



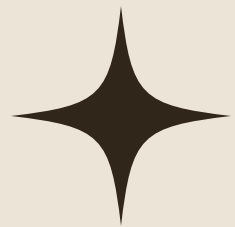
LABORATORIO

179

Ignacio Suárez

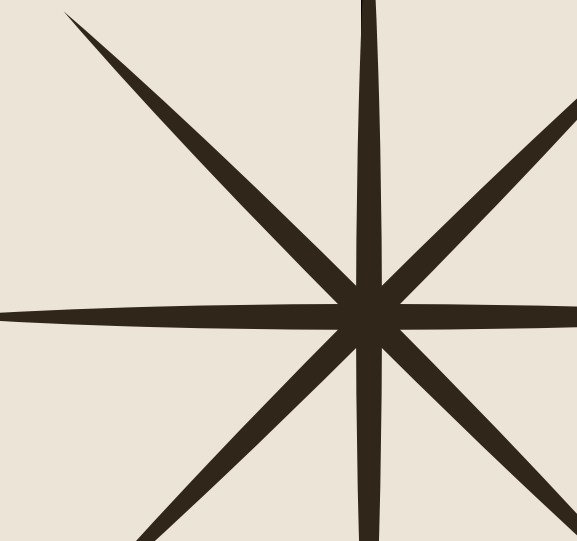
Objetivos // del laboratorio

- Crear una instancia MariaDB de Amazon RDS mediante el uso de la AWS CLI.
- Migrar datos de una base de datos MariaDB en una instancia de EC2 a una instancia MariaDB de Amazon RDS.
- Supervisar la instancia de Amazon RDS mediante las métricas de Amazon CloudWatch.





Tarea 1: generar datos de pedido en el sitio web de la cafetería

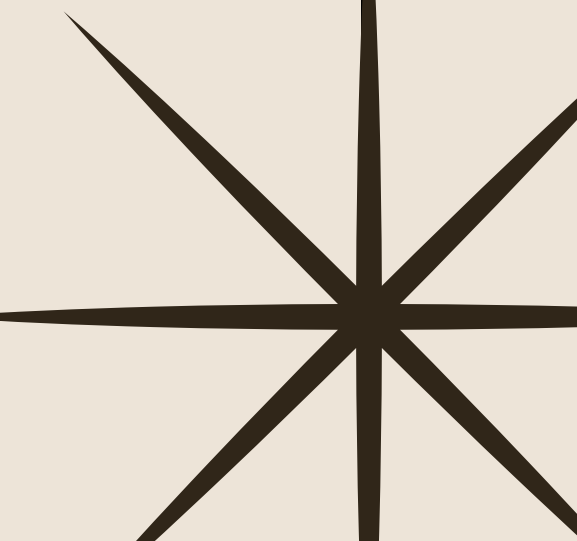
- En esta tarea, navegará por el sitio web de la cafetería y realizará algunos pedidos que están almacenados en la base de datos existente. Al realizar pedidos se crean datos para la aplicación antes de que esta se migre a una instancia de Amazon RDS nueva.
 - Ahora, abra la aplicación web de la cafetería y realice algunos pedidos.
- 

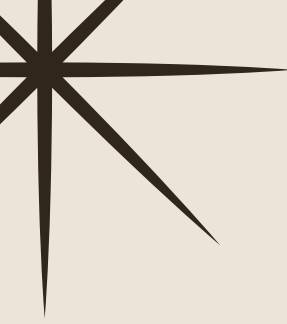
Pedidos realizados en la app

Order Number: 3		Date: 2024-07-02	Time: 19:00:26	Total Amount: \$6.00	
Item		Price	Quantity	Amount	
Chocolate Chip Cookie		\$2.50	1	\$2.50	
Strawberry Tart		\$3.50	1	\$3.50	
Order Number: 2		Date: 2024-07-02	Time: 19:00:21	Total Amount: \$13.50	
Item		Price	Quantity	Amount	
Donut		\$1.00	1	\$1.00	
Muffin		\$3.00	1	\$3.00	
Strawberry Blueberry Tart		\$3.50	1	\$3.50	
Coffee		\$3.00	1	\$3.00	
Hot Chocolate		\$3.00	1	\$3.00	
Order Number: 1		Date: 2024-07-02	Time: 19:00:07	Total Amount: \$14.50	
Item		Price	Quantity	Amount	
Croissant		\$1.50	1	\$1.50	
Donut		\$1.00	1	\$1.00	
Chocolate Chip Cookie		\$2.50	1	\$2.50	
Coffee		\$3.00	1	\$3.00	
Hot Chocolate		\$3.00	1	\$3.00	
Latte		\$3.50	1	\$3.50	



