

Despliegue Cloud

Despliegue aplicación play-products-manager en
aws

Celia Herrera Ferreira
cherrerafe.inf@upsa.es

Marzo 2021

Descripción del sistema

En esta parte de la asignatura se va a utilizar el proveedor AWS para desplegar la aplicación que se creó en la anterior parte de la asignatura, en mi caso, la API REST play-products-manager desarrollada con el framework play.

Para ello se requerirá la utilización de herramientas como Packer y Terraform tanto para la creación y despliegue de la base de datos como del propio API REST. Se utilizarán también diferentes servicios ofrecidos por AWS, como son el servicio de bases de datos relacionales (RDS) y también el servicio EC2.

El despliegue se ha hecho en la zona de disponibilidad de París (eu-west-3).

Base de datos

Para poder realizar el despliegue completo de la aplicación, es necesario crear una base de datos que permita la persistencia de los datos que se van a guardar y gestionar en la aplicación. Esta base de datos, que en nuestro caso será una base de datos Postgres, se creará automáticamente utilizando Terraform, para ello se hace uso del script de Terraform existente en la carpeta RDS, el archivo *main.tf*.

En este fichero se configura que la base de datos sea de tipo Postgres y la versión (en este caso la 12.5), que tenga 20gb de memoria, el tipo de la instancia que en nuestro caso utilizamos la opción gratuita de db.t2.micro, además de indicar el usuario (postgres) y la contraseña de ese usuario (celiar00t). Por otro lado se configura el puerto de entrada (5432) y también se añade un grupo de seguridad, que en este caso permite el acceso a la instancia por el puerto 5432 de conexión TCP. Además se incluye en el fichero la configuración del servicio route53 para esta instancia, en el que se indica el

nombre de la zona primaria del la Hosted Zone además del nombre que se le va a dar a la dns estática, ese record es de tipo CNAME que es el que permite el renombramiento de la dns creada por la base de datos por el que se configura en este archivo.

En él está comprendida toda la creación de la instancia de base de datos. Esta base de datos se creará con un endpoint dado de manera automática por Amazon, el cual, posteriormente habrá que indicar en el despliegue de play a donde deberá apuntar. Para evitar este cambio manual se ha configurado el servicio ofrecido por Amazon de **route53**, con este servicio se pretende crear una Hosted Zone en la que se enlaza el endpoint generado de la base de datos por una dns estática a la que siempre se apuntaría en la aplicación. Aunque se ha llegado a poder crear la Hosted Zone, no ha sido posible acceder a esa dns estática. Esto es debido a que esa dns estática, aún siendo pública, debería de estar alojada en un dominio público, el cual solo es posible utilizarlo previo pago. Debido a esto, aunque se añade el desarrollo de la creación de la Hosted Zone, no se ha terminado realizando la conexión de esta manera, y a la hora del despliegue de la aplicación habrá que cambiar la url a la que apuntar.

productsmanager.db.com

Eliminar zona

Probar el registro

Configurar el registro de consultas

Info

► Detalles de la zona alojada

Editar zona alojada

Registros (3)

Firma DNSSEC

Etiquetas de zona hospedada (0)

Registros (3)

Info

El modo Automatic es el comportamiento actual de búsqueda que se ha optimizado para obtener los mejores resultados del filtrado. Para cambiar los modos, vaya a la configuración.

🔄

Eliminar registro

Importar archivo de zona

Crear un registro

🔍 Filtrar registros por propiedad o valor

Tipo ▼

Política de d... ▼

Alias ▼

< 1 > ⓘ

<input type="checkbox"/>	Nombre del registro ▼	Tipo ▼	Politic... ▼	Difer... ▼	Valor/Dirigir el tráfico a ▼
<input type="checkbox"/>	productsmanager.db.com	NS	Simple	-	ns-1810.awsdns-34.co.uk. ns-382.awsdns-47.com. ns-772.awsdns-32.net. ns-1209.awsdns-23.org.
<input type="checkbox"/>	productsmanager.db.com	SOA	Simple	-	ns-1810.awsdns-34.co.uk. awsdns-hostmaster.amazon.com. 1 7200
<input type="checkbox"/>	www.test.productsmanager.db.com	CNAME	Simple	-	first-hair-manager.ccxonjbmavx.eu-west-3.rds.amazonaws.com

