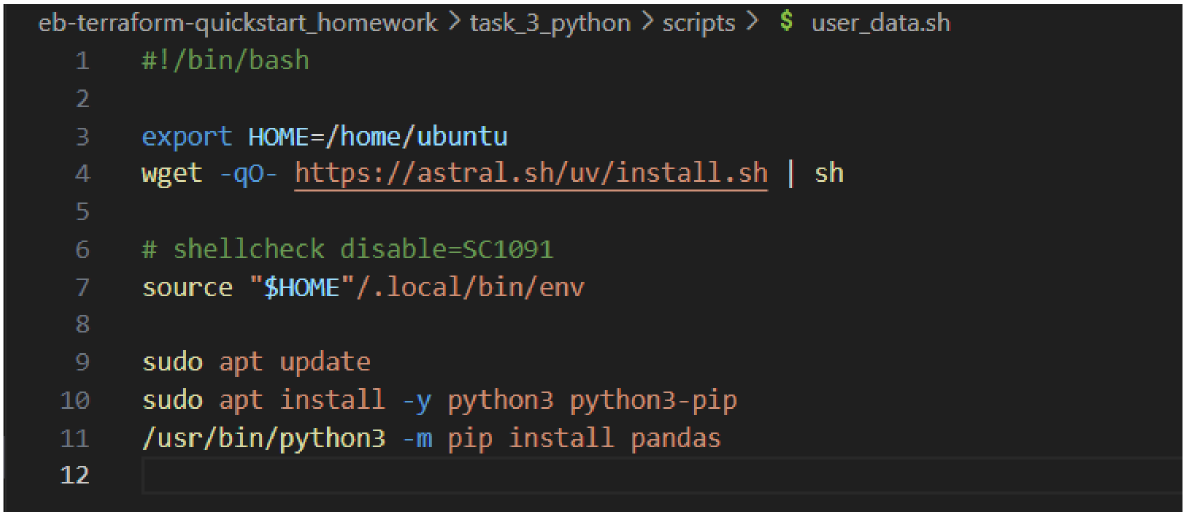
EVIDENCIAS LABORATORIO #3 TERRAFORM

1. Una instancia de EC2 que tenga Python y Pandas instalado.

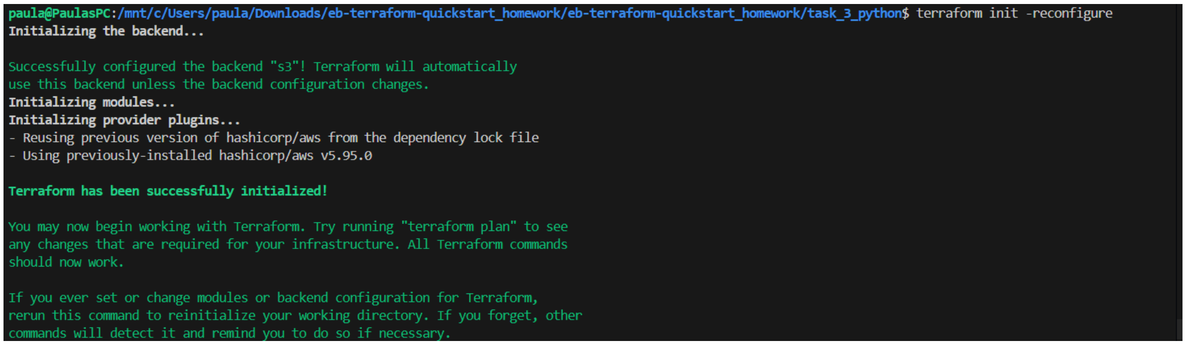
Configuración EC2:



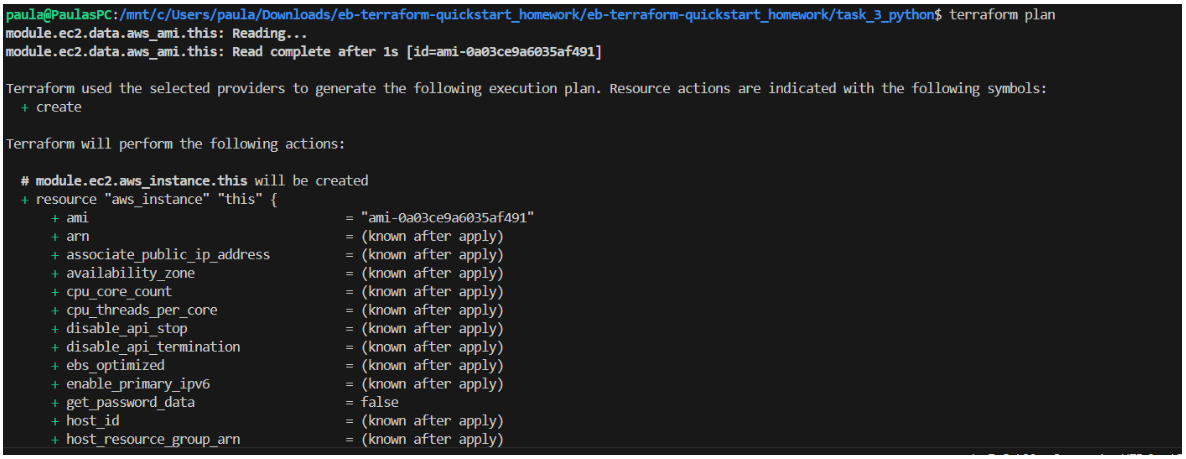
eb-terraform-quickstart_homework > task_3_python > scripts > $ user_data.sh

```
#!/bin/bash

export HOME=/home/ubuntu
wget -qO- https://astral.sh/uv/install.sh | sh

# shellcheck disable=SC1091
source "$HOME"/.local/bin/env

sudo apt update
sudo apt install -y python3 python3-pip
/usr/bin/python3 -m pip install pandas

```

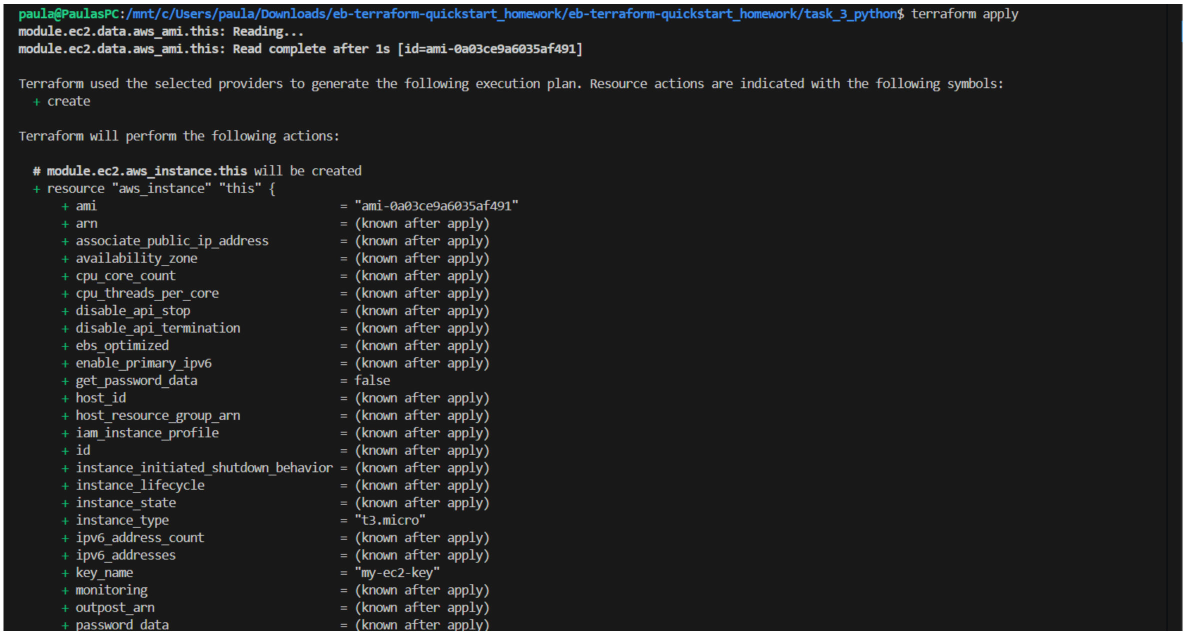
Consola:



