

Implementación de un clúster de Neptune

El primer paso que vamos a hacer es crear un único clúster de Neptune independiente. Una vez que se aprovisiona el clúster, cargaremos algunos datos e instalaremos y configuraremos un cliente Gremlin para permitirnos consultar los datos en la base de datos.

Comencemos por crear nuestro Clúster de Amazon Neptune, de la siguiente manera:

Haga clic en **Iniciar Amazon Neptune**, como se ilustra en la siguiente captura de pantalla:

Graph database

Create an Amazon Neptune database cluster

Launch Amazon Neptune

Complete el formulario **Crear base de datos**, de la siguiente manera. Si no se menciona un valor, déjelo como predeterminado:

- **Configuración | Identificador de clúster de base de datos:** neptunenombre
- **Plantillas:** Development and Testing
- **Tamaño de instancia de base de datos | Clase de instancia de base de datos:** db.t3.medium
- **Conectividad | Nube privada virtual (VPC):** seleccione la VPC default
- **Conectividad | Configuración de conectividad adicional | Grupo de seguridad de VPC:** cree uno nuevo e ingresedbcert-neptune-sg-nombre
- **Configuración del cuaderno:** desmarque **Crear cuaderno**
- Configuración adicional desmarque **Registro de auditoría**

Haga clic en **Crear base de datos**. La base de datos tardará alrededor de 10 minutos en mostrar un estado **Disponible**, que puede ver en la siguiente captura de pantalla:

Databases										
<input type="text" value="Filter databases"/> Group resources Modify Actions Create database										
	DB identifier	Role	Engine	Region & AZ	Size	Status	CPU	Current activity	Maintenance	VPC
<input type="radio"/>	dbcert-neptune	Cluster	Neptune	eu-west-1	-	Available	-	-	none	-
<input type="radio"/>	dbcert-neptune-instance-1	Writer	Neptune	eu-west-1b	db.t3.medium	Available	7.99%	-	none	vpc-0f21c54b

Serás capaz de ver el clúster y un solo nodo Writer

Ahora que tenemos nuestra base de datos de Neptune en funcionamiento, cargaremos algunos datos y ejecutaremos algunas consultas. Comenzaremos cargando datos en un depósito S3 y otorgando permisos de lectura a Neptune. Proceder de la siguiente:

Descargue los archivos airports.csv y flight_routes.csv.

Regrese a la consola de AWS y navegue hasta **S3**.

Haga clic en **Crear depósito** en el lado derecho, como se ilustra en la siguiente captura de pantalla:

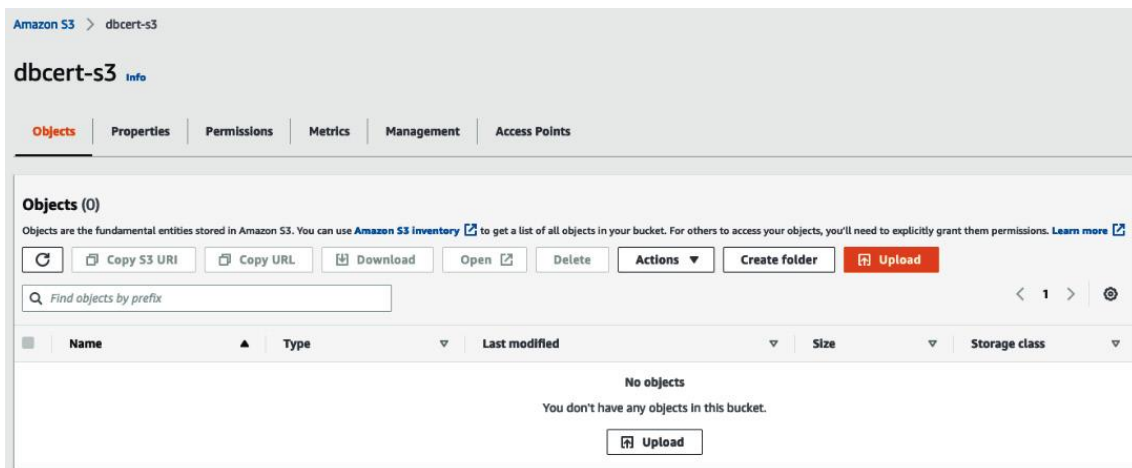


Completa el formulario **Crear depósito**, de la siguiente manera. Deje los valores no mencionados como predeterminados:

