

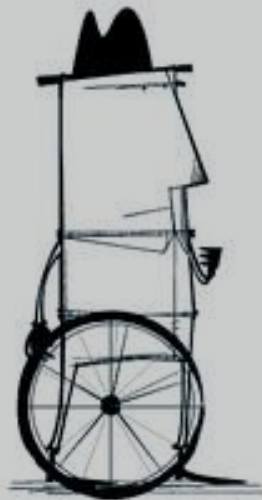
ASYS

Por una humanidad sustentable

Dev Ops

Modulo 7

ERRR...



**CAN'T STOP.
TOO BUSY!!**



TOO BUSY TO IMPROVE?

MAPA

DEMO

PRESENTACIÓN

REPETIR DEMO

—

Objetivos del Modulo

1. Repaso
2. Hilo Rojo
3. Terraform
4. Kahoot



Qué es DevOps ?

The Netflix logo is displayed in a bold, black, sans-serif font. The letters are filled with a dense, white, pixelated or grainy texture, giving it a digital or film-like appearance. The logo is centered within a solid gray rectangular background.

NETFLIX

The Google logo is rendered in its classic, multi-colored, rounded sans-serif typeface. The letters are a uniform light gray color. The logo is centered horizontally and occupies a significant portion of the middle section of the slide.

Google

Desarrollar, Operar y Mejorar Sistemas de Excelente Calidad

Flujo

**Hacer el
trabajo**

**Lo más
eficiente
posible**

**Retro
Alimentación**

**Observar
Medir
Notificar**

**Mejora
Continua**

**Incrementar
el valor de
nuestro
trabajo**

Cultura

Flujo

Intro

Trabajo

Diseño

Calidad

Infraestructura

FrontEnd

BackEnd

DataEnd

Infraestructura de Producción on Demand

Pipeline - C.I. - C.D. - C.Deployment

Cultura

Retroalimentación

Telemetría FrontEnd

Telemetría BackEnd

Telemetría Negocio

Telemetría Infraestructura

Mejora Continua

Operaciones

Experimentación

Ingeniería del Caos

Infraestructura como código



Será posible ser uno feliz administrando infraestructura ?

Terraform

Herramienta

Para la administración de Recursos de infraestructura tales como (por nombrar solo algunos)

- Computadoras físicas
- Computadoras virtuales
- Redes
- DNS
- CDNs
- Administradores de Contenedores
- Balanceadores de carga

Providers

Son proveedores de recursos tales como

- AWS
- AZURE
- GCP
- OpenStack
- Heroku
- DigitalOcean
- CloudFlare
- DNSimple

Estos hacen módulos y los ponen en el registro de terraform. Estos módulos son los responsables de interactuar con las API's del proveedor y exponer sus recursos

Como instalar Terraform



Instalar

1 Descargar binario

2 Ubicar el archivo en un directorio adecuado

3 Si hace falta : agregar el directorio al \$PATH del sistema operativo



macOS

64-bit



FreeBSD

32-bit | 64-bit | Arm



Linux

32-bit | 64-bit | Arm



OpenBSD

32-bit | 64-bit



Solaris

64-bit



Windows

32-bit | 64-bit

```
abernal@Alejandros-MacBook-Pro ~$ which terraform
/Users/abernal/bin/terraform
abernal@Alejandros-MacBook-Pro ~$ echo $PATH
/Library/Frameworks/Python.framework/Versions/3.6/bin:/Users/abernal/bin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:/usr/libexec:/Library/Frameworks/Mono.framework/Versions/Current/Commands:/Applications/Wireshark.app/Contents/MacOS
abernal@Alejandros-MacBook-Pro ~$ terraform
Usage: terraform [-version] [-help] <command> [args]
```

The available commands for execution are listed below. The most common, useful commands are shown first, followed by less common or more advanced commands. If you're just getting started with Terraform, stick with the common commands. For the other commands, please read the help and docs before usage.

Common commands:

