Implementación y gestión de aplicaciones de automatización de procesos empresariales en Amazon EKS: una guía práctica

Contenido

Abstracto

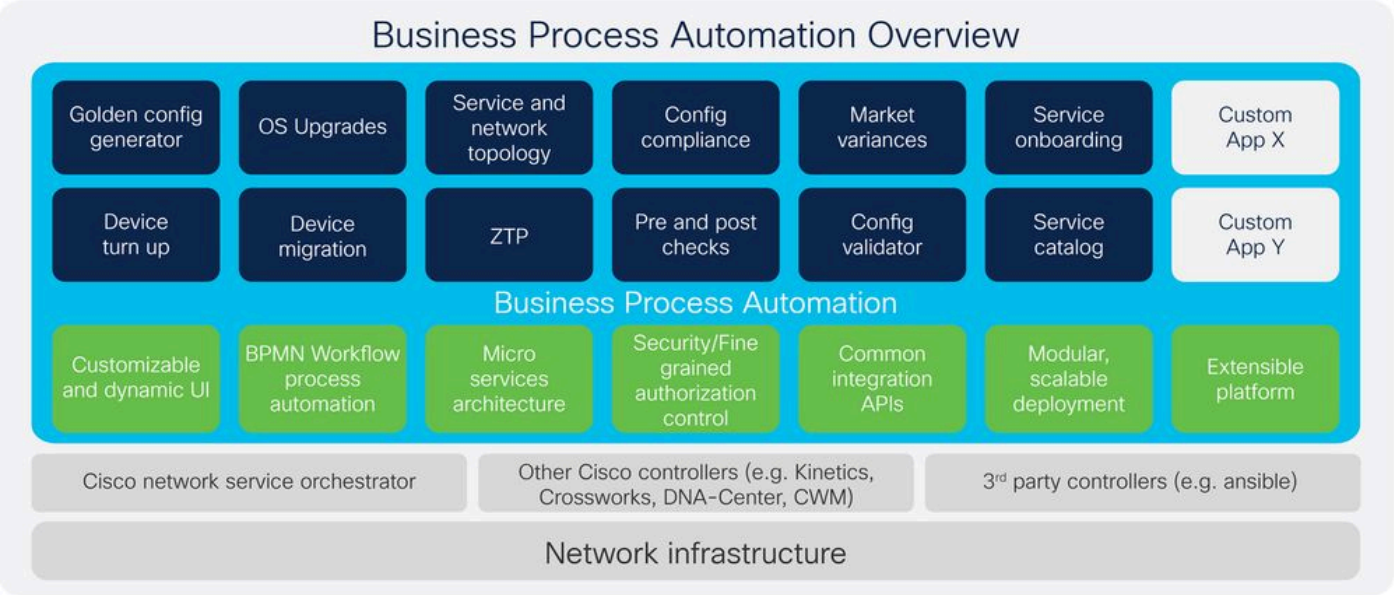
Este documento presenta una guía completa sobre la implementación y la gestión de aplicaciones de automatización de procesos empresariales (BPA) mediante el servicio Amazon Elastic Kubernetes Service (EKS). Describe los requisitos previos, destaca las ventajas de utilizar EKS y proporciona instrucciones paso a paso para configurar un clúster EKS, una base de datos Amazon RDS y un sistema MongoDB Atlas. Además, el documento profundiza en la arquitectura de implementación y especifica los requisitos del entorno, ofreciendo un recurso exhaustivo para las organizaciones que pretenden aprovechar EKS para sus aplicaciones BPA en contenedores.

Palabras clave

Amazon EKS, Kubernetes, AWS, RDS, MongoDB Atlas, DevOps, Cloud Computing, automatización de procesos empresariales.

Introducción

BPA



En la era digital actual, las empresas buscan agilizar y automatizar los procesos empresariales complejos en una amplia gama de entornos de TI. La automatización de procesos empresariales (BPA) se ha convertido en una tecnología fundamental que permite a las organizaciones mejorar la eficacia operativa, reducir los errores y mejorar la prestación de servicios. BPA presenta varias innovaciones y mejoras clave destinadas a avanzar en la automatización del flujo de trabajo, el aprovisionamiento de servicios y las aplicaciones de automatización estándar.

La plataforma BPA aloja aplicaciones y casos de uso operativos y de TI/empresariales, como actualizaciones de SO, aprovisionamiento de servicios e integración con motores de orquestación. Los clientes tienen acceso a un ciclo de vida de servicios y funciones de BPA que incluyen asesoría, implementación, servicios empresariales críticos y asistencia para soluciones a través de expertos de Cisco, prácticas recomendadas y técnicas y metodologías probadas que ayudan a automatizar sus procesos empresariales y a eliminar el riesgo de sus sistemas.

Estas capacidades del ciclo de vida pueden estar basadas en suscripciones o personalizarse según las necesidades individuales. Los servicios de implementación ayudan a definir, integrar e implementar herramientas y procesos para acelerar la automatización. Los expertos de Cisco llevan a cabo un proceso formal para recopilar requisitos, diseñar y desarrollar historias de usuarios basadas en procesos ágiles y herramientas de integración continua y prestación continua (CICD), e implementa servicios flexibles con pruebas automatizadas de flujos de trabajo, dispositivos y servicios nuevos o existentes. Con la Asistencia para soluciones, los clientes pueden acceder a una asistencia centralizada las 24 horas del día, los 7 días de la semana, que se centra en los problemas centrados en el software, junto con la asistencia de varios proveedores y código abierto que se ofrece a través del modelo de software por niveles de Cisco. Los expertos en asistencia para soluciones de Cisco le ayudan a gestionar su caso desde la primera llamada hasta la resolución final y actúan como el principal punto de contacto al trabajar con varios proveedores simultáneamente. Podrá experimentar hasta un 44% menos de problemas al trabajar con expertos en soluciones, lo que le ayudará a mantener la continuidad empresarial y a obtener un retorno de la inversión en BPA más rápido.

Las características técnicas clave, como la compatibilidad con FMC y dispositivos administrados con

Ansible, las ejecuciones paralelas mediante Advanced Queuing Framework (AQF) y el cumplimiento ampliado de la configuración para los dispositivos NDFC y FMC, posicionan a BPA como una solución completa para la automatización empresarial a gran escala. Con capacidades añadidas en gestión de SD-WAN, incorporación de dispositivos y administración de políticas de firewall, la versión aborda aspectos críticos de la seguridad y la automatización de la red, satisfaciendo las demandas de entornos de varios proveedores a gran escala.

EKS

Amazon Elastic Kubernetes Service (EKS) es un servicio de Kubernetes totalmente administrado proporcionado por Amazon Web Services (AWS). EKS, que se lanzó en 2018, simplifica el proceso de implementación, gestión y ampliación de aplicaciones en contenedores mediante Kubernetes, una plataforma de orquestación de contenedores de código abierto. EKS abstrae las complejidades de la gestión de clústeres de Kubernetes, lo que permite a los desarrolladores centrarse en crear y ejecutar aplicaciones sin necesidad de gestionar la infraestructura subyacente.

