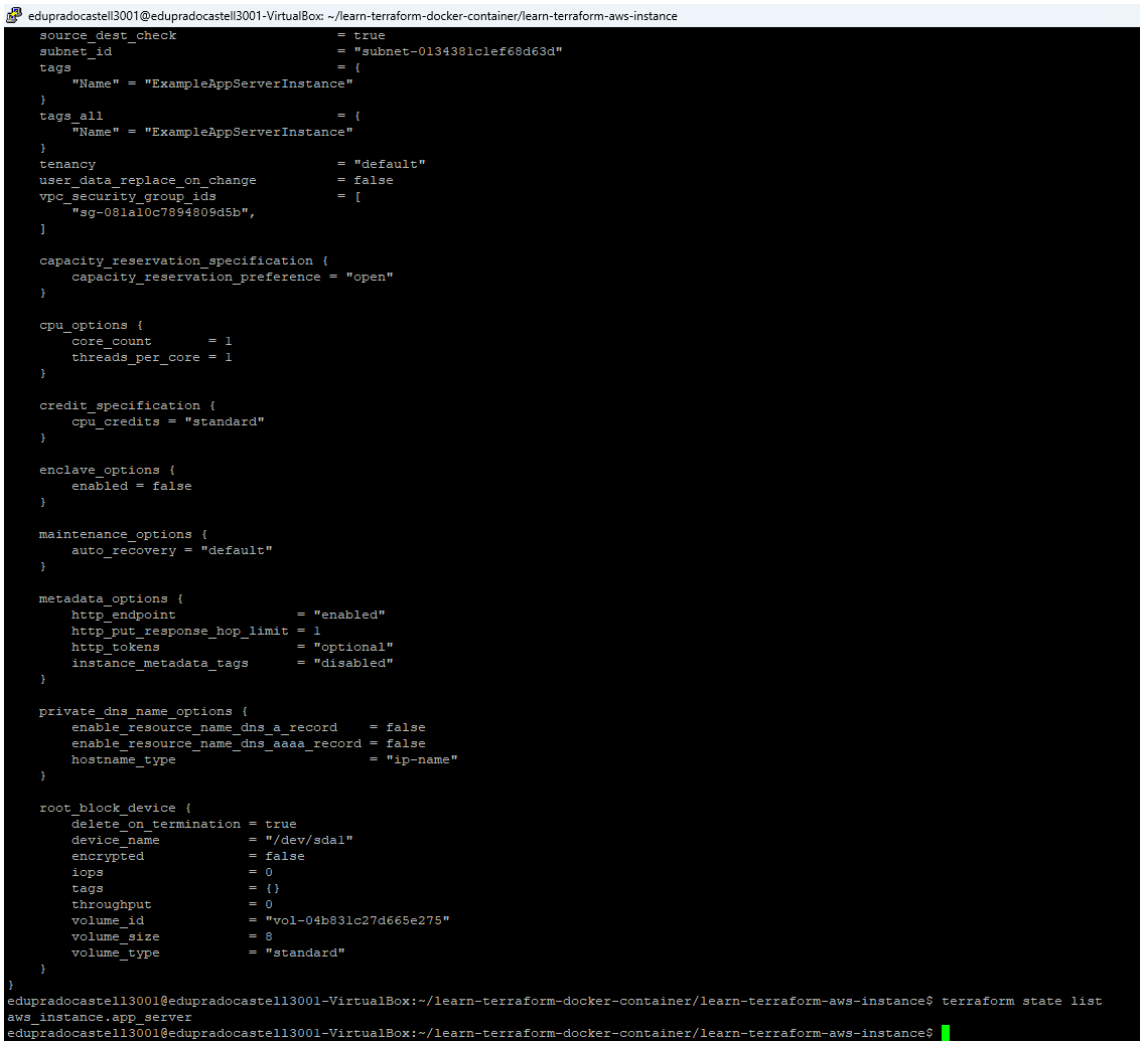
Cheking



Show detail about instance



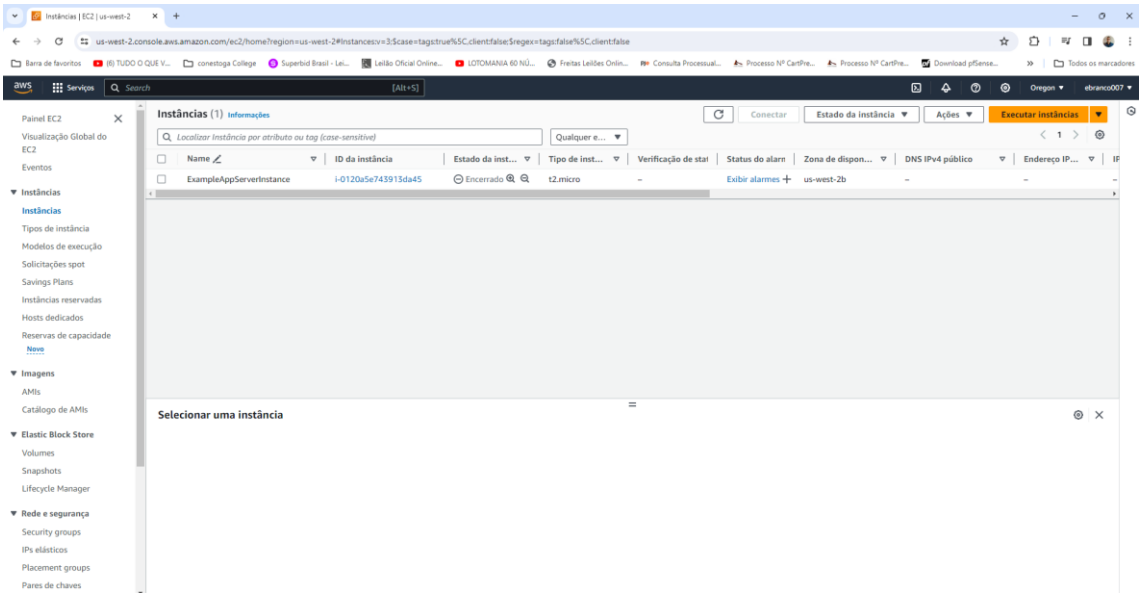
Details about instance;