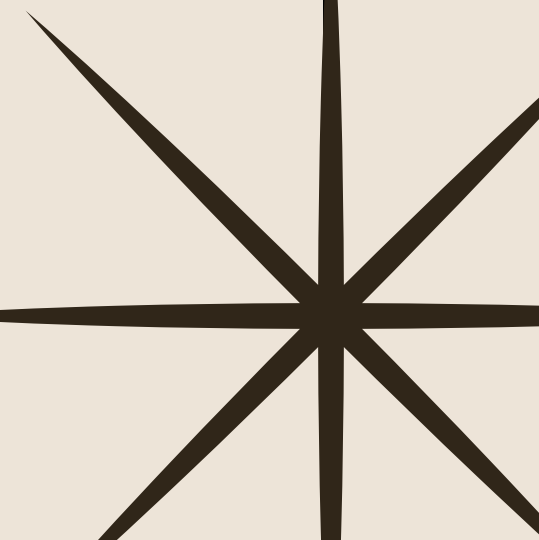
Tarea 2: crear una instancia de Amazon RDS mediante el uso de la AWS CLI

- En esta tarea, creará una instancia de Amazon RDS mediante el uso de la AWS CLI. Para comenzar, utilizará EC2 Instance Connect para conectarse de forma segura con la instancia CLI Host que ya se le aprovisionó. Esta instancia ya cuenta con la AWS CL instalada como parte del aprovisionamiento. Luego puede ejecutar los comandos de la AWS CLI para hacer lo siguiente:
 - Configurar la AWS CLI.
 - Crear los siguientes componentes que se requieren previamente para crear la instancia de Amazon RDS:
 - Un firewall del grupo de seguridad para la instancia de Amazon RDS
 - Dos subredes privadas y un grupo de subredes de base de datos
 - Crear la instancia MariaDB de Amazon RDS.
- 



Tarea 2.1: conectarse a la instancia CLI Host

- En esta tarea, usará EC2 Instance Connect para conectarse a la instancia CLI Host de EC2. Utilizará esta instancia para ejecutar comandos de la AWS CLI.



Conexión con la instancia

Connect to instance [Info](#)

Connect to your instance i-0c576f6fb13d42186 (CLI Host) using any of these options


EC2 Instance Connect

Session Manager

SSH client

EC2 serial console

Instance ID


 i-0c576f6fb13d42186 (CLI Host)

Connection Type

☒ **Connect using EC2 Instance Connect**
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.


☐ **Connect using EC2 Instance Connect Endpoint**
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address

 54.188.97.33

Username

Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ec2-user.

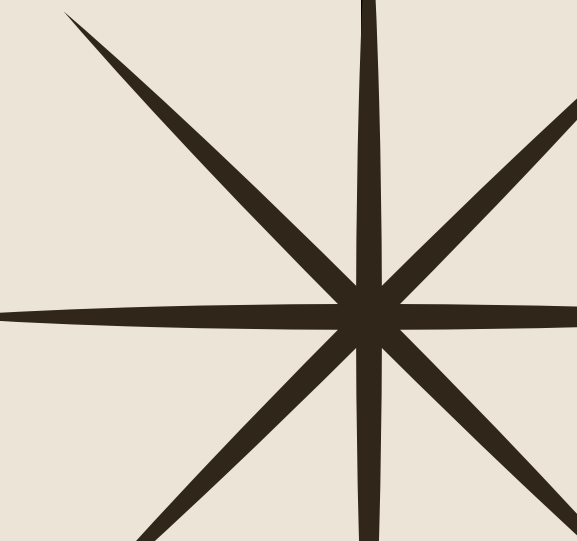
 **Note:** In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Cancel

