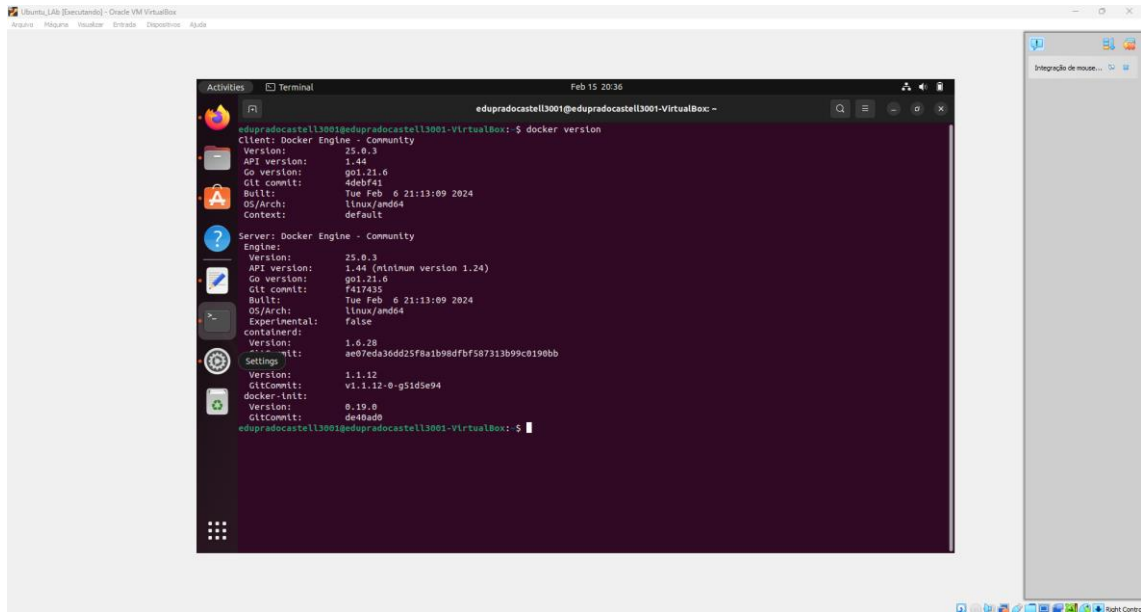
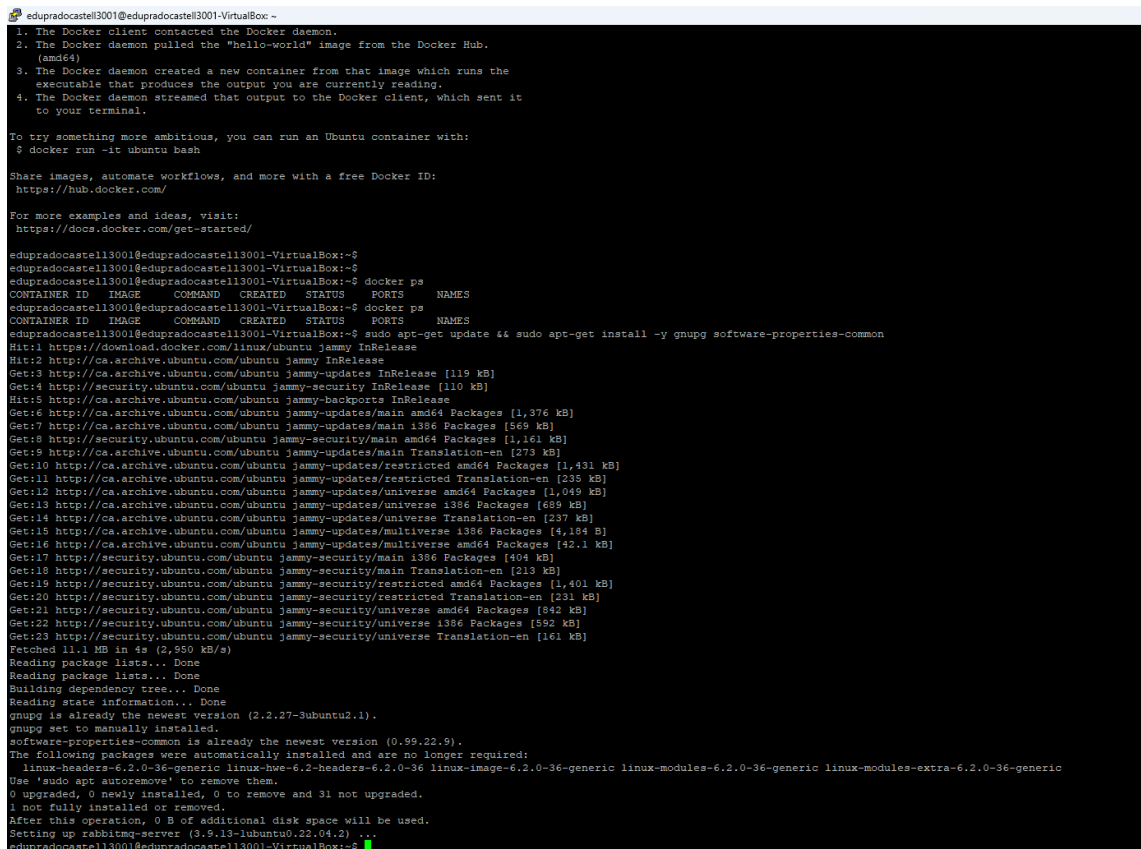


Terraform

Ubuntu 22.04 Run my VirtualBox.



Frist steps for installation Terraform.



