

CURSO: PIAD-626_TECNOLOGÍA CLOUD CON AWS

Tarea – HT-04

Configura la infraestructura.

Operaciones:

1. Configurar la infraestructura de AWS.
2. Crear soluciones usando herramientas de automatización.
3. Crear instancias usando servidores cloud.
4. Planificar el escalamiento de la infraestructura.
5. Configurar contenedores en la nube
6. Crear base de datos en la nube.
7. Crear soluciones con Amazon VPC.

Objetivo de la Tarea

Al concluir la tarea el participante estará en condiciones de configurar una infraestructura completa en Amazon Web Services (AWS), utilizando herramientas de automatización, contenedores, bases de datos gestionadas y redes personalizadas, así como planificar su escalabilidad para soportar aplicaciones modernas en la nube de forma segura y eficiente.

Caso Práctico

La empresa tecnológica EduCode, especializada en plataformas educativas en línea, está por lanzar una nueva aplicación para cursos interactivos en tiempo real. Necesita una infraestructura cloud robusta, modular y escalable que permita desplegar contenedores, bases de datos relacionales, automatización en el aprovisionamiento, y una red segura con control total del tráfico interno y externo. El equipo de desarrollo ha optado por construir la solución completa sobre AWS.

Por lo que se requiere que se desarrolle: Configurar la infraestructura de AWS, crear soluciones usando herramientas de automatización, crear instancias usando servidores cloud, planificar el escalamiento de la infraestructura, configurar contenedores en la nube, crear base de datos en la nube y crear soluciones con Amazon VPC.

Materiales/ Instrumentos/ Equipos/Herramientas/ Reactivos/ Insumos/ Colorantes.

Las siguientes listas son de referencia.

El instructor puede variar los requerimientos, con fin de desarrollar la tarea.

Materiales:	
Nombre	Cantidad
Guía de buenas prácticas AWS	1
Manual de Terraform básico	1

Instrumentos y Equipos:	
Nombre	Cantidad
Laptop o PC con conexión a Internet	1
Acceso a cuenta de AWS Free Tier	1

Herramientas:	
Nombre	Cantidad
AWS Management Console	1
AWS CLI	1
Terraform (o AWS CloudFormation)	1
Docker	1
Visual Studio Code o IDE equivalente	1
MySQL Workbench o cliente SQL	1

Desarrollo de la Práctica

OPERACIÓN 01: Configurar la infraestructura de AWS

- Crear una VPC personalizada.
- Crear dos subnets (pública y privada).
- Configurar Internet Gateway y tabla de rutas.
- Asociar la tabla con la subnet pública.

```
aws ec2 create-vpc --cidr-block 10.0.0.0/16
```

```
aws ec2 create-subnet --vpc-id <VPC-ID> --cidr-block 10.0.1.0/24
```

```
PS C:\Users\18-TC-404>
```

```
[ec2-user@ip-10-0-1-241 ~]$ 0_
```

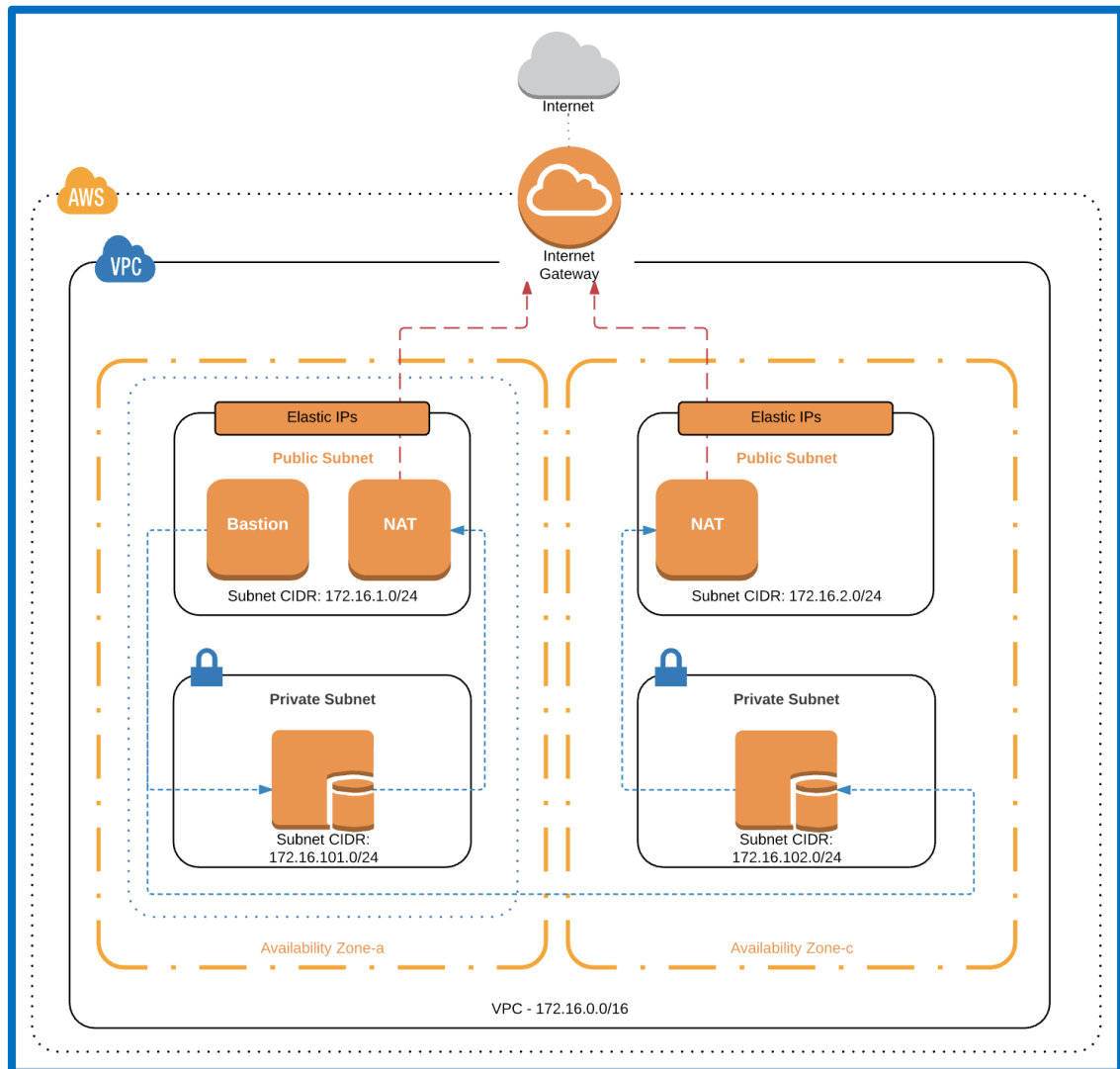


Diagrama de VPC con Subnets públicas y privadas, Bastión, NAT Gateways con Terraform

OPERACIÓN 02: Crear soluciones usando herramientas de automatización

- Crear un archivo main.tf con Terraform para lanzar una instancia EC2.