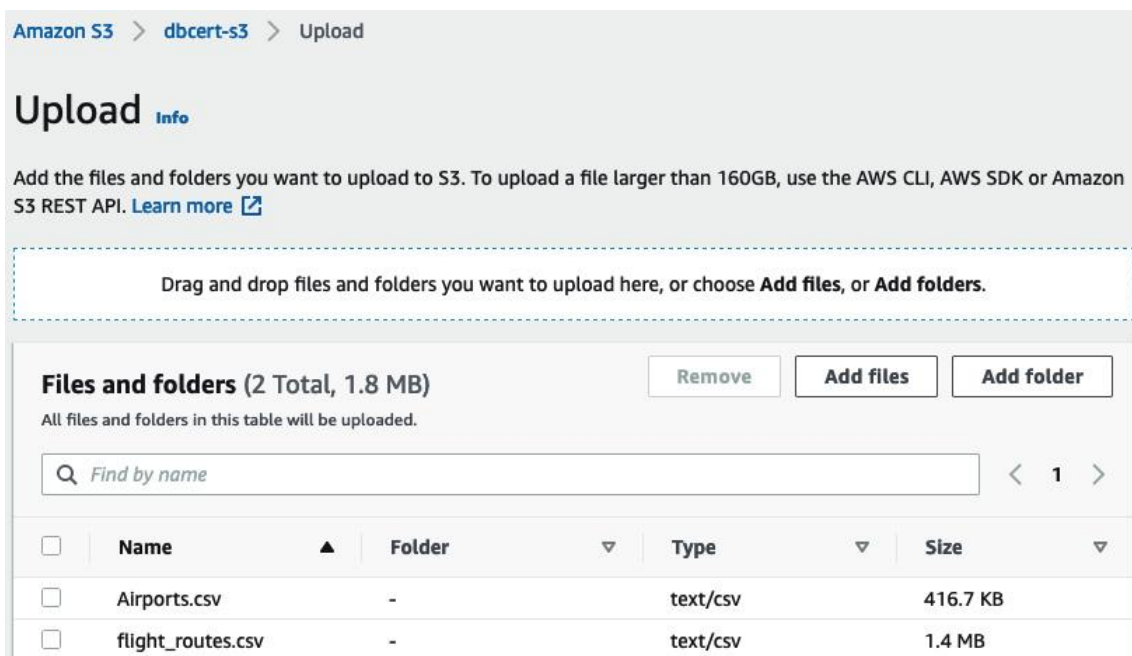
- **Configuración general | Nombre del depósito:** dbcert-s3-<nombre>. Los depósitos de S3 deben tener nombres únicos a nivel mundial.
- **Configuración general | Región AWS:** EU (Ireland) eu-west-1.

Haz clic en **Crear depósito**.

Haga clic en el nombre del depósito que acaba de crear desde el tablero y seleccione **Cargar**, como se ilustra en la siguiente captura de pantalla:



Localiza los archivos que descargaste y cargalos en el bucket, como se ilustra en la siguiente captura de pantalla:



Haga clic en **Cargar** en la parte inferior y espera a que se complete la transferencia. Tomará alrededor de un minuto con una conexión a Internet estándar. Espera hasta que ambos archivos se muestren como **Correcto**.

Ahora necesitamos crear un punto final S3 para que Neptune pueda acceder a los datos en el bucket. Proceder de la siguiente:

Navega hasta **el servicio VPC** en la consola de AWS.

En el panel de navegación izquierdo, elija **Endpoints**.

Haga clic en **Crear endpoint**.