Connect



Tarea 2.2: configurar la AWS CLI

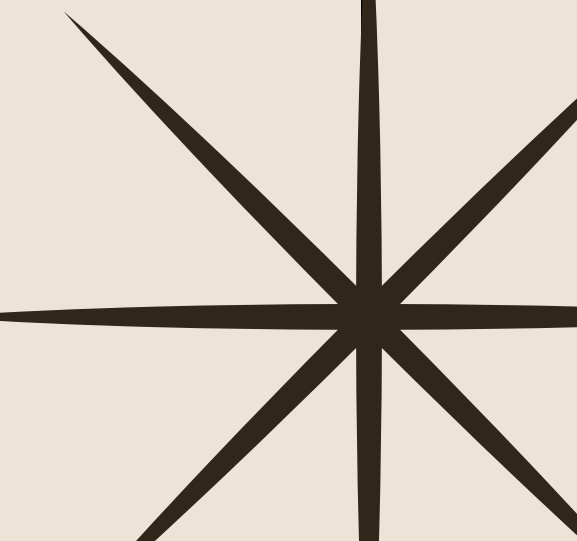
- En esta tarea, configurará la AWS CLI estableciendo los parámetros de configuración que se le proporcionaron cuando se aprovisionó el laboratorio. Después de la configuración, ejecutará los comandos de la AWS CLI para interactuar con servicios de AWS.
- 

Configuración de AWS en la instancia

```
[ec2-user@ip-10-200-0-129 ~]$ aws configure
AWS Access Key ID [None]: AKIAQ2Z4X76O2SHOJWBV
AWS Secret Access Key [None]: 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
Default region name [None]: us-west-2
Default output format [None]: json
[ec2-user@ip-10-200-0-129 ~]$
```



Tarea 2.3: crear componentes de requisitos previos

- En esta tarea, creará los componentes de infraestructura de requisitos previos para la instancia de Amazon RDS. En concreto, creará los siguientes componentes que se muestran en el diagrama de arquitectura final:
 - CafeDatabaseSG (grupo de seguridad para la base de datos de Amazon RDS)
 - Subred privada 1 de CafeDB
 - Subred privada 2 de CafeDB
 - Grupo de subred de CafeDB (Grupo de subredes de la base de datos)
 - Ahora ejecutará los comandos de la AWS CLI en el terminal de EC2 Instance Connect.
- 

Creación del security group

```
[ec2-user@ip-10-200-0-129 ~]$ aws ec2 create-security-group --group-name CafeDatabaseSG --description "Security group for Cafe database" --vpc-id vpc-0664867d0753d2698
{
  "GroupId": "sg-0500336532737267a"
}
[ec2-user@ip-10-200-0-129 ~]$
```

Crear regla de entrada

```
[ec2-user@ip-10-200-0-129 ~]$
[ec2-user@ip-10-200-0-129 ~]$ aws ec2 authorize-security-group-ingress --group-id sg-0500336532737267a --protocol tcp --port 3306 --source-group sg-0067bd925e5304a52
[ec2-user@ip-10-200-0-129 ~]$
```

Vemos
la regla

```
[ec2-user@ip-10-200-0-129 ~]$ aws ec2 describe-security-groups --query "SecurityGroups[*].[GroupName,GroupId,IpPermissions]" --filters "Name=group-name,Values='CafeDatabaseSG'"
[
  [
    "CafeDatabaseSG",
    "sg-0500336532737267a",
    [
      {
        "PrefixListIds": [],
        "FromPort": 3306,
        "IpRanges": [],
        "ToPort": 3306,
        "IpProtocol": "tcp",
        "UserIdGroupPairs": [
          {
            "UserId": "057572589469",
            "GroupId": "sg-0067bd925e5304a52"
          }
        ],
        "Ipv6Ranges": []
      }
    ]
  ]
]
[ec2-user@ip-10-200-0-129 ~]$
```