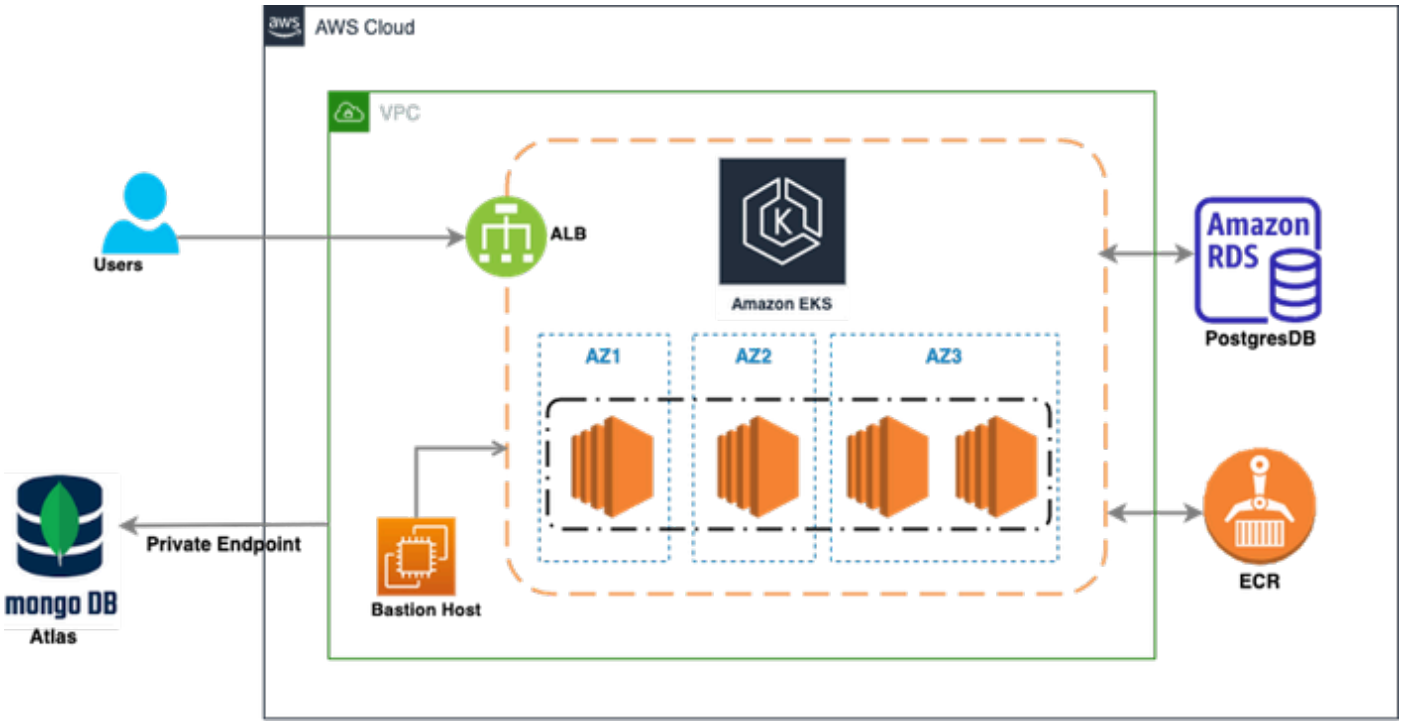
Ventajas del uso de Amazon EKS para la implementación de aplicaciones

Amazon EKS ofrece varias ventajas para la implementación de aplicaciones, lo que lo convierte en una opción popular para las organizaciones que aprovechan aplicaciones y microservicios en contenedores.

Entre las principales ventajas se incluyen:

- **Plano de control de Kubernetes administrado**: EKS se encarga de la implementación, la ampliación y el mantenimiento del plano de control de Kubernetes, lo que reduce la carga operativa.
- **Gestión de clústeres simplificada**: EKS abstrae las complejidades de la configuración y gestión de clústeres de Kubernetes.
- **Escalabilidad**: EKS permite una sencilla ampliación de clústeres para adaptarse a cargas de trabajo cada vez mayores.
- **Alta disponibilidad**: EKS admite implementaciones de zonas de disponibilidad múltiple, lo que mejora la disponibilidad y la tolerancia a fallos.
- **Integración con los servicios de AWS**: EKS se integra a la perfección con varios servicios de AWS.
- **DevOps Automation**: EKS admite la integración continua y la implementación continua (CI/CD) para aplicaciones en contenedores.

Arquitectura de implementación de BPA



Esta imagen representa una arquitectura de alto nivel de una infraestructura basada en la nube implementada en **AWS**, utilizando varios componentes clave. Aquí hay un desglose del diagrama:

1. **Amazon EKS (Elastic Kubernetes Service)**: en el núcleo del diagrama, Amazon EKS se implementa en tres zonas de disponibilidad (AZ1, AZ2, AZ3), con nodos de trabajo de Kubernetes dentro de cada zona. Esto indica una configuración de alta disponibilidad y tolerante a fallos, ya que las cargas de trabajo se distribuyen en varias zonas de disponibilidad.

2. **ALB (equilibrador de carga de aplicaciones)**: se sitúa en la parte frontal, recibe el tráfico de los usuarios y lo distribuye por el clúster EKS para gestionar las cargas de trabajo de las aplicaciones. El equilibrador de carga garantiza que las solicitudes se distribuyan de manera uniforme y que puedan gestionar la escalabilidad según la demanda del tráfico.

3. **Amazon RDS (Servicio de base de datos relacional)** - PostgreSQL: En el lado derecho del diagrama, hay una instancia de Amazon RDS que ejecuta PostgreSQL. A esta base de datos pueden tener acceso las aplicaciones que se ejecutan dentro del clúster EKS.

4. **ECR (Elastic Container Registry)**: Aquí es donde se almacenan y administran las imágenes del contenedor Docker, que luego se implementan en Amazon EKS para ejecutar las cargas de trabajo.

5. **MongoDB Atlas**: En el lado izquierdo, MongoDB Atlas se integra en la arquitectura a través de un terminal privado. MongoDB Atlas es un servicio de base de datos NoSQL alojado en la nube, utilizado aquí para manejar los requisitos de base de datos basados en documentos. El terminal privado garantiza una comunicación segura y privada entre la instancia de MongoDB Atlas y otros componentes de AWS.

6. **Host de bastión**: ubicado en la VPC (nube privada virtual), un host de bastión proporciona un punto de entrada seguro para que los administradores accedan a los recursos dentro de la VPC sin exponerlos directamente a Internet.