Ingresa s3 en la barra de búsqueda y seleccione com.amazonaws.eu-west-1.s3 el valor del **Nombre del servicio** . Elija un valor para **el tipo de puerta de enlace**.

Elija la VPC que contiene su instancia de base de datos de Neptune.

Seleccione la casilla de verificación junto a las tablas de rutas que están asociadas con las subredes relacionadas con su clúster de base de datos de Neptune. Si solo tiene una tabla de rutas, debe seleccionar esa casilla.

Si lo deseas, revisa la declaración de política que define este extremo.

Haz clic en **Crear endpoint**

Un punto final será creado inmediatamente y lo verás en la vista del tablero, como se ilustra en la siguiente captura de pantalla:



The screenshot shows the AWS Endpoints console. At the top, there is a search bar with the text "Filter by tags and attributes or search by keyword". Below this is a table with the following columns: Name, Endpoint ID, VPC ID, Service name, Endpoint type, Status, and Creation time. A single row is displayed with the following values: a checkbox (checked), a truncated Name "vpce-06ba7ad6b3...", a truncated Endpoint ID "vpce-0f21c54bc90f...", the Service name "com.amazonaws.eu-west-1.s3", the Endpoint type "Gateway", the Status "available", and the Creation time "December 14, 2021 at 5:07:25 PM UTC".

<input type="checkbox"/>	Name	Endpoint ID	VPC ID	Service name	Endpoint type	Status	Creation time
<input checked="" type="checkbox"/>	vpce-06ba7ad6b3...	vpce-0f21c54bc90f...		com.amazonaws.eu-west-1.s3	Gateway	available	December 14, 2021 at 5:07:25 PM UTC

El siguiente paso es configurar los roles **de administración de acceso e identidad (IAM)** para permitir que la instancia de Neptune se comuniquen con el depósito S3.

Navega hasta **IAM**.

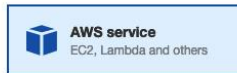
Haz clic en **Roles** en el menú y luego seleccione **Crear rol**.

Haz clic en el botón **de servicio de AWS** y luego seleccione **S3** de la lista, como se ilustra en la siguiente captura de pantalla. Seleccione **S3** nuevamente en la opción **Seleccione su caso de uso** en la parte inferior y luego haga clic en **Siguiente: Permisos** :

Create role

1 2 3 4

Select type of trusted entity



Allows AWS services to perform actions on your behalf. [Learn more](#)

Choose a use case

Common use cases

EC2

Allows EC2 instances to call AWS services on your behalf.

Lambda

Allows Lambda functions to call AWS services on your behalf.

Or select a service to view its use cases

API Gateway	CloudWatch Events	EMR	IoT SiteWise	RAM
AWS Backup	CodeBuild	EMR Containers	IoT Things Graph	RDS
AWS Chatbot	CodeDeploy	ElastiCache	KMS	Redshift
AWS Marketplace	CodeGuru	Elastic Beanstalk	Kinesis	Rekognition
AWS Support	CodeStar Notifications	Elastic Container Registry	Lake Formation	RoboMaker
Amazon OpenSearch Service	Comprehend	Elastic Container Service	Lambda	S3
Amplify	Config	Elastic Transcoder	Lex	SMS
AppStream 2.0	Connect	ElasticLoadBalancing	License Manager	SNS
	DMS	EventBridge	MQ	SWF

Ingresa AmazonS3ReadOnlyAccess en el cuadro de búsqueda y luego marca la casilla de verificación en la tabla a continuación, como ilustrado en la siguiente captura de pantalla. Luego, haz clic en **Siguiente: Etiquetas**:

Create role

1 2 3 4

▼ Attach permissions policies

Choose one or more policies to attach to your new role.

Create policy

Filter policies ▼

Q AmazonS3ReadOnlyAccess

Showing 1 result

	Policy name ▼	Used as
<input checked="" type="checkbox"/>	AmazonS3ReadOnlyAccess	Permissions policy (1)

Haz clic en **Revisar** .