HCL

```
provider "aws" {
  region = "us-east-1"
}

resource "aws_instance" "app_server" {
  ami          = "ami-0c02fb55956c7d316"
  instance_type = "t2.micro"
  tags = {
    Name = "EduCodeApp"
  }
}
```

```
}
```

- Ejecutar:

terraform init

terraform apply

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\WINDOWS\system32> terraform -version
>>
Terraform v1.13.0
on windows_amd64
PS C:\WINDOWS\system32>
```

```
PS C:\Users\18-TC-404\Documents\terraform_1.13.0_windows_amd64> cd C:\Users\18-TC-404\Documents\terraform-proyecto
PS C:\Users\18-TC-404\Documents\terraform-proyecto> C:\Users\18-TC-404\Documents\terraform_1.13.0_windows_amd64\terraform.exe init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v6.10.0...
- Installed hashicorp/aws v6.10.0 (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.
PS C:\Users\18-TC-404\Documents\terraform-proyecto> C:\Users\18-TC-404\Documents\terraform_1.13.0_windows_amd64\terraform.exe apply

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

Terraform will perform the following actions:

# aws_internet_gateway.igw will be created
+ resource "aws_internet_gateway" "igw" {
  + arn      = (known after apply)
  + id       = (known after apply)
  + owner_id = (known after apply)
  + region   = "us-east-1"
  + tags     = {
    + "Name" = "seguroya-igw"
  }
}
```

```
# aws_vpc.seguroya_vpc will be created
+ resource "aws_vpc" "seguroya_vpc" {
  + arn                        = (known after apply)
  + cidr_block                = "10.0.0.0/16"
  + default_network_acl_id    = (known after apply)
  + default_route_table_id    = (known after apply)
  + default_security_group_id = (known after apply)
  + dhcp_options_id           = (known after apply)
  + enable_dns_hostnames      = (known after apply)
  + enable_dns_support        = true
  + enable_network_address_usage_metrics = (known after apply)
  + id                        = (known after apply)
  + instance_tenancy          = "default"
  + ipv6_association_id       = (known after apply)
  + ipv6_cidr_block           = (known after apply)
  + ipv6_cidr_block_network_border_group = (known after apply)
  + main_route_table_id       = (known after apply)
  + owner_id                  = (known after apply)
  + region                    = "us-east-1"
  + tags                      = {
    + "Name" = "seguroya-vpc"
  }
  + tags_all                  = {
    + "Name" = "seguroya-vpc"
  }
}
```

Plan: 6 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_vpc.seguroya_vpc: Creating...

aws_vpc.seguroya_vpc: Creation complete after 2s [id=vpc-0cd86c9ae934a1a36]

aws_internet_gateway.igw: Creating...

aws_subnet.public_subnet: Creating...

aws_subnet.private_subnet: Creating...

aws_internet_gateway.igw: Creation complete after 1s [id=igw-064e55a08edcc18ca]

aws_route_table.public_rt: Creating...

aws_subnet.private_subnet: Creation complete after 1s [id=subnet-05e7275f9beff49cc]

aws_subnet.public_subnet: Creation complete after 1s [id=subnet-045de890d8d849cd1]

aws_route_table.public_rt: Creation complete after 1s [id=rtb-061fd8bd485f3322d]

aws_route_table_association.public_assoc: Creating...

aws_route_table_association.public_assoc: Creation complete after 1s [id=rtbassoc-031fd2c248cd1e4e8]

Apply complete! Resources: 6 added, 0 changed, 0 destroyed.

PS C:\Users\18-TC-404\Documents\terraform-proyecto> |

```
PS C:\Users\18-TC-404> cd C:\Users\18-TC-404\Documents\terraform-proyecto
PS C:\Users\18-TC-404\Documents\terraform-proyecto> terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v6.10.0
```

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.

```
PS C:\Users\18-TC-404\Documents\terraform-proyecto> terraform plan
aws_vpc.seguroya_vpc: Refreshing state... [id=vpc-0cd86c9ae934a1a36]
aws_internet_gateway.igw: Refreshing state... [id=igw-064e55a08edcc18ca]
aws_subnet.private_subnet: Refreshing state... [id=subnet-05e7275f9beff49cc]
aws_subnet.public_subnet: Refreshing state... [id=subnet-045de890d8d849cd1]
aws_route_table.public_rt: Refreshing state... [id=rtb-061fd8bd485f3322d]
aws_route_table_association.public_assoc: Refreshing state... [id=rtbassoc-031fd2c248cd1e4e8]
```

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.

```
PS C:\Users\18-TC-404\Documents\terraform-proyecto> terraform apply
aws_vpc.seguroya_vpc: Refreshing state... [id=vpc-0cd86c9ae934a1a36]
aws_internet_gateway.igw: Refreshing state... [id=igw-064e55a08edcc18ca]
aws_subnet.public_subnet: Refreshing state... [id=subnet-045de890d8d849cd1]
aws_subnet.private_subnet: Refreshing state... [id=subnet-05e7275f9beff49cc]
aws_route_table.public_rt: Refreshing state... [id=rtb-061fd8bd485f3322d]
aws_route_table_association.public_assoc: Refreshing state... [id=rtbassoc-031fd2c248cd1e4e8]
```

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.

```
PS C:\Users\18-TC-404\Documents\terraform-proyecto> terraform plan
aws_vpc.seguroya_vpc: Refreshing state... [id=vpc-0cd86c9ae934a1a36]
aws_internet_gateway.igw: Refreshing state... [id=igw-064e55a08edcc18ca]
aws_subnet.private_subnet: Refreshing state... [id=subnet-05e7275f9beff49cc]
aws_subnet.public_subnet: Refreshing state... [id=subnet-045de890d8d849cd1]
aws_route_table.public_rt: Refreshing state... [id=rtb-061fd8bd485f3322d]
aws_route_table_association.public_assoc: Refreshing state... [id=rtbassoc-031fd2c248cd1e4e8]
```

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.

```
PS C:\Users\18-TC-404\Documents\terraform-proyecto> terraform apply
aws_vpc.seguroya_vpc: Refreshing state... [id=vpc-0cd86c9ae934a1a36]
aws_internet_gateway.igw: Refreshing state... [id=igw-064e55a08edcc18ca]
aws_subnet.public_subnet: Refreshing state... [id=subnet-045de890d8d849cd1]
aws_subnet.private_subnet: Refreshing state... [id=subnet-05e7275f9beff49cc]
aws_route_table.public_rt: Refreshing state... [id=rtb-061fd8bd485f3322d]
aws_route_table_association.public_assoc: Refreshing state... [id=rtbassoc-031fd2c248cd1e4e8]
```

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.

```
PS C:\Users\18-TC-404\Documents\terraform-proyecto> terraform plan
aws_vpc.seguroya_vpc: Refreshing state... [id=vpc-0cd86c9ae934a1a36]
aws_internet_gateway.igw: Refreshing state... [id=igw-064e55a08edcc18ca]
aws_subnet.public_subnet: Refreshing state... [id=subnet-045de890d8d849cd1]
aws_subnet.private_subnet: Refreshing state... [id=subnet-05e7275f9beff49cc]
aws_route_table.public_rt: Refreshing state... [id=rtb-061fd8bd485f3322d]
aws_route_table_association.public_assoc: Refreshing state... [id=rtbassoc-031fd2c248cd1e4e8]
```

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.

```
PS C:\Users\18-TC-404\Documents\terraform-proyecto> terraform apply
aws_vpc.seguroya_vpc: Refreshing state... [id=vpc-0cd86c9ae934a1a36]
aws_internet_gateway.igw: Refreshing state... [id=igw-064e55a08edcc18ca]
aws_subnet.private_subnet: Refreshing state... [id=subnet-05e7275f9beff49cc]
aws_subnet.public_subnet: Refreshing state... [id=subnet-045de890d8d849cd1]
aws_route_table.public_rt: Refreshing state... [id=rtb-061fd8bd485f3322d]
aws_route_table_association.public_assoc: Refreshing state... [id=rtbassoc-031fd2c248cd1e4e8]
```

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.