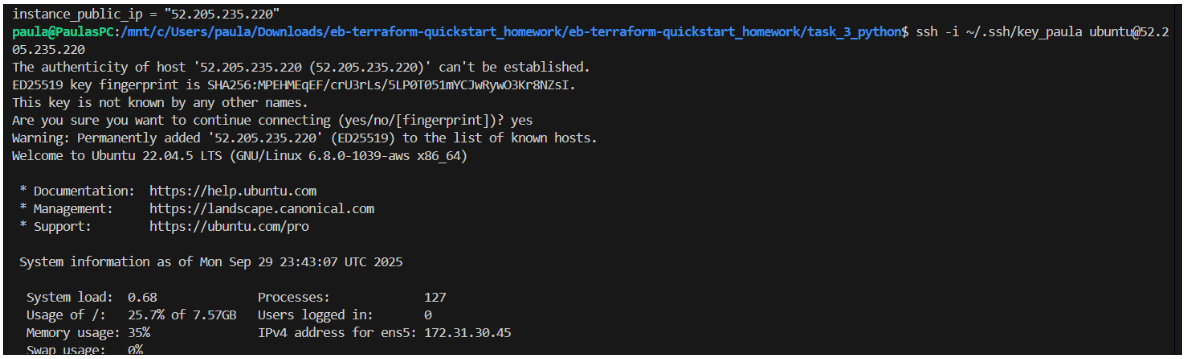
```
paula@PaulasPC:/mnt/c/Users/paula/Downloads/eb-terraform-quickstart_homework/eb-terraform-quickstart_homework/task_3_python$ terraform init -reconfigure
Initializing the backend...

Successfully configured the backend "s3"! Terraform will automatically
use this backend unless the backend configuration changes.
Initializing modules...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.95.0

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.
```



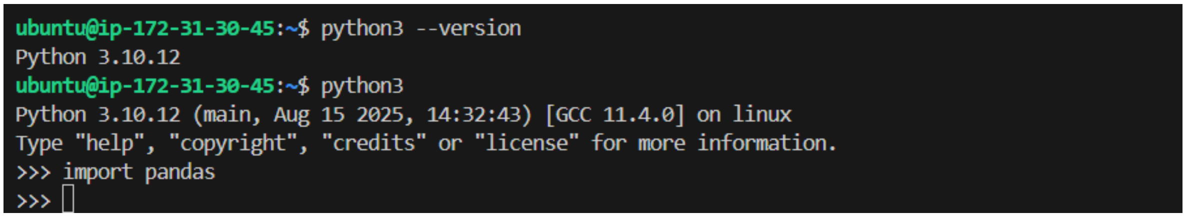
```
paula@PaulasPC:/mnt/c/Users/paula/Downloads/eb-terraform-quickstart_homework/eb-terraform-quickstart_homework/task_3_python$ terraform plan
module.ec2.data.aws_ami.this: Reading...
module.ec2.data.aws_ami.this: Read complete after 1s [id=ami-0a03ce9a6035af491]

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

Terraform will perform the following actions:

  # module.ec2.aws_instance.this will be created
  + resource "aws_instance" "this" {
      + ami                                  = "ami-0a03ce9a6035af491"
      + 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)
      + enable_primary_ipv6                  = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
```



```
paula@PaulasPC:/mnt/c/Users/paula/Downloads/eb-terraform-quickstart_homework/eb-terraform-quickstart_homework/task_3_python$ terraform apply
module.ec2.data.aws_ami.this: Reading...
module.ec2.data.aws_ami.this: Read complete after 1s [id=ami-0a03ce9a6035af491]

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

Terraform will perform the following actions:

  # module.ec2.aws_instance.this will be created
  + resource "aws_instance" "this" {
      + ami                                  = "ami-0a03ce9a6035af491"
      + 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)
      + enable_primary_ipv6                  = (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_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t3.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = "my-ec2-key"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
```



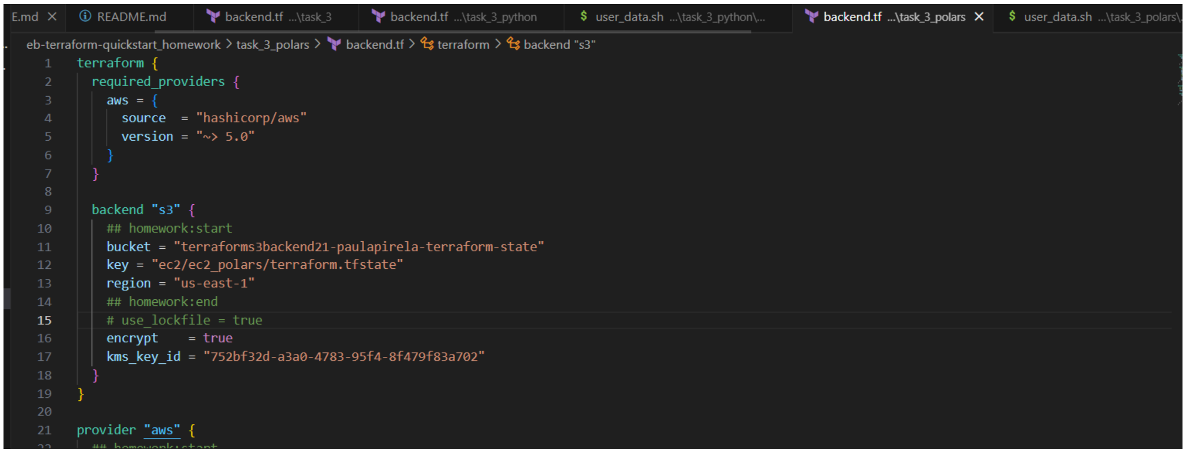
```
instance_public_ip = "52.205.235.220"
paula@PaulasPC:/mnt/c/Users/paula/Downloads/eb-terraform-quickstart_homework/eb-terraform-quickstart_homework/task_3_python$ ssh -i ~/.ssh/key_paula ubuntu@52.2
05.235.220
The authenticity of host '52.205.235.220 (52.205.235.220)' can't be established.
ED25519 key fingerprint is SHA256:MPEHMEqEF/crU3rLs/5LP0T051mYCJwRywO3Kr8NZsI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '52.205.235.220' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-1039-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Mon Sep 29 23:43:07 UTC 2025

  System load:  0.68               Processes:             127
  Usage of /:   25.7% of 7.57GB    Users logged in:       0
  Memory usage: 35%                IPv4 address for ens5: 172.31.30.45
  Swap usage:   0%
```



```
ubuntu@ip-172-31-30-45:~$ python3 --version
Python 3.10.12
ubuntu@ip-172-31-30-45:~$ python3
Python 3.10.12 (main, Aug 15 2025, 14:32:43) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import pandas
>>> 
```

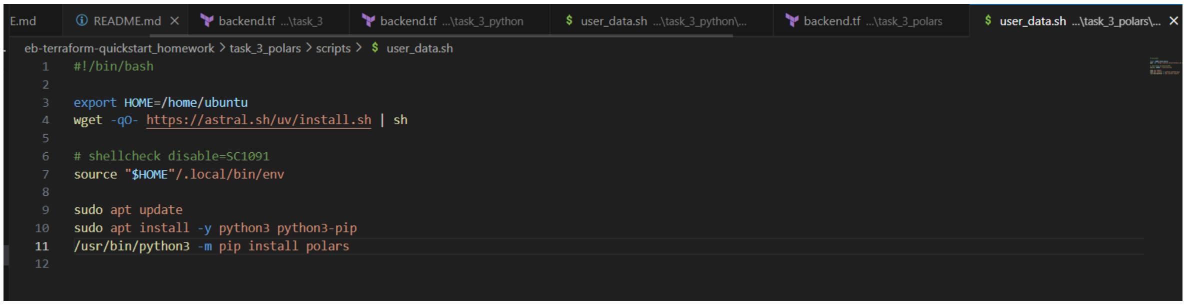
2. Una instancia de EC2 que tenga Python y Polars instalado.