En general, esta arquitectura proporciona una solución segura, escalable y de alta disponibilidad para implementar y administrar aplicaciones en contenedores mediante Amazon EKS, con compatibilidad con bases de datos relacionales (PostgreSQL) y NoSQL (MongoDB).

- **Configuración del clúster EKS**
  Para crear un clúster de Amazon EKS mediante la CLI de AWS, se puede utilizar la utilidad de línea de comandos `eksctl`. Este es un ejemplo de comando:

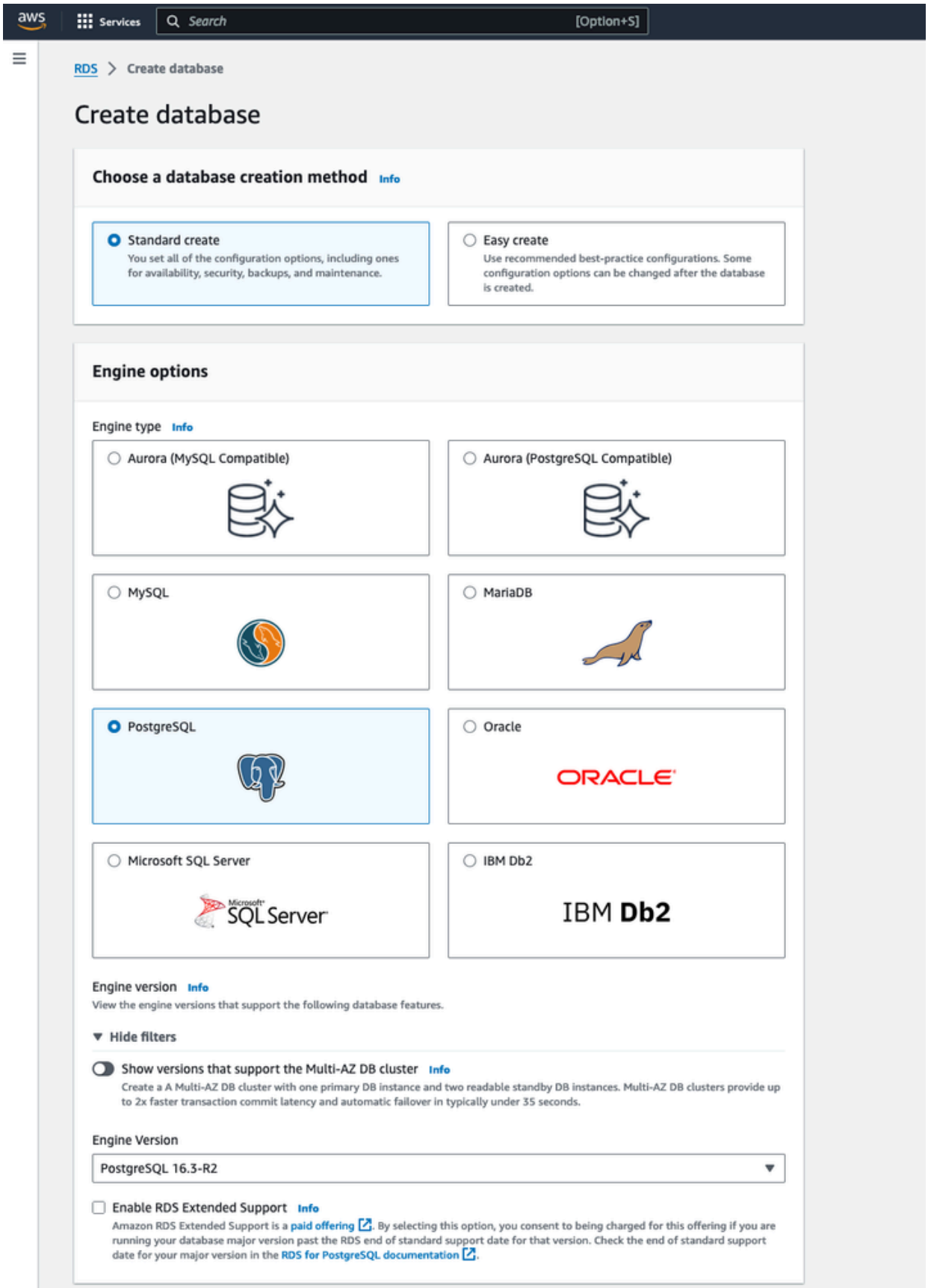
```
eksctl create cluster \
  --name
```

```
\ --region us-west-2 \ --nodegroup-name standard-workers \ --node-type t3.medium \ --node
```

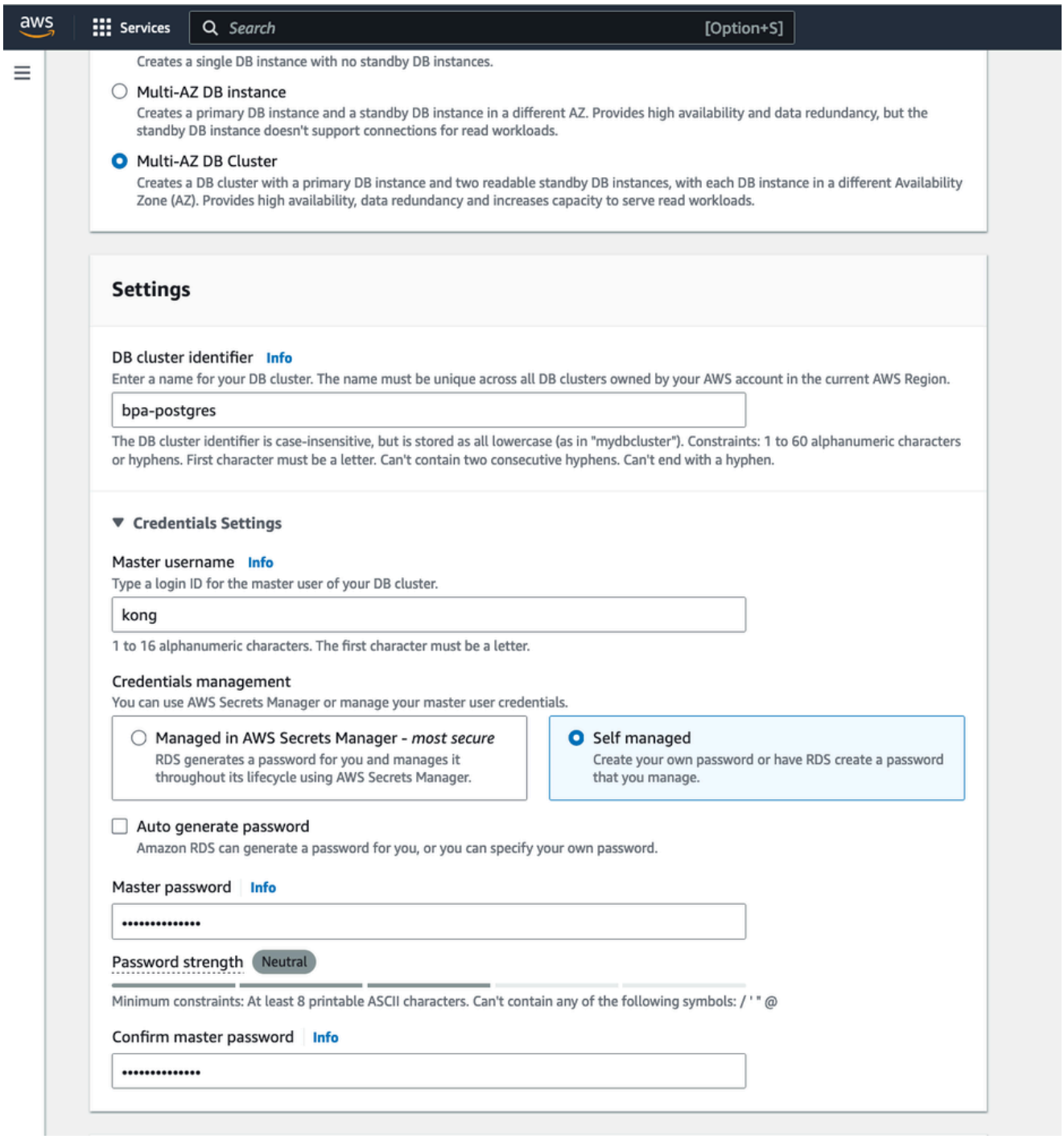
- **Configuración de base de datos RDS**