```
PS C:\Users\18-TC-404\Documents\terraform-proyecto>
```


OPERACIÓN 03: Crear instancias usando servidores cloud

- Desde la consola de AWS o Terraform, lanzar una instancia EC2 (Ubuntu Server).
- Conectarse vía SSH.
- Instalar el entorno Node.js:

```
sudo apt update
sudo apt install nodejs npm -y
```

- Subir y ejecutar una app de prueba

Resumen de instancia de i-016e90df614775046 (seguroya-web-ec2)

Información

Se ha actualizado hace 3 minutes

ID de la instancia

i-016e90df614775046

Dirección IPv6

–

Tipo de nombre de anfitrión

Nombre de IP: ip-10-0-1-241.ec2.internal

Responder al nombre DNS de recurso privado

–

Dirección IP asignada automáticamente

3.84.182.237 [IP pública]

Rol de IAM

–

IMDSv2

Required

Operador

–

Dirección IPv4 pública

3.84.182.237 | [dirección abierta](#)

Estado de la instancia

En ejecución

Nombre DNS de IP privada (solo IPv4)

ip-10-0-1-241.ec2.internal

Tipo de instancia

t3.micro

ID de VPC

vpc-05d30ef21557da1d9 (seguroya-vpc)

ID de subred

subnet-09070d552cf1a2dbd (Subnet pública)

ARN de instancia

arn:aws:ec2:us-east-1:359104569356:instance/i-016e90df614775046

Direcciones IPv4 privadas

10.0.1.241

DNS público

–

Direcciones IP elásticas

–

Hallazgo de AWS Compute Optimizer

[Suscribirse a AWS Compute Optimizer para recibir recomendaciones.](#)
[Más información](#)

Nombre del grupo de Auto Scaling

–

Administradas

falso

[illegible]


```
[ec2-user@ip-10-0-1-241 ~]$ sudo dnf install -y git unzip
Last metadata expiration check: 0:00:21 ago on Wed Aug 27 19:15:20 2025.
Package git-2.50.1-1.amzn2023.0.1.x86_64 is already installed.
Package unzip-6.0-57.amzn2023.0.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-10-0-1-241 ~]$ sudo dnf uptade -y
No such command: uptade. Please use /usr/bin/dnf --help
It could be a DNF plugin command, try: "dnf install 'dnf-command(uptade)'"
[ec2-user@ip-10-0-1-241 ~]$ sudo dnf update -y
Last metadata expiration check: 0:01:08 ago on Wed Aug 27 19:15:20 2025.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-10-0-1-241 ~]$ sudo systemctl start nginx
[ec2-user@ip-10-0-1-241 ~]$ sudo systemctl enable nginx
[ec2-user@ip-10-0-1-241 ~]$ systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: disabled)
   Active: active (running) since Sat 2025-08-23 01:52:32 UTC; 4 days ago
     Main PID: 192110 (nginx)
        Tasks: 3 (limit: 1057)
       Memory: 6.1M
          CPU: 6.089s
      CGroup: /system.slice/nginx.service
              └─192110 "nginx: master process /usr/sbin/nginx"
                 └─192111 "nginx: worker process"
                    └─192112 "nginx: worker process"

Aug 23 01:52:32 ip-10-0-1-241.ec2.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
Aug 23 01:52:32 ip-10-0-1-241.ec2.internal nginx[192108]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Aug 23 01:52:32 ip-10-0-1-241.ec2.internal nginx[192108]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Aug 23 01:52:32 ip-10-0-1-241.ec2.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
lines 1-16/16 (END)...skipping...
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: disabled)
   Active: active (running) since Sat 2025-08-23 01:52:32 UTC; 4 days ago
     Main PID: 192110 (nginx)
        Tasks: 3 (limit: 1057)
       Memory: 6.1M
          CPU: 6.089s
      CGroup: /system.slice/nginx.service
              └─192110 "nginx: master process /usr/sbin/nginx"
                 └─192111 "nginx: worker process"
                    └─192112 "nginx: worker process"

Aug 23 01:52:32 ip-10-0-1-241.ec2.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
Aug 23 01:52:32 ip-10-0-1-241.ec2.internal nginx[192108]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Aug 23 01:52:32 ip-10-0-1-241.ec2.internal nginx[192108]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Aug 23 01:52:32 ip-10-0-1-241.ec2.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
```

```
[ec2-user@ip-10-0-1-241 ~]$ sudo dnf update -y
Last metadata expiration check: 0:09:50 ago on Wed Aug 27 19:15:20 2025.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-10-0-1-241 ~]$ sudo dnf install -y nodejs npm
Last metadata expiration check: 0:10:13 ago on Wed Aug 27 19:15:20 2025.
Package nodejs-1:18.20.8-1.amzn2023.0.2.x86_64 is already installed.
Package nodejs-npm-1:10.8.2-1.18.20.8.1.amzn2023.0.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-10-0-1-241 ~]$ node -v
v18.20.8
[ec2-user@ip-10-0-1-241 ~]$ npm -v
10.8.2
```

```
GNU nano 8.3 app.js Modified
const http = require('http');

const hostname = '0.0.0.0';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('¡Hola desde el servidor Node.js en AWS EC2!\n');
});

server.listen(port, hostname, () => {
  console.log('Servidor corriendo en http://${hostname}:${port}/');
});
```



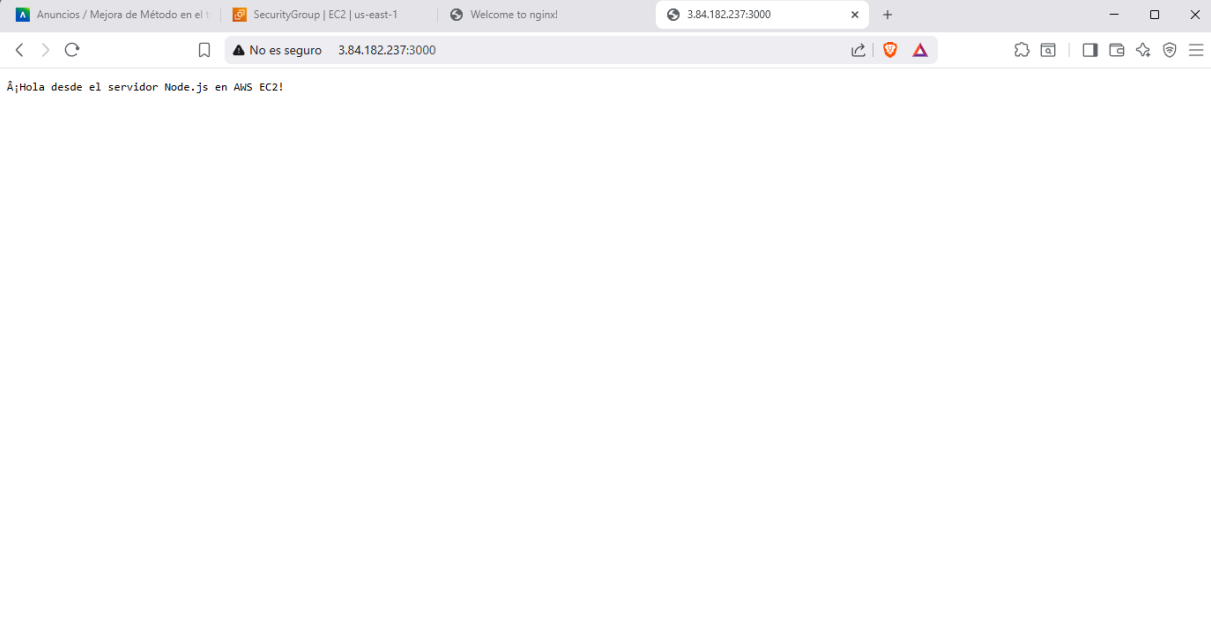
```
Environment=PM2_HOME=/home/ec2-user/.pm2
PIDFile=/home/ec2-user/.pm2/pm2.pid
Restart=on-failure