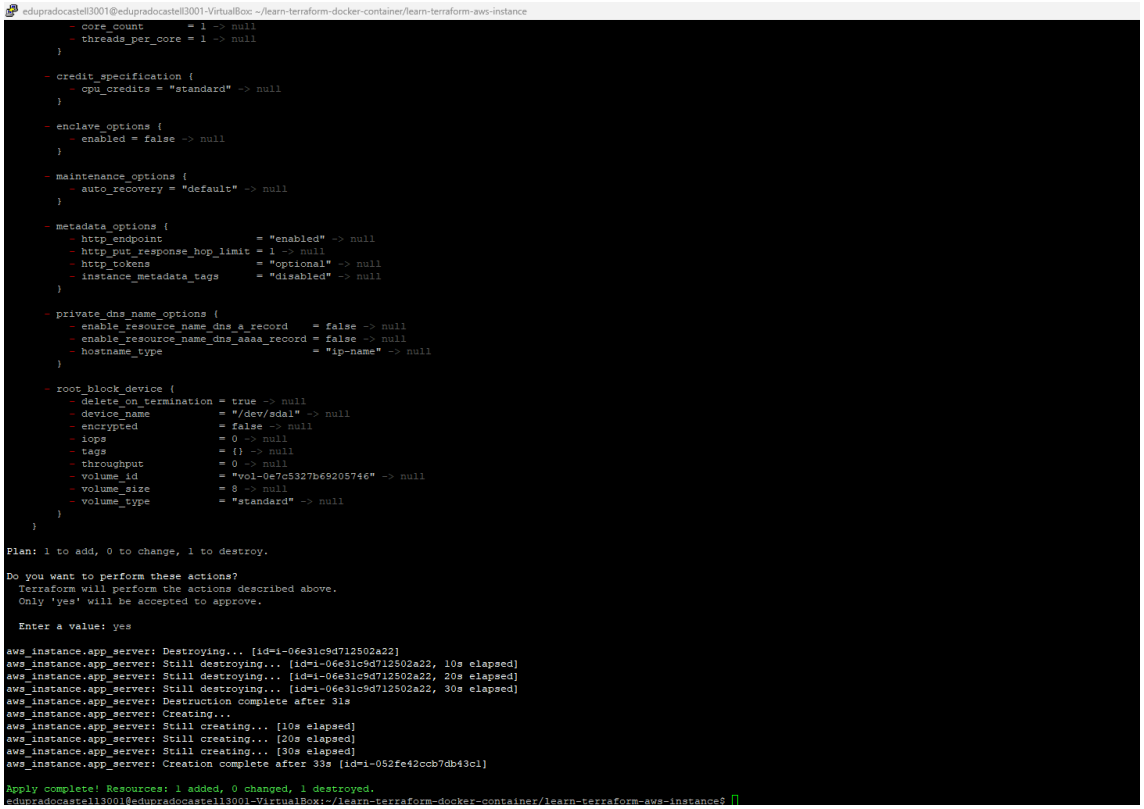
Destroying

```
edupradocastell3001@edupradocastell3001-VirtualBox: ~/learn-terraform-docker-container/learn-terraform-aws-instance
      + user_data_base64                      = (known after apply)
      ~ vpc_security_group_ids                = [
          - "sg-081a10c7894809d5b",
        ] -> (known after apply)
        # (5 unchanged attributes hidden)

      - capacity_reservation_specification {
          - capacity_reservation_preference = "open" -> null
        }

      - cpu_options {
          - core_count       = 1 -> null
          - threads_per_core = 1 -> null
        }

      - credit_specification {
          - cpu_credits = "standard" -> null
        }

      - enclave_options {
          - enabled = false -> null
        }

      - maintenance_options {
          - auto_recovery = "default" -> null
        }

      - metadata_options {
          - http_endpoint               = "enabled" -> null
          - http_put_response_hop_limit = 1 -> null
          - http_tokens                 = "optional" -> null
          - instance_metadata_tags      = "disabled" -> null
        }

      - private_dns_name_options {
          - enable_resource_name_dns_a_record    = false -> null
          - enable_resource_name_dns_aaaa_record = false -> null
          - hostname_type                        = "ip-name" -> null
        }

      - root_block_device {
          - delete_on_termination = true -> null
          - device_name           = "/dev/sda1" -> null
          - encrypted             = false -> null
          - iops                  = 0 -> null
          - tags                  = {} -> null
          - throughput            = 0 -> null
          - volume_id             = "vol-04b831c27d665e275" -> null
          - volume_size           = 8 -> null
          - volume_type           = "standard" -> null
        }
    }

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

aws_instance.app_server: Destroying... [id=i-0120a5e743913da45]
```

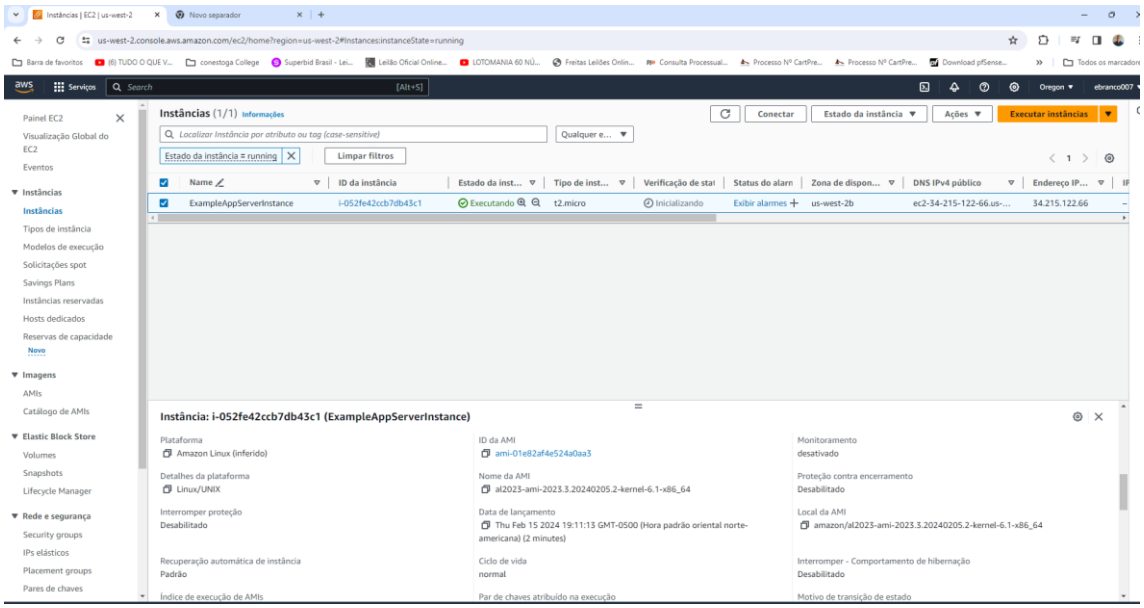
Just confirmation



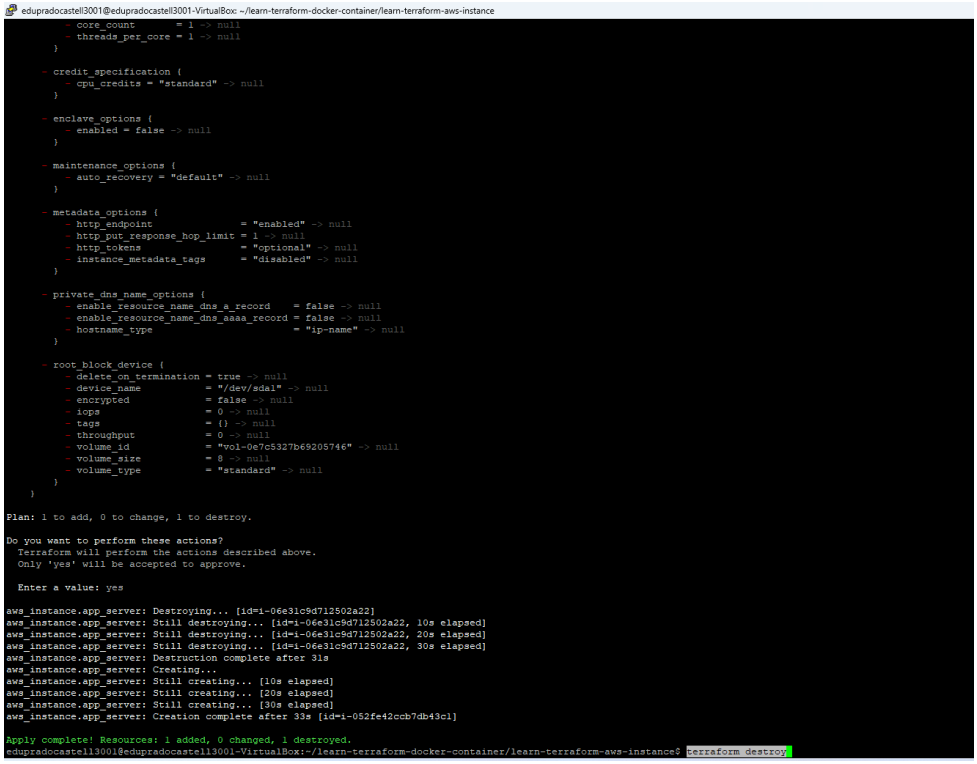
Now, after edit file, as can see change AMI. So, destroying and creating a new instance.