Backend:


E.md
README.md
backend.tf ...\task_3
backend.tf ...\task_3_python
user_data.sh ...\task_3_python\...
backend.tf ...\task_3_polars
user_data.sh ...\task_3_polars\...
eb-terraform-quickstart_homework > task_3_polars > backend.tf > terraform > backend "s3"

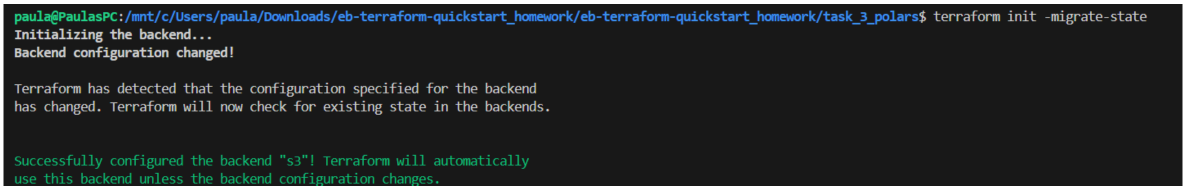
```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 5.0"
    }
  }

  backend "s3" {
    ## homework:start
    bucket = "terraforms3backend21-paulapirela-terraform-state"
    key = "ec2/ec2_polars/terraform.tfstate"
    region = "us-east-1"
    ## homework:end
    # use_lockfile = true
    encrypt     = true
    kms_key_id = "752bf32d-a3a0-4783-95f4-8f479f83a702"
  }
}

provider "aws" {
  ## homework:start
```

User_Data:

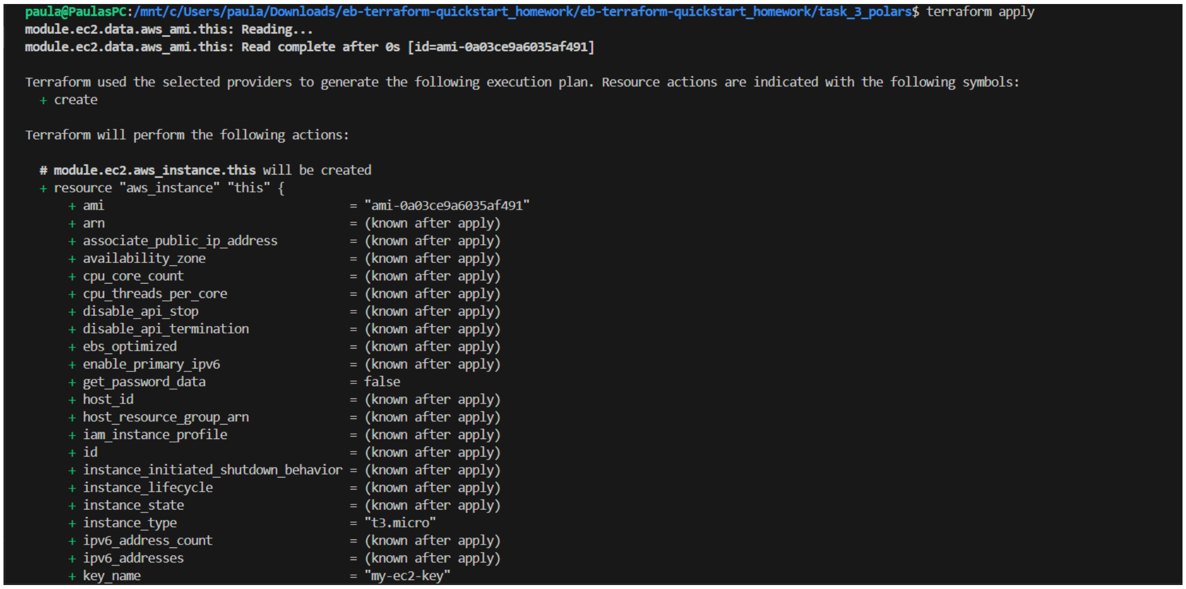

E.md
README.md
backend.tf ...\task_3
backend.tf ...\task_3_python
user_data.sh ...\task_3_python\...
backend.tf ...\task_3_polars
user_data.sh ...\task_3_polars\...
eb-terraform-quickstart_homework > task_3_polars > scripts > user_data.sh

```
#!/bin/bash

export HOME=/home/ubuntu
wget -qO- https://astral.sh/uv/install.sh | sh

# shellcheck disable=SC1091
source "$HOME"/.local/bin/env

sudo apt update
sudo apt install -y python3 python3-pip
/usr/bin/python3 -m pip install polars

```

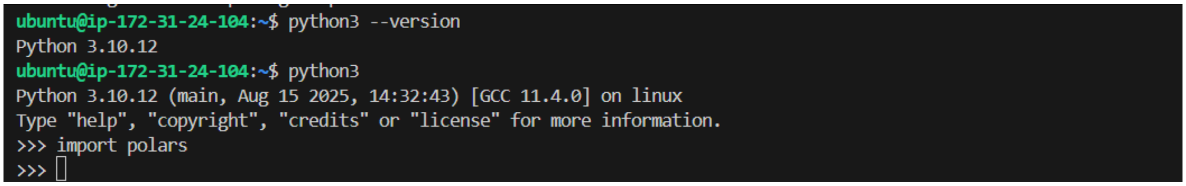
En consola:



```
paula@PaulasPC:/mnt/c/Users/paula/Downloads/eb-terraform-quickstart_homework/eb-terraform-quickstart_homework/task_3_polars$ terraform init -migrate-state
Initializing the backend...
Backend configuration changed!

Terraform has detected that the configuration specified for the backend
has changed. Terraform will now check for existing state in the backends.


Successfully configured the backend "s3"! Terraform will automatically
use this backend unless the backend configuration changes.
```