ExecStart=/usr/lib/nodejs18/lib/node_modules/pm2/bin/pm2 resurrect
ExecReload=/usr/lib/nodejs18/lib/node_modules/pm2/bin/pm2 reload all
ExecStop=/usr/lib/nodejs18/lib/node_modules/pm2/bin/pm2 kill

[Install]
WantedBy=multi-user.target

Target path
/etc/systemd/system/pm2-ec2-user.service
Command list
[ 'systemctl enable pm2-ec2-user' ]
[PM2] Writing init configuration in /etc/systemd/system/pm2-ec2-user.service
[PM2] Making script booting at startup...
[PM2] [-] Executing: systemctl enable pm2-ec2-user...
Created symlink /etc/systemd/system/multi-user.target.wants/pm2-ec2-user.service → /etc/systemd/system/pm2-ec2-user.service.
[PM2] [v] Command successfully executed.
-----+
[PM2] Freeze a process list on reboot via:
$ pm2 save

[PM2] Remove init script via:
$ pm2 unstartup systemd
[ec2-user@ip-10-0-1-241 mi-app]$ S_
```



<http://3.84.182.237:3000/>

OPERACIÓN 04: Planificar el escalamiento de la infraestructura

- Crear un Launch Template.
- Configurar un Auto Scaling Group:
 - Min: 1, Max: 4 instancias.
 - Escalado por CPU > 60%.
- Asociar un Load Balancer (ALB).

seguroya-template (lt-04980d8a681bd7cfe)

[Acciones](#)[Eliminar plantilla](#)

Detalles de la plantilla de lanzamiento

ID de la plantilla de lanzamiento
lt-04980d8a681bd7cfe

Nombre de la plantilla de lanzamiento
seguroya-template

Versión predeterminada
1

Propietario
arn:aws:iam::359104569356:root

[Detalles](#)[Versiones](#)[Etiquetas de la plantilla](#)

Detalles de la versión de la plantilla de lanzamiento

[Acciones](#)[Eliminar versión de plantilla](#)

Versión

1 (predeterminado)

Descripción

Plantilla para Auto Scaling con Nginx en Amazon Linux 2023.

Fecha de creación

2025-08-23T04:05:42.000Z

Creada por

arn:aws:iam::359104569356:root

[Detalles de la instancia](#)[Almacenamiento](#)[Etiquetas de recursos](#)[Interfaces de red](#)[Detalles avanzados](#)

ID de AMI

ami-00ca32bbc84273381

Tipo de instancia

t3.micro

Zona de disponibilidad

-

ID de zona de disponibilidad

-

Nombre del par de claves

seguroya-key

Grupos de seguridad

-

ID de grupo de seguridad

sg-0bd8a9b1686f3e6ac

seguroya-asg

seguroya-asg Descripción general de la capacidad

[Editar](#)

arn:aws:autoscaling:us-east-1:359104569356:autoScalingGroup:91067253-cd51-45bf-b34b-e431eef0f05:autoScalingGroupName/seguroya-asg

Capacidad deseada

1

Límites de escalamiento (Mín. - Máx.)

1 - 4

Tipo de capacidad deseado

Unidades (número de instancias)

Estado

Actualización de capacidad

Fecha de creación

Sat Aug 23 2025 00:40:04 GMT-0500 (hora estándar de Perú)

[Detalles](#)[Integraciones - nueva](#)[Escalado automático](#)[Administración de instancias](#)[Actualización de instancias](#)[Actividad](#)[Monitoreo](#)

Las políticas de escalado ajustan el tamaño del grupo de escalado automático para satisfacer los cambios de demanda. Con las políticas de escalado dinámico reactivas, puede realizar un seguimiento de métricas específicas de CloudWatch y tomar medidas cuando se alcance el umbral de alarma de CloudWatch. Utilice políticas de escalado predictivo junto con políticas de escalado dinámico en las siguientes situaciones: cuando la demanda de la aplicación cambie rápidamente, pero con un patrón recurrente, o cuando las instancias de EC2 necesiten más tiempo para inicializarse.

Políticas de escalado dinámico (1) [Info](#)

[Acciones](#)[Crear una política de escalado dinámico](#)[1](#)

CPU-Target-Scaling

Tipo de política

Escalado de seguimiento de destino

Habilitado o deshabilitado

Habilitado

Ejecutar la política cuando

Según sea necesario para mantener Utilización promedio de la CPU en 60

Realizar la acción

Agregar o eliminar unidades de capacidad según sea necesario

Las instancias necesitan

300 segundos para prepararse antes de incluirse en la métrica

Escalado descendente

Habilitado

seguroya-asg

seguroya-asg Descripción general de la capacidad

am:aws:autoscaling:us-east-1:359104569356:autoScalingGroup:91067253-cd51-45bf-b34b-e431eec0f05e:autoScalingGroupName/seguroya-asg

Capacidad deseada

1

Límites de escalamiento (Mín. - Máx.)

1 - 4

Tipo de capacidad deseado

Unidades (número de instancias)

Estado

Actualización de capacidad

Fecha de creación

Sat Aug 23 2025 00:40:04 GMT-0500 (hora estándar de Perú)

Detalles

Integraciones - nueva

Escalado automático

Administración de instancias

Actualización de instancias

Actividad

Monitoreo

Plantilla de lanzamiento

Plantilla de lanzamiento

lt-04980da681bd7cfe

seguroya-template

Propietario

arn:aws:iam::359104569356:root

Horario de creación

Fri Aug 22 2025 23:05:42 GMT-0500 (hora estándar de Perú)

Solicitar instancias de spot

No

Versión

Latest

ID de AMI

ami-00ca32bbc84273381

ID de grupos de seguridad

sg-0bd8a9b1686f3e6ac

Nombre del par de claves

seguroya-key

Almacenamiento (volumenes)

-

Grupos de seguridad

-

Descripción

Plantilla para Auto Scaling con Nginx en Amazon Linux 2023.

Ver detalles en la consola de la plantilla de lanzamiento

Red

Zonas de disponibilidad

use1-az4 (us-east-1a)
use1-az6 (us-east-1b)

ID de subred

subnet-09070d552cf1a2dbd
subnet-0e99fd9599693be29

Distribución de zonas de disponibilidad

Mejor esfuerzo equilibrado

Requisitos de tipo de instancias

El grupo de Auto Scaling se adhiere a la plantilla de lanzamiento para la opción de compra y el tipo de instancia.

Las opciones de equilibrio de carga y VPC Lattice se han trasladado a la nueva pestaña de integraciones.

Ver pestaña de integraciones

Comprobaciones de estado

Tipo de comprobación de estado

EC2, ELB

Periodo de gracia de la comprobación de estado

300

seguroya-asg

seguroya-asg Descripción general de la capacidad

am:aws:autoscaling:us-east-1:359104569356:autoScalingGroup:91067253-cd51-45bf-b34b-e431eec0f05e:autoScalingGroupName/seguroya-asg

Capacidad deseada

1

Límites de escalamiento (Mín. - Máx.)

1 - 4

Tipo de capacidad deseado

Unidades (número de instancias)

Estado

Actualización de capacidad

Fecha de creación

Sat Aug 23 2025 00:40:04 GMT-0500 (hora estándar de Perú)

Detalles

Integraciones - nueva

Escalado automático

Administración de instancias

Actualización de instancias

Actividad

Monitoreo

Balace de carga

Grupos de destino del balanceador de carga

seguroya-tg

Balaceadores de carga clásicos

-

Opciones de integración de VPC Lattice

Grupos de destino de VPC Lattice

-

Cambio de zona del controlador de recuperación de aplicaciones (ARC) - nueva

Cuando la zona de disponibilidad no tiene un estado adecuado, la instancia de destino se lanza en otras zonas de disponibilidad en buen estado.

Desplazamiento de cambio de zona ARC

Deshabilitado

SENATI

Seminario de Complementación Práctica – S6

RDA

04

OPERACIÓN 05: Configurar contenedores en la nube

- Crear una imagen Docker para el backend.
- Subirla a Amazon Elastic Container Registry (ECR).
- Configurar Amazon ECS con Fargate para ejecutar contenedores.