Introduzca un nombre para la función, como dbcert-neptune-s3 y haz clic en **Crear rol** .

Una vez que se haya creado el rol, haga clic en **Roles** en el menú de la izquierda e ingrese el nombre del rol que acaba de crear en el cuadro de búsqueda. Haga clic en el nombre del rol que creó.

Haz clic en la pestaña **Relaciones de confianza** , como ilustra la siguiente captura de pantalla:

Roles > dbcert-neptune-s3 Delete role

Summary

Role ARN	arn:aws:iam::653375240923:role/dbcert-neptune-s3
Role description	Allows S3 to call AWS services on your behalf. Edit
Instance Profile ARNs	
Path	/
Creation time	2021-12-15 13:33 UTC
Last activity	Not accessed in the tracking period
Maximum session duration	1 hour Edit

Permissions Trust relationships Tags Access Advisor Revoke sessions

You can view the trusted entities that can assume the role and the access conditions for the role. [Show policy document](#)

[Edit trust relationship](#)

Trusted entities

The following trusted entities can assume this role.

Trusted entities
The identity provider(s) s3.amazonaws.com

Conditions

The following conditions define how and when trusted entities can assume the role.

There are no conditions associated with this role.

Haz clic en **Editar relación de confianza** y reemplaza el código con el siguiente código:

```
[
  {
    "Version": "2012-10-17",
    "Statement": [
      {
        "Sid": "",
        "Effect": "Allow",
        "Principal": {
          "Service": [
            "rds.amazonaws.com"
          ]
        },
        "Action": "sts:AssumeRole"
      }
    ]
  }
]
```



Este código permite que una instancia de RDS utilice este rol.

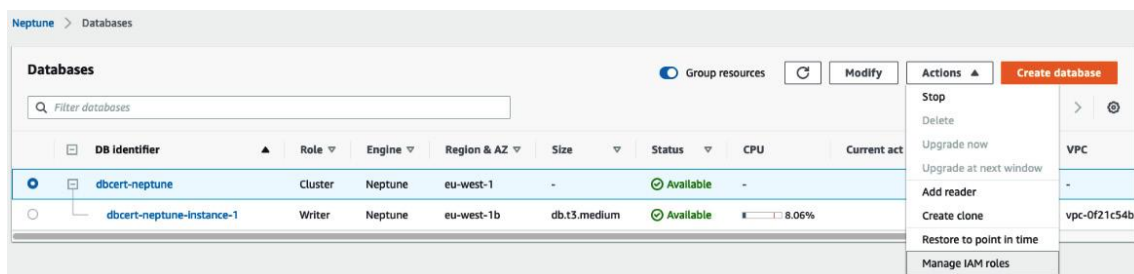
Haz clic en **Actualizar política de confianza**.

Toma nota de el valor **del ARN del rol**, como se muestra en la siguiente captura de pantalla, ya que lo necesitaremos más adelante:



Navega a **Neptune** desde la consola de AWS.

Selecciona la base de datos con el rol de clúster que creó anteriormente y luego selecciona **Administrar roles de IAM** en el menú desplegable **Acciones**, como se ilustra en la siguiente captura de pantalla:



Selecciona el rol que acaba de crear y haga clic en **Listo**.

Ahora crearemos una instancia **de Elastic Compute Cloud (EC2)** desde la cual podemos ejecutar comandos en la base de datos de Neptune.