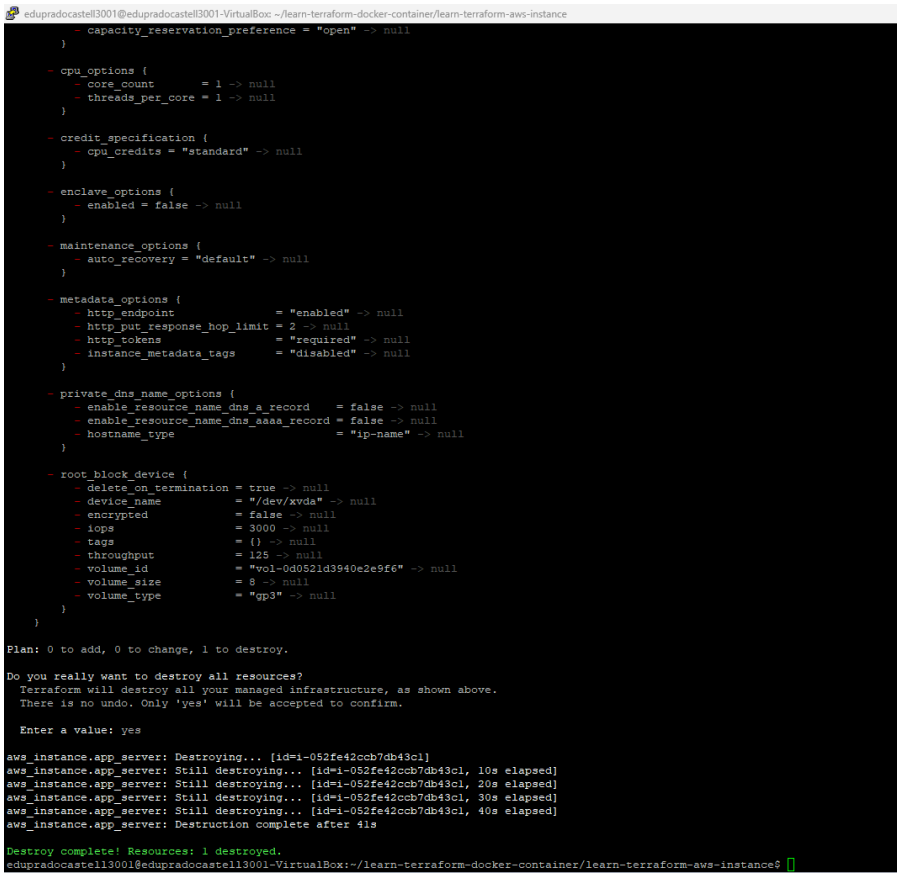
Show another AMI - CODE



Destroy instance

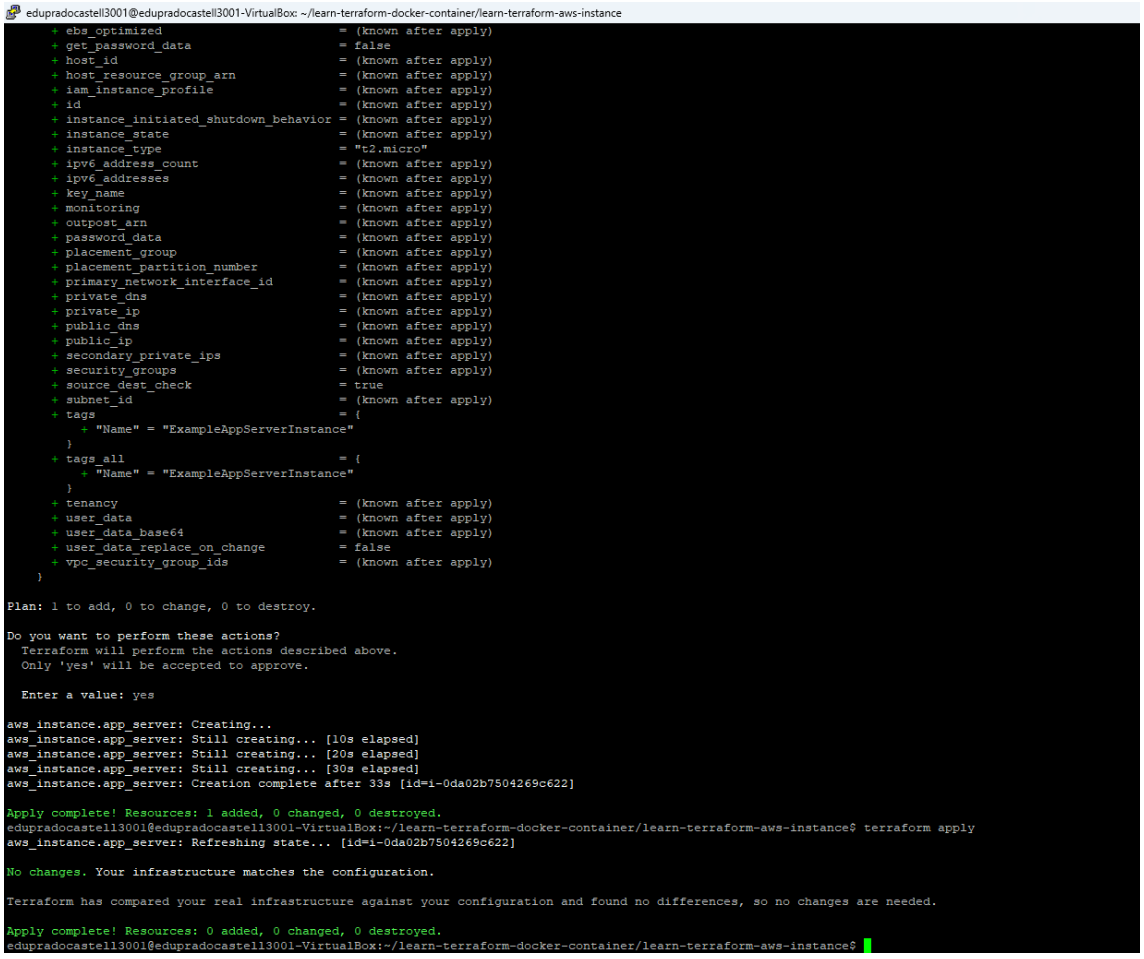