Creamos la subred

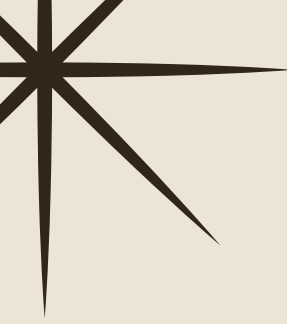
```
[ec2-user@ip-10-200-0-129 ~]$ aws ec2 create-subnet --vpc-id vpc-0664867d0753d2698 --cidr-block 10.200.2.0/23 --availability-zone us-west-2a
{
  "Subnet": {
    "MapPublicIpOnLaunch": false,
    "AvailabilityZoneId": "usw2-az1",
    "AvailableIpAddressCount": 507,
    "DefaultForAz": false,
    "SubnetArn": "arn:aws:ec2:us-west-2:057572589469:subnet/subnet-0463cc64ba237f285",
    "Ipv6CidrBlockAssociationSet": [],
    "VpcId": "vpc-0664867d0753d2698",
    "State": "available",
    "AvailabilityZone": "us-west-2a",
    "SubnetId": "subnet-0463cc64ba237f285",
    "OwnerId": "057572589469",
    "CidrBlock": "10.200.2.0/23",
    "AssignIpv6AddressOnCreation": false
  }
}
[ec2-user@ip-10-200-0-129 ~]$
```

Creamos la otra subred

```
[ec2-user@ip-10-200-0-129 ~]$ aws ec2 create-subnet --vpc-id vpc-0664867d0753d2698 --cidr-block 10.200.10.0/23 --availability-zone us-west-2b
{
  "Subnet": {
    "MapPublicIpOnLaunch": false,
    "AvailabilityZoneId": "usw2-az2",
    "AvailableIpAddressCount": 507,
    "DefaultForAz": false,
    "SubnetArn": "arn:aws:ec2:us-west-2:057572589469:subnet/subnet-0775bbae270d3246d",
    "Ipv6CidrBlockAssociationSet": [],
    "VpcId": "vpc-0664867d0753d2698",
    "State": "available",
    "AvailabilityZone": "us-west-2b",
    "SubnetId": "subnet-0775bbae270d3246d",
    "OwnerId": "057572589469",
    "CidrBlock": "10.200.10.0/23",
    "AssignIpv6AddressOnCreation": false
  }
}
[ec2-user@ip-10-200-0-129 ~]$
```

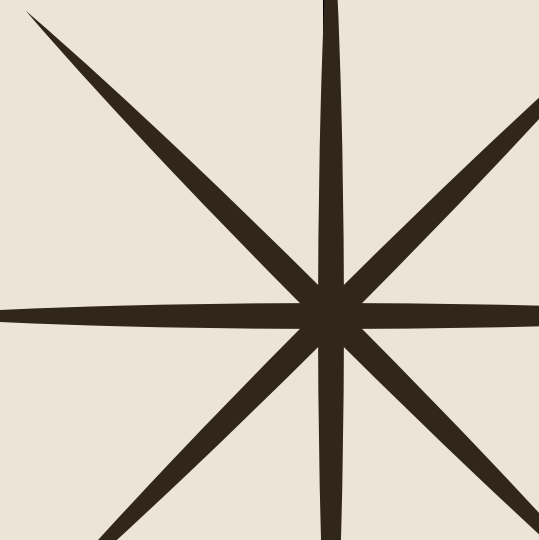
Creamos la subnet de DB

```
[ec2-user@ip-10-200-0-129 ~]$ aws rds create-db-subnet-group --db-subnet-group-name "CafeDB Subnet Group" --db-subnet-group-description "DB subnet group for Cafe" --subnet-ids subnet-0463cc64ba237f285 subnet-0775bbae270d3246d --tags "Key=Name,Value= CafeDatabaseSubnetGroup"
{
  "DBSubnetGroup": {
    "Subnets": [
      {
        "SubnetStatus": "Active",
        "SubnetIdentifier": "subnet-0775bbae270d3246d",
        "SubnetOutpost": {},
        "SubnetAvailabilityZone": {
          "Name": "us-west-2b"
        }
      },
      {
        "SubnetStatus": "Active",
        "SubnetIdentifier": "subnet-0463cc64ba237f285",
        "SubnetOutpost": {},
        "SubnetAvailabilityZone": {
          "Name": "us-west-2a"
        }
      }
    ],
    "VpcId": "vpc-0664867d0753d2698",
    "DBSubnetGroupDescription": "DB subnet group for Cafe",
    "SubnetGroupStatus": "Complete",
    "DBSubnetGroupArn": "arn:aws:rds:us-west-2:057572589469:subgrp:cafedb subnet group",
    "DBSubnetGroupName": "cafedb subnet group"
  }
}
[ec2-user@ip-10-200-0-129 ~]$
```