```
# Dockerfile
FROM node:16
WORKDIR /app
COPY . .
RUN npm install
CMD ["npm", "start"]
EXPOSE 3000
```

- Usar ecs-cli o consola para crear un servicio ECS con balanceo de carga.

```
~ $ ls -l
total 4
-rw-r--r-- 1 cloudshell-user cloudshell-user 90 Sep  2 22:32 Dockerfile
~ $ mkdir seguroya-backend && cd seguroya-backend
seguroya-backend $ cat <<EOF > package.json
> {
>   "name": "seguroya-backend",
>   "version": "1.0.0",
>   "main": "index.js",
>   "scripts": {
>     "start": "node index.js"
>   }
> }
> EOF
seguroya-backend $ cat <<EOF > index.js
> const express = require('express');
> const app = express();
> app.get('/', (req, res) => res.send('SeguroYA Backend funcionando 🚀'));
> app.listen(3000, () => console.log('Servidor en puerto 3000'));
> EOF
seguroya-backend $ cat <<EOF > Dockerfile
> FROM node:16
> WORKDIR /app
> COPY . .
> RUN npm install express
> CMD ["npm", "start"]
> EXPOSE 3000
> EOF
seguroya-backend $ docker build -t seguroya-backend .
[+] Building 57.5s (9/9) FINISHED
It
-> [internal] load build definition from Dockerfile
0s
-> transferring dockerfile: 184B
0s
-> [internal] load metadata for docker.io/library/node:16
2s
-> [internal] load .dockerignore
0s
-> transferring context: 2B
0s
-> [1/4] FROM docker.io/library/node:16@sha256:f77a1aef2da8d83e45ec990f45df50f1a286c5fc8bbfb8c6e4246c6389705c0b
0s
-> [internal] load build context
0s
-> transferring context: 698B
0s
-> CACHED [2/4] WORKDIR /app
0s
-> [3/4] COPY . .
0s
-> [4/4] RUN npm install express
9s
-> exporting to image
2s
-> exporting layers
2s
-> writing image sha256:4abbba88e7761c2d6cfc874a3a5f6e569e18ff75202adc8967883850bbc029d9f
0s
-> naming to docker.io/library/seguroya-backend
0s
seguroya-backend $ docker run -p 3000:3000 seguroya-backend

> seguroya-backend@1.0.0 start
> node index.js

Servidor en puerto 3000
```



```
EC2 > Grupos de seguridad > sg-0bd8a9b1686f3e6ac - seguridad-sg > Editar reglas de entrada

CloudShell

us-east-1 +

project $ docker run -it --rm project-image .
1.713 so will have to try and fail each time (and thus, yum will be much
1.713 slower). If it is a very temporary problem though, this is often a nice
1.713 compromise:
1.713 yum-config-manager --save --setopt=repoid.skip_if_unavailable=true
1.713
1.713 Cannot find a valid baseurl for repo: base77/x86_64
-----
Dockerfile:5
-----
3 |
4 | # Instala el paquete del servidor HTTP Apache desde el repositorio de CentOS
5 |>>> RUN yum install httpd -y
6 |
7 | # Copia el archivo index.html desde el contexto de construcción de Docker al contenedor