Was destroyed

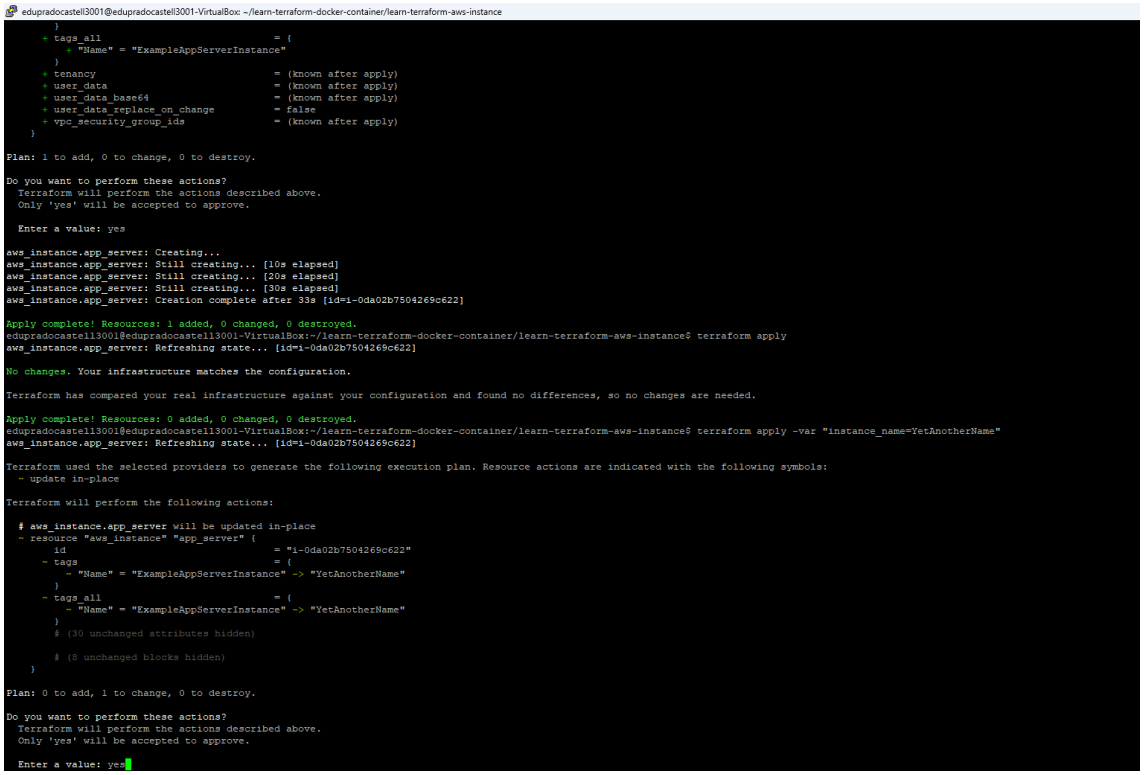


Create file `variables.tf`

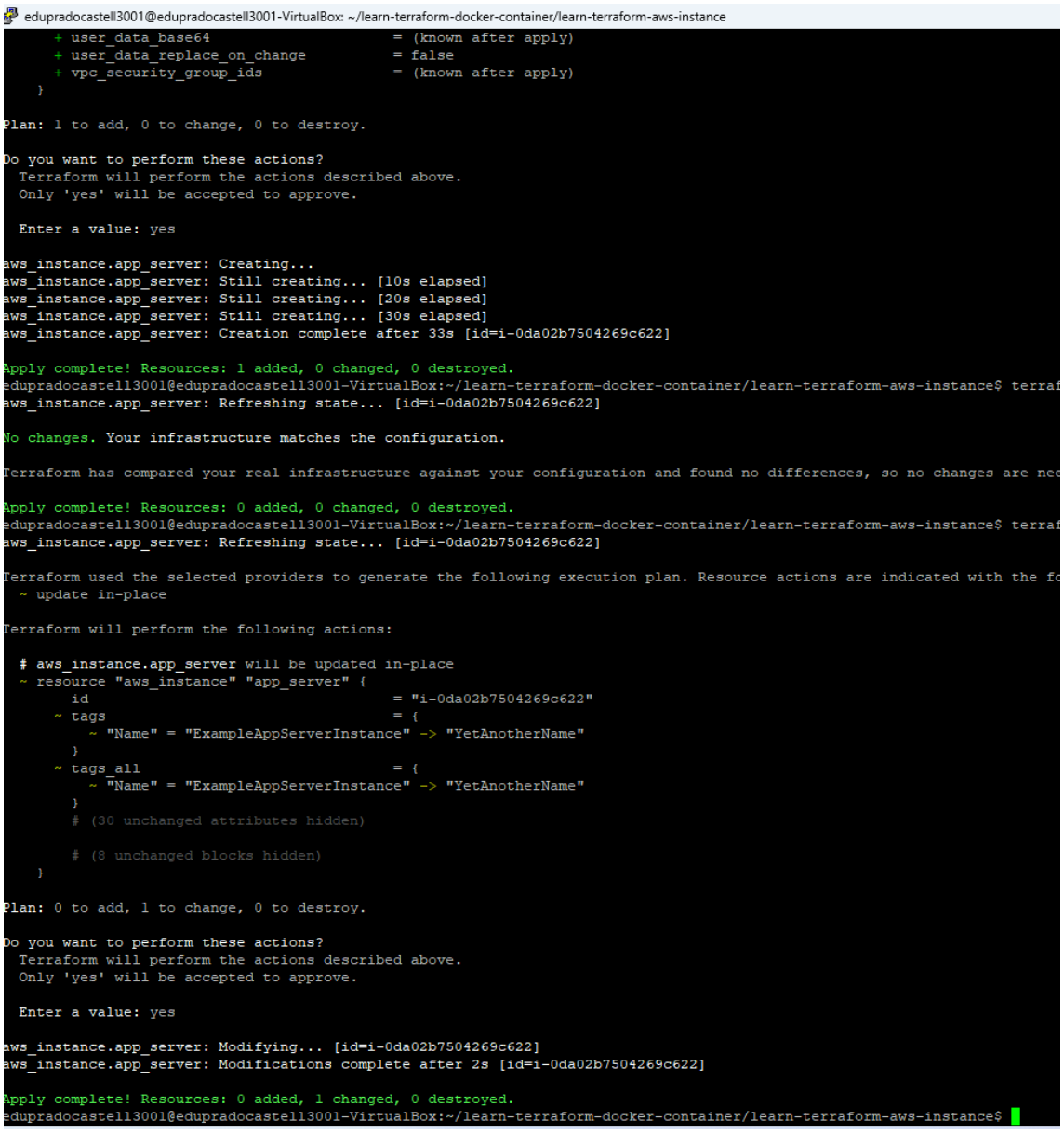
Creating..