Tarea 2.4: crear la instancia de MariaDB de Amazon RDS

- Ahora puede crear la CafeDBInstance que aparece en la arquitectura final. Con la AWS CLI, creará una instancia MariaDB de Amazon RDS con los siguientes ajustes de configuración




Creamos la instancia RDS


```
[ec2-user@ip-10-200-0-129 ~]$ aws rds create-db-instance \
> --db-instance-identifier CafeDBInstance \
> --engine mariadb \
> --db-instance-class db.t3.micro \
> --allocated-storage 20 \
> --availability-zone us-west-2a \
> --db-subnet-group-name "CafeDB Subnet Group" \
> --vpc-security-group-ids sg-0500336532737267a \
> --no-publicly-accessible \
> --master-username root --master-user-password 'Re:Start!9'
{
  "DBInstance": {
    "PubliclyAccessible": false,
    "MasterUsername": "root",
    "MonitoringInterval": 0,
    "LicenseModel": "general-public-license",
    "VpcSecurityGroups": [
      {
        "Status": "active",
        "VpcSecurityGroupId": "sg-0500336532737267a"
      }
    ],
    "CopyTagsToSnapshot": false,
    "OptionGroupMemberships": [
      {
        "Status": "in-sync",
        "OptionGroupName": "default:mariadb-10-11"
      }
    ],
    "PendingModifiedValues": {
      "MasterUserPassword": "*****"
    }
  }
}
```


Revisamos el estado de la BD


```
[ec2-user@ip-10-200-0-129 ~]$ aws rds describe-db-instances \
> --db-instance-identifier CafeDBInstance \
> --query "DBInstances[*].[Endpoint.Address,AvailabilityZone,PreferredBackupWindow,BackupRetentionPeriod,DBInstanceStatus]"
[
  [
    null,
    "us-west-2a",
    "12:46-13:16",
    1,
    "creating"
  ]
]
[ec2-user@ip-10-200-0-129 ~]$
```

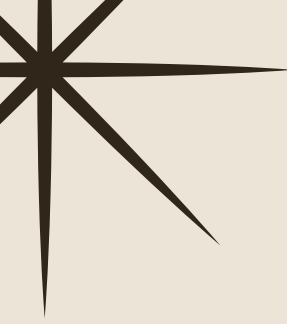


```
[ec2-user@ip-10-200-0-129 ~]$ aws rds describe-db-instances --db-instance-identifier CafeDBInstance --query "DBInstances[*].[Endpoint.Address,AvailabilityZone,PreferredBackupWindow,BackupRetentionPeriod,DBInstanceStatus]"
[
  [
    "cafedbinstance.czptnzn9net.us-west-2.rds.amazonaws.com",
    "us-west-2a",
    "12:46-13:16",
    1,
    "backing-up"
  ]
]
[ec2-user@ip-10-200-0-129 ~]$
```



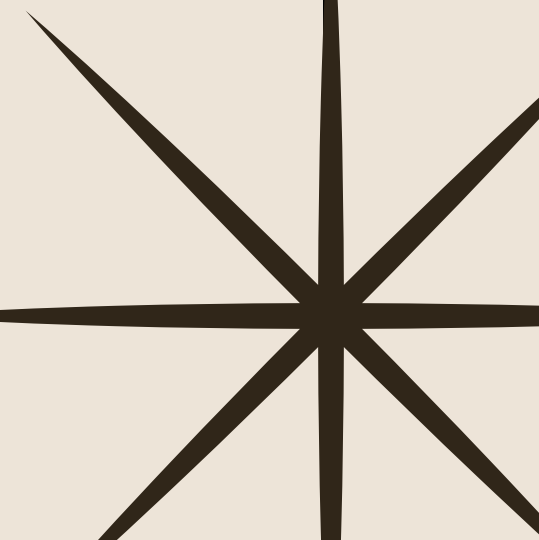
```
[ec2-user@ip-10-200-0-129 ~]$ aws rds describe-db-instances --db-instance-identifier CafeDBInstance --query "DBInstances[*].[Endpoint.Address,AvailabilityZone,PreferredBackupWindow,BackupRetentionPeriod,DBInstanceStatus]"
[
  [
    "cafedbinstance.czptnzn9net.us-west-2.rds.amazonaws.com",
    "us-west-2a",
    "12:46-13:16",
    1,
    "available"
  ]
]
[ec2-user@ip-10-200-0-129 ~]$
```