Inicia sesión en la Consola de AWS y vaya a **EC2**

Haz clic en **Iniciar instancia** desde el panel de **EC2**

Selecciona **Amazon Linux 2 AMI** como se ilustra en la siguiente captura de pantalla. Cualquier versión del kernel está bien. Los detalles de su **imagen de máquina de Amazon (AMI)** variarán según la región que elija:

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type

Free tier eligible

ami-0e8cb4bdc5bb2e6c0 (64-bit (x86)) / ami-00cf3a525c4693c5f (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2 Kernel 5.10 AMI 2.0.20220406.1 x86_64 HVM gp2

Architecture

AMI ID

64-bit (x86)

ami-0e8cb4bdc5bb2e6c0

Elije t2.micro, que es parte del nivel gratuito, como se ilustra en la siguiente captura de pantalla:

▼ Instance type [Info](#)

Instance type

t2.micro

Free tier eligible

Family: t2 1 vCPU 1 GiB Memory

On-Demand Linux pricing: 0.0126 USD per Hour

On-Demand Windows pricing: 0.0172 USD per Hour

[Compare instance types](#)

Haz clic en **Siguiente: Configurar detalles de la instancia**

En **Red**, selecciona la misma VPC en la que se lanzó la base de datos de Neptune.

Selecciona cualquier subred pública del menú desplegable **Subred** como se ilustra en la siguiente captura de pantalla: puede saber si es una subred pública si el valor predeterminado de **Asignación automática de IP pública** es **(Habilitar)** . Esto es importante para que podamos conectarnos a esta instancia EC2. Deje todos los demás valores como predeterminados y haga clic en **Revisar y lanzar**

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instances, and more.

Number of instances

1

Launch into Auto Scaling Group

Purchasing option

☐ Request Spot instances

Network

vpc-0f21c54bc90f5f4d8 | NoDNS

Create new VPC

Subnet

subnet-08eed0e68c3422bb5 | eu-west-1a

Create new subnet

Auto-assign Public IP

Use subnet setting (Disable)

Hostname type

Use subnet setting (IP name)

DNS Hostname

☒ Enable IP name IPv4 (A record) DNS requests
☒ Enable resource-based IPv4 (A record) DNS requests
☐ Enable resource-based IPv6 (AAAA record) DNS requests

En la página **Revisar lanzamiento de instancia** haga clic en **Editar grupos de seguridad**

Crear un nuevo grupo de seguridad llamado. Edite la regla para permitir **Secure Shell (SSH)** en el puerto 22 desde todas las direcciones de **Protocolo de Internet (IP)**. Esto es para que podamos acceder a la instancia. El proceso se ilustra en la siguiente captura de pantalla:

The screenshot shows the AWS Management Console interface for editing a security group rule. The 'Security group name' field is set to 'dbcert-sg-neptune'. Below it, a note states: 'This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&;{}!\$*'. The 'Description' field is also set to 'dbcert-sg-neptune'. Under 'Inbound security groups rules', there is a rule named 'Security group rule 1 (TCP, 22, 0.0.0.0/0)'. The rule configuration is as follows:

Type	Protocol	Port range	Source type	Source	Description
ssh	TCP	22	Anywhere	0.0.0.0/0	e.g. SSH for admin desktop

Haz clic en **Revisar y lanzar** y luego haga clic en **Lanzar**

Seleccione **Crear un nuevo par de claves** en el menú desplegable e ingrese un nombre, como. Haga clic en **Descargar par de claves** y luego en **Iniciar instancias**

Vuelva al **panel de EC2** y espere a que el valor **del estado de la instancia** se muestre como **En ejecución**. Esto tomará unos pocos minutos.

Ahora necesitamos agregar una regla a la seguridad de nuestro clúster de base de datos de Neptune para permitir que la instancia EC2 se conecte a él. Navegue hasta el servicio **VPC** desde el menú principal de la consola.

Seleccione **Grupos de seguridad** en el menú de la izquierda y localice su grupo de seguridad de Neptune,

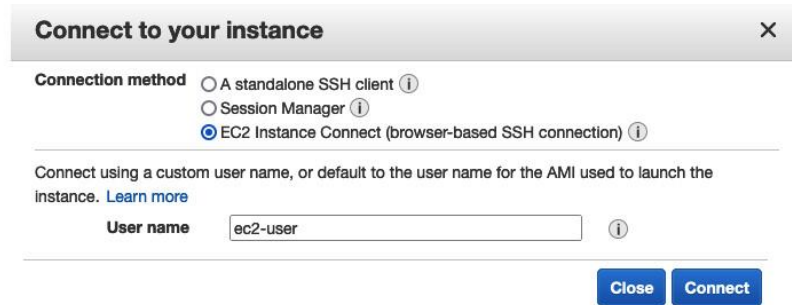
Agregue una regla para permitir conectarse a él en el puerto 8182.