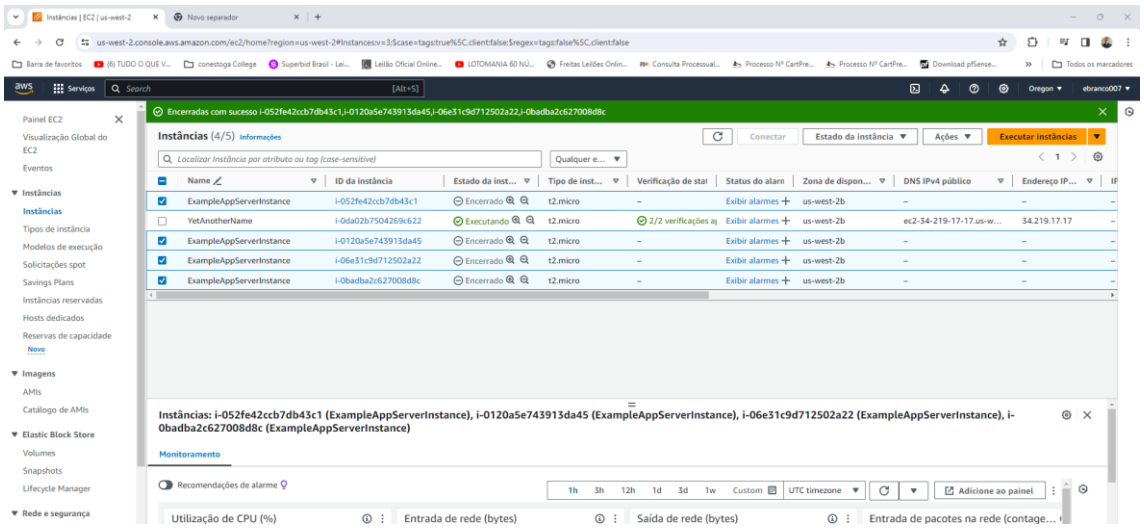


Just confirmations Yes



Just change Name

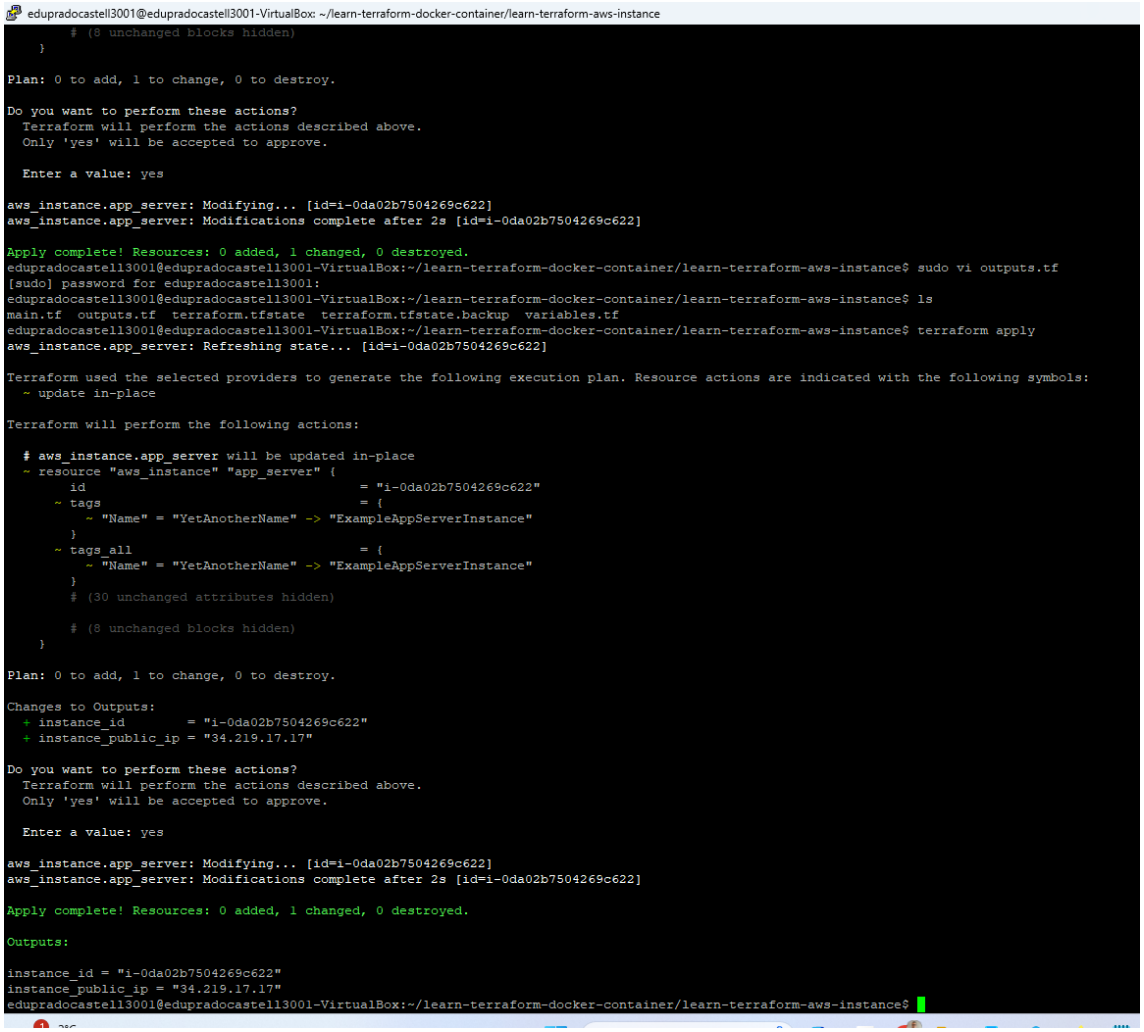




Outputs

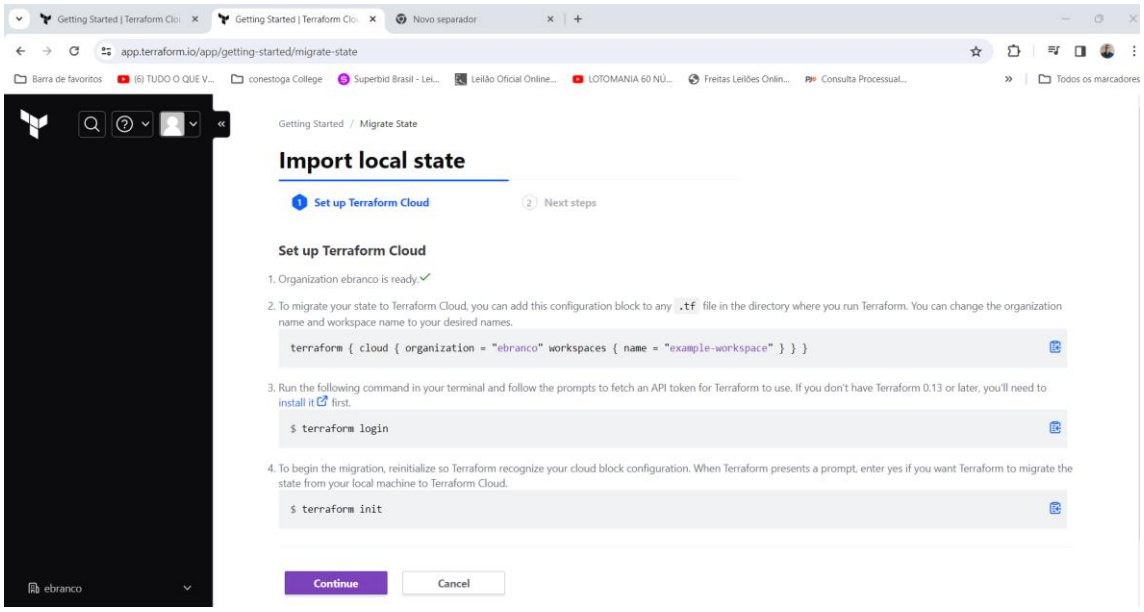
```
eduradocastell3001@edupradocastell3001-VirtualBox: ~/learn-terraform-docker-container/learn-terraform-aws-instance

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.app_server: Creating...
aws_instance.app_server: Still creating... [10s elapsed]
aws_instance.app_server: Still creating... [20s elapsed]
aws_instance.app_server: Still creating... [30s elapsed]
aws_instance.app_server: Creation complete after 33s [id=i-0da02b7504269c622]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
eduradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ terraform apply
aws_instance.app_server: Refreshing state... [id=i-0da02b7504269c622]

No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and found no differences, so no changes are needed.

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
eduradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ terraform apply -var "instance_name=YetAnotherName"
aws_instance.app_server: Refreshing state... [id=i-0da02b7504269c622]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  ~ update in-place

Terraform will perform the following actions:

  # aws_instance.app_server will be updated in-place
  ~ resource "aws_instance" "app_server" {
        id                                   = "i-0da02b7504269c622"
      ~ tags                                 = {
          ~ "Name" = "ExampleAppServerInstance" -> "YetAnotherName"
        }
      ~ tags_all                             = {
          ~ "Name" = "ExampleAppServerInstance" -> "YetAnotherName"
        }
        # (30 unchanged attributes hidden)

        # (8 unchanged blocks hidden)
    }

Plan: 0 to add, 1 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.app_server: Modifying... [id=i-0da02b7504269c622]
aws_instance.app_server: Modifications complete after 2s [id=i-0da02b7504269c622]

Apply complete! Resources: 0 added, 1 changed, 0 destroyed.
eduradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ sudo vi outputs.tf
[sudo] password for eduradocastell3001:
eduradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ ls
main.tf  outputs.tf  terraform.tfstate  terraform.tfstate.backup  variables.tf
eduradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$
```