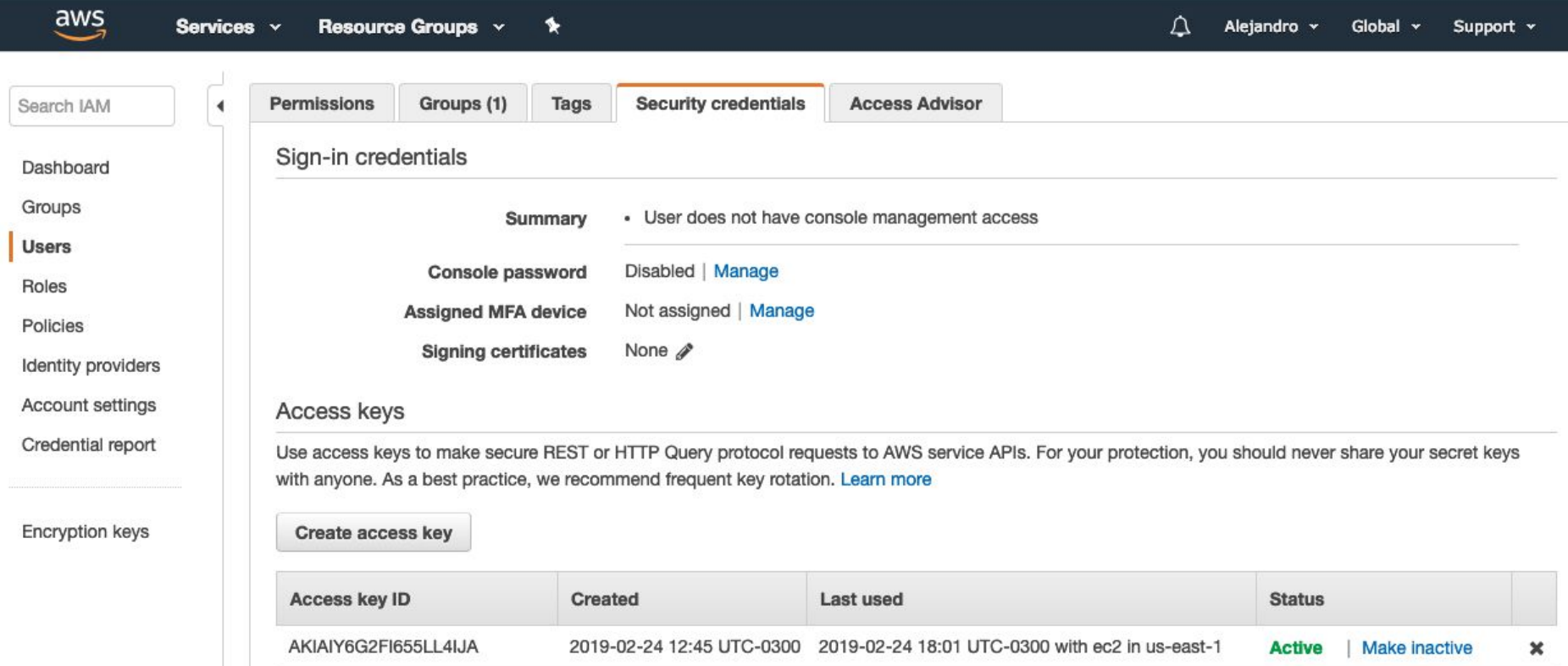
apply	Builds or changes infrastructure
console	Interactive console for Terraform interpolations
destroy	Destroy Terraform-managed infrastructure
env	Workspace management
fmt	Rewrites config files to canonical format
get	Download and install modules for the configuration
graph	Create a visual graph of Terraform resources
import	Import existing infrastructure into Terraform
init	Initialize a Terraform working directory
output	Read an output from a state file
plan	Generate and show an execution plan
providers	Prints a tree of the providers used in the configuration
push	Upload this Terraform module to Atlas to run
refresh	Update local state file against real resources
show	Inspect Terraform state or plan
taint	Manually mark a resource for recreation
untaint	Manually unmark a resource as tainted
validate	Validates the Terraform files
version	Prints the Terraform version
workspace	Workspace management

All other commands:

Como crear una VM en AWS



1 Obtener las credenciales en AWS para interactuar con las API's



The screenshot shows the AWS IAM console interface. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information (Alejandro, Global, Support). The left sidebar contains navigation links: Search IAM, Dashboard, Groups, Users (highlighted), Roles, Policies, Identity providers, Account settings, Credential report, and Encryption keys.

The main content area is titled 'Security credentials' and contains two sections:

- Sign-in credentials**
 - Summary**: User does not have console management access.
 - Console password**: Disabled | [Manage](#)
 - Assigned MFA device**: Not assigned | [Manage](#)
 - Signing certificates**: None [✎](#)
- Access keys**

Use access keys to make secure REST or HTTP Query protocol requests to AWS service APIs. For your protection, you should never share your secret keys with anyone. As a best practice, we recommend frequent key rotation. [Learn more](#)

[Create access key](#)

Access key ID	Created	Last used	Status	
AKIAIY6G2FI655LL4IJA	2019-02-24 12:45 UTC-0300	2019-02-24 18:01 UTC-0300 with ec2 in us-east-1	Active	Make inactive ✕

2 Crear un archivo con extensión *.tf

```
provider "aws" {  
  access_key = "ACCESS_KEY_HERE"  
  secret_key = "SECRET_KEY_HERE"  
  region     = "us-east-1"  
}  
  
resource "aws_instance" "example" {  
  ami           = "ami-2757f631"  
  instance_type = "t2.micro"  
}
```

3 Ejecutar el comando terraform init

```
abernal@Alejandros-MacBook-Pro ~/tmp/terraform-ppt$ terraform init
```

```
Initializing provider plugins...
```

- Checking for available provider plugins on <https://releases.hashicorp.com...>
- Downloading plugin for provider "aws" (1.60.0)...