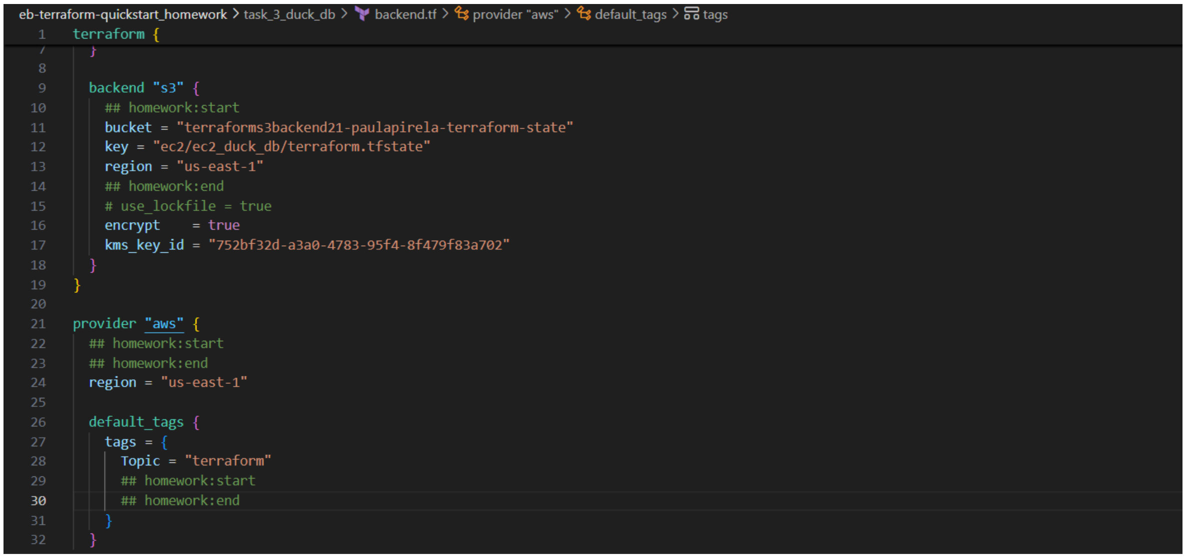
```
paula@PaulasPC:/mnt/c/Users/paula/Downloads/eb-terraform-quickstart_homework/eb-terraform-quickstart_homework/task_3_polars$ terraform apply
module.ec2.data.aws_ami.this: Reading...
module.ec2.data.aws_ami.this: Read complete after 0s [id=ami-0a03ce9a6035af491]

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

Terraform will perform the following actions:

  # module.ec2.aws_instance.this will be created
  + resource "aws_instance" "this" {
      + ami                                  = "ami-0a03ce9a6035af491"
      + 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)
      + enable_primary_ipv6                  = (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_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t3.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = "my-ec2-key"
```



```
ubuntu@ip-172-31-24-104:~$ python3 --version
Python 3.10.12
ubuntu@ip-172-31-24-104:~$ python3
Python 3.10.12 (main, Aug 15 2025, 14:32:43) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import polars
>>> 
```

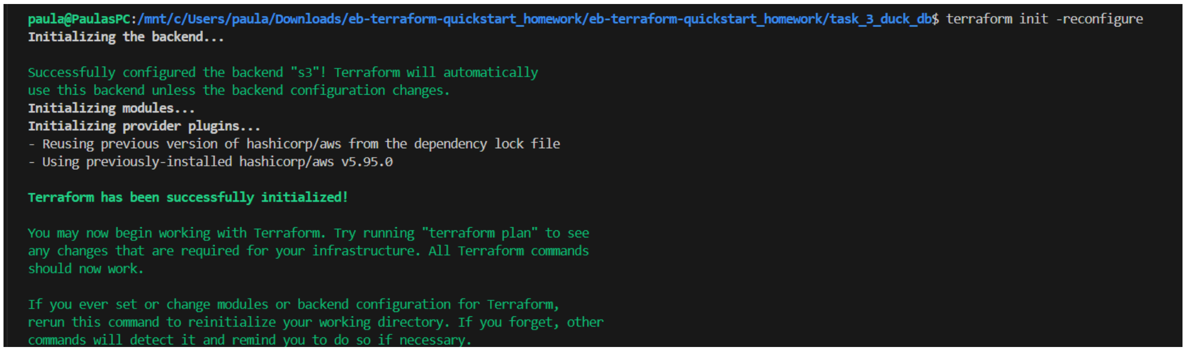
3. Una instancia de EC2 que tenga Python y DuckDB instalado.

Backend:

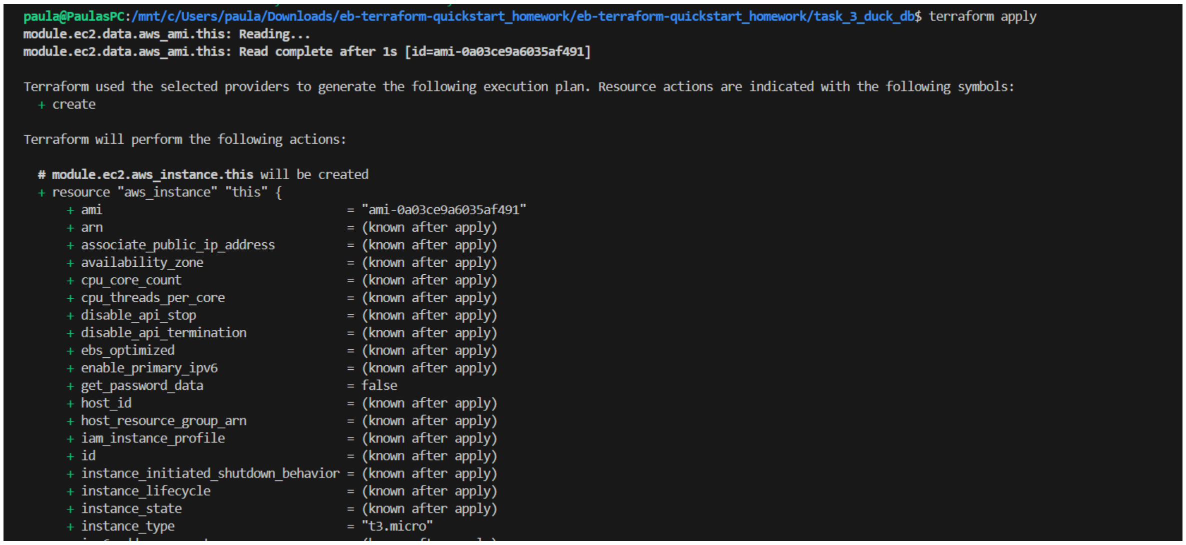

eb-terraform-quickstart_homework > task_3_duck_db > backend.tf > provider "aws" > default_tags > tags

```
terraform {
  }

  backend "s3" {
    ## homework:start
    bucket = "terraforms3backend21-paulapirela-terraform-state"
    key = "ec2/ec2_duck_db/terraform.tfstate"
    region = "us-east-1"
    ## homework:end
    # use_lockfile = true
    encrypt     = true
    kms_key_id = "752bf32d-a3a0-4783-95f4-8f479f83a702"
  }
}

provider "aws" {
  ## homework:start
  ## homework:end
  region = "us-east-1"

  default_tags {
    tags = {
      Topic = "terraform"
      ## homework:start
      ## homework:end
    }
  }
```