Show instance Id / Ip-address

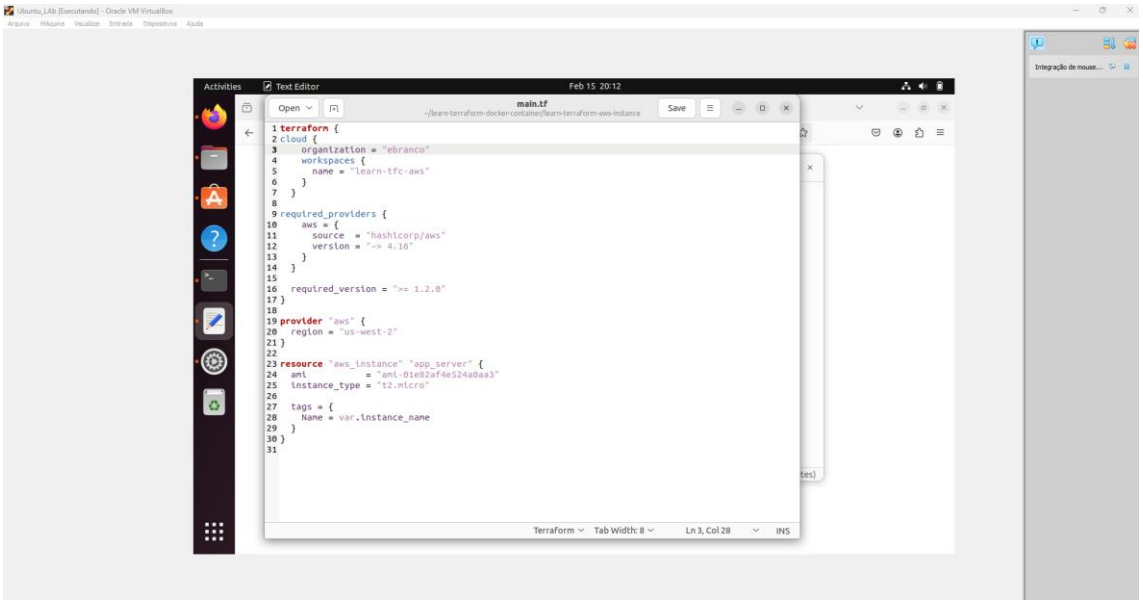


```
edupradocastell3001@edupradocastell3001-VirtualBox: ~/learn-terraform-docker-container/learn-terraform-aws-instance
Plan: 0 to add, 1 to change, 0 to destroy.

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.app_server: Modifying... [id=i-0da02b7504269c622]
aws_instance.app_server: Modifications complete after 2s [id=i-0da02b7504269c622]

Apply complete! Resources: 0 added, 1 changed, 0 destroyed.
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ sudo vi outputs.tf
[sudo] password for edupradocastell3001:
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ ls
main.tf  outputs.tf  terraform.tfstate  terraform.tfstate.backup  variables.tf
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ terraform apply
aws_instance.app_server: Refreshing state... [id=i-0da02b7504269c622]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  ~ update in-place

Terraform will perform the following actions:

  # aws_instance.app_server will be updated in-place
  ~ resource "aws_instance" "app_server" {
        id                                   = "i-0da02b7504269c622"
      ~ tags                                 = {
          ~ "Name" = "YetAnotherName" -> "ExampleAppServerInstance"
        }
      ~ tags_all                             = {
          ~ "Name" = "YetAnotherName" -> "ExampleAppServerInstance"
        }
        # (30 unchanged attributes hidden)

        # (8 unchanged blocks hidden)
    }

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

Changes to Outputs:
  + instance_id        = "i-0da02b7504269c622"
  + instance_public_ip = "34.219.17.17"

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.app_server: Modifying... [id=i-0da02b7504269c622]
aws_instance.app_server: Modifications complete after 2s [id=i-0da02b7504269c622]

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

Outputs:

instance_id = "i-0da02b7504269c622"
instance_public_ip = "34.219.17.17"
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ terraform output
instance_id = "i-0da02b7504269c622"
instance_public_ip = "34.219.17.17"
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$
```