ERROR: failed to solve: process "/bin/sh -c yum install httpd -y" did not complete successfully: exit code: 1
project $ docker run -it --rm project-image /usr/www/html/
Unable to find image 'project-image:latest' locally
docker: Error response from daemon: pull access denied for project-image, repository does not exist or may require 'docker login': denied: requested access to the resource is denied.
See 'docker run --help'.
project $ FROM httpd:2.4
-bash: FROM: command not found
project $
project $ # Copia tu index.html al directorio de Apache
project $ COPY index.html /usr/local/apache2/htdocs/
-bash: COPY: command not found
project $
project $ EXPOSE 80
-bash: EXPOSE: command not found
project $ cd -w /project
project $ nano Dockerfile
project $ docker build -t project-image .
[*] Building 7.7s (7/7) FINISHED
--> [internal] load build definition from Dockerfile
--> [internal] load metadata for docker.io/library/httpd:2.4
--> [internal] load .dockerignore
--> [internal] transferring context: 28
--> [internal] load build context
--> [internal] transferring context: 876
--> [1/2] FROM docker.io/library/httpd:2.4@sha256:1198c1839e1a875f80818083758a7635f1ae999f0601f302f308c2ac99c3
--> resolve docker.io/library/httpd:2.4@sha256:1198c1839e1a875f80818083758a7635f1ae999f0601f302f308c2ac99c3
--> sha256:1198c1839e1a875f80818083758a7635f1ae999f0601f302f308c2ac99c3 / 7.85KB / 7.85KB
--> sha256:4f4f0906f5461cf40271a08d909c1e0c055774846d75e08d38acc1 328 / 328
--> sha256:70bbac3823c8c06d17fca838f016f8b1f4a99908704cc0c8ca6420522f 11.43MB / 11.43MB
--> sha256:70bbac3823c8c06d17fca838f016f8b1f4a99908704cc0c8ca6420522f 2.09KB / 2.09KB
--> sha256:19861d576362cdcd18956c3d4f7952c0b4a18058d3086a2c3b372f0117c 29.77MB / 29.77MB
--> sha256:4072221232674134270808dc852d0811d0f7977f4830b3440e302321 145B / 145B
--> sha256:d2b1a5a0e8d3f1040b4d7a3024610240d44d4a0ec4a40f30fa0779c404117 1.99MB / 1.99MB
--> sha256:1198c1839e1a875f80818083758a7635f1ae999f0601f302f308c2ac99c3 10.14KB / 10.14KB
--> sha256:779cd8139720f936ca3e104157a5af26cd24c07ce58f525c3108a0f0772 293B / 293B
--> extracting sha256:779cd8139720f936ca3e104157a5af26cd24c07ce58f525c3108a0f0772 0.0117c
--> extracting sha256:4072221232674134270808dc852d0811d0f7977f4830b3440e302321 180b352521
--> extracting sha256:4f4f0906f5461cf40271a08d909c1e0c055774846d75e08d38acc1
--> extracting sha256:d2b1a5a0e8d3f1040b4d7a3024610240d44d4a0ec4a40f30fa0779c404117
--> extracting sha256:70bbac3823c8c06d17fca838f016f8b1f4a99908704cc0c8ca6420522f
--> extracting sha256:779cd8139720f936ca3e104157a5af26cd24c07ce58f525c3108a0f0772
--> [2/2] COPY index.html /usr/local/apache2/htdocs/
--> exporting to image
--> exporting layers
--> writing image sha256:27b780f9a1cbfb7c93759f87ed732898f7c91bd0970b94a854a907194c57
--> name to docker.io/library/project-image
project $ docker run -d -p 80:80 --name project-container project-image
2a3ef3fc76d361690d2b8a93b5f6434fde038e8fe0809377b16f085d5aef16c
project $ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
2a3ef3fc76d3 project-image "httpd-foreground" About a minute ago Up About a minute 0.0.0.0:80->80/tcp, :::80->80/tcp project-container
project $ curl http://169.254.169.254/latest/meta-data/public-ipv4
project $ curl -s http://169.254.169.254/latest/meta-data/public-ipv4
project $
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
2a3ef3fc76d3 project-image "httpd-foreground" About an hour ago Up About an hour 0.0.0.0:80->80/tcp, :::80->80/tcp project-container
project $ docker start project-container
project $ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
2a3ef3fc76d3 project-image "httpd-foreground" About an hour ago Up About an hour 0.0.0.0:80->80/tcp, :::80->80/tcp project-container
project $ curl http://localhost
<!-- index.html -->
<html>
<head>
<title>Amazon ECS Sample App</title>
<style>
body {
margin-top: 40px;
background-color: #333;
color: white;
text-align: center;
}
</style>
</head>
<body>
<div>
<h1>Amazon ECS Sample App</h1>
<h2>Felicidades!</h2>
<p>Tu aplicación está ejecutándose en un contenedor AWS ECS.</p>
</div>
</body>
</html>
project $
```


aws Buscar [Alt+S] Estados Unidos (Norte de Virginia) ID de cuenta: 3591-0456-9356 jhorvyn_Reyes

Aurora and RDS > Bases de datos > educode-db

Aurora and RDS

Panel

Bases de datos

Información sobre rendimiento

Instantáneas de

Exportaciones en Amazon S3

Copias de seguridad automatizadas

Instancias reservadas

Proxies

Grupos de subredes

Grupos de parámetros

Grupos de opciones

Versiones de motor personalizadas

Integraciones sin extracción, transformación y carga (ETL) [Nuevo](#)

Eventos

Suscripciones a eventos

Recomendaciones [1](#) [1](#)

Actualización del certificado

educode-db [Modificar](#) [Acciones](#)

Resumen

Identificador de base de datos educode-db	Estado Modificando	Rol Instancia	Motor MySQL Community	Recomendaciones
CPU 3.27%	Clase db.t4g.micro	Actividad actual 0 Conexiones	Región y AZ us-east-1b	

Conectividad y seguridad | Supervisión | Registros y eventos | Configuración | Integraciones sin extracción, transformación y carga

Conectividad y seguridad

Punto de enlace y puerto	Redes	Seguridad
Punto de enlace educode-db.cib8ouk2jf1.us-east-1.rds.amazonaws.com	Zona de disponibilidad us-east-1b	Grupos de seguridad de la VPC default (sg-0afc701a1dfd6bf1a) Información Activo
Puerto 3306	VPC vpc-0b354acca1c39a717	Accesible públicamente No
	Grupo de subredes default-vpc-0b354acca1c39a717	Entidad de certificación Información rds-ca-rsa2048-g1
	Subredes subnet-0cd5e957fb9b8e502 subnet-0fe61cbd4d1270f1f subnet-0b27518eb7ffa29d4 subnet-0eae1e0d5f600603 subnet-03b5a690828a8c013 subnet-044ce2121ba475a18	Fecha de la entidad de certificación May 25, 2061, 18:34 (UTC-05:00)
	Tipo de red IPv4	Fecha de expiración del certificado de instancia de base de datos September 09, 2026, 13:31 (UTC-05:00)

Manage Server Connections

MySQL Connections

- Local instance MySQL80
- AWS RDS Books
- educode-db

Connection Name: educode-db

Connection Remote Management System Profile

Connection Method: Standard (TCP/IP) Method to use to connect to the RDBMS

Parameters SSL Advanced

Hostname: educode-db.cib8ouk2jf1.us Port: 3306 Name or IP address of the server host - and TCP/IP port.

Username: admin Name of the user to connect with.

Password: [Store in Vault ...](#) [Clear](#) The user's password. Will be requested later if it's not set.

Default Schema: The schema to use as default schema. Leave blank to select it later.

[New](#) [Delete](#) [Duplicate](#) [Move Up](#) [Move Down](#) [Test Connection](#) [Close](#)

MySQL Workbench

educode-db x educode-db (educode) x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

educode

Tables

Views

Stored Procedures

Functions

sys

Query 1

SHOW DATABASES;

Limit to 1000 rows

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid

Filter Rows:

Export:

Wrap Cell Contents: 1

Database

educode

information_schema

mysql

performance_schema

sys

Administration Schemas

Information

No object selected

Object Info Session

MySQL Workbench

Result 5

Read Only Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
3	14:20:38	CREATE DATABASE IF NOT EXISTS educode	1 row(s) affected	0.109 sec
4	14:20:47	USE educode	0 row(s) affected	0.094 sec
5	14:20:57	CREATE TABLE students (id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100))	0 row(s) affected	0.141 sec
6	14:22:14	CREATE TABLE students (id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100))	Error Code: 1050. Table 'students' already exists	0.094 sec
7	14:22:48	SHOW DATABASES	5 row(s) returned	0.094 sec / 0.000 sec
8	14:22:56	SHOW TABLES	1 row(s) returned	0.094 sec / 0.000 sec
9	14:23:07	SHOW DATABASES	5 row(s) returned	0.110 sec / 0.000 sec
10	14:23:17	DESCRIBE students	3 row(s) returned	0.093 sec / 0.000 sec
11	14:23:39	SHOW DATABASES	5 row(s) returned	0.094 sec / 0.000 sec

MySQL Workbench

educode-db x educode-db (educode) x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

educode

Tables

Views

Stored Procedures

Functions

sys

Query 1

DESCRIBE students;

Limit to 1000 rows

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid

Filter Rows:

Export:

Wrap Cell Contents: 1

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI		auto_increment
name	varchar(100)	YES			
email	varchar(100)	YES			

Administration Schemas

Information

No object selected

Object Info Session

MySQL Workbench

Result 7

Read Only Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
5	14:20:57	CREATE TABLE students (id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100))	0 row(s) affected	0.141 sec
6	14:22:14	CREATE TABLE students (id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100))	Error Code: 1050. Table 'students' already exists	0.094 sec
7	14:22:48	SHOW DATABASES	5 row(s) returned	0.094 sec / 0.000 sec
8	14:22:56	SHOW TABLES	1 row(s) returned	0.094 sec / 0.000 sec
9	14:23:07	SHOW DATABASES	5 row(s) returned	0.110 sec / 0.000 sec
10	14:23:17	DESCRIBE students	3 row(s) returned	0.093 sec / 0.000 sec
11	14:23:39	SHOW DATABASES	5 row(s) returned	0.094 sec / 0.000 sec
12	14:24:02	SHOW TABLES	1 row(s) returned	0.094 sec / 0.000 sec
13	14:24:14	DESCRIBE students	3 row(s) returned	0.094 sec / 0.000 sec

MySQL Workbench

educode-db x educode-db (educode) x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

educode

Tables

Views

Stored Procedures

Functions

sys

Query 1