Una vez que la instancia esen ejecución, intentaremos conectarnos a él mediante la herramienta de conexión de instancias EC2. También puede intentar usar

un sshcomando desde su computadora local, pero esto está más allá del alcance de esta práctica de laboratorio. Proceder de la siguiente:

Seleccione la instancia que acaba de crear y luego haga clic en **Conectar**

Seleccione **EC2 Instance Connect** y deje el nombre de usuario como ec2-user, como se ilustra en la siguiente captura de pantalla:



Connect to your instance [X]

Connection method

- ☐ A standalone SSH client ⓘ
- ☐ Session Manager ⓘ
- ☒ EC2 Instance Connect (browser-based SSH connection) ⓘ

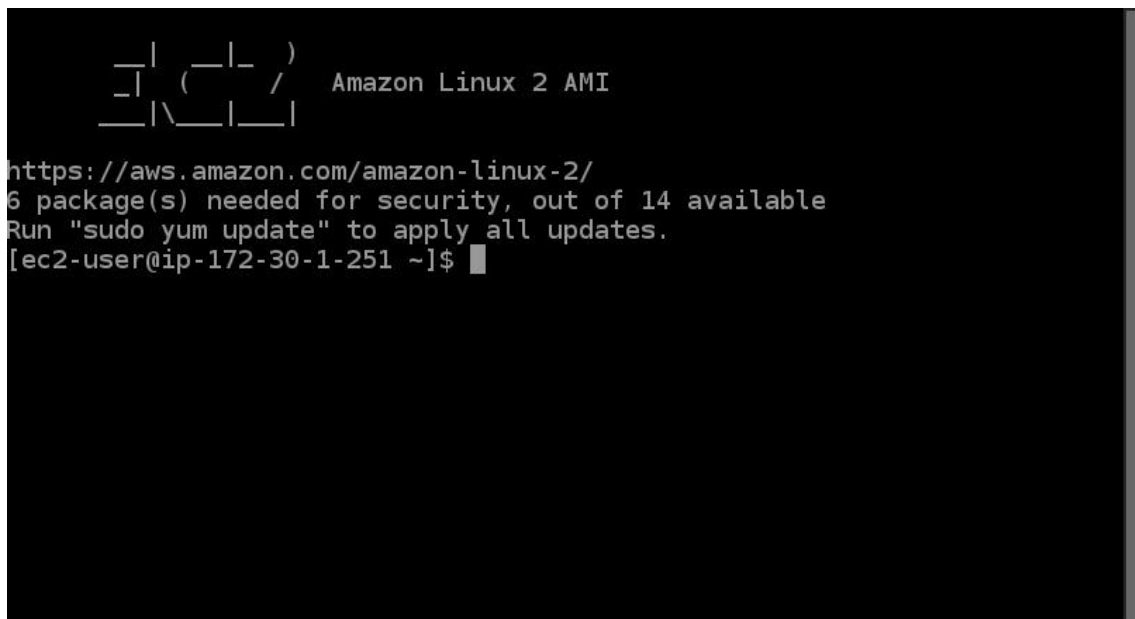
Connect using a custom user name, or default to the user name for the AMI used to launch the instance. [Learn more](#)

User name ⓘ

Close **Connect**

Haga clic en **Conectar** .

Debería aparecer una pantalla similar a esta:



```

  _ | _ | _ )
 _ | ( _ | /   Amazon Linux 2 AMI
 _ | \ _ | _ |

https://aws.amazon.com/amazon-linux-2/
6 package(s) needed for security, out of 14 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-30-1-251 ~]$
```

Ahora vamos a usar una herramienta llamada curl que permite ejecutar comandos usando **el Protocolo de transferencia de hipertexto (HTTP)**.