Accessing my account

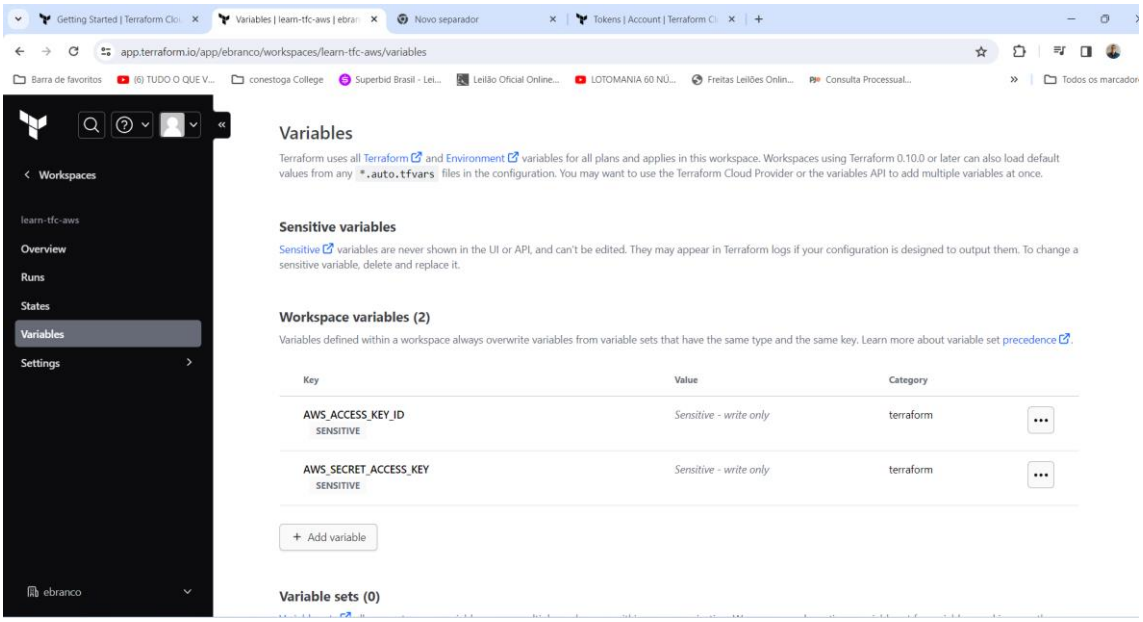


Edit file with my account details

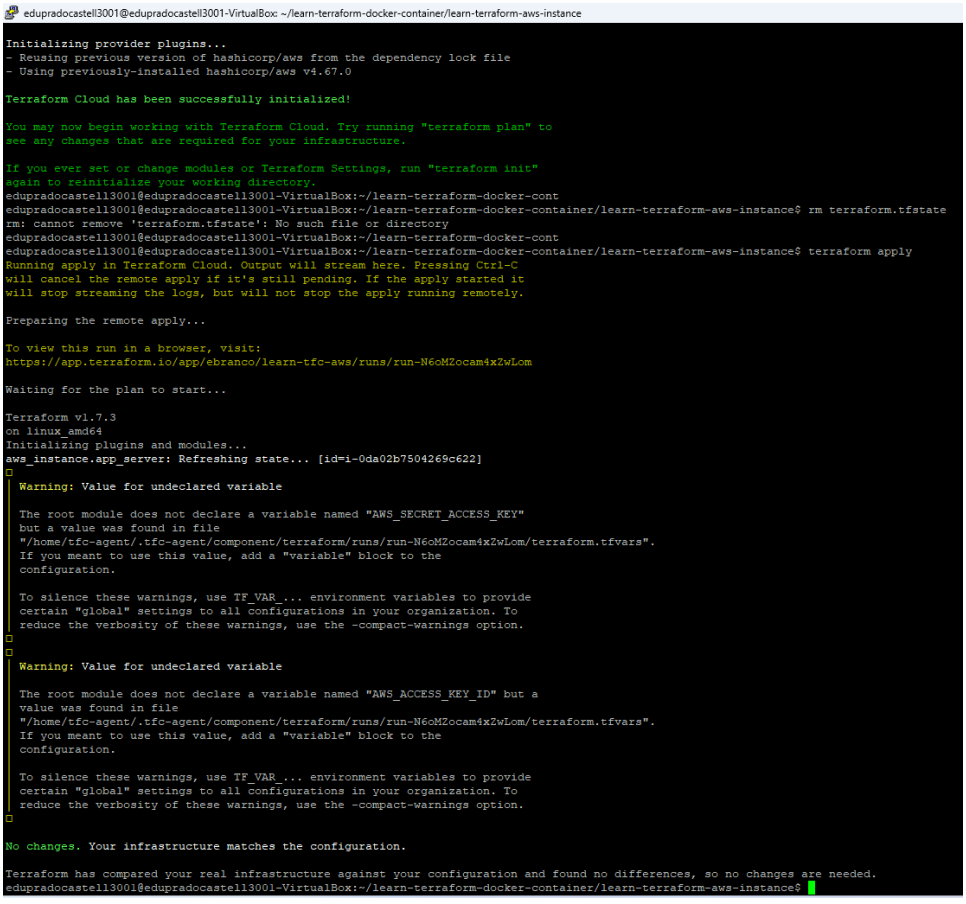


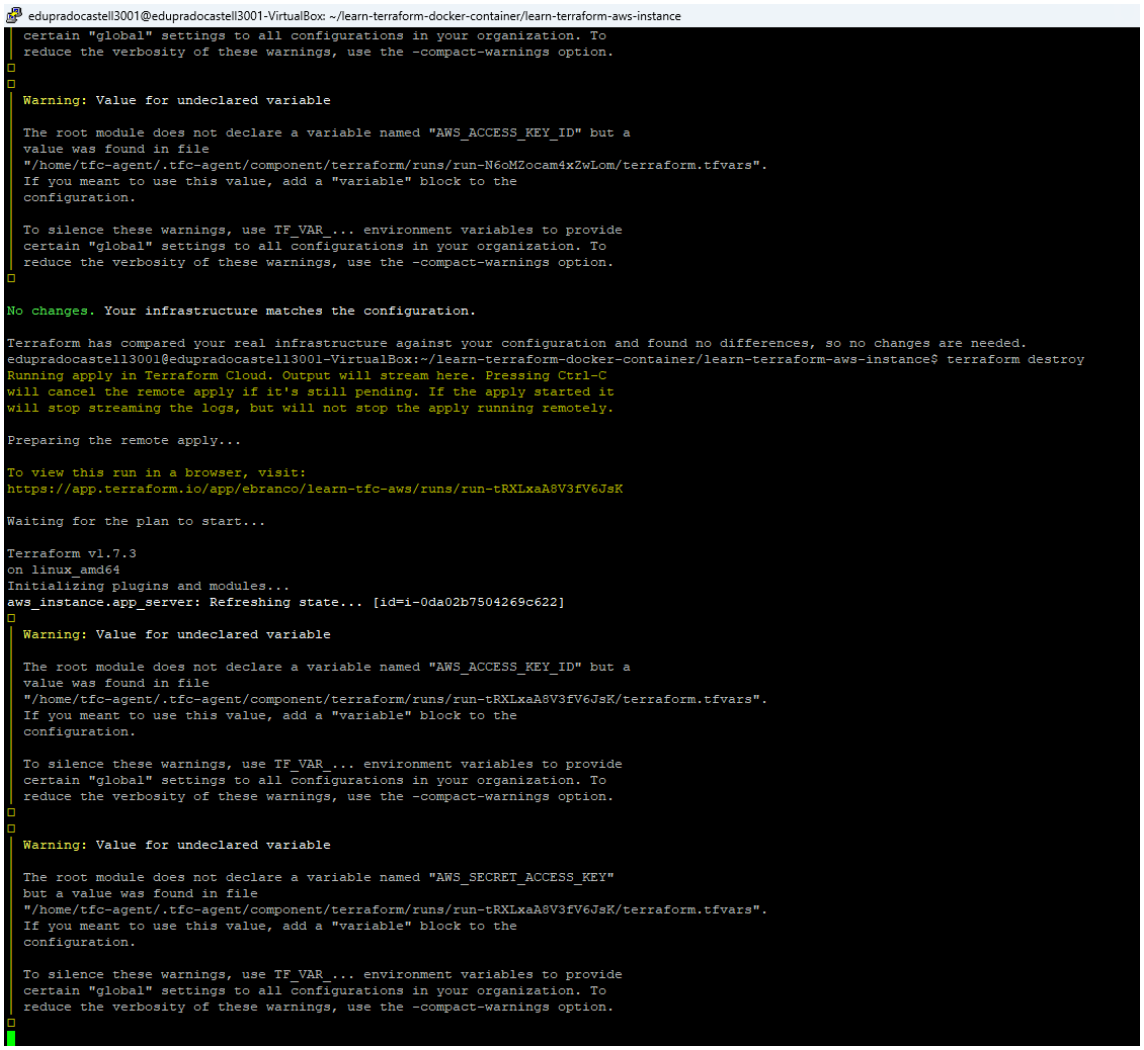
Step to complete access;

Configure aws key/secret;



Moving to cloud.