Run command ->> Terraform init

```
edupradocastell3001@edupradocastell3001-VirtualBox: ~/learn-terraform-docker-container
configuration, instructs Terraform to generate HCL
for any imported resources not already present. The
configuration is written to a new file at PATH,
which must not already exist. Terraform may still
attempt to write configuration if the plan errors.

-input=true          Ask for input for variables if not directly set.
-lock=false         Don't hold a state lock during the operation. This
                   is dangerous if others might concurrently run
                   commands against the same workspace.
-lock-timeout=0s    Duration to retry a state lock.
-no-color           If specified, output won't contain any color.
-out-path           Write a plan file to the given path. This can be
                   used as input to the "apply" command.
-parallelism=n      Limit the number of concurrent operations. Defaults
                   to 10.
-state=statefile    A legacy option used for the local backend only.
                   See the local backend's documentation for more
                   information.
edupradocastell3001@edupradocastell3001-VirtualBox:~$ touch ~/.bashrc
edupradocastell3001@edupradocastell3001-VirtualBox:~$ terraform -install-autocomplete
edupradocastell3001@edupradocastell3001-VirtualBox:~$ mkdir learn-terraform-docker-container
edupradocastell3001@edupradocastell3001-VirtualBox:~$ cd learn-terraform-docker-container
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container$ sudo vi main.tf
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container$ ls
main.tf
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding kreuzwerker/docker versions matching "~> 3.0.1"...
- Installing kreuzwerker/docker v3.0.2...
- Installed kreuzwerker/docker v3.0.2 (self-signed, key ID BD080C4571C6104C)

Partner and community providers are signed by their developers.
If you'd like to know more about provider signing, you can read about it here:
https://www.terraform.io/docs/cli/plugins/signing.html

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$ terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create
```

Confirmations YES

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

Terraform will perform the following actions:

# 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
+   endpoint         = (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"
+   restart_policy   = 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: 
```

```
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
  + endpoint              = (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"
  }
}

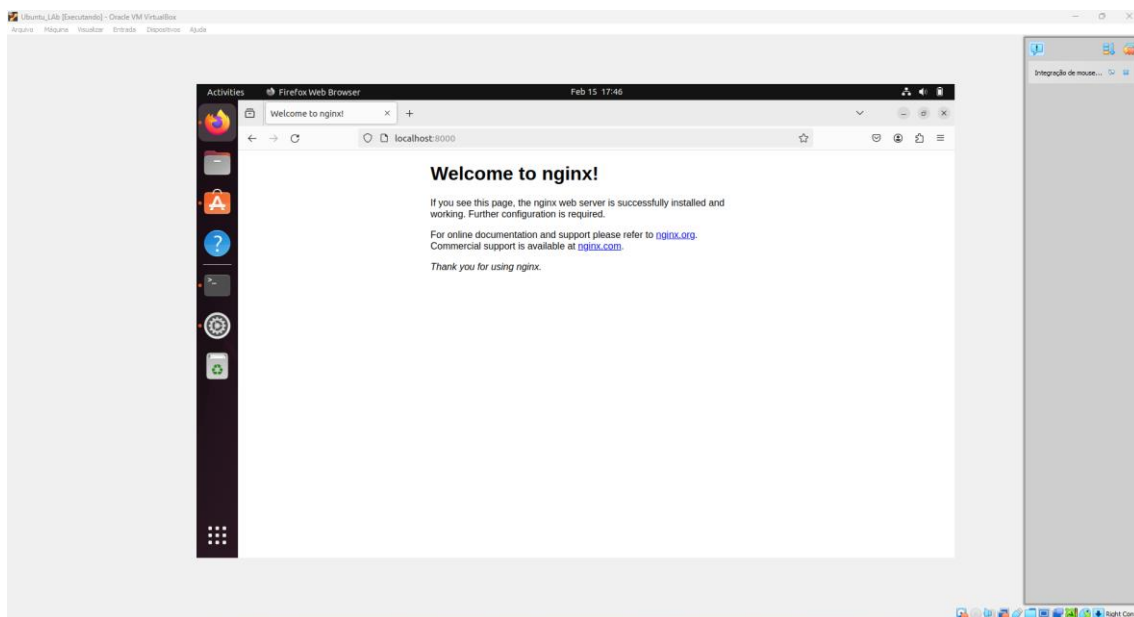
# docker_image.nginx will be created
+ resource "docker_image" "nginx" {
  + id            = (known after apply)
  + image_id      = (known after apply)
  + keep_locally = false
  + name          = "nginx"
  + repo_digest   = (known after apply)
}

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.

```
edupradocastell3001@edupradocastell3001-VirtualBox: ~ - learn-terraform-docker-container
- id = (known after apply)
- image = (known after apply)
- init = (known after apply)
- ipc_mode = (known after apply)
- log_driver = (known after apply)
- logging = false
- maintain_mode = 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)
- sysctl = false
- tmpfs = false
- 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...
docker_image.nginx: Still creating... [10s elapsed]
docker_image.nginx: Still creating... [20s elapsed]
docker_image.nginx: Creation complete after 21s [id=sha256:e4720093a3c1381245b53a5a51b417963b3c4472d3f47fc301930a4f3b17666e6a93a]
docker_container.nginx: Creating...
docker_container.nginx: Creation complete after 3s [id=4b13666e277e4720093a3c1381245b53a5a51b417963b3c4472d3f47fc301930a4f3b17666e6a93a]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
edupradocastell3001@edupradocastell3001-VirtualBox: ~ - learn-terraform-docker-container$ terraform destroy
CONTAINER ID        IMAGE               COMMAND              CREATED        STATUS        PORTS        NAMES
4b13666e277e4720093a3c1381245b53a5a51b417963b3c4472d3f47fc301930a4f3b17666e6a93a   nginx              "/docker-entrypoint..."   3 minutes ago   Up 3 minutes   0.0.0.0:8000->80/tcp   tutorial

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

Stop container

```
edupradocastell3001@edupradocastell3001-VirtualBox: ~ - learn-terraform-docker-container
- global_ipv6_address = ""
- global_ipv6_prefix_length = 0
- ip_address = "172.17.0.2"
- ip_prefix_length = 16
- ipv6_gateway = ""
- mac_address = "02:42:ac:11:00:02"
- network_name = "bridge"
}

- network_mode = "default" -> null
- privileged = false -> null
- publish_all_ports = false -> null
- read_only = false -> null
- remove_volumes = true -> null
- restart = "no" -> null
- rm = false -> null
- runtime = "runc" -> null
- security_opts = [] -> null
- shm_size = 64 -> null
- start = true -> null
- stdin_open = false -> null
- stop_signal = "SIGQUIT" -> null
- stop_timeout = 0 -> null
- storage_opts = {} -> null
- sysctls = {} -> null
- tmpfs = {} -> null
- tty = false -> null
- wait = false -> null
- wait_timeout = 60 -> null