```

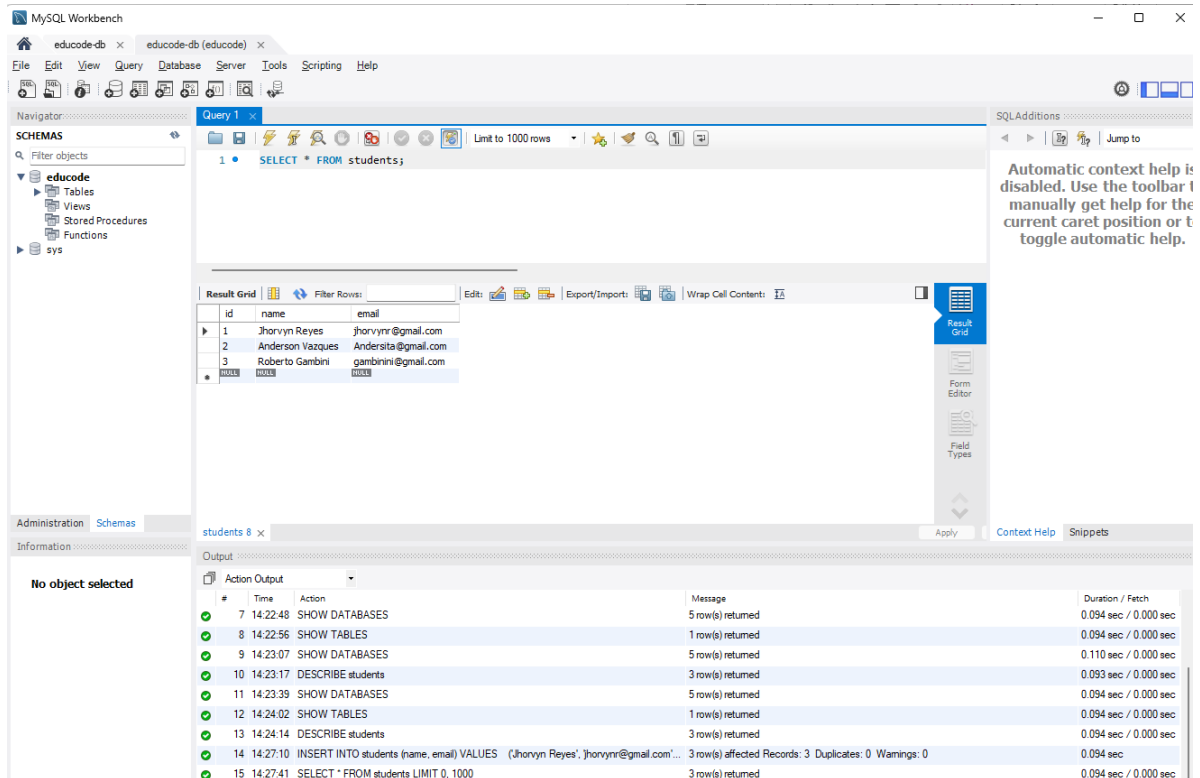
1 INSERT INTO students (name, email)
2 VALUES
3 ('Jhorvyn Reyes', 'jhorvynr@gmail.com'),
4 ('Anderson Vazquez', 'Andersita@gmail.com'),
5 ('Roberto Gambini', 'gambinini@gmail.com');
6

```

Limit to 1000 rows

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.



Endpoint: educode-db.cibs8ouk2jf1.us-east-1.rds.amazonaws.com

Usuario: admin

Password: Educ0de2025!

OPERACIÓN 07: Crear soluciones con Amazon VPC

- VPC con CIDR personalizado.
- Subnets segmentadas por funciones (EC2, RDS, Load Balancer)
- Grupos de seguridad definidos:
 - o EC2: HTTP, HTTPS, SSH.
 - o RDS: solo conexiones internas desde EC2.
- Asociar servicios correctamente para aislamiento y seguridad.

Panel de VPC < rtb-0f8d63caff7cdf66e / seguroya-private-rt

Vista global de EC2 [Filtrar por VPC](#)

▼ Nube virtual privada

Sus VPC

Subredes

Tablas de enrutamiento

Puertas de enlace de Internet

Puerta de enlace de Internet de solo salida

Gateways de operador

Conjuntos de opciones de DHCP

Direcciones IP elásticas

Listas de prefijos administradas

Gateways NAT

Interconexiones

Servidores de ruta [Nueva](#)

▼ Seguridad

ACL de red

Grupos de seguridad

▼ PrivateLink y Lattice

Introducción [Actualizado](#)

Detalles Información

ID de tabla de enrutamiento [rtb-0f8d63caff7cdf66e](#)

VPC [vpc-05d30ef21557da1d9 / seguroya-vpc](#)

Principal [No](#)

ID de propietario [359104569356](#)

Asociaciones de subredes explícitas [subnet-038323bcc02b8e111 / Subnet privada](#)

Asociaciones de borde -

Rutas [Asociaciones de subredes](#) Asociaciones de borde Propagación de rutas Etiquetas

Asociaciones de subredes explícitas (1) [Editar asociaciones de subredes](#)

Buscar asociación de subredes

Nombre	ID de subred	CIDR IPv4	CIDR IPv6
Subnet privada	subnet-038323bcc02b8e111	10.0.2.0/24	-

Subredes sin asociaciones explícitas (1) [Editar asociaciones de subredes](#)

Las siguientes subredes no se han asociado explícitamente con ninguna tabla de enrutamiento y, por lo tanto, están asociadas a la tabla de enrutamiento principal:

Buscar asociación de subredes

Nombre	ID de subred	CIDR IPv4	CIDR IPv6
seguroya-alb	subnet-0e99fd9599693be29	10.0.3.0/24	-

Panel de VPC < Tablas de enrutamiento

Vista global de EC2 [Filtrar por VPC](#)

▼ Nube virtual privada

Sus VPC

Subredes

Tablas de enrutamiento

Puertas de enlace de Internet

Puerta de enlace de Internet de solo salida

Gateways de operador

Conjuntos de opciones de DHCP

Direcciones IP elásticas

Listas de prefijos administradas

Gateways NAT

Interconexiones

Servidores de ruta [Nueva](#)

▼ Seguridad

ACL de red

Grupos de seguridad

▼ PrivateLink y Lattice

Introducción [Actualizado](#)

Tablas de enrutamiento (5) Información

Buscar tablas de enrutamiento por atributo o etiqueta

Nombre	ID de tabla de enrutam...	Asociaciones de subredes explícitas	Asociaciones de...	Princ...	VPC	ID de propietario
-	rtb-00b34ef411545411f	-	-	Si	vpc-0c86c9ae934a1a36 / seguroya-vpc	359104569356
-	rtb-011ba151b09764db7	-	-	Si	vpc-0b35dacc1c39a717 / seguroya-vpc	359104569356
-	rtb-0813da7f66122d809	-	-	Si	vpc-05d30ef21557da1d9 / seguroya-vpc	359104569356
seguroya-private-rt	rtb-0f8d63caff7cdf66e	subnet-038323bcc02b8e111 / Subnet privada	-	No	vpc-05d30ef21557da1d9 / seguroya-vpc	359104569356
seguroya-public-rt	rtb-011450abcf9464ade	subnet-09070d532f1a2dbd / Subnet pública	-	No	vpc-05d30ef21557da1d9 / seguroya-vpc	359104569356
seguroya-public-rt	rtb-06168bb4685f332d	subnet-045da90d8d849cd1 / seguroya-public-subnet	-	No	vpc-0c86c9ae934a1a36 / seguroya-vpc	359104569356

Panel de VPC < Grupos de seguridad

Vista global de EC2 [Filtrar por VPC](#)

▼ Nube virtual privada

Sus VPC

Subredes

Tablas de enrutamiento

Puertas de enlace de Internet

Puerta de enlace de Internet de solo salida

Gateways de operador

Conjuntos de opciones de DHCP

Direcciones IP elásticas

Listas de prefijos administradas

Gateways NAT

Interconexiones

Servidores de ruta [Nueva](#)

▼ Seguridad

ACL de red

Grupos de seguridad