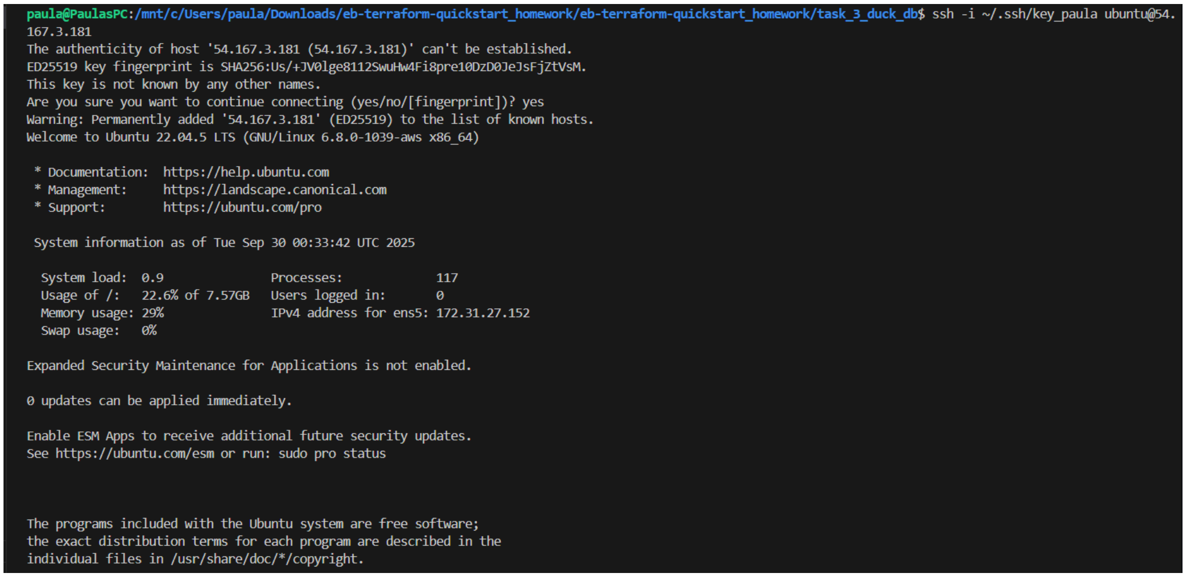
Consola:



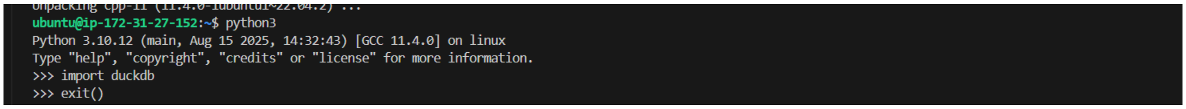
```
paula@PaulasPC:/mnt/c/Users/paula/Downloads/eb-terraform-quickstart_homework/eb-terraform-quickstart_homework/task_3_duck_db$ terraform init -reconfigure
Initializing the backend...

Successfully configured the backend "s3"! Terraform will automatically
use this backend unless the backend configuration changes.
Initializing modules...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.95.0

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.
```



```
paula@PaulasPC:/mnt/c/Users/paula/Downloads/eb-terraform-quickstart_homework/eb-terraform-quickstart_homework/task_3_duck_db$ terraform apply
module.ec2.data.aws_ami.this: Reading...
module.ec2.data.aws_ami.this: Read complete after 1s [id=ami-0a03ce9a6035af491]

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

Terraform will perform the following actions:

  # module.ec2.aws_instance.this will be created
  + resource "aws_instance" "this" {
      + ami                                  = "ami-0a03ce9a6035af491"
      + 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)
      + enable_primary_ipv6                  = (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_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t3.micro"
```



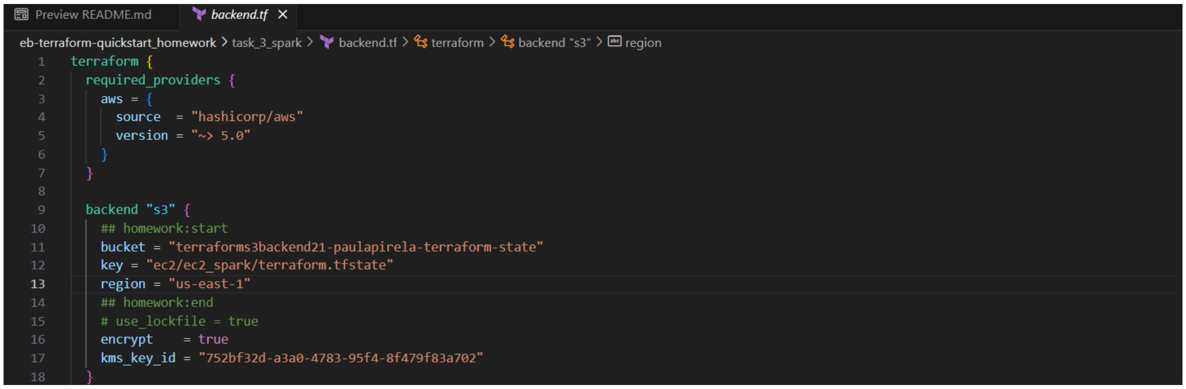
```
paula@PaulasPC:/mnt/c/Users/paula/Downloads/eb-terraform-quickstart_homework/eb-terraform-quickstart_homework/task_3_duck_db$ ssh -i ~/.ssh/key_paula ubuntu@54.
167.3.181
The authenticity of host '54.167.3.181 (54.167.3.181)' can't be established.
ED25519 key fingerprint is SHA256:Us/+JV0lge8112SwuHw4Fi8pre10DzD0JeJsFjZtVsM.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.167.3.181' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-1039-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Tue Sep 30 00:33:42 UTC 2025

  System load:  0.9               Processes:             117
  Usage of /:   22.6% of 7.57GB   Users logged in:       0
  Memory usage: 29%               IPv4 address for ens5: 172.31.27.152
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
```



```
Unpacking cpp-11 (11.4.0-1ubuntu1~22.04.2) ...
ubuntu@ip-172-31-27-152:~$ python3
Python 3.10.12 (main, Aug 15 2025, 14:32:43) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import duckdb
>>> exit()
```

4. Una instancia de EC2 que tenga Python y Spark instalado.

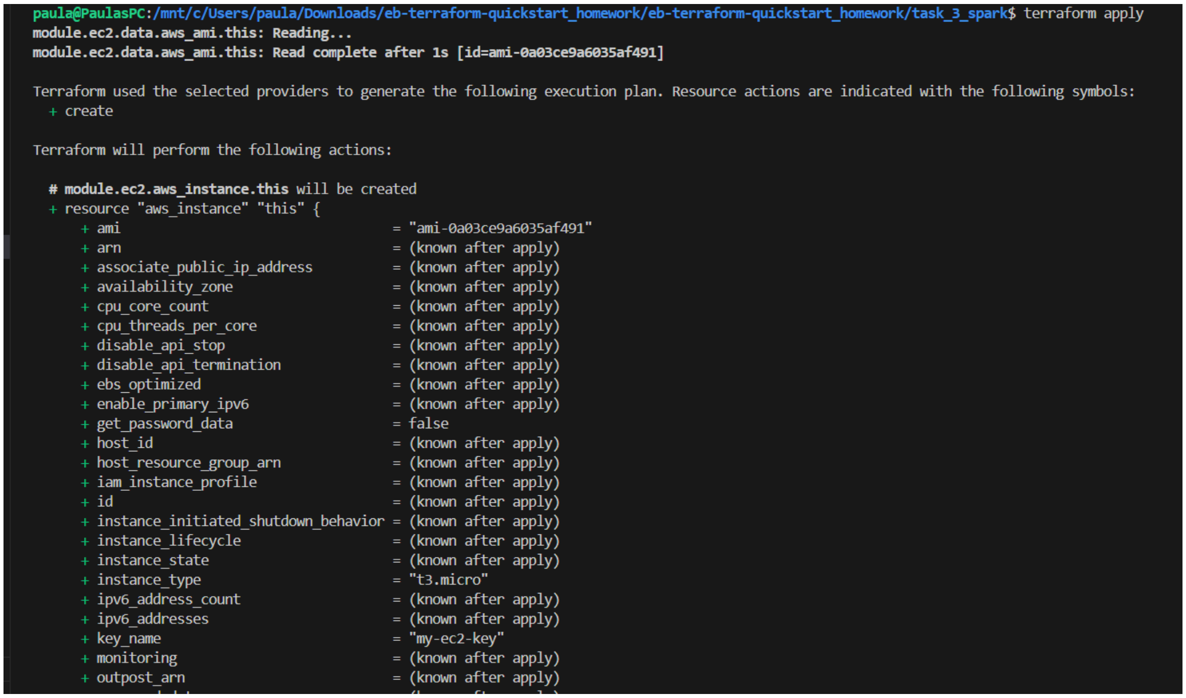
Backend:



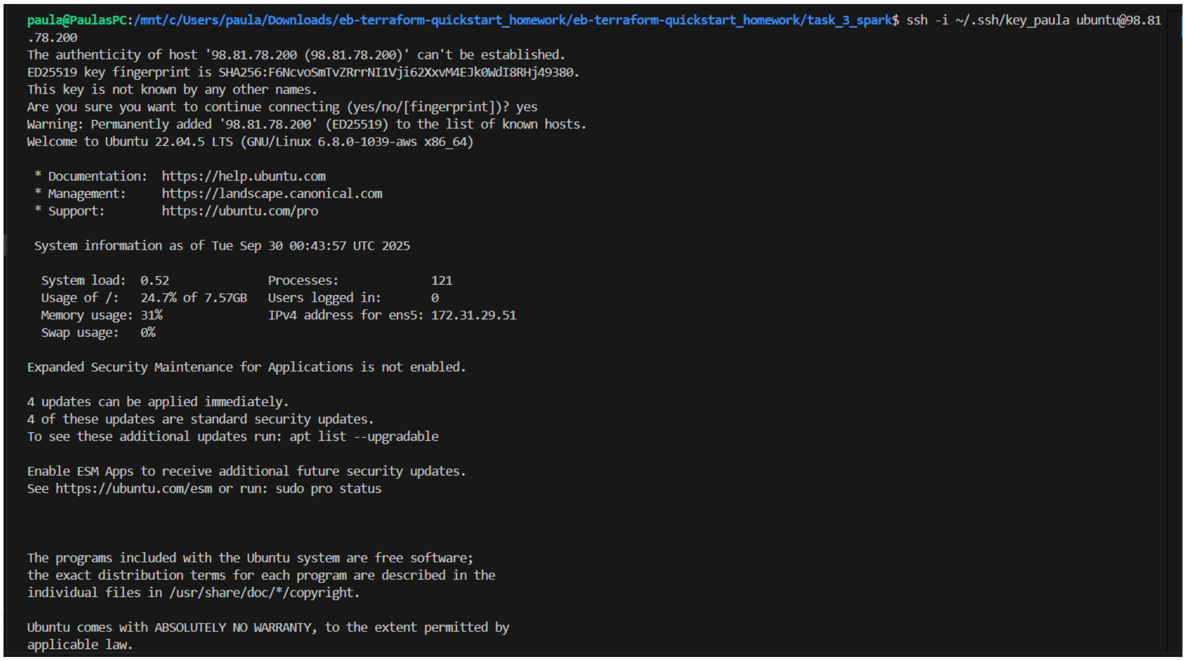
Preview README.md
backend.tf
eb-terraform-quickstart_homework > task_3_spark > backend.tf > terraform > backend "s3" > region

```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 5.0"
    }
  }

  backend "s3" {
    ## homework:start
    bucket = "terraforms3backend21-paulapirela-terraform-state"
    key = "ec2/ec2_spark/terraform.tfstate"
    region = "us-east-1"
    ## homework:end
    # use_lockfile = true
    encrypt     = true
    kms_key_id = "752bf32d-a3a0-4783-95f4-8f479f83a702"
  }
```

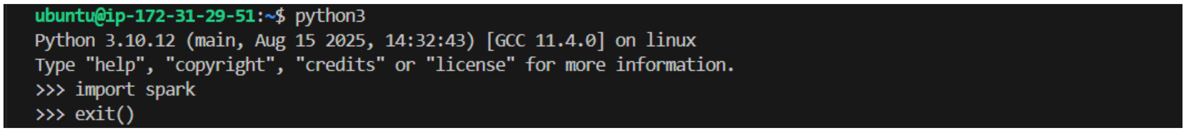
Consola:



```
paula@PaulasPC:/mnt/c/Users/paula/Downloads/eb-terraform-quickstart_homework/eb-terraform-quickstart_homework/task_3_spark$ terraform apply
module.ec2.data.aws_ami.this: Reading...
module.ec2.data.aws_ami.this: Read complete after 1s [id=ami-0a03ce9a6035af491]

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

Terraform will perform the following actions:

  # module.ec2.aws_instance.this will be created
  + resource "aws_instance" "this" {
      + ami                                  = "ami-0a03ce9a6035af491"
      + 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)
      + enable_primary_ipv6                  = (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_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t3.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = "my-ec2-key"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
```



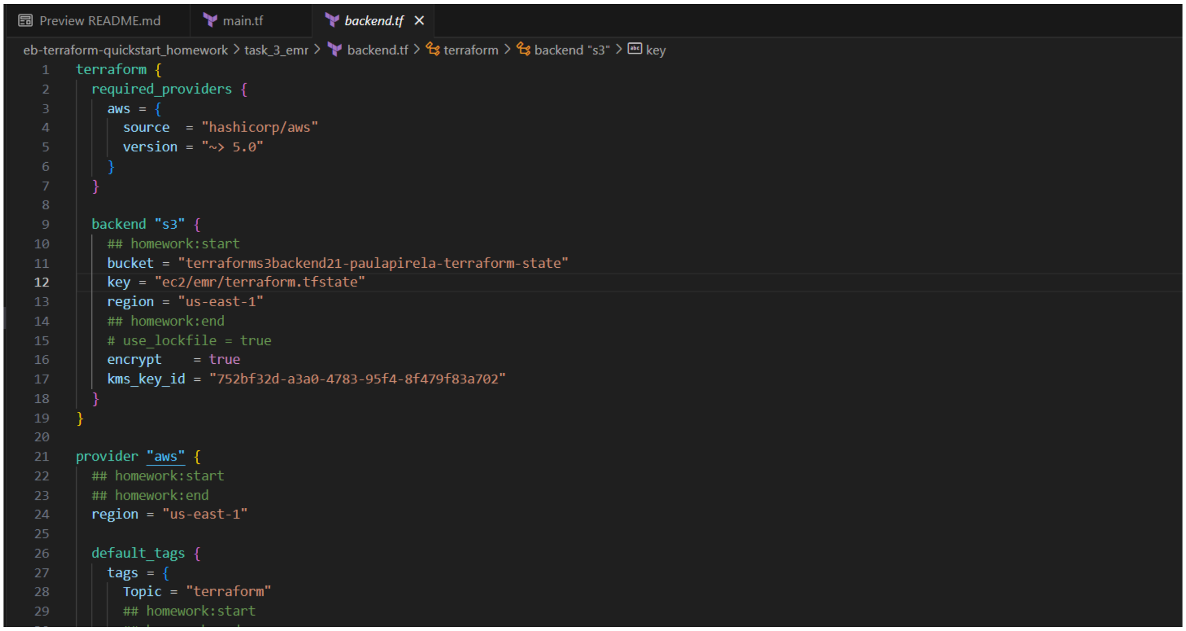
```
paula@PaulasPC:/mnt/c/Users/paula/Downloads/eb-terraform-quickstart_homework/eb-terraform-quickstart_homework/task_3_spark$ ssh -i ~/.ssh/key_paula ubuntu@98.81
.78.200
The authenticity of host '98.81.78.200 (98.81.78.200)' can't be established.
ED25519 key fingerprint is SHA256:F6NcvoSmTvZRrrNI1Vji62XxvM4EJk0WdI8RHj49380.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '98.81.78.200' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-1039-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Tue Sep 30 00:43:57 UTC 2025

  System load:  0.52              Processes:             121
  Usage of /:   24.7% of 7.57GB   Users logged in:       0
  Memory usage: 31%               IPv4 address for ens5: 172.31.29.51
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

4 updates can be applied immediately.
4 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```