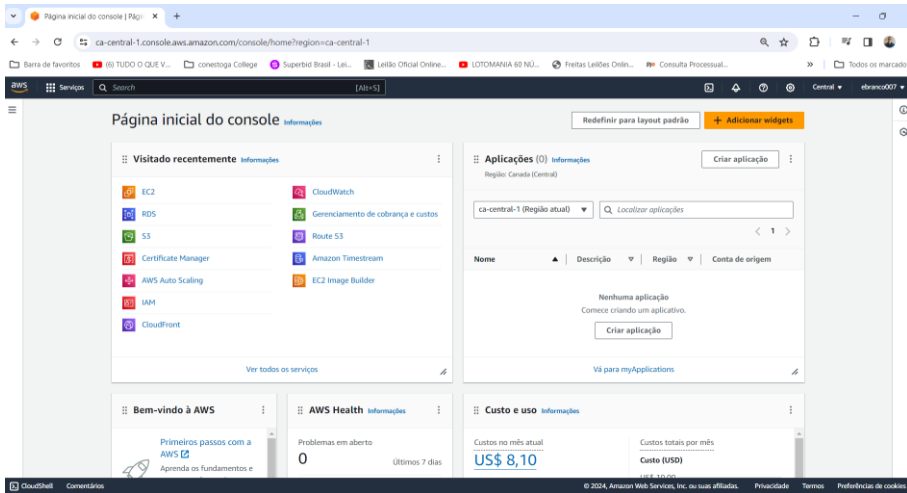
ports {
  - external = 8000 -> null
  - internal = 80 -> null
  - ip = "0.0.0.0" -> null
  - protocol = "tcp" -> null
}

# docker_image.nginx will be destroyed
- resource "docker_image" "nginx" {
  id = "sha256:e4720093a3c1381245b53a5a51b417963b3c4472d3f47fc301930a4f3b17666e6a93a" -> null
  image_id = "sha256:e4720093a3c1381245b53a5a51b417963b3c4472d3f47fc301930a4f3b17666e6a93a" -> null
  keep_locally = false -> null
  name = "nginx" -> null
  repo_digest = "nginx@sha256:25ff478171a2fd27d61a1774d97672bb7c13e88749fc70c711e207be34d370a" -> null
}

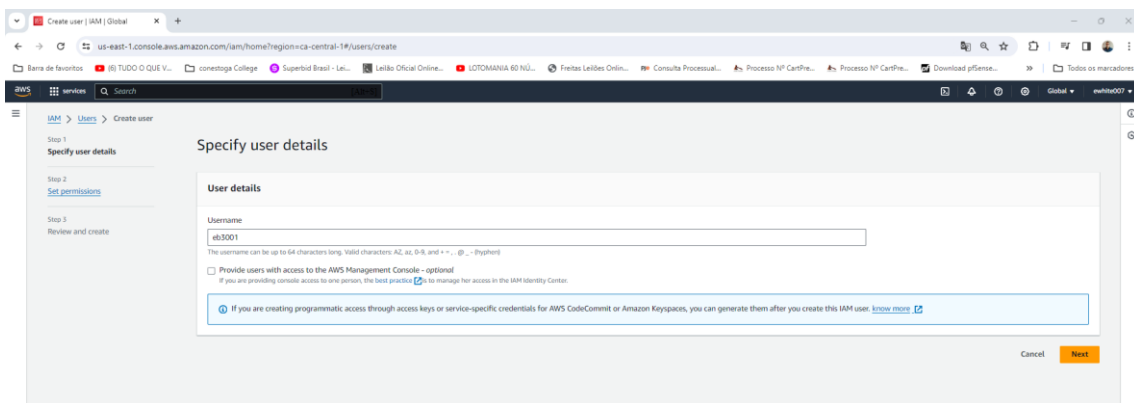
Plan: 0 to add, 0 to change, 2 to destroy.

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.