RDS > Databases > first-hair-manager

first-hair-manager Modify Actions

Summary

DB identifier first-hair-manager	CPU 3.39%	Status Available	Class db.t2.micro
Role Instance	Current activity 0 Sessions	Engine PostgreSQL	Region & AZ eu-west-3a

Connectivity & security | Monitoring | Logs & events | Configuration | Maintenance & backups | Tags

Connectivity & security

Endpoint & port Endpoint first-hair-manager.ccwoxnjbmvx.eu-west-3.rds.amazonaws.com Port 5432	Networking Availability zone eu-west-3a VPC vpc-08c03460 Subnet group default-vpc-08c03460	Security VPC security groups first-manager-sg (sg-0253ecd1941cfe10) (active) Public accessibility No Certificate authority
--	---	--

SWs Servicios | Buscar servicios, características, productos del Marketplace y doc. [Opción+S] | CellaMIMO | Global | Soporte

1: Buscar un dominio | **Elegir un nombre de dominio** | Carro de la compra

2: Detalles de contacto | 3: Verificar y comprar

Para registrar un nombre de dominio, emplace buscando uno que esté disponible (como example.com), elija una extensión (como .com u .org) y haga clic en Comprobar.

Comprobar

Popular

- .com - 12,00 \$
- .net - 11,00 \$
- .org - 12,00 \$
- .com.au - 15,00 \$
- .co.uk - 9,00 \$
- .io - 39,00 \$
- .info - 21,00 \$
- .co - 25,00 \$
- .me - 9,00 \$