Tarea 3: migrar los datos de la aplicación a la instancia de Amazon RDS

- En esta tarea, migrará los datos de la base de datos local existente a la base de datos de Amazon RDS recién creada. Específicamente, haga lo siguiente:
 - Conéctese a CafeInstance mediante EC2 Instance Connect.
 - Utilice la herramienta mysqldump para crear un respaldo de la base de datos local.
 - Restaure el respaldo en la base de datos de Amazon RDS.
 - Pruebe la migración de datos.



Creamos un backup de la BD local

```
[ec2-user@ip-10-200-0-49 ~]$ mysqldump --user=root --password='Re:Start!9' \  
> --databases cafe_db --add-drop-database > cafedb-backup.sql  
[ec2-user@ip-10-200-0-49 ~]$
```

Restauramos los datos del respaldo de la RDS

```
[ec2-user@ip-10-200-0-49 ~]$ mysql --user=root --password='Re:Start!9' \  
> --host=cafedbinstance.czptnzn9net.us-west-2.rds.amazonaws.com \  
> < cafedb-backup.sql  
[ec2-user@ip-10-200-0-49 ~]$
```

Verificamos que la BD en RDS se creó y rellenó correctamente

```
[ec2-user@ip-10-200-0-49 ~]$ mysql --user=root --password='Re:Start!9' \
> --host=cafeINSTANCE.czptnzn9net.us-west-2.rds.amazonaws.com \
> cafe_db
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

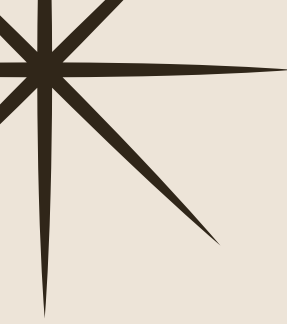
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 58
Server version: 10.11.6-MariaDB-log managed by https://aws.amazon.com/rds/

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

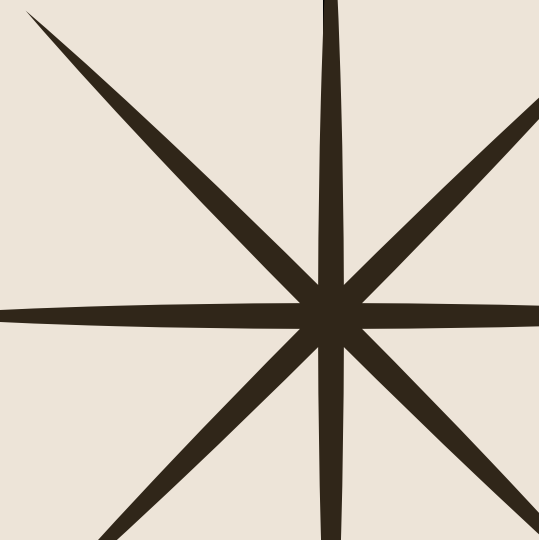
MariaDB [cafe_db]> select * from product;
+----+-----+-----+-----+-----+-----+
| id | product_name | description | price | product_group | image_url |
+----+-----+-----+-----+-----+-----+
| 1 | Croissant | Fresh, buttery and fluffy... Simply delicious! | 1.50 | 1 | images/Croissants.jpg |
| 2 | Donut | We have more than half-a-dozen flavors! | 1.00 | 1 | images/Donuts.jpg |
| 3 | Chocolate Chip Cookie | Made with Swiss chocolate with a touch of Madagascar vanilla | 2.50 | 1 | images/Chocolate-Chip-Cookies.jpg |
| 4 | Muffin | Banana bread, blueberry, cranberry or apple | 3.00 | 1 | images/Muffins.jpg |
| 5 | Strawberry Blueberry Tart | Bursting with the taste and aroma of fresh fruit | 3.50 | 1 | images/Strawberry-Blueberry-Tarts.jpg |
| 6 | Strawberry Tart | Made with fresh ripe strawberries and a delicious whipped cream | 3.50 | 1 | images/Strawberry-Tarts.jpg |
| 7 | Coffee | Freshly-ground black or blended Colombian coffee | 3.00 | 2 | images/Coffee.jpg |
| 8 | Hot Chocolate | Rich and creamy, and made with real chocolate | 3.00 | 2 | images/Cup-of-Hot-Chocolate.jpg |
| 9 | Latte | Offered hot or cold and in various delicious flavors | 3.50 | 2 | images/Latte.jpg |
+----+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)

MariaDB [cafe_db]> 
```