▼ PrivateLink y Lattice

Introducción [Actualizado](#)

sg-02f112e832d3135f2 - seguroya-rds-sg

El grupo de seguridad (sg-02f112e832d3135f2 / seguroya-rds-sg) se ha creado correctamente

Detalles

Nombre del grupo de seguridad [seguroya-rds-sg](#)

ID del grupo de seguridad [sg-02f112e832d3135f2](#)

Descripción [Grupo de seguridad para RDS, acceso solo desde EC2](#)

ID de la VPC [vpc-05d30ef21557da1d9](#)

Propietario [359104569356](#)

Número de reglas de entrada [1 Entrada de permisos](#)

Número de reglas de salida [1 Entrada de permisos](#)

Reglas de entrada Reglas de salida Compartiendo: [novedad](#) Asociaciones de VPC: [novedad](#) Etiquetas

Reglas de entrada (1) [Administrar etiquetas](#) [Editar reglas de entrada](#)

Nombre	ID de la regla del gr...	Versión de IP	Tipo	Protocolo	Intervalo de puertos	Origen	Descripción
-	sgr-0ca548db325a44	-	MySQL/Aurora	TCP	3306	sg-0b08a9b1d8613a5c...	-

Panel de VPC < Grupos de seguridad

Vista global de EC2 [Filtrar por VPC](#)

▼ Nube virtual privada

Sus VPC

Subredes

Tablas de enrutamiento

Puertas de enlace de Internet

Puerta de enlace de Internet de solo salida

Gateways de operador

Conjuntos de opciones de DHCP

Direcciones IP elásticas

Listas de prefijos administradas

Gateways NAT

Interconexiones

Servidores de ruta [Nueva](#)

▼ Seguridad

ACL de red

Grupos de seguridad

▼ PrivateLink y Lattice

Introducción [Actualizado](#)

sg-04192cc7601f8346 - educode-rds-sg

El grupo de seguridad (sg-04192cc7601f8346 / educode-rds-sg) se ha creado correctamente

Detalles

Nombre del grupo de seguridad [educode-rds-sg](#)

ID del grupo de seguridad [sg-04192cc7601f8346](#)

Descripción [Grupo de seguridad para RDS, acceso solo desde EC2](#)

ID de la VPC [vpc-0b35dacc1c39a717](#)

Propietario [359104569356](#)

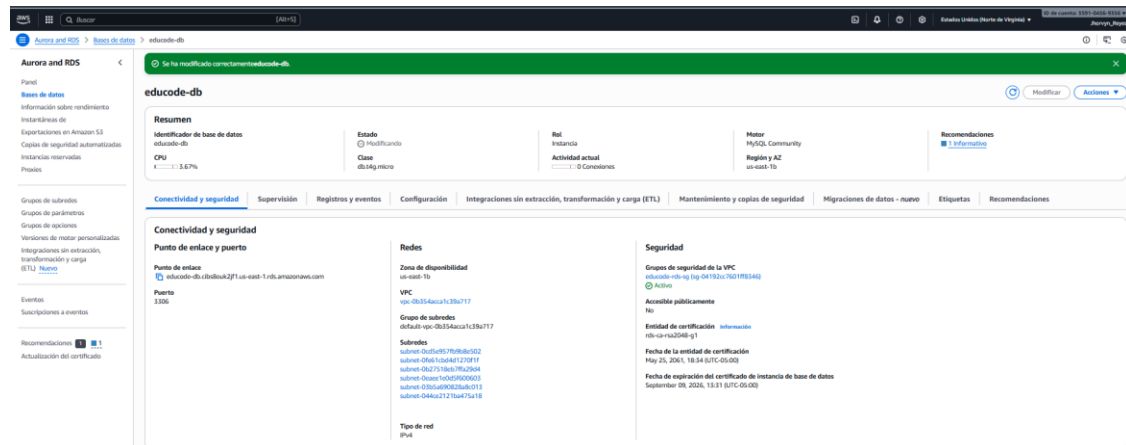
Número de reglas de entrada [1 Entrada de permisos](#)

Número de reglas de salida [1 Entrada de permisos](#)

Reglas de entrada Reglas de salida Compartiendo: [novedad](#) Asociaciones de VPC: [novedad](#) Etiquetas

Reglas de entrada (1) [Administrar etiquetas](#) [Editar reglas de entrada](#)

Nombre	ID de la regla del gr...	Versión de IP	Tipo	Protocolo	Intervalo de puertos	Origen	Descripción
-	sgr-083c8b0834ac244	-	MySQL/Aurora	TCP	3306	sg-0af701a1d8613a5c...	Solo desde EC2



Actividades para el Estudiante

1. ¿Qué ventajas encontraste al usar Terraform en lugar de configuraciones manuales?

- La principal ventaja es que me ahorró mucho tiempo porque pude crear toda la infraestructura de AWS con un solo archivo, sin tener que configurar cada recurso manualmente. Además, me permitió repetir la misma configuración en otras regiones sin errores

2. ¿Por qué es importante segmentar la red en subnets públicas y privadas?

- Esto es importante porque protege los recursos más sensibles, como la base de datos, dejándola en una subnet privada donde solo los servidores internos pueden acceder. Mientras tanto, los servidores que necesitan recibir tráfico desde internet están en la subnet pública.

3. ¿Qué aprendiste al integrar la base de datos RDS con los servicios EC2/ECS?

- Al principio fue confuso asociar subnets a las tablas de enrutamiento correctas y configurar los grupos de seguridad. Lo solucioné revisando cada subnet, asegurándome de que las privadas y públicas estuvieran bien asignadas, y creando un grupo de seguridad específico para RDS..

4. ¿Qué dificultades enfrentaste al configurar VPC y cómo las solucionaste?

- Tuve problemas al asociar subnets con las tablas de enrutamiento y al elegir grupos de seguridad. Lo solucioné revisando cuidadosamente las asociaciones de cada subnet y creando un grupo de seguridad específico para RDS.

5. ¿Cómo aplicarías este conocimiento en un entorno real de trabajo?

- Este conocimiento me serviría en un trabajo real para desplegar infraestructuras en la nube de manera segura y ordenada. Podría crear entornos replicables, controlar accesos y tráfico de la red, y mantener todo protegido sin depender de configuraciones manuales.