```
ubuntu@ip-172-31-29-51:~$ python3
Python 3.10.12 (main, Aug 15 2025, 14:32:43) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import spark
>>> exit()
```

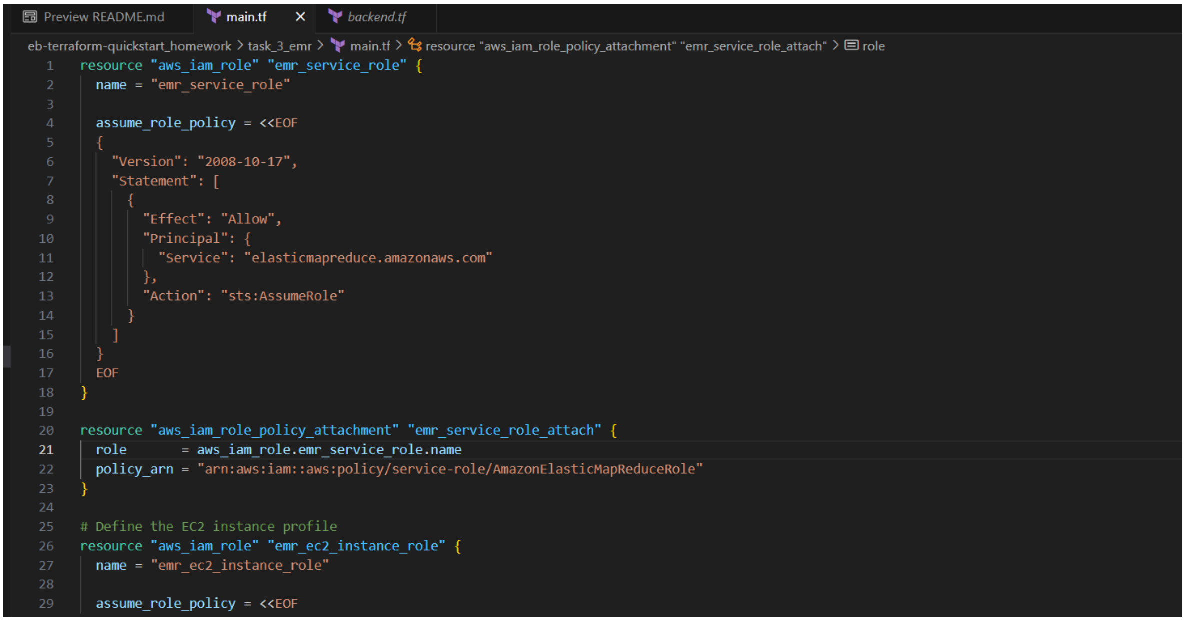
5. Una cluster de EMR en el que puedan correr Spark distribuido.

Backend:


Preview README.md
main.tf
backend.tf
eb-terraform-quickstart_homework > task_3_emr > backend.tf > terraform > backend "s3" > key

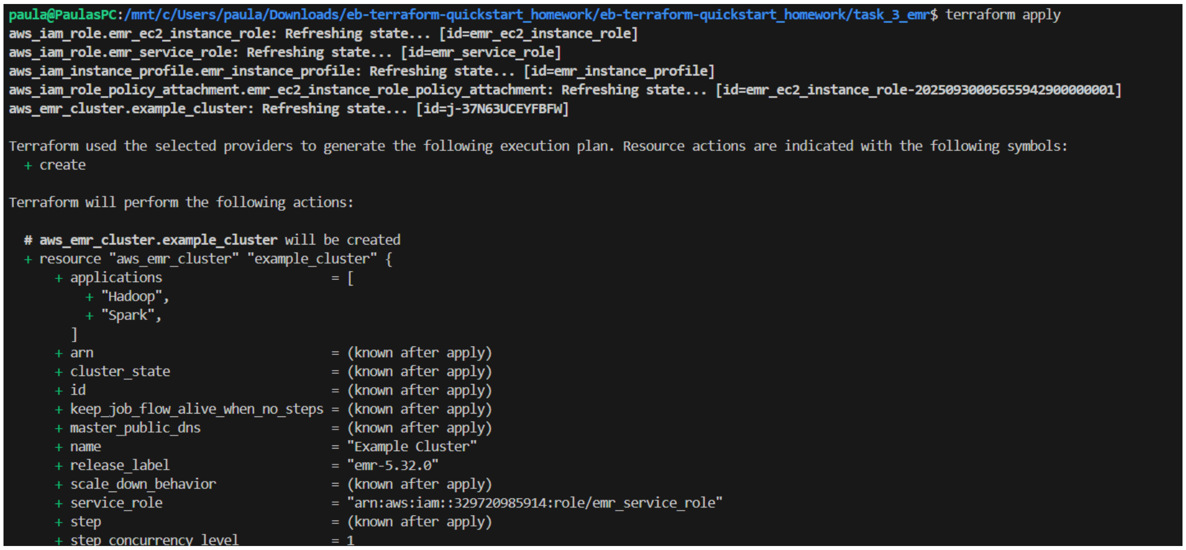
```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 5.0"
    }
  }

  backend "s3" {
    ## homework:start
    bucket = "terraforms3backend21-paulapirela-terraform-state"
    key = "ec2/emr/terraform.tfstate"
    region = "us-east-1"
    ## homework:end
    # use_lockfile = true
    encrypt     = true
    kms_key_id = "752bf32d-a3a0-4783-95f4-8f479f83a702"
  }
}

provider "aws" {
  ## homework:start
  ## homework:end
  region = "us-east-1"

  default_tags {
    tags = {
      Topic = "terraform"
      ## homework:start
```

Main:

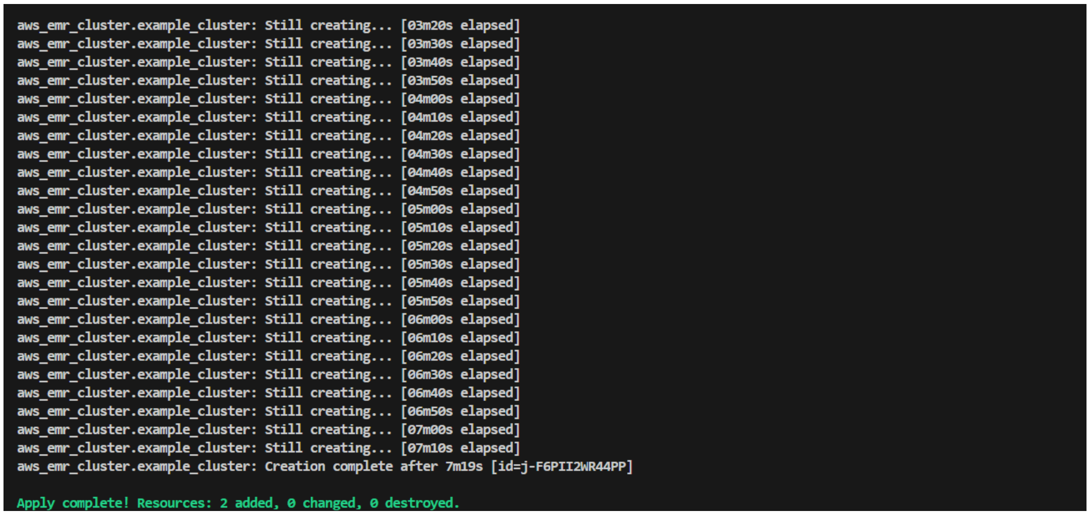

Preview README.md
main.tf
backend.tf
eb-terraform-quickstart_homework > task_3_emr > main.tf > resource "aws_iam_role_policy_attachment" "emr_service_role_attach" > role