Tarea 4: configurar el sitio web para utilizar la instancia de Amazon RDS

- Ahora ya puede configurar el sitio web de la cafetería y utilizar la instancia de Amazon RDS. Este paso es sencillo, porque el diseñador de la aplicación siguió las prácticas recomendadas y externalizó la información de conexión de la base de datos como parámetros en el almacén de parámetros, que es una capacidad de AWS Systems Manager. En esta tarea, cambiará el parámetro de la URL de la base de datos de la aplicación de la cafetería, para que apunte a la dirección del punto de enlace de la instancia de RDS.



Cambiamos la URL de la BD de la APP mediante Systems Manager

Name

/cafe/dbUrl

When naming a parameter, you can use forward slashes (/) to organize it into a hierarchy. [Learn more about hierarchies](#)

Description — *Optional*

Database URL

Tier

Parameter Store offers standard and advanced parameters.

Standard

Store up to 10,000 standard parameters. Store parameter values up to 4 KB. Parameter policies and sharing with other AWS accounts are not available. No additional charge.

Advanced

Store up to 100,000 advanced parameters. Store parameter values up to 8 KB. Add parameter policies. Share with other AWS accounts. Charges apply.

Standard parameters cannot be shared with other AWS accounts. [Learn more](#)

Type

String

Data type

text

Value

cafedbinstance.czptnzn9net.us-west-2.rds.amazonaws.com

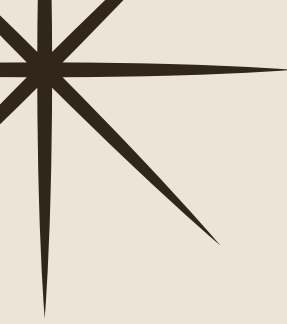
Maximum length 4096 characters.

Cancel

Save changes

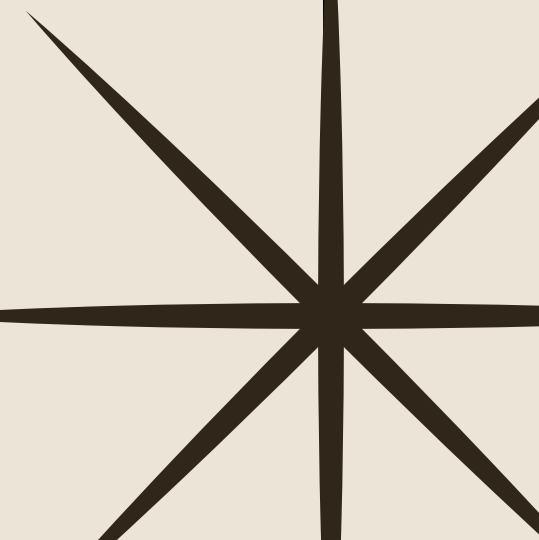
Verificamos que los datos de la BD de RDS son los mismos que la local

Order Number: 3		Date: 2024-07-02	Time: 19:00:26	Total Amount: \$6.00	
Item		Price	Quantity	Amount	
Chocolate Chip Cookie		\$2.50	1	\$2.50	
Strawberry Tart		\$3.50	1	\$3.50	
Order Number: 2		Date: 2024-07-02	Time: 19:00:21	Total Amount: \$13.50	
Item		Price	Quantity	Amount	
Donut		\$1.00	1	\$1.00	
Muffin		\$3.00	1	\$3.00	
Strawberry Blueberry Tart		\$3.50	1	\$3.50	
Coffee		\$3.00	1	\$3.00	
Hot Chocolate		\$3.00	1	\$3.00	
Order Number: 1		Date: 2024-07-02	Time: 19:00:07	Total Amount: \$14.50	
Item		Price	Quantity	Amount	
Croissant		\$1.50	1	\$1.50	
Donut		\$1.00	1	\$1.00	
Chocolate Chip Cookie		\$2.50	1	\$2.50	
Coffee		\$3.00	1	\$3.00	
Hot Chocolate		\$3.00	1	\$3.00	
Latte		\$3.50	1	\$3.50	