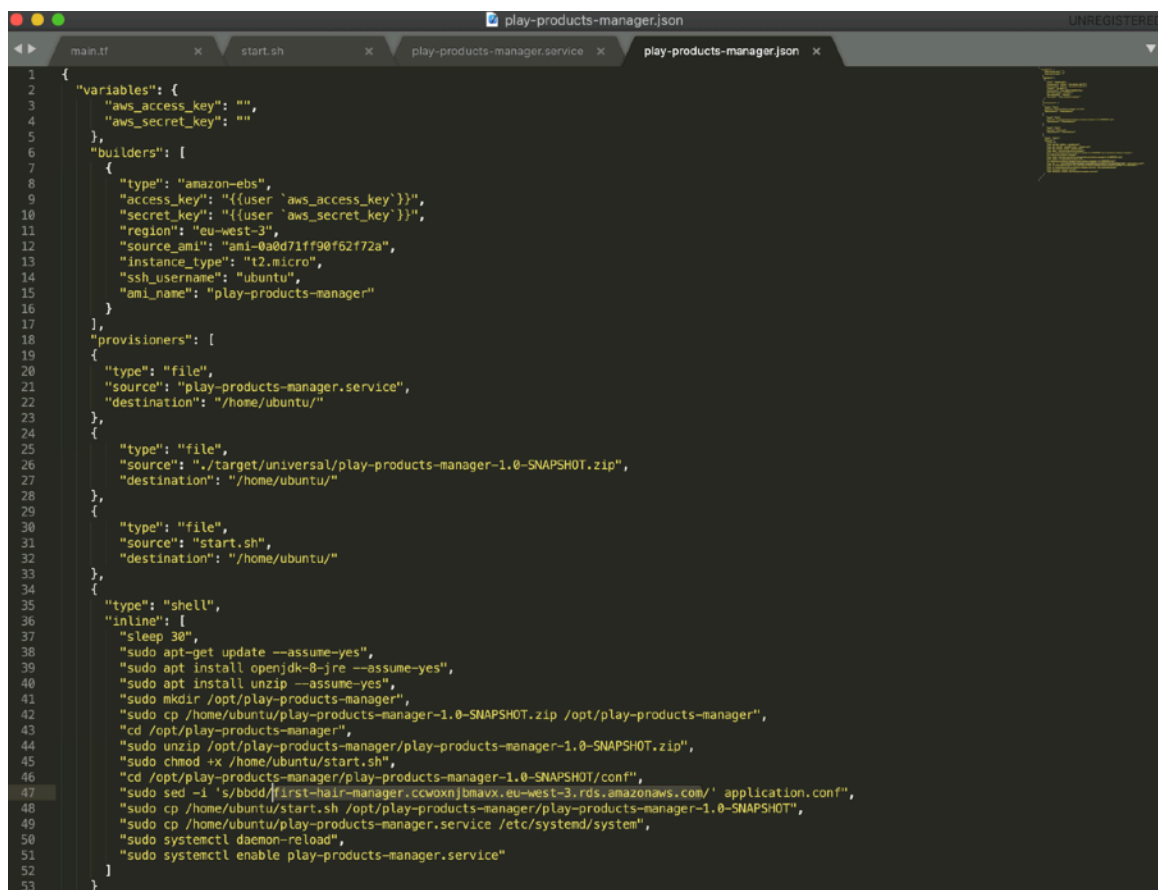
Cancelar Continuar

Posteriormente a la creación de la instancia RDS, habrá que generar la base de datos de manera manual. Para ello, habrá que acceder a la instancia creada mediante un cliente de Postgres, con el cual se creará la base de datos en ella, con el nombre de *productsmanager* y tras ello se deberá proceder a la introducción de todas las tablas de la base de datos, así como sus relaciones. Estas están contenidas en el archivo *ProductsManager.sql* contenido en la misma carpeta que el archivo *main.tf*.

Aplicación

Tras haber realizado el despliegue y configuración de la base de datos se procederá a desplegar la aplicación. Para realizar dicho despliegue lo primero que se hará será crear una AMI, es decir, una imagen de una máquina de Amazon, en nuestro caso de una instancia EC2. Esta AMI se creará de manera automática mediante la utilización de Packer, utilizando un fichero contenido en la carpeta de Aplicación, en concreto el fichero *play-products-manager.json*.

Antes de proceder a ejecutarlo habrá que editarlo cambiando el endpoint de la base de datos que se nos ha creado automáticamente en el proceso anterior.



```
1 {
2   "variables": {
3     "aws_access_key": "",
4     "aws_secret_key": ""
5   },
6   "builders": [
7     {
8       "type": "amazon-ebs",
9       "access_key": "{{user `aws_access_key`}}",
10      "secret_key": "{{user `aws_secret_key`}}",
11      "region": "eu-west-3",
12      "source_ami": "ami-0a0d71ff90f62f72a",
13      "instance_type": "t2.micro",
14      "ssh_username": "ubuntu",
15      "ami_name": "play-products-manager"
16    }
17  ],
18  "provisioners": [
19    {
20      "type": "file",
21      "source": "play-products-manager.service",
22      "destination": "/home/ubuntu/"
23    },
24    {
25      "type": "file",
26      "source": "../target/universal/play-products-manager-1.0-SNAPSHOT.zip",
27      "destination": "/home/ubuntu/"
28    },
29    {
30      "type": "file",
31      "source": "start.sh",
32      "destination": "/home/ubuntu/"
33    },
34    {
35      "type": "shell",
36      "inline": [
37        "sleep 30",
38        "sudo apt-get update --assume-yes",
39        "sudo apt install openjdk-8-jre --assume-yes",
40        "sudo apt install unzip --assume-yes",
41        "sudo mkdir /opt/play-products-manager",
42        "sudo cp /home/ubuntu/play-products-manager-1.0-SNAPSHOT.zip /opt/play-products-manager",
43        "cd /opt/play-products-manager",
44        "sudo unzip /opt/play-products-manager/play-products-manager-1.0-SNAPSHOT.zip",
45        "sudo chmod +x /home/ubuntu/start.sh",
46        "cd /opt/play-products-manager/play-products-manager-1.0-SNAPSHOT/conf",
47        "sudo sed -i 's/bbdd/first-hair-manager.ccvoxnjbnavx.eu-west-3.rds.amazonaws.com/' application.conf",
48        "sudo cp /home/ubuntu/start.sh /opt/play-products-manager/play-products-manager-1.0-SNAPSHOT",
49        "sudo cp /home/ubuntu/play-products-manager.service /etc/systemd/system",
50        "sudo systemctl daemon-reload",
51        "sudo systemctl enable play-products-manager.service"
52      ]
53    }
54  ]
55 }
```