  La implementación de una base de datos relacional en Amazon RDS implica estos pasos:

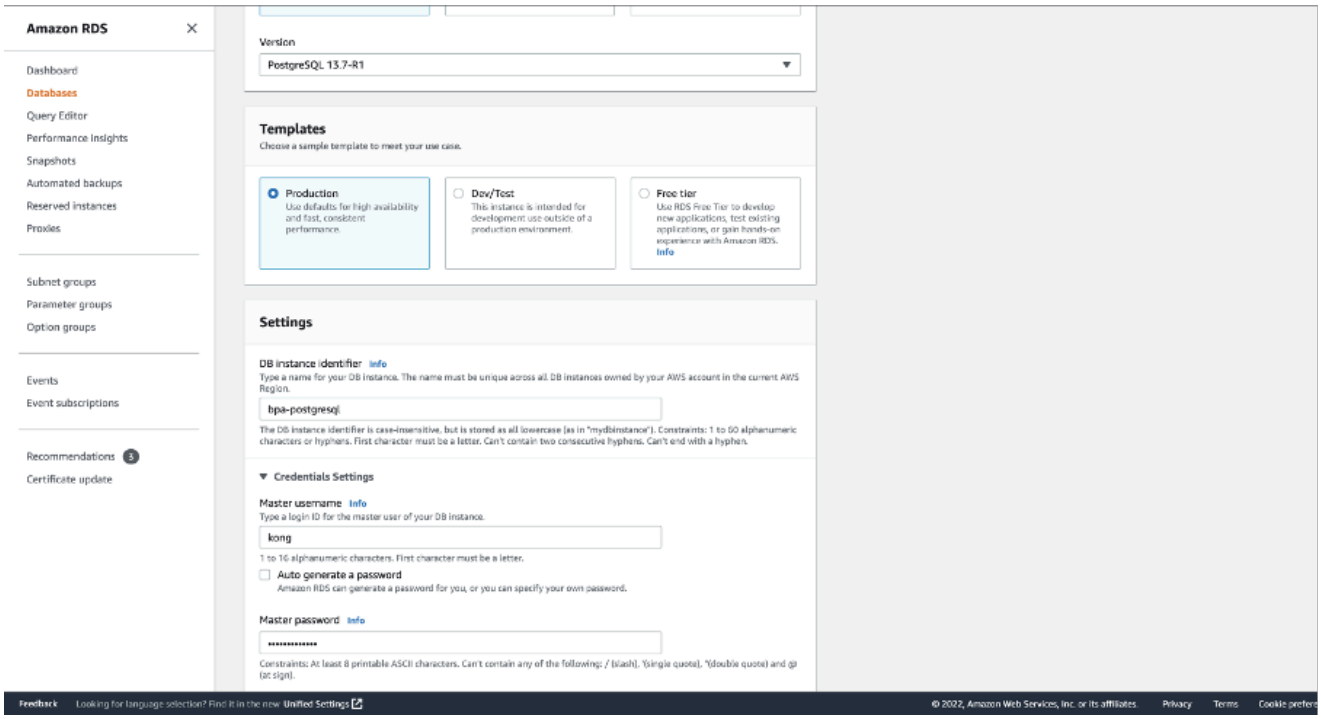
  - Acceda a la consola de administración de AWS y navegue hasta el servicio Amazon RDS.
  - Cree una nueva instancia de base de datos con las especificaciones deseadas.
  - Configure el grupo de seguridad para permitir conexiones entrantes desde el clúster de Amazon EKS.



En el menú desplegable, seleccione la versión más reciente de PostgreSQL. En nuestro caso, es "PostgreSQL 16.3-R1."



Para ello, asigne un nombre a la instancia de base de datos y cree un nombre de usuario y una contraseña.