The following providers do not have any version constraints in configuration, so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking changes, it is recommended to add version = "... constraints to the corresponding provider blocks in configuration, with the constraint strings suggested below.

```
* provider.aws: version = "~> 1.60"
```

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.

4 Ejecutar el comando terraform plan

```
abernal@Alejandros-MacBook-Pro ~/tmp/terraform-ppt terraform plan
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.

-----

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

+ aws_instance.example
  id: <computed>
  ami: "ami-2757f631"
```

5 Ejecutar el comando terraform apply

```
abernal@Alejandros-MacBook-Pro ~/tmp/terraform-ppt terraform apply

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

+ aws_instance.example
  id: <computed>
  ami: "ami-2757f631"
  arn: <computed>
  associate_public_ip_address: <computed>
  availability_zone: <computed>
  cpu_core_count: <computed>
  cpu_threads_per_core: <computed>
  ebs_block_device.#: <computed>
  ephemeral_block_device.#: <computed>
  get_password_data: "false"
  host_id: <computed>
  instance_state: <computed>
  instance_type: "t2.micro"
  ipv6_address_count: <computed>
  ipv6_addresses.#: <computed>
  key_name: <computed>
  network_interface.#: <computed>
  network_interface_id: <computed>
  password_data: <computed>
  placement_group: <computed>
  primary_network_interface_id: <computed>
  private_dns: <computed>
  private_ip: <computed>
  public_dns: <computed>
  public_ip: <computed>
  root_block_device.#: <computed>
  security_groups.#: <computed>
  source_dest_check: "true"
  subnet_id: <computed>
  tenancy: <computed>
  volume_tags.%: <computed>
  vpc_security_group_ids.#: <computed>

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: 
```

6 Confirmar ejecución

```
abernal@Alejandros-MacBook-Pro ~/tmp/terraform-ppt terraform apply
```

```
An execution plan has been generated and is shown below.  
Resource actions are indicated with the following symbols:
```

```
+ create
```

```
Terraform will perform the following actions:
```

```
+ aws_instance.example  
  id:                               <computed>  
  ami:                              "ami-2757f631"  
  arn:                               <computed>  
  associate_public_ip_address:      <computed>  
  availability_zone:                 <computed>  
  cpu_core_count:                    <computed>  
  cpu_threads_per_core:              <computed>  
  ebs_block_device.#:                <computed>  
  ephemeral_block_device.#:          <computed>  
  get_password_data:                 "false"  
  host_id:                           <computed>  
  instance_state:                    <computed>  
  instance_type:                     "t2.micro"  
  ipv6_address_count:                 <computed>  
  ipv6_addresses.#:                  <computed>  
  key_name:                           <computed>  
  network_interface.#:                <computed>  
  network_interface_id:               <computed>  
  password_data:                      <computed>  
  placement_group:                   <computed>  
  primary_network_interface_id:        <computed>  
  private_dns:                        <computed>  
  private_ip:                         <computed>  
  public_dns:                         <computed>  
  public_ip:                          <computed>  
  root_block_device.#:                <computed>  
  security_groups.#:                  <computed>  
  source_dest_check:                  "true"  
  subnet_id:                          <computed>  
  tenancy:                            <computed>  
  volume_tags.%:                      <computed>  
  vpc_security_group_ids.#:           <computed>
```

```
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
```