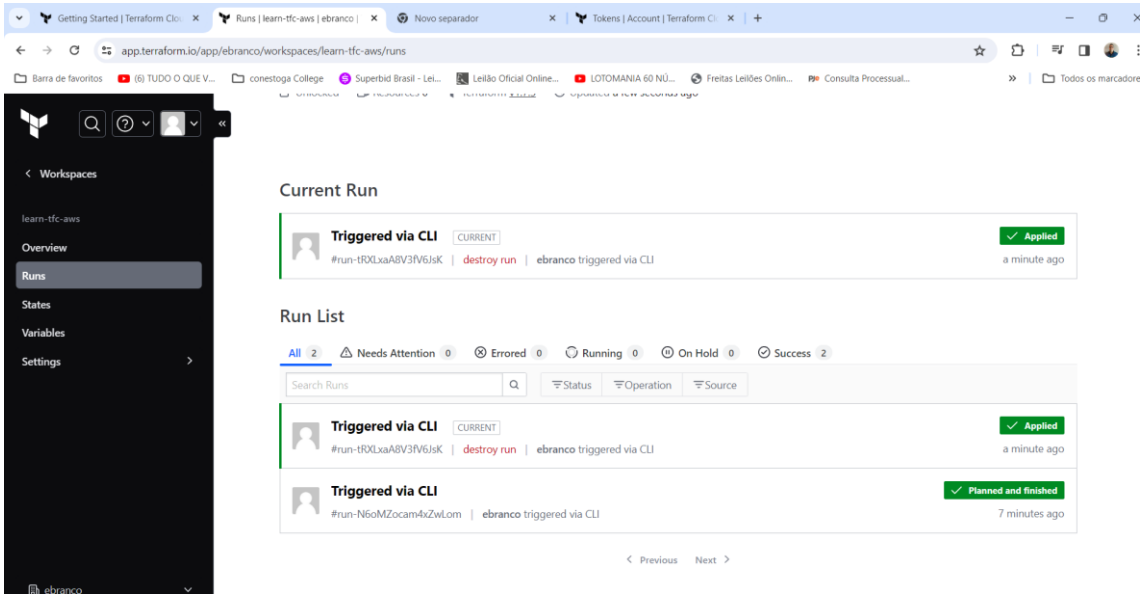


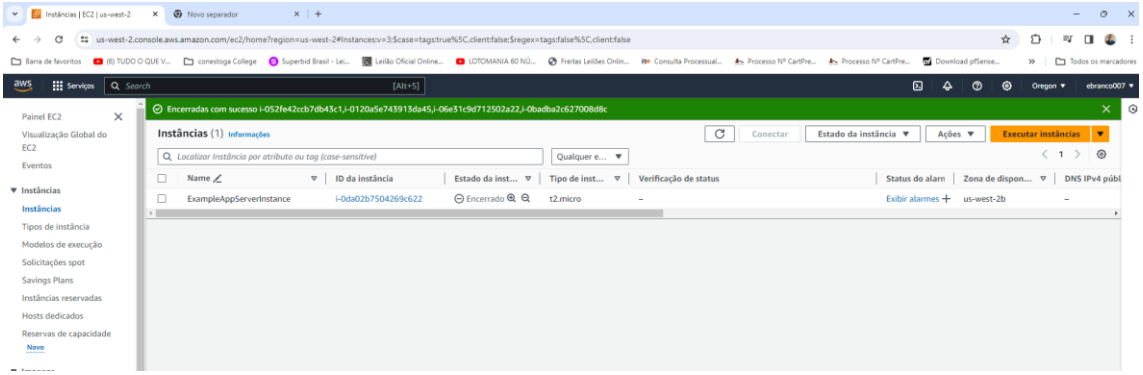

edupradocastell3001@edupradocastell3001-VirtualBox: ~/learn-terraform-docker-container/learn-terraform-aws-instance
certain "global" settings to all configurations in your organization. To
reduce the verbosity of these warnings, use the -compact-warnings option.
Warning: Value for undeclared variable
The root module does not declare a variable named "AWS_ACCESS_KEY_ID" but a
value was found in file
"/home/tfc-agent/.tfc-agent/component/terraform/runs/run-N6oMZocam4xZwLom/terraform.tfvars".
If you meant to use this value, add a "variable" block to the
configuration.
To silence these warnings, use TF_VAR_... environment variables to provide
certain "global" settings to all configurations in your organization. To
reduce the verbosity of these warnings, use the -compact-warnings option.
No changes. Your infrastructure matches the configuration.
Terraform has compared your real infrastructure against your configuration and found no differences, so no changes are needed.
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ terraform destroy
Running apply in Terraform Cloud. Output will stream here. Pressing Ctrl-C
will cancel the remote apply if it's still pending. If the apply started it
will stop streaming the logs, but will not stop the apply running remotely.
Preparing the remote apply...
To view this run in a browser, visit:
https://app.terraform.io/app/ebranco/learn-tfc-aws/runs/run-tRXLxaA8V3fV6JsK
Waiting for the plan to start...
Terraform v1.7.3
on linux_amd64
Initializing plugins and modules...
aws_instance.app_server: Refreshing state... [id=i-0da02b7504269c622]
Warning: Value for undeclared variable
The root module does not declare a variable named "AWS_ACCESS_KEY_ID" but a
value was found in file
"/home/tfc-agent/.tfc-agent/component/terraform/runs/run-tRXLxaA8V3fV6JsK/terraform.tfvars".
If you meant to use this value, add a "variable" block to the
configuration.
To silence these warnings, use TF_VAR_... environment variables to provide
certain "global" settings to all configurations in your organization. To
reduce the verbosity of these warnings, use the -compact-warnings option.
Warning: Value for undeclared variable
The root module does not declare a variable named "AWS_SECRET_ACCESS_KEY"
but a value was found in file
"/home/tfc-agent/.tfc-agent/component/terraform/runs/run-tRXLxaA8V3fV6JsK/terraform.tfvars".
If you meant to use this value, add a "variable" block to the
configuration.
To silence these warnings, use TF_VAR_... environment variables to provide
certain "global" settings to all configurations in your organization. To
reduce the verbosity of these warnings, use the -compact-warnings option.

```
edupradocastell3001@edupradocastell3001-VirtualBox: ~/learn-terraform-docker-container/learn-terraform-aws-instance
          - threads_per_core = 1 -> null
        }

      - credit_specification {
          - cpu_credits = "standard" -> null
        }

      - enclave_options {
          - enabled = false -> null
        }

      - maintenance_options {
          - auto_recovery = "default" -> null
        }

      - metadata_options {
          - http_endpoint               = "enabled" -> null
          - http_put_response_hop_limit = 2 -> null
          - http_tokens                 = "required" -> null
          - instance_metadata_tags      = "disabled" -> null
        }

      - private_dns_name_options {
          - enable_resource_name_dns_a_record    = false -> null
          - enable_resource_name_dns_aaaa_record = false -> null
          - hostname_type                        = "ip-name" -> null
        }

      - root_block_device {
          - delete_on_termination = true -> null
          - device_name           = "/dev/xvda" -> null
          - encrypted             = false -> null
          - iops                  = 3000 -> null
          - tags                  = {} -> null
          - throughput            = 125 -> null
          - volume_id             = "vol-0d45d37afeb3374ce" -> null
          - volume_size           = 8 -> null
          - volume_type           = "gp3" -> null
        }
    }

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

Changes to Outputs:
  - instance_id        = "i-0da02b7504269c622" -> null
  - instance_public_ip = "34.219.17.17" -> null

Do you really want to destroy all resources in workspace "learn-tfc-aws"?
  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

aws_instance.app_server: Destroying... [id=i-0da02b7504269c622]
aws_instance.app_server: Still destroying... [10s elapsed]
aws_instance.app_server: Still destroying... [20s elapsed]
aws_instance.app_server: Still destroying... [30s elapsed]
aws_instance.app_server: Destruction complete after 30s

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

edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-cont
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$
```

So, this runs from cli, shown in the terraform cloud.

Amazing Lab.




Instâncias | EC2 | us-west-2
Novo separador
us-west-2.console.aws.amazon.com/ec2/home?region=us-west-2#Instances:v=3;$case=tags:true%5C,client:false;$regex=tags:false%5C,client:false
Serviços
Search
[Alt+S]
Oregon
ebranco007
Painel EC2
Visualização Global do EC2
Eventos
Instâncias
Instâncias
Tipos de instância
Modelos de execução
Solicitações spot
Savings Plans
Instâncias reservadas
Hosts dedicados
Reservas de capacidade
Novo
Encerradas com sucesso i-052fe42ccb7dbd43c1,i-0120a5e743915da45,i-06e31c9d712502a22,i-0badba2c627008d8c
Instâncias (1) Informações
Conectar
Estado da instância
Ações
Executar instâncias
Localizar instância por atributo ou tag (case-sensitive)
Qualquer e...
Name
ID da instância
Estado da inst...
Tipo de inst...
Verificação de status
Status do alarm
Zona de dispon...
DNS IPv4 públ
ExampleAppServerInstance
i-0da02b7504269c622
Encerrado
t2.micro
–
Exibir alarmes
us-west-2b
–