En una ventana o pestaña diferente del navegador, navegue hasta el panel de control **de Neptune**

Haz clic en el nodo y tome nota de los valores de **Endpoint** y **Puerto** en la pestaña **Conectividad y seguridad** .

Regresa a su sesión de EC2 y ejecute el siguiente comando curl. Deberá modificar las secciones resaltadas para que coincidan con sus propios valores:

```
curl -X POST \
  -H 'Content-Type: application/json' \
  https://dbcert-neptune-instance-1.cdhcmbt6wawh.eu-west-
  1.neptune.amazonaws.com:8182/loader -d '{
    "source" : "s3://dbcert-s3-kgawron",
    "format" : "csv",
    "iamRoleArn" : "arn:aws:iam::653375240923:role/dbcert-neptune-s3",
    "region" : "eu-west-1",
    "failOnError" : "FALSE",
    "parallelism" : "MEDIUM",
    "updateSingleCardinalityProperties" : "FALSE",
    "queueRequest" : "FALSE"
  }
```

Ahora que tenemos algunos datos en la base de datos, necesitamos instalar Gremlin para poder ejecutarlo.

Desde su instancia EC2, instale Java, así:

```
sudo yum install java-1.8.0-devel
```

Ahora, descargue e instale la consola Gremlin, de la siguiente manera:

```
wget https://archive.apache.org/dist/tinkerpop/3.4.8/apache-tinkerpop-gremlin-console-3.4.8-bin.zip
unzip apache-tinkerpop-gremlin-console-3.4.8-bin.zip
```

Cambiar directorio a apache-tinkerpop-gremlin-console-3.4.8.

Descargar e instalar certificados de Amazon. Neptune utiliza una conexión de **capa de sockets seguros (SSL)**, por lo que para conectarnos necesitamos un certificado SSL. Cambie la ubicación de Java para que coincida con la que ha descargado. El código se ilustra en el siguiente fragmento:

```
wget https://www.amazontrust.com/repository/SFSRootCAG2.cer
mkdir /tmp/certs/
cp /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.312.b07-1.amzn2.0.1.x86_64/jre/lib/security/cacerts /tmp/certs/cacerts
sudo keytool -importcert \
    -alias neptune-tests-ca \
    -keystore /tmp/certs/cacerts \
    -file /home/ec2-user/apache-tinkerpop-gremlin-console-3.4.8/SFSRootCAG2.cer \
    -noprompt \
    -storepass changeit
```

Cambie al directorio y cree un archivo llamado neptune-con.yaml con los siguientes contenidos. Deberá cambiar los valores resaltados para que coincidan con su propia base de datos de Neptune.

```
hosts: [dbcert-neptune.cluster-cdhcmbt6wawh.eu-west-1.neptune.amazonaws.com]
port: 8182
connectionPool: { enableSsl: true, trustStore: /tmp/certs/cacerts }
serializer: { className: org.apache.tinkerpop.gremlin.driver.ser.GryoMessageSerializerV3d0,
config: { serializeResultToString: true }}
```

Vuelva al directorio `apache-tinkerpop-gremlin-console-3.4.8` y ejecute `bin/gremlin.sh`. Esto cargará la consola de Gremlin.

Cuando se le solicite `gremlin>`, ingrese el siguiente código para cargar su configuración de Neptune y luego díglelo para usarlo:

```
:remote connect tinkerpop.server conf/neptune-con.yaml
:remote console
```

Ahora podemos ejecutar consultas Gremlin, como averiguar cuántos componentes hay, de la siguiente manera:

```
g.V().label().groupCount()
```

O bien, podemos averiguar a dónde podemos volar directamente desde Londres Heathrow, como en este ejemplo:

```
g.V().has('code','LHR').out().path().by('code')
```

Ahora puede limpiar el entorno, incluida la instancia EC2 que creó.