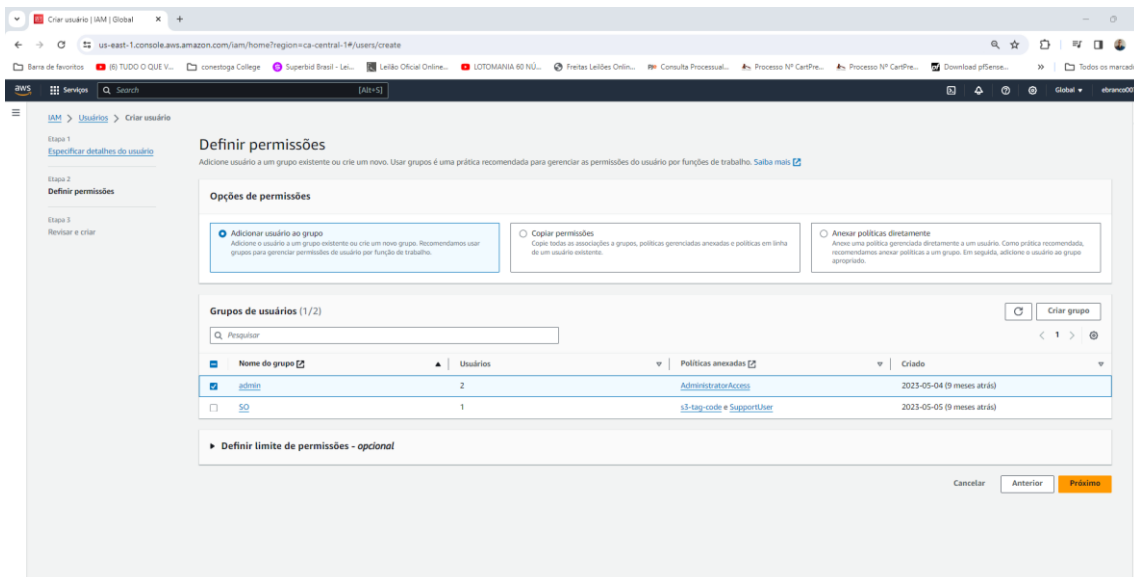
Enter a value: ^Z
[2]+  Stopped                  terraform destroy
edupradocastell3001@edupradocastell3001-VirtualBox: ~ - learn-terraform-docker-container$ docker ps
CONTAINER ID        IMAGE               COMMAND              CREATED        STATUS        PORTS        NAMES
4b13666e277e4720093a3c1381245b53a5a51b417963b3c4472d3f47fc301930a4f3b17666e6a93a   nginx              "/docker-entrypoint..."   3 minutes ago   Up 3 minutes   0.0.0.0:8000->80/tcp   tutorial
edupradocastell3001@edupradocastell3001-VirtualBox: ~ - learn-terraform-docker-container$ docker stop 4b13666e277e4720093a3c1381245b53a5a51b417963b3c4472d3f47fc301930a4f3b17666e6a93a
edupradocastell3001@edupradocastell3001-VirtualBox: ~ - learn-terraform-docker-container$ docker ps
CONTAINER ID        IMAGE               COMMAND              CREATED        STATUS        PORTS        NAMES
edupradocastell3001@edupradocastell3001-VirtualBox: ~ - learn-terraform-docker-container$
```



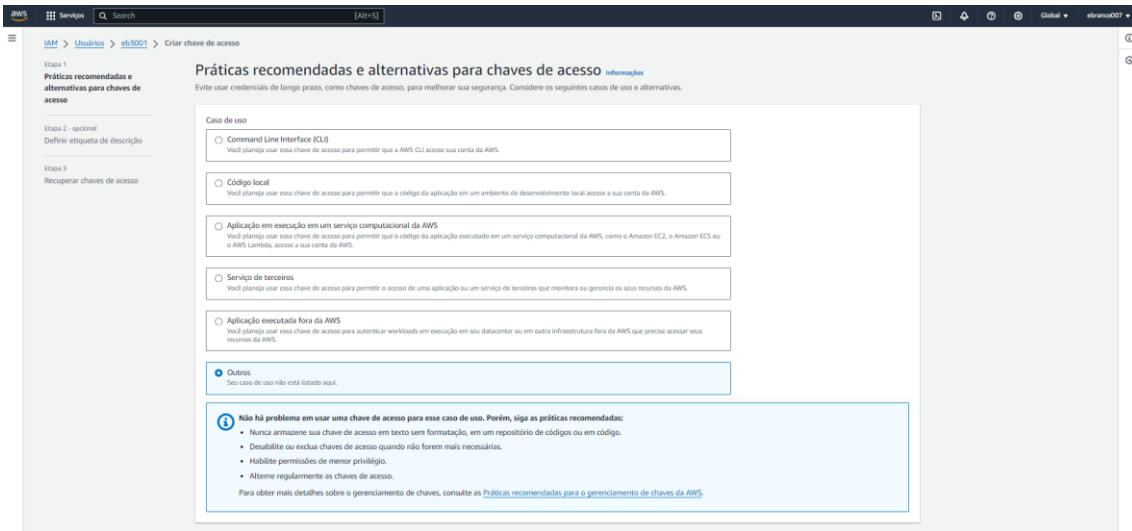
Create User EB3001 (it is my initials and last digit number of Conestoga)



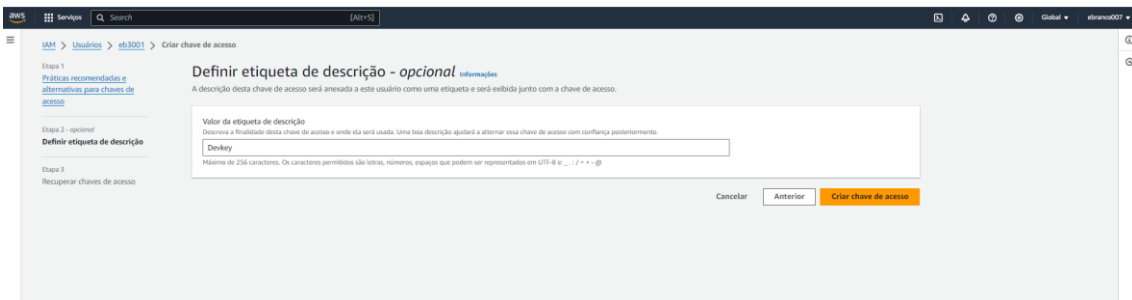
Give permissions



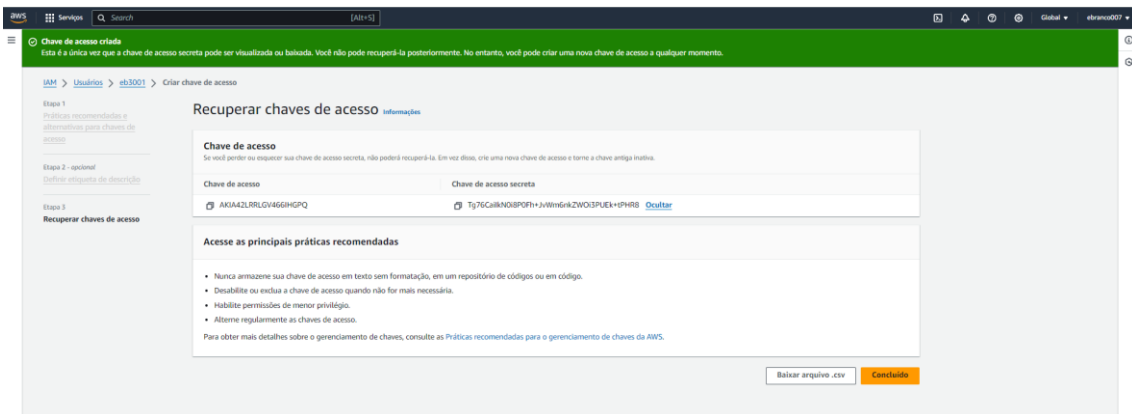
Another method for auth.



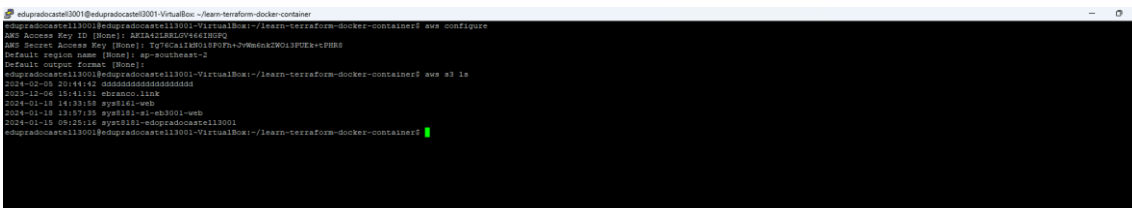
Create key



ACCESSkey and Secret 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]: AKIA42LRRLGV466IHGPO
AWS Secret Access Key [None]: Tg76CaIKN0i8P0Fn+JvWm6nkZW0i3PUEk+tPHR8
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 dddddd
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
[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 = "ap-southeast-2"
}

resource "aws_instance" "app_server" {
  ami           = "ami-830c9e33"
  instance_type = "t3.micro"
}

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

Terraform init