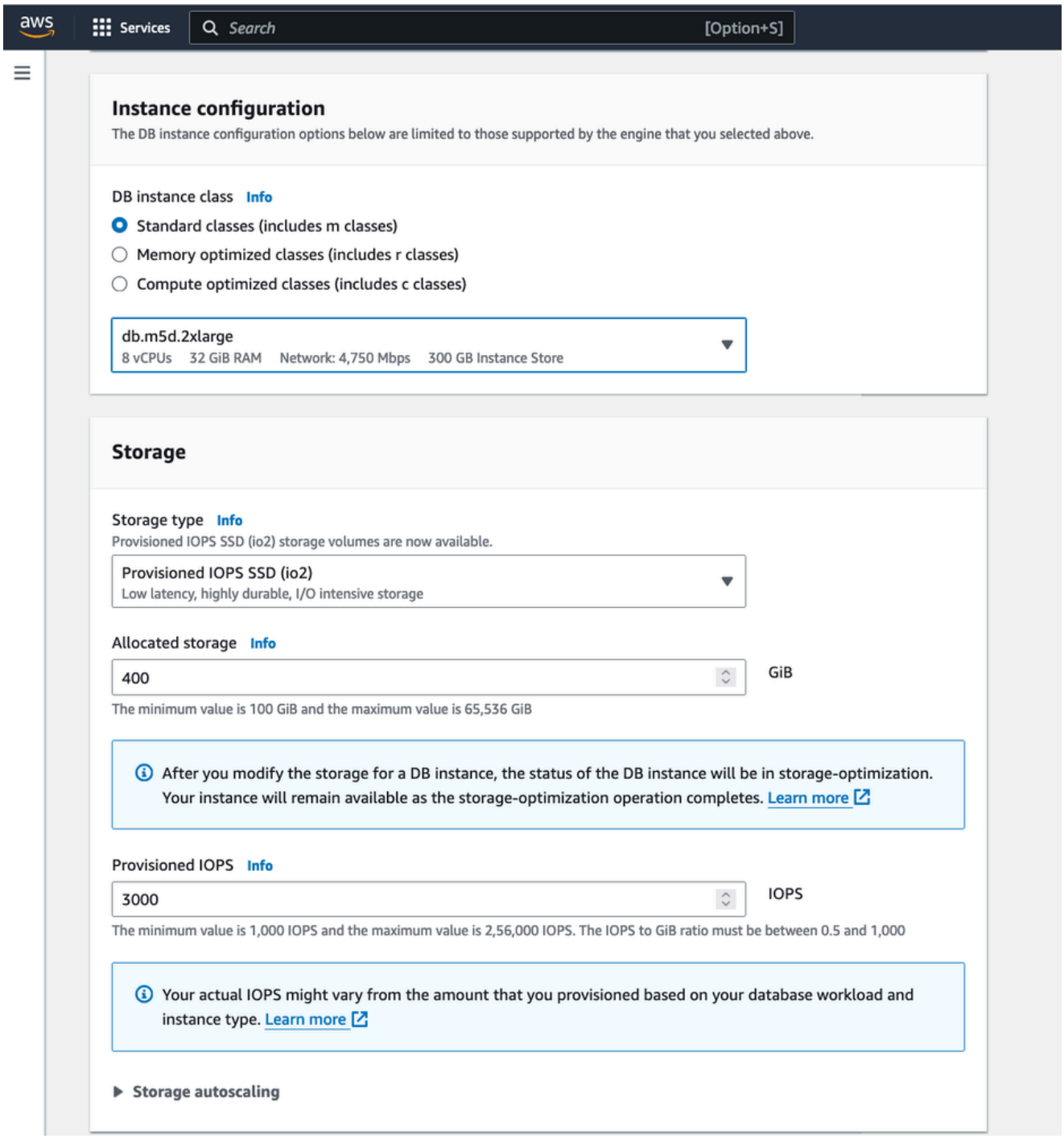


Asegúrese de que la configuración predeterminada para "DB instance size" (Tamaño de instancia de base de datos) y "Storage" (Almacenamiento) esté seleccionada.

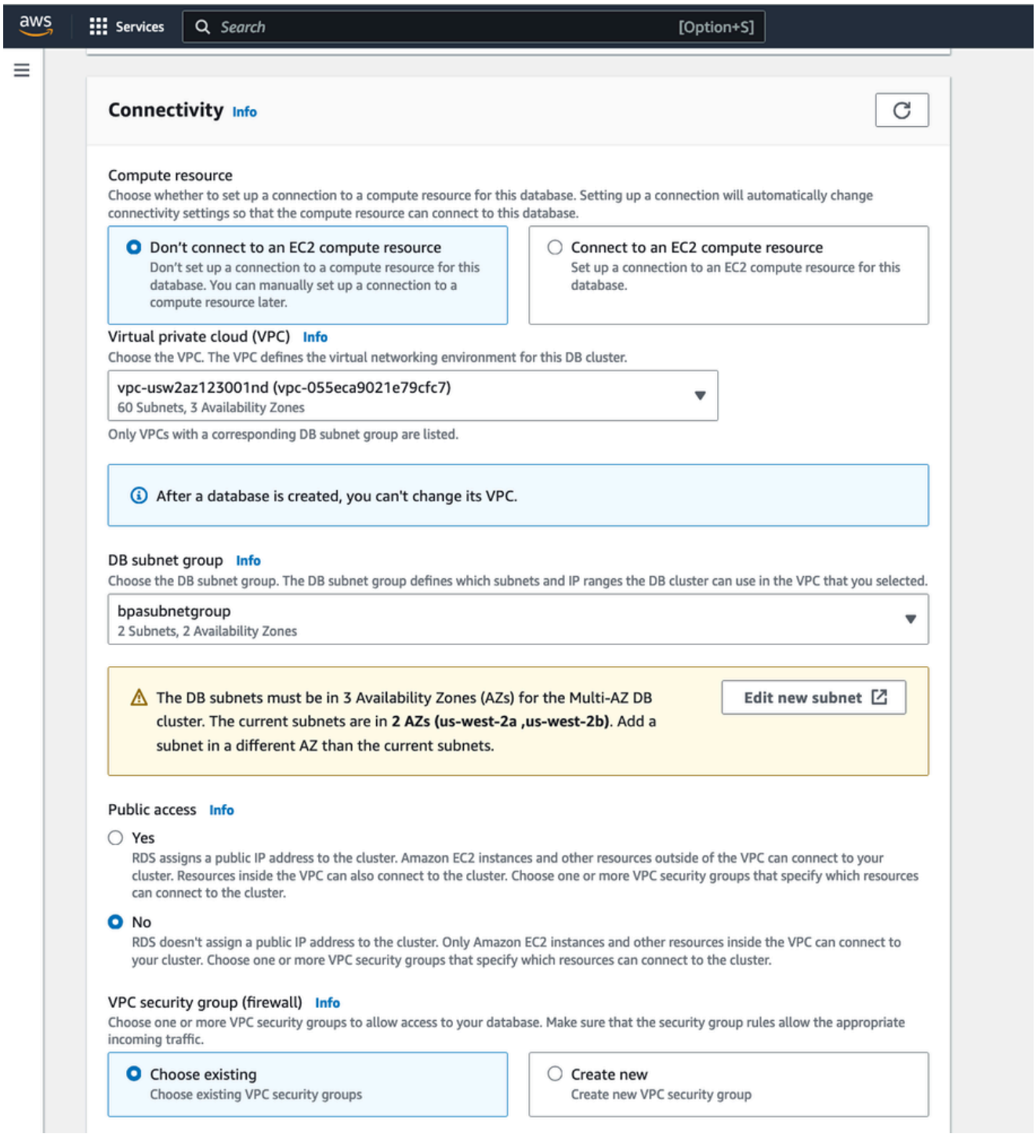
En función del tamaño del clúster y de los requisitos de datos, seleccione el tamaño de instancia de base de datos y el tipo de almacenamiento adecuados.