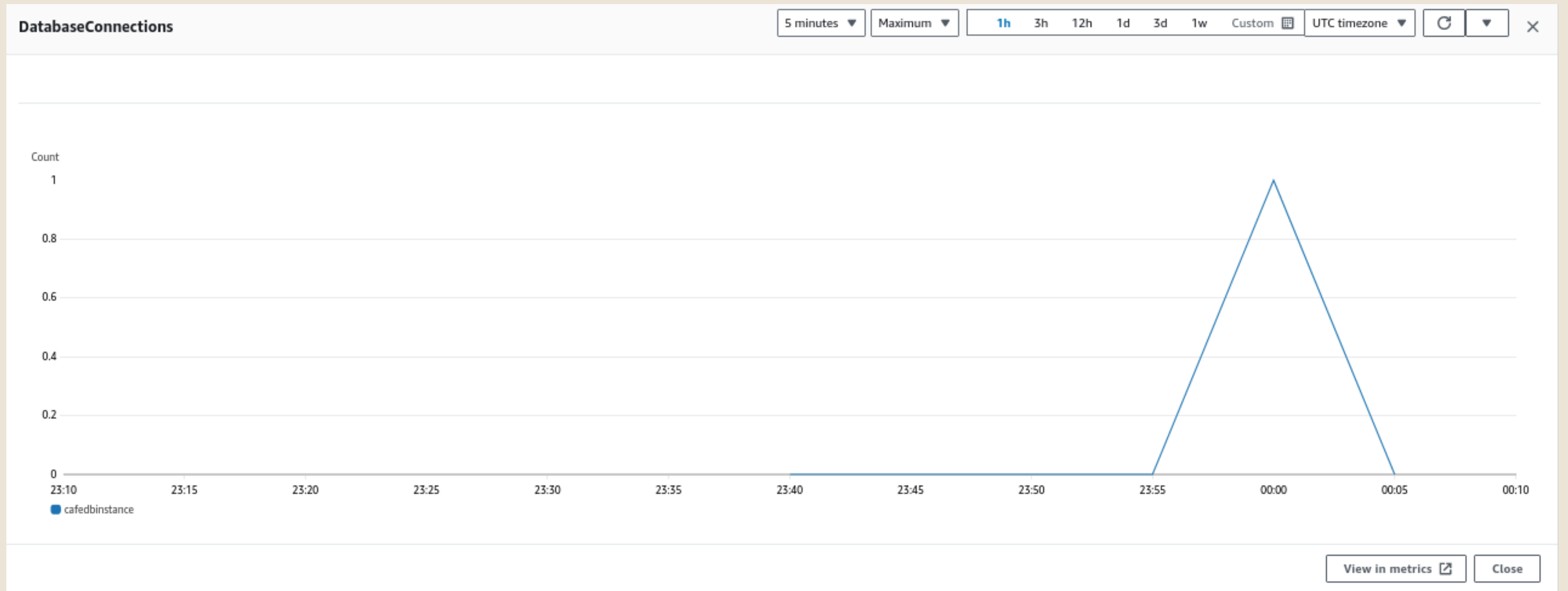


Tarea 5: supervisar la base de datos de Amazon RDS

- Uno de los beneficios de utilizar Amazon RDS es la posibilidad de supervisar el rendimiento de una instancia de base de datos. Amazon RDS envía métricas a CloudWatch de forma automática cada minuto para cada base de datos activa. En esta tarea, identificará algunas de estas métricas de rendimiento y aprenderá a supervisar una métrica en la consola de Amazon RDS.



Nos fijamos que se registró nuestra conexión correctamente



Conclusiones

- Creó una instancia MariaDB de Amazon RDS mediante el uso de la CLI de AWS
- Migró datos de una base de datos MariaDB en una instancia de EC2 a una instancia MariaDB de Amazon RDS
- Supervisó la instancia de Amazon RDS mediante las métricas de CloudWatch

¡Muchas gracias!

Hecho por: Ignacio Suárez
Realizado en: canva.com