```
edupradocastell3001@edupradocastell3001-VirtualBox ~/learn-terraform-docker-container/learn-terraform-aws-instance
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ aws configure
AWS Access Key ID [None]: AKIA42LRRLGV466IHGPO
AWS Secret Access Key [None]: Tg76CaIKN0i8P0Fn+JvWm6nkZW0i3PUEk+tPHR8
Default region name [None]: ap-southeast-2
Default output format [None]:
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ aws s3 ls
2024-02-05 20:44:42 dddddd
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/learn-terraform-aws-instance$ export AWS_ACCESS_KEY_ID=
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ export AWS_SECRET_ACCESS_KEY=
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ mkdir learn-terraform-aws-instance
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ 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
[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$ tee
telnet 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]: AKIA42LRRLGV466IHGFQ
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$ tee
tee telnet 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

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

Terraform will perform the following actions:

# aws_instance.app_server will be created
+ resource "aws_instance" "app_server" {
+   ami                    = "ami-830c94e3"
+   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)
+   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          = "c2.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...
```

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.

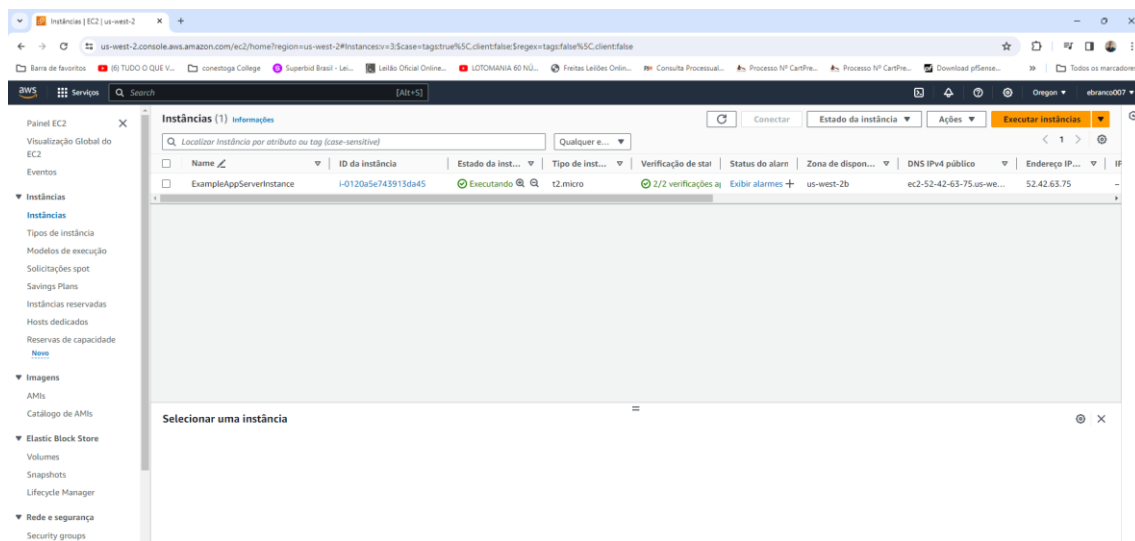
Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  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

```
edupradocastell3001@edupradocastell3001-VirtualBox: ~/learn-terraform-docker-container/learn-terraform-aws-instance
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ terraform show
# aws_instance.app_server:
resource "aws_instance" "app_server" {
  ami                    = "ami-830c94e3"
  arn                    = "arn:aws:ec2:us-west-2:801243281835:instance/i-0120a5e743913da45"
  associate_public_ip_address = true
  availability_zone      = "us-west-2b"
  cpu_core_count         = 1
  cpu_threads_per_core   = 1
  disable_api_stop       = false
  disable_api_termination = false
  ebs_optimized          = false
  get_password_data      = false
  hibernation            = false
  id                     = "i-0120a5e743913da45"
  instance_initiated_shutdown_behavior = "stop"
  instance_state         = "running"
  instance_type          = "t2.micro"
  ipv6_address_count     = 0
  ipv6_addresses        = []
  monitoring             = false
  placement_partition_number = 0
  primary_network_interface_id = "eni-06b592acb65219037"
  private_dns            = "ip-172-31-31-108.us-west-2.compute.internal"
  private_ip             = "172.31.31.108"
  public_dns              = "ec2-52-42-63-75.us-west-2.compute.amazonaws.com"
  public_ip              = "52.42.63.75"
  secondary_private_ips  = []
  security_groups        = [
    "default",
  ]
  source_dest_check      = true
  subnet_id              = "subnet-0134381c1ef6d63d"
  tags                   = {
    "Name" = "ExampleAppServerInstance"
  }
  tags_all               = {
    "Name" = "ExampleAppServerInstance"
  }
  tenancy                 = "default"
  user_data_replace_on_change = false
  vpc_security_group_ids = [
    "sg-081a10c7894809d5b",
  ]
  capacity_reservation_specification {
    capacity_reservation_preference = "open"
  }
  cpu_options {
    core_count       = 1
    threads_per_core = 1
  }
  credit_specification {
    cpu_credits = "standard"
  }
  enclave_options {
    enabled = false
  }
}
```

Details about instance;

```
edupradocastell3001@edupradocastell3001-VirtualBox: ~/learn-terraform-docker-container/learn-terraform-aws-instance
source_dest_check = true
subnet_id         = "subnet-0134381c1ef68d63d"
tags              = {
  "Name" = "ExampleAppServerInstance"
}
tags_all          = {
  "Name" = "ExampleAppServerInstance"
}
tenancy           = "default"
user_data_replace_on_change = false
vpc_security_group_ids = [
  "sg-081a10c7894809d5b",
]

capacity_reservation_specification {
  capacity_reservation_preference = "open"
}

cpu_options {
  core_count = 1
  threads_per_core = 1
}

credit_specification {
  cpu_credits = "standard"
}

enclave_options {
  enabled = false
}

maintenance_options {
  auto_recovery = "default"
}

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

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

root_block_device {
  delete_on_termination = true
  device_name = "/dev/sda1"
  encrypted = false
  iops = 0
  tags = {}
  throughput = 0
  volume_id = "vol-04b831c27d665e275"
  volume_size = 8
  volume_type = "standard"
}
}
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ terraform state list
aws_instance.app_server
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-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-04b931c27d665e275" -> null
  - volume_size = 8 -> null
  - volume_type = "standard" -> null
}