En función de nuestro caso práctico, hemos elegido la siguiente configuración:

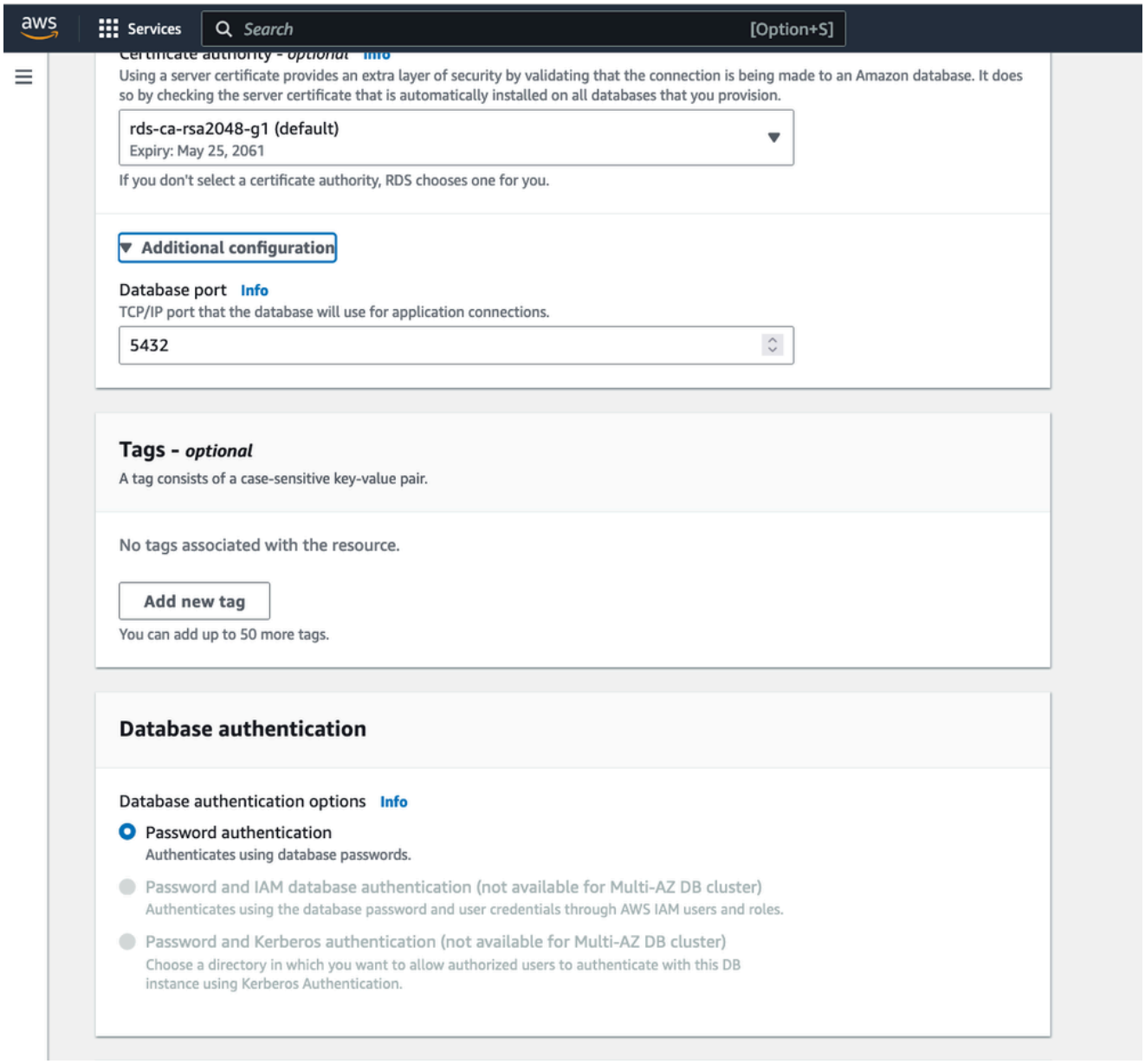
- **Tamaño de instancia de BD**: db.m5d.2xlarge
  - 8 vCPU
  - 32 GiB RAM
  - Red: 4750 Mbps
  - Almacén de instancias de 300 GB