La parte señalada es el endpoint a cambiar

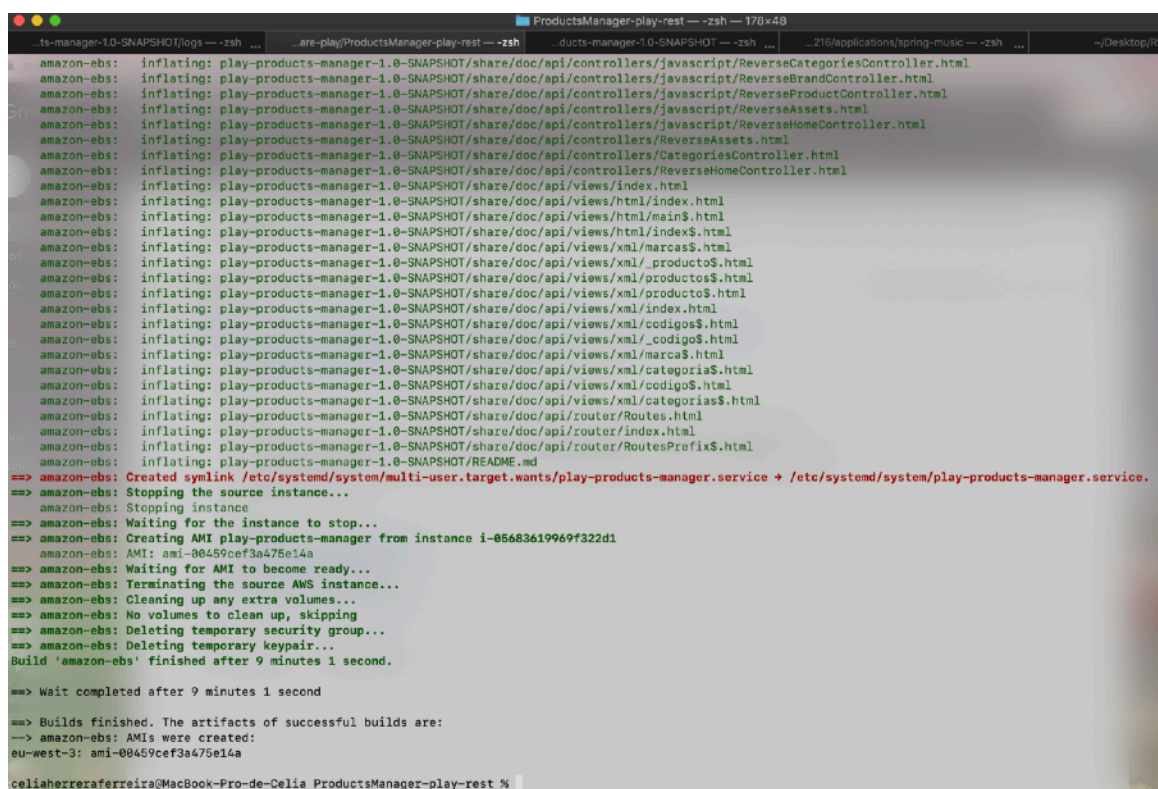
Este cambio es necesario ya que cuando se ejecute Packer este valor cambiará dentro de la configuración de la aplicación Play para que de esta

manera se apunte correctamente a la base de datos. En mi caso he realizado el despliegue en la zona de disponibilidad de París (eu-west-3) por lo que la `source_ami` introducido depende de la instancia que tomé como base en esa zona de disponibilidad.

Dentro de este fichero `play-products-manager.json` se puede observar que está configurada la subida de un servicio (`play-products-manager.service`) que permitirá la ejecución de la automática de la aplicación y que se encuentra también en la carpeta, la aplicación propiamente dicha comprimida en formato zip (ya que aunque el desarrollo sea en Java, Play es un framework basado en Scala) y el script `start.sh` (también contenido en la carpeta) que realizará la ejecución y correcta configuración del puerto cuando el servicio (`play-products-manager.service`) inicie el despliegue. Se puede comprobar que tras la subida están todos los pasos necesarios descritos para realizar la correcta configuración de la imagen para que el despliegue de ella sea automático.

Para ejecutar este archivo se deberá ejecutar introduciendo las variables tanto de la clave privada de aws como de la clave secreta con el comando:

```
Packer build -var 'aws_access_key=$ACCESSKEY' -var 'aws_secret_key=$SECRETKEY'
```