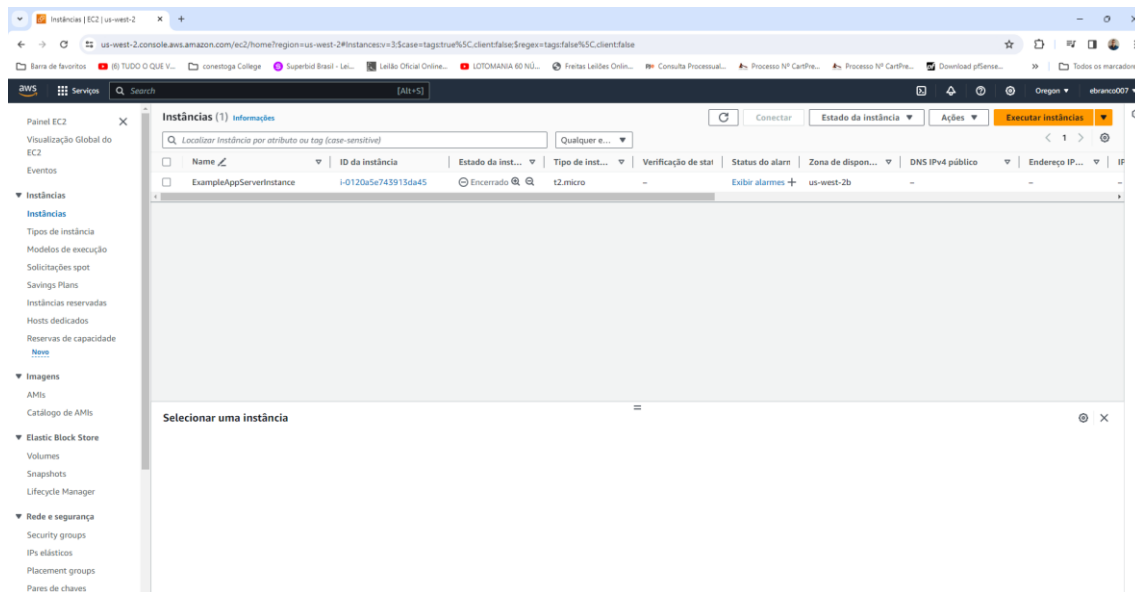
Plan: 1 to add, 0 to change, 1 to destroy.

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

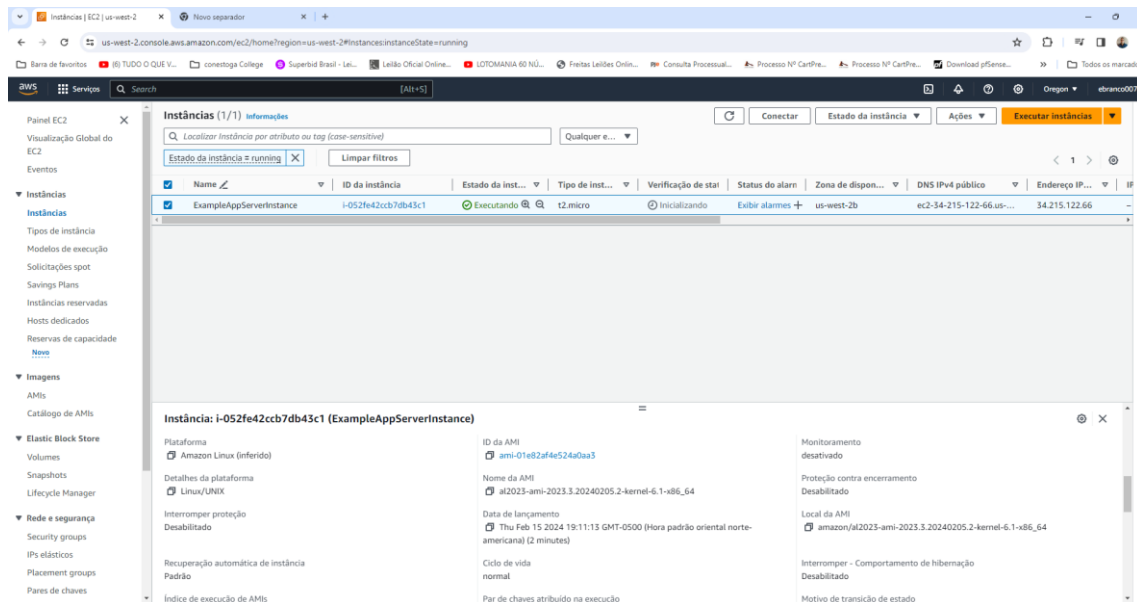


Now, after edit file, as can see change AMI. So, destroying and creating a new instance.

```
edupradocastel13001@edupradocastel13001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance
- 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_sockets = "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-0e7e5327b69205746" -> 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-06e31c9d712502a22]
aws_instance.app_server: Still destroying... [id=i-06e31c9d712502a22, 10s elapsed]
aws_instance.app_server: Still destroying... [id=i-06e31c9d712502a22, 20s elapsed]
aws_instance.app_server: Still destroying... [id=i-06e31c9d712502a22, 30s elapsed]
aws_instance.app_server: Destruction complete after 31s
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 39s [id=i-052fe42ccb7db43c1]
Apply complete! Resources: 1 added, 0 changed, 1 destroyed.
edupradocastel13001@edupradocastel13001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$
```

Show another AMI - CODE



Destroy instance

```
edupradocastell3001@edupradocastell3001-VirtualBox: ~/learn-terraform-docker-container/learn-terraform-aws-instance
- 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-0e7c5327b6920574e" -> 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-06e31c9d712502a22]
aws_instance.app_server: Still destroying... [id=i-06e31c9d712502a22, 10s elapsed]
aws_instance.app_server: Still destroying... [id=i-06e31c9d712502a22, 20s elapsed]
aws_instance.app_server: Still destroying... [id=i-06e31c9d712502a22, 30s elapsed]
aws_instance.app_server: Destruction complete after 31s
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-052fe42ccb7db43c1]
Apply complete! Resources: 1 added, 0 changed, 1 destroyed.
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ terraform destroy
```