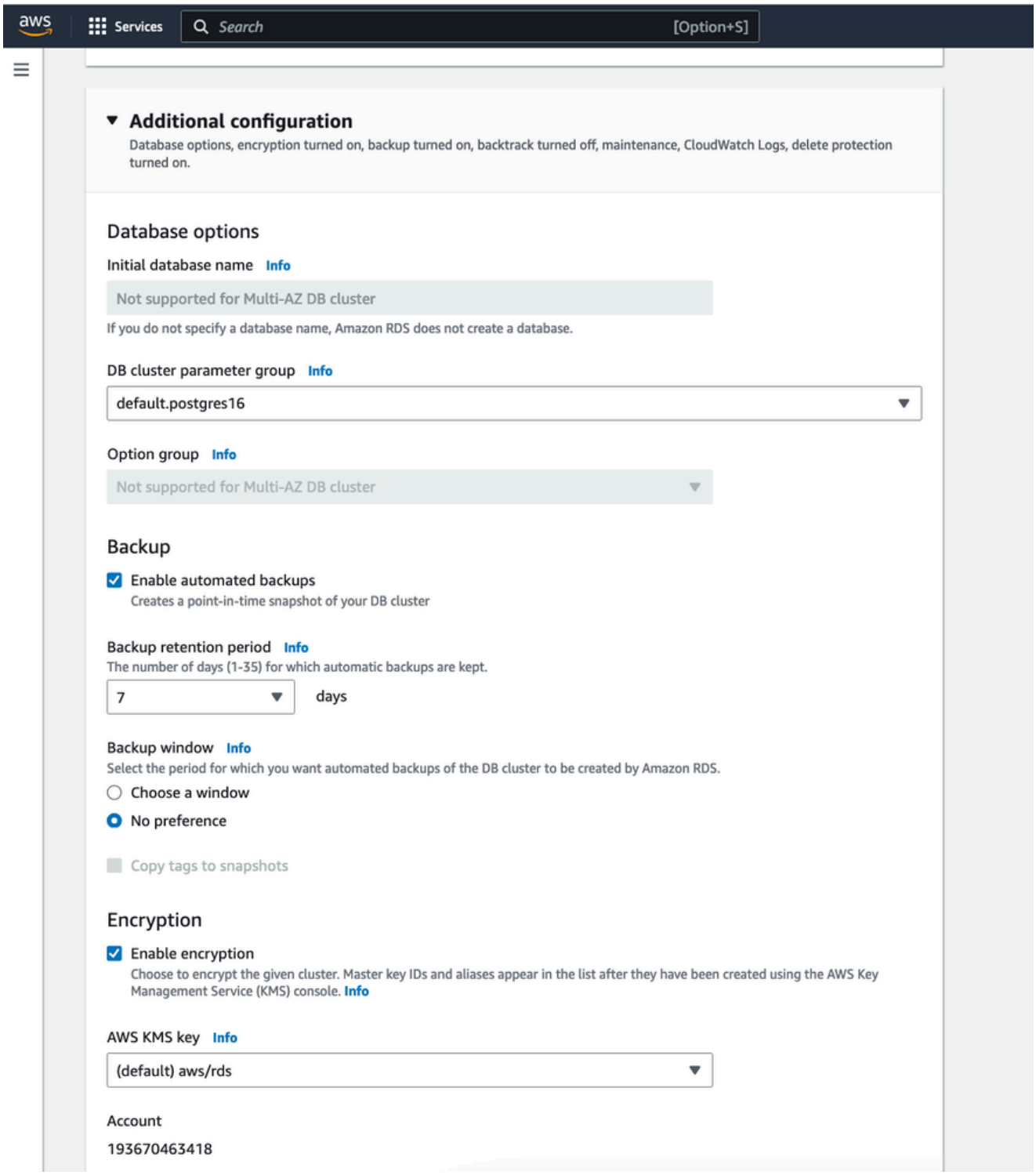


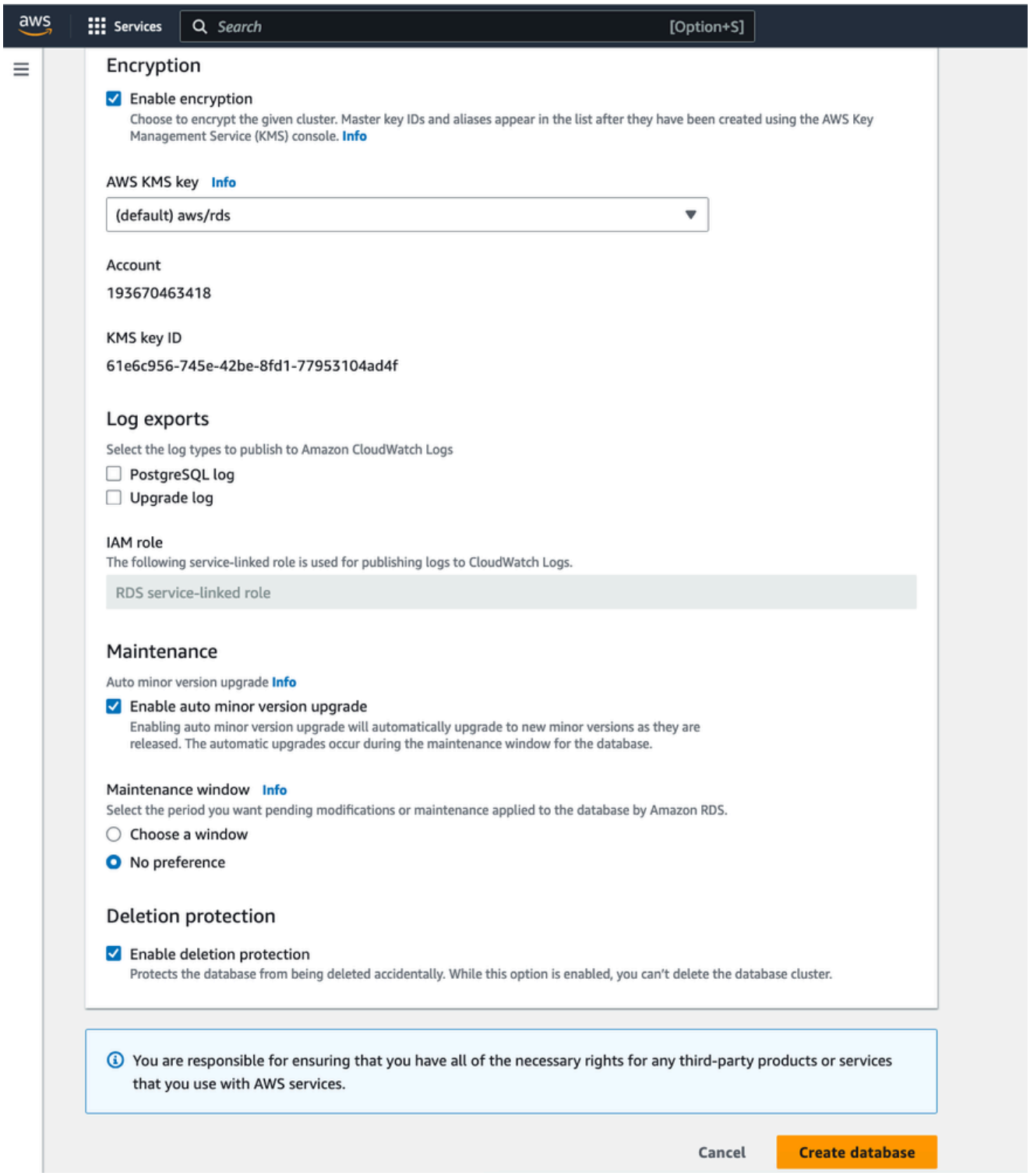
Seleccione los valores adecuados en función de su caso práctico. Hemos seleccionado los valores predeterminados.



Asegúrese de que en "Autenticación de base de datos" hemos seleccionado la autenticación de contraseña. Autentica mediante contraseñas de base de datos.


aws
Services
Search
[Option+S]
Certificate authority - optional Info
Using a server certificate provides an extra layer of security by validating that the connection is being made to an Amazon database. It does so by checking the server certificate that is automatically installed on all databases that you provision.
rds-ca-rsa2048-g1 (default)
Expiry: May 25, 2061
If you don't select a certificate authority, RDS chooses one for you.
Additional configuration
Database port Info
TCP/IP port that the database will use for application connections.
5432
Tags - optional
A tag consists of a case-sensitive key-value pair.
No tags associated with the resource.
Add new tag
You can add up to 50 more tags.
Database authentication
Database authentication options Info
Password authentication
Authenticates using database passwords.
Password and IAM database authentication (not available for Multi-AZ DB cluster)
Authenticates using the database password and user credentials through AWS IAM users and roles.
Password and Kerberos authentication (not available for Multi-AZ DB cluster)
Choose a directory in which you want to allow authorized users to authenticate with this DB instance using Kerberos Authentication.