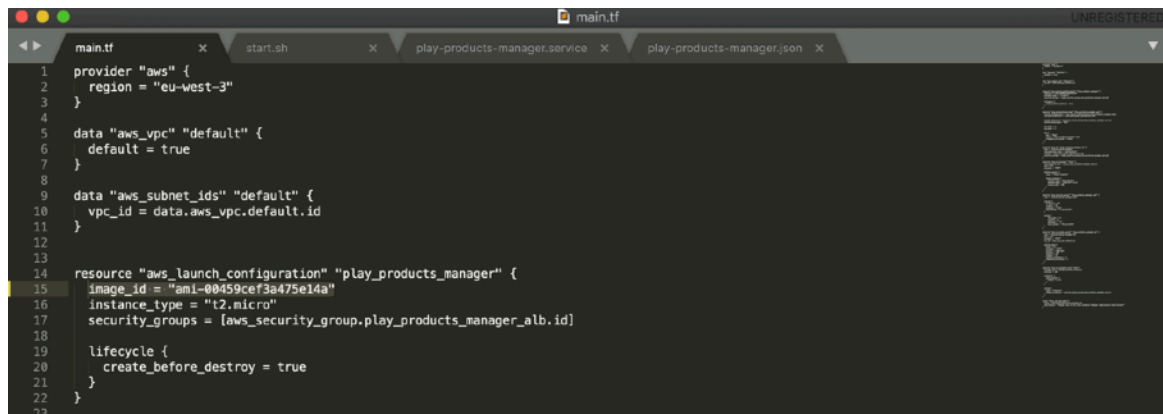
```
ts-manager-1.0-SNAPSHOT/logs --- zsh ... .are-playProductsManager-play-rest --- zsh ... ducts-manager-1.0-SNAPSHOT --- zsh ... ..216/applications/spring-music --- zsh ... ~/Desktop/RTD
amazon-ebs: inflating: play-products-manager-1.0-SNAPSHOT/share/doc/api/controllers/javascript/ReverseCategoriesController.html
amazon-ebs: inflating: play-products-manager-1.0-SNAPSHOT/share/doc/api/controllers/javascript/ReverseBrandController.html
amazon-ebs: inflating: play-products-manager-1.0-SNAPSHOT/share/doc/api/controllers/javascript/ReverseProductController.html
amazon-ebs: inflating: play-products-manager-1.0-SNAPSHOT/share/doc/api/controllers/javascript/ReverseAssets.html
amazon-ebs: inflating: play-products-manager-1.0-SNAPSHOT/share/doc/api/controllers/javascript/ReverseHomeController.html
amazon-ebs: inflating: play-products-manager-1.0-SNAPSHOT/share/doc/api/controllers/ReverseAssets.html
amazon-ebs: inflating: play-products-manager-1.0-SNAPSHOT/share/doc/api/controllers/CategoriesController.html
amazon-ebs: inflating: play-products-manager-1.0-SNAPSHOT/share/doc/api/controllers/ReverseHomeController.html
amazon-ebs: inflating: play-products-manager-1.0-SNAPSHOT/share/doc/api/views/index.html
amazon-ebs: inflating: play-products-manager-1.0-SNAPSHOT/share/doc/api/views/html/index.html
amazon-ebs: inflating: play-products-manager-1.0-SNAPSHOT/share/doc/api/views/html/main.html
amazon-ebs: inflating: play-products-manager-1.0-SNAPSHOT/share/doc/api/views/html/index$.html
amazon-ebs: inflating: play-products-manager-1.0-SNAPSHOT/share/doc/api/views/xml/marcas$.html
amazon-ebs: inflating: play-products-manager-1.0-SNAPSHOT/share/doc/api/views/xml/productos$.html
amazon-ebs: inflating: play-products-manager-1.0-SNAPSHOT/share/doc/api/views/xml/productos$.html
amazon-ebs: inflating: play-products-manager-1.0-SNAPSHOT/share/doc/api/views/xml/index.html
amazon-ebs: inflating: play-products-manager-1.0-SNAPSHOT/share/doc/api/views/xml/codigos$.html
amazon-ebs: inflating: play-products-manager-1.0-SNAPSHOT/share/doc/api/views/xml/codigos$.html
amazon-ebs: inflating: play-products-manager-1.0-SNAPSHOT/share/doc/api/views/xml/marca$.html
amazon-ebs: inflating: play-products-manager-1.0-SNAPSHOT/share/doc/api/views/xml/categorias$.html
amazon-ebs: inflating: play-products-manager-1.0-SNAPSHOT/share/doc/api/views/xml/codigos$.html
amazon-ebs: inflating: play-products-manager-1.0-SNAPSHOT/share/doc/api/router/Routes.html
amazon-ebs: inflating: play-products-manager-1.0-SNAPSHOT/share/doc/api/router/index.html
amazon-ebs: inflating: play-products-manager-1.0-SNAPSHOT/share/doc/api/router/RoutesPrefix$.html
amazon-ebs: inflating: play-products-manager-1.0-SNAPSHOT/README.md
==> amazon-ebs: Created symlink /etc/systemd/system/multi-user.target.wants/play-products-manager.service → /etc/systemd/system/play-products-manager.service.
==> amazon-ebs: Stopping the source instance...
amazon-ebs: Stopping instance
==> amazon-ebs: Waiting for the instance to stop...
==> amazon-ebs: Creating AMI play-products-manager from instance i-05683619969f322d1
amazon-ebs: AMI: ami-00459cef3a475e14a
==> amazon-ebs: Waiting for AMI to become ready...
==> amazon-ebs: Terminating the source AWS instance...
==> amazon-ebs: Cleaning up any extra volumes...
==> amazon-ebs: No volumes to clean up, skipping
==> amazon-ebs: Deleting temporary security group...
==> amazon-ebs: Deleting temporary keypair...
Build 'amazon-ebs' finished after 9 minutes 1 second.

==> Wait completed after 9 minutes 1 second

==> Builds finished. The artifacts of successful builds are:
--> amazon-ebs: AMIs were created:
eu-west-3: ami-00459cef3a475e14a

celiahereraferreira@MacBook-Pro-de-Celia ProductsManager-play-rest %
```