```
resource "aws_iam_role" "emr_service_role" {
  name = "emr_service_role"

  assume_role_policy = <<EOF
  {
    "Version": "2008-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
          "Service": "elasticmapreduce.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
      }
    ]
  }
  EOF
}

resource "aws_iam_role_policy_attachment" "emr_service_role_attach" {
  role       = aws_iam_role.emr_service_role.name
  policy_arn = "arn:aws:iam::aws:policy/service-role/AmazonElasticMapReduceRole"
}

# Define the EC2 instance profile
resource "aws_iam_role" "emr_ec2_instance_role" {
  name = "emr_ec2_instance_role"

  assume_role_policy = <<EOF
```

Consola:

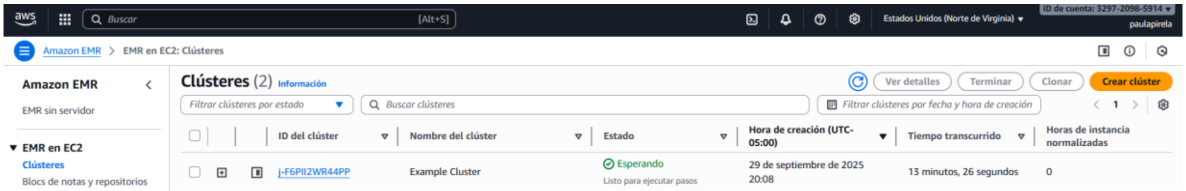


```
paula@PaulasPC:/mnt/c/Users/paula/Downloads/eb-terraform-quickstart_homework/eb-terraform-quickstart_homework/task_3_emr$ terraform apply
aws_iam_role.emr_ec2_instance_role: Refreshing state... [id=emr_ec2_instance_role]
aws_iam_role.emr_service_role: Refreshing state... [id=emr_service_role]
aws_iam_instance_profile.emr_instance_profile: Refreshing state... [id=emr_instance_profile]
aws_iam_role_policy_attachment.emr_ec2_instance_role_policy_attachment: Refreshing state... [id=emr_ec2_instance_role-20250930005655942900000001]
aws_emr_cluster.example_cluster: Refreshing state... [id=j-37N63UCEYFBFW]

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

Terraform will perform the following actions:

  # aws_emr_cluster.example_cluster will be created
  + resource "aws_emr_cluster" "example_cluster" {
      + applications                      = [
          + "Hadoop",
          + "Spark",
        ]
      + arn                               = (known after apply)
      + cluster_state                     = (known after apply)
      + id                                = (known after apply)
      + keep_job_flow_alive_when_no_steps = (known after apply)
      + master_public_dns                 = (known after apply)
      + name                              = "Example Cluster"
      + release_label                     = "emr-5.32.0"
      + scale_down_behavior               = (known after apply)
      + service_role                      = "arn:aws:iam::329720985914:role/emr_service_role"
      + step                              = (known after apply)
      + step_concurrency_level            = 1
```



```
aws_emr_cluster.example_cluster: Still creating... [03m20s elapsed]
aws_emr_cluster.example_cluster: Still creating... [03m30s elapsed]
aws_emr_cluster.example_cluster: Still creating... [03m40s elapsed]
aws_emr_cluster.example_cluster: Still creating... [03m50s elapsed]
aws_emr_cluster.example_cluster: Still creating... [04m00s elapsed]
aws_emr_cluster.example_cluster: Still creating... [04m10s elapsed]
aws_emr_cluster.example_cluster: Still creating... [04m20s elapsed]
aws_emr_cluster.example_cluster: Still creating... [04m30s elapsed]
aws_emr_cluster.example_cluster: Still creating... [04m40s elapsed]
aws_emr_cluster.example_cluster: Still creating... [04m50s elapsed]
aws_emr_cluster.example_cluster: Still creating... [05m00s elapsed]
aws_emr_cluster.example_cluster: Still creating... [05m10s elapsed]
aws_emr_cluster.example_cluster: Still creating... [05m20s elapsed]
aws_emr_cluster.example_cluster: Still creating... [05m30s elapsed]
aws_emr_cluster.example_cluster: Still creating... [05m40s elapsed]
aws_emr_cluster.example_cluster: Still creating... [05m50s elapsed]
aws_emr_cluster.example_cluster: Still creating... [06m00s elapsed]
aws_emr_cluster.example_cluster: Still creating... [06m10s elapsed]
aws_emr_cluster.example_cluster: Still creating... [06m20s elapsed]
aws_emr_cluster.example_cluster: Still creating... [06m30s elapsed]
aws_emr_cluster.example_cluster: Still creating... [06m40s elapsed]
aws_emr_cluster.example_cluster: Still creating... [06m50s elapsed]
aws_emr_cluster.example_cluster: Still creating... [07m00s elapsed]
aws_emr_cluster.example_cluster: Still creating... [07m10s elapsed]
aws_emr_cluster.example_cluster: Creation complete after 7m19s [id=j-F6PII2WR44PP]

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

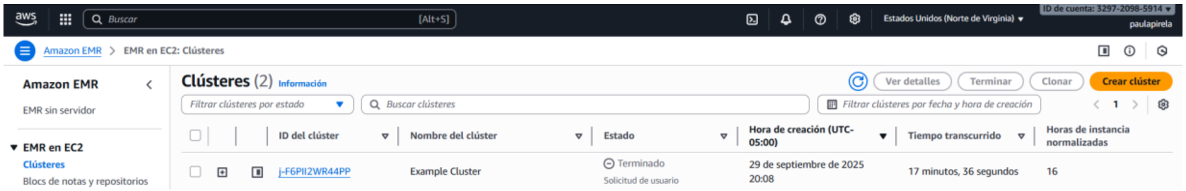
AWS:



aws
Buscar
[Alt+S]
Estados Unidos (Norte de Virginia)
ID de cuenta: 3297-2098-5914
paulapirela
Amazon EMR > EMR en EC2: Clústeres
Amazon EMR
EMR sin servidor
EMR en EC2
Clústeres
Blocs de notas y repositorios
Ver detalles
Terminar
Clonar
Crear clúster
Filtrar clústeres por estado
Buscar clústeres
Filtrar clústeres por fecha y hora de creación
1

Clústeres (2) Información

□			ID del clúster	Nombre del clúster	Estado	Hora de creación (UTC-05:00)	Tiempo transcurrido	Horas de instancia normalizadas
□	⊞	◨	j-F6PII2WR44PP	Example Cluster	✓ Esperando Listo para ejecutar pasos	29 de septiembre de 2025 20:08	13 minutos, 26 segundos	0



aws
Buscar
[Alt+S]
Estados Unidos (Norte de Virginia)
ID de cuenta: 3297-2098-5914
paulapirela
Amazon EMR > EMR en EC2: Clústeres
Amazon EMR
EMR sin servidor
EMR en EC2
Clústeres
Blocs de notas y repositorios
Ver detalles
Terminar
Clonar
Crear clúster
Filtrar clústeres por estado
Buscar clústeres
Filtrar clústeres por fecha y hora de creación
1

Clústeres (2) Información

□			ID del clúster	Nombre del clúster	Estado	Hora de creación (UTC-05:00)	Tiempo transcurrido	Horas de instancia normalizadas
□	⊞	◨	j-F6PII2WR44PP	Example Cluster	⊖ Terminado Solicitud de usuario	29 de septiembre de 2025 20:08	17 minutos, 36 segundos	16