Was destroyed

```
edupradocastell3001@edupradocastell3001-VirtualBox: ~/learn-terraform-docker-container/learn-terraform-aws-instance
- 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 = 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-0d0521d3940e2ef6" -> null
  - volume_size = 8 -> null
  - volume_type = "gp3" -> null
}
}
Plan: 0 to add, 0 to change, 1 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
aws_instance.app_server: Destroying... [id=i-052fe42ccb7db43c1]
aws_instance.app_server: Still destroying... [id=i-052fe42ccb7db43c1, 10s elapsed]
aws_instance.app_server: Still destroying... [id=i-052fe42ccb7db43c1, 20s elapsed]
aws_instance.app_server: Still destroying... [id=i-052fe42ccb7db43c1, 30s elapsed]
aws_instance.app_server: Still destroying... [id=i-052fe42ccb7db43c1, 40s elapsed]
aws_instance.app_server: Destruction complete after 41s
Destroy complete! Resources: 1 destroyed.
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$
```


Create file variables.tf

```
edupradocastell3001@edupradocastell3001-VirtualBox: ~/learn-terraform-docker-container/learn-terraform-aws-instance
variable "instance_name" {
  description = "Value of the Name tag for the EC2 instance"
  type        = string
  default     = "ExampleAppServerInstance"
}
"variables.tf" [New File]
```

Creating..

```
edupradocastell3001@edupradocastell3001-VirtualBox: ~/learn-terraform-docker-container/learn-terraform-aws-instance
+ 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: Creation complete after 33s [id=i-0da02b7504269c622]

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

Just confirmations Yes

```
edupradocastell3001@edupradocastell3001-VirtualBox: ~/learn-terraform-docker-container/learn-terraform-aws-instance
}
+ 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: Creation complete after 33s [id=i-0da02b7504269c622]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
edupradocastell3001@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.
edupradocastell3001@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
```

Just change Name

```
edupradocastell3001@edupradocastell3001-VirtualBox: ~/learn-terraform-docker-container/learn-terraform-aws-instance
+ 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: Creation complete after 33s [id=i-0da02b7504269c622]

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

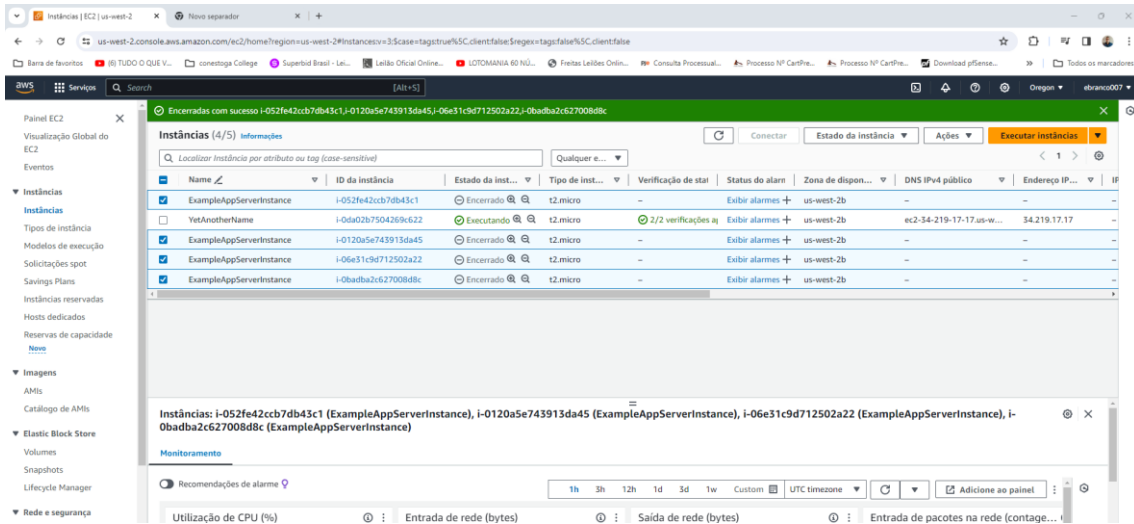
Plan: 0 to add, 1 to change, 0 to destroy.

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



Outputs

```

edupradocastell3001@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.
edupradocastell3001@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.
edupradocastell3001@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)

    # (0 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.
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$

```

Show instance Id / Ip-address

```
edupradocastell3001@edupradocastell3001-VirtualBox: ~/learn-terraform-docker-container/learn-terraform-aws-instance
# (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.
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$
```

```
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

Getting Started / Migrate State

Import local state

1 Set up Terraform Cloud 2 Next steps

Set up Terraform Cloud

1. Organization ebranco is ready. ✓
2. To migrate your state to Terraform Cloud, you can add this configuration block to any `.tf` file in the directory where you run Terraform. You can change the organization name and workspace name to your desired names.

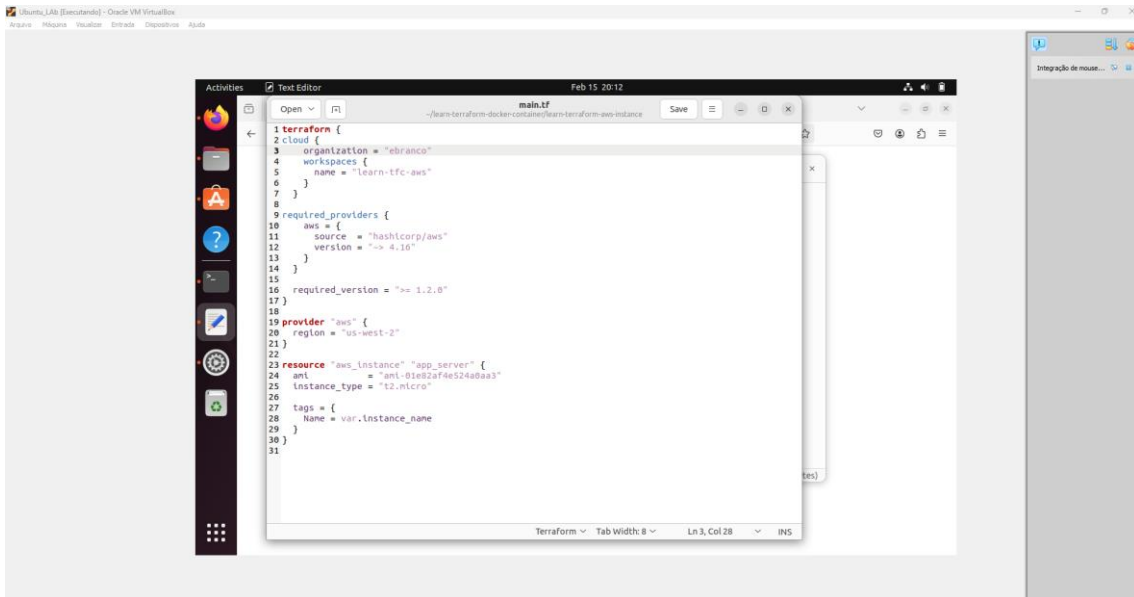
```
terraform { cloud { organization = "ebranco" workspaces { name = "example-workspace" } } }
```
3. Run the following command in your terminal and follow the prompts to fetch an API token for Terraform to use. If you don't have Terraform 0.13 or later, you'll need to install it first.