Tras esta ejecución se procede a ejecutar el fichero de Terraform (*main.tf*), el cual se encuentra también contenido en la carpeta y que habrá que editar para introducir el id de la AMI que se haya creado y sustituirlo en la variable "image_id" para que la tome de referencia, que será distinta en cada ejecución.



```
1 provider "aws" {
2   region = "eu-west-3"
3 }
4
5 data "aws_vpc" "default" {
6   default = true
7 }
8
9 data "aws_subnet_ids" "default" {
10  vpc_id = data.aws_vpc.default.id
11 }
12
13
14 resource "aws_launch_configuration" "play_products_manager" {
15  image_id = "ami-00459cef3a475e14a"
16  instance_type = "t2.micro"
17  security_groups = [aws_security_group.play_products_manager_alb.id]
18
19  lifecycle {
20    create_before_destroy = true
21  }
22 }
23
```

En este fichero se encuentra toda la configuración que se va a realizar para el despliegue de Cloud final de la aplicación. En él se puede observar que se va a crear un grupo de autoescalado, para tres instancias de EC2, ya que de momento se cree conveniente que existan tres instancias para soportar la capacidad deseada de uso de la aplicación, entre otras cosas por la arquitectura en la que se ha desarrollado la aplicación. En caso de que una falle, Amazon se encargará de crear otra para suplirla. Se creará también un balanceador que permita la distribución automática del tráfico a la aplicación entre las tres instancias que se crearán. También se crea un listener para ese balanceador, que en caso de que exista algún problema con el servicio, ofrecerá una pantalla de fallo con el error 404 (para nuestro caso en HTTP solamente). Se configuran también unos grupos de seguridad que permitirán la gestión de las instancias, en concreto se permitirá el acceso por el puerto 80 desde cualquier punto. Por otro lado, se creará un grupo de destino que permita al balanceador enviar las solicitudes a los destinos registrados, en nuestro caso por el puerto 80 con el protocolo HTTP. En este Target group se configura un comprobación del estado de las instancias, donde se va a esperar

durante 60 segundos la respuesta correcta de cada instancia en intervalos de 30 segundos. En el caso de que una instancia falle durante 3 veces seguidas, se creará otra para suplirla, debido a que se considera que esa instancia no está funcionando correctamente y se terminará con ella. Además, para comprobar que una instancia está funcionando correctamente, deberá de responder 3 veces consecutivas de manera correcta. De esta manera se espera obtener una tolerancia a fallos correcta de la aplicación.

En el caso de que en algún momento se decida que no es suficiente con ese número de instancias para el servicio, se podrá aumentar o disminuir gracias a este archivo.

Para ejecutar este archivo de Terraform simplemente habrá que tener introducidas de igual manera la clave privada y la clave secreta, y tras iniciar Terraform con `terraform init`, se podrá comprobar todas las configuraciones que se van a realizar con el comando `terraform plan`, y finalmente se ejecutará con el comando `terraform apply`.

Tras su ejecución aparecerá la dns a la que acceder para ver el despliegue realizado de la aplicación.