7 Verificando

```
aws_instance.example: Creating...
  ami: "" => "ami-2757f631"
  arn: "" => "<computed>"
  associate_public_ip_address: "" => "<computed>"
  availability_zone: "" => "<computed>"
  cpu_core_count: "" => "<computed>"
  cpu_threads_per_core: "" => "<computed>"
  ebs_block_device.#: "" => "<computed>"
  ephemeral_block_device.#: "" => "<computed>"
  get_password_data: "" => "false"
  host_id: "" => "<computed>"
  instance_state: "" => "<computed>"
  instance_type: "" => "t2.micro"
  ipv6_address_count: "" => "<computed>"
  ipv6_addresses.#: "" => "<computed>"
  key_name: "" => "<computed>"
  network_interface.#: "" => "<computed>"
  network_interface_id: "" => "<computed>"
  password_data: "" => "<computed>"
  placement_group: "" => "<computed>"
  primary_network_interface_id: "" => "<computed>"
  private_dns: "" => "<computed>"
  private_ip: "" => "<computed>"
  public_dns: "" => "<computed>"
  public_ip: "" => "<computed>"
  root_block_device.#: "" => "<computed>"
  security_groups.#: "" => "<computed>"
  source_dest_check: "" => "true"
  subnet_id: "" => "<computed>"
  tenancy: "" => "<computed>"
  volume_tags.%: "" => "<computed>"
  vpc_security_group_ids.#: "" => "<computed>"
aws_instance.example: Still creating... (10s elapsed)
aws_instance.example: Still creating... (20s elapsed)
aws_instance.example: Still creating... (30s elapsed)
aws_instance.example: Still creating... (40s elapsed)
aws_instance.example: Creation complete after 41s (ID: i-0446dbfa55e20bd92)

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


7 Verificando

The screenshot displays the AWS Management Console interface. The top navigation bar includes the AWS logo, 'Services' with a dropdown arrow, 'Resource Groups' with a dropdown arrow, and a star icon. On the right side of the navigation bar are a notification bell, the user name 'Alejandro' with a dropdown arrow, the region 'N. Virginia' with a dropdown arrow, and 'Support' with a dropdown arrow.

The main content area is divided into three sections:

- EC2 Dashboard** (left sidebar):
 - Events
 - Tags
 - Reports
 - Limits
 - INSTANCES (expanded):
 - Instances
 - Launch Templates
- Resources** (center):

You are using the following Amazon EC2 resources in the US East (N. Virginia) region:

1 Running Instances	0 Elastic IPs
0 Dedicated Hosts	0 Snapshots
1 Volumes	0 Load Balancers
0 Key Pairs	1 Security Groups
0 Placement Groups	
- Account Attributes** (right sidebar):
 - Supported Platforms
 - VPC
 - Default VPC
 - vpc-2076c95a
 - Resource ID length management
 - Console experiments

Como eliminar nuestra VM en AWS



1 Ejecutar el comando terraform destroy

```
abernal@Alejandros-MacBook-Pro ~/tmp/terraform-ppt$ terraform destroy
aws_instance.example: Refreshing state... (ID: i-0446dbfa55e20bd92)
```

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:

- destroy

Terraform will perform the following actions:

- **aws_instance.example**

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

1 Ejecutar el comando terraform destroy

```
aws_instance.example: Destroying... (ID: i-0446dbfa55e20bd92)
aws_instance.example: Still destroying... (ID: i-0446dbfa55e20bd92, 10s elapsed)
aws_instance.example: Still destroying... (ID: i-0446dbfa55e20bd92, 20s elapsed)
aws_instance.example: Still destroying... (ID: i-0446dbfa55e20bd92, 30s elapsed)
aws_instance.example: Still destroying... (ID: i-0446dbfa55e20bd92, 40s elapsed)
aws_instance.example: Still destroying... (ID: i-0446dbfa55e20bd92, 50s elapsed)
aws_instance.example: Still destroying... (ID: i-0446dbfa55e20bd92, 1m0s elapsed)
aws_instance.example: Destruction complete after 1m6s

Destroy complete! Resources: 1 destroyed.
```

2 Verificando



The screenshot displays the AWS Management Console interface. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and a star icon. On the right, there are links for 'Alejandro', 'N. Virginia', and 'Support'. The left sidebar shows the 'EC2 Dashboard' with links to 'Events', 'Tags', 'Reports', 'Limits', 'INSTANCES', 'Instances', and 'Launch Templates'. The main content area is divided into two panels. The 'Resources' panel, titled 'Resources', states 'You are using the following Amazon EC2 resources in the US East (N. Virginia) region:' and lists the following counts: 0 Running Instances, 0 Elastic IPs, 0 Dedicated Hosts, 0 Snapshots, 0 Volumes, 0 Load Balancers, 0 Key Pairs, 1 Security Groups, and 0 Placement Groups. The 'Account Attributes' panel, titled 'Account Attributes', lists 'Supported Platforms' (VPC), 'Default VPC' (vpc-2076c95a), and 'Resource ID length management' (Console experiments).

Resource Type	Count
Running Instances	0
Elastic IPs	0
Dedicated Hosts	0
Snapshots	0
Volumes	0
Load Balancers	0
Key Pairs	0
Security Groups	1
Placement Groups	0

Attribute	Value
Supported Platforms	VPC
Default VPC	vpc-2076c95a
Resource ID length management	Console experiments

RESUMEN

PARA QUÉ SIRVE

`terraform init`

`terraform plan`

`terraform apply`

`terraform show`

`terraform show`

TERRAFORM WEB SITE

LIBRO

PARROQUIALES