```
$ terraform login
```
4. To begin the migration, reinitialize so Terraform recognizes your cloud block configuration. When Terraform presents a prompt, enter yes if you want Terraform to migrate the state from your local machine to Terraform Cloud.

```
$ terraform init
```

[Continue](#) [Cancel](#)

Edit file with my account details



Step to complete access;



Configure aws key/secret;

Variables

Terraform uses all [Terraform](#) and [Environment](#) variables for all plans and applies in this workspace. Workspaces using Terraform 0.10.0 or later can also load default values from any `*.auto.tfvars` files in the configuration. You may want to use the Terraform Cloud Provider or the variables API to add multiple variables at once.

Sensitive variables

[Sensitive](#) variables are never shown in the UI or API, and can't be edited. They may appear in Terraform logs if your configuration is designed to output them. To change a sensitive variable, delete and replace it.

Workspace variables (2)

Variables defined within a workspace always overwrite variables from variable sets that have the same type and the same key. [Learn more about variable set precedence](#).

Key	Value	Category	
AWS_ACCESS_KEY_ID SENSITIVE	Sensitive - write only	terraform	...
AWS_SECRET_ACCESS_KEY SENSITIVE	Sensitive - write only	terraform	...

[+ Add variable](#)

Variable sets (0)

Moving to cloud.

```
edupradocastell3001@edupradocastell3001-VirtualBox ~:/learn-terraform-docker-container/learn-terraform-aws-instance
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v4.67.0

Terraform Cloud has been successfully initialized!

You may now begin working with Terraform Cloud. Try running "terraform plan" to
see any changes that are required for your infrastructure.

If you ever set or change modules or Terraform Settings, run "terraform init"
again to reinitialize your working directory.
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-cont
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-container/learn-terraform-aws-instance$ rm terraform.tfstate
rm: cannot remove 'terraform.tfstate': No such file or directory
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-cont
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-cont
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-cont
edupradocastell3001@edupradocastell3001-VirtualBox:~/learn-terraform-docker-cont$ terraform apply
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/edupradocastell3001/learn-terraform-aws/runs/run-N6cM2ocam4x2wLom

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_SECRET_ACCESS_KEY"
but a value was found in file
"/home/tfc-agent/.tfc-agent/component/terraform/runs/run-N6cM2ocam4x2wLom/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_ACCESS_KEY_ID" but a
value was found in file
"/home/tfc-agent/.tfc-agent/component/terraform/runs/run-N6cM2ocam4x2wLom/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$
```



```
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-N6cM2ocam4xZwLom/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/eb Branco/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

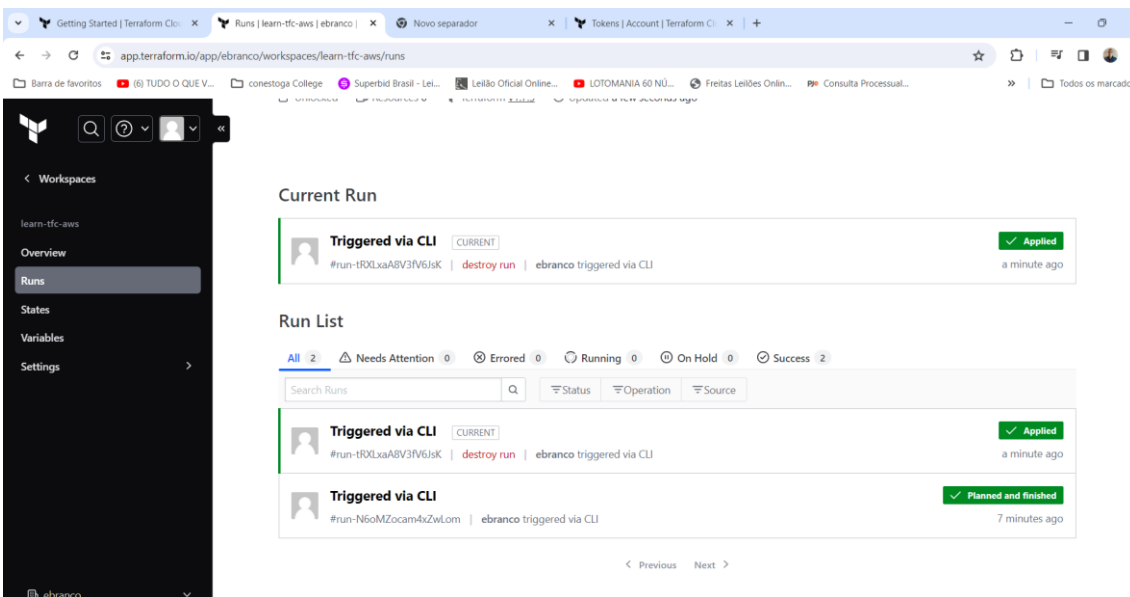
aws_instance.app_server: Destroying... [id=i-0da02b7504269c622]
aws_instance.app_server: Still destroying... [10s elapsed]
aws_instance.app_server: Still destroying... [20s elapsed]
aws_instance.app_server: Still destroying... [30s elapsed]
aws_instance.app_server: Destruction complete after 30s

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

us-west-2.console.aws.amazon.com/ec2/home?region=us-west-2#instances:vs=35&case=tag:true&f%5C.clientfalse&f%5C.tag:true&f%5C.clientfalse

Encerradas com sucesso i-052fe42cc7b043c1j-0120a56743913da45j-06e31c9d712502a22j-0ba0ba2c64700b8bc

Instâncias (1) Informações

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 IP v4 públ
ExampleAppServerInstance	i-0da02b7504269c622	Encerrado	t2.micro	-	-	us-west-2b	-

Encerradas com sucesso