Welcome to haircare product manager


```
ProductsManager-play-rest — zsh — 178x75
...s-manager-10-SNAPSHOT/logs — zsh ... ..are-play/ProductsManager-play-rest — zsh ... ..ducts-manager-10-SNAPSHOT — zsh ... ...216/applications/spring-music — zsh ... ~/Desktop/RDS — zsh ... 4
aws_lb.play_products_manager_lb: Still creating... [1m40s elapsed]
aws_lb.play_products_manager_lb: Still creating... [1m50s elapsed]
aws_lb.play_products_manager_lb: Still creating... [2m0s elapsed]
aws_lb.play_products_manager_lb: Still creating... [2m10s elapsed]
aws_lb.play_products_manager_lb: Creation complete after 2m13s [id=arn:aws:elasticloadbalancing:eu-west-3:588938888761:loadbalancer/app/play-products-manager/ef27aaf3a96d4cc6]
aws_lb.listener.http: Creating...
aws_lb.listener.http: Creation complete after 0s [id=arn:aws:elasticloadbalancing:eu-west-3:588938888761:listener/app/play-products-manager/ef27aaf3a96d4cc6/e4284cbac9de62d1/2c68e5fb4641bcd6]
aws_lb.listener_rule.asp: Creating...
aws_lb.listener_rule.asp: Creation complete after 0s [id=arn:aws:elasticloadbalancing:eu-west-3:588938888761:listener-rule/app/play-products-manager/ef27aaf3a96d4cc6/e4284cbac9de62d1/2c68e5fb4641bcd6]

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

Outputs:
play_alb_dns_name = {
  "access_logs" = tolist([
    {
      "bucket" = ""
      "enabled" = false
      "prefix" = ""
    },
  ])
  "arn" = "arn:aws:elasticloadbalancing:eu-west-3:588938888761:loadbalancer/app/play-products-manager/ef27aaf3a96d4cc6"
  "arn_suffix" = "app/play-products-manager/ef27aaf3a96d4cc6"
  "customer_owned_ipv4_pool" = ""
  "dns_name" = "play-products-manager-273487466.eu-west-3.elb.amazonaws.com"
  "drop_invalid_header_fields" = false
  "enable_cross_zone_load_balancing" = tobool(null)
  "enable_deletion_protection" = false
  "enable_http2" = true
  "id" = "arn:aws:elasticloadbalancing:eu-west-3:588938888761:loadbalancer/app/play-products-manager/ef27aaf3a96d4cc6"
  "idle_timeout" = 60
  "internal" = false
  "ip_address_type" = "ipv4"
  "load_balancer_type" = "application"
  "name" = "play-products-manager"
  "name_prefix" = tostring(null)
  "security_groups" = toset([
    "sg-89299b12192ca6d6be",
  ])
  "subnet_mapping" = toset([
    {
      "allocation_id" = ""
      "ipv6_address" = ""
      "outpost_id" = ""
      "private_ipv4_address" = ""
      "subnet_id" = "subnet-1d8ba166"
    },
    {
      "allocation_id" = ""
      "ipv6_address" = ""
      "outpost_id" = ""
      "private_ipv4_address" = ""
      "subnet_id" = "subnet-3596c978"
    },
    {
      "allocation_id" = ""
      "ipv6_address" = ""
      "outpost_id" = ""
      "private_ipv4_address" = ""
      "subnet_id" = "subnet-b4e8fbdd"
    },
  ])
  "subnets" = toset([
    "subnet-1d8ba166",
    "subnet-3596c978",
    "subnet-b4e8fbdd",
  ])
  "tags" = tomap(null) /* of string */
  "timeouts" = null /* object */
  "vpc_id" = "vpc-08c03460"
  "zone_id" = "Z3077PNBOS71R4"
}
celiaherreraferreira@MacBook-Pro-de-Celia ProductsManager-play-rest % terraform destroy
```

Otras consideraciones

El desarrollo de la práctica se encuentra tanto en los ficheros que se adjuntan en la entrega como en un repositorio de GitHub en la rama de deploy.

<https://github.com/CeliaHFerreira/ProductsManager-play-rest/tree/deploy>

En los archivos entregados está por una parte la carpeta RDS que contiene tanto el fichero de terraform para la creación de la base de datos como el propio archivo SQL de la base de datos.

En la carpeta Archivos despliegue se encuentran separados los archivos `start.sh`, `play-products-manager.json`, `play-products-manager.service`, `main.tf` y el proyecto para poder acceder a ellos de manera rápida, el despliegue aquí explicado si se desea replicar debe hacerse desde el fichero comprimido `ProductsManager-play-rest.zip`.