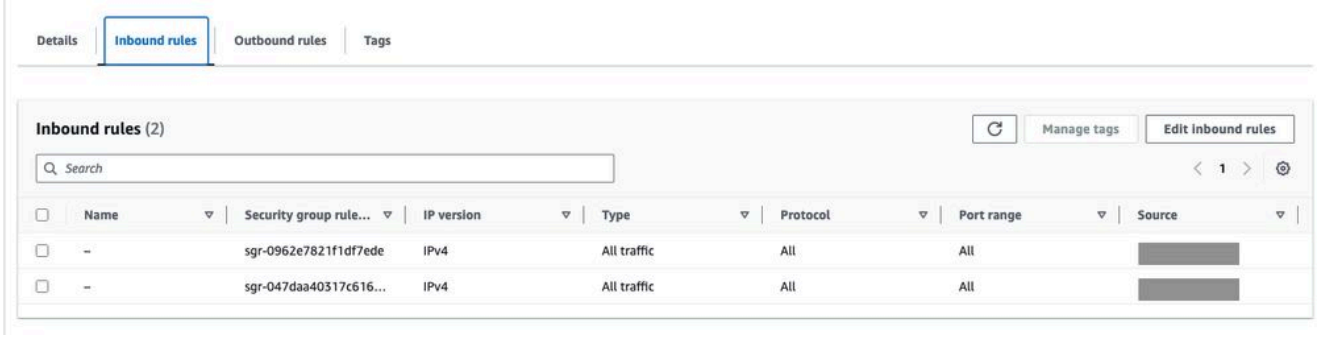

aws
Services
Search
[Option+S]
Additional configuration
Database options, encryption turned on, backup turned on, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned on.
Database options
Initial database name Info
Not supported for Multi-AZ DB cluster
If you do not specify a database name, Amazon RDS does not create a database.
DB cluster parameter group Info
default.postgres16
Option group Info
Not supported for Multi-AZ DB cluster
Backup
Enable automated backups
Creates a point-in-time snapshot of your DB cluster
Backup retention period Info
The number of days (1-35) for which automatic backups are kept.
7
days
Backup window Info
Select the period for which you want automated backups of the DB cluster to be created by Amazon RDS.
Choose a window
No preference
Copy tags to snapshots
Encryption
Enable encryption
Choose to encrypt the given cluster. Master key IDs and aliases appear in the list after they have been created using the AWS Key Management Service (KMS) console. Info
AWS KMS key Info
(default) aws/rds
Account
193670463418



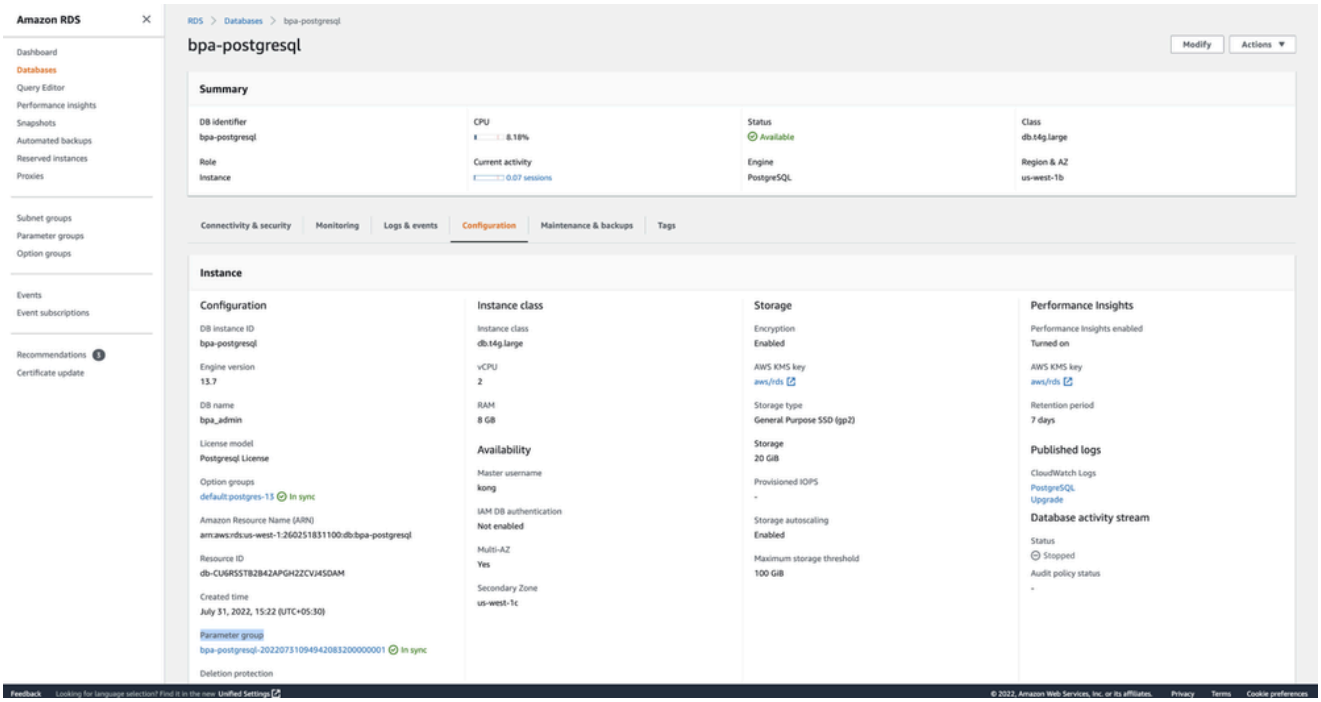
Una vez verificado, estamos listos para crear la base de datos. Vuelva al panel de Amazon RDS. Confirme que la instancia está disponible para su uso.

Reglas de grupo de seguridad

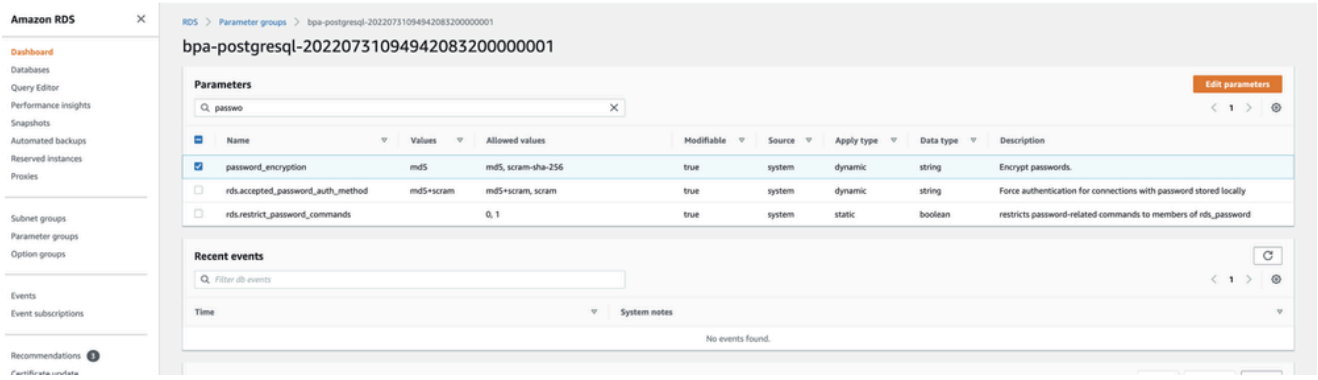
Actualice el grupo de seguridad entrante con el POD CIDR y el bloque CIDR del nodo.



En RDS -> Databases -> DB-NAME, haga clic en configuration y consulte la sección Parameter Group y haga clic en el grupo de parámetros para ver.



Busque "password_encryption" y cambie el valor a md5 desde blank / other value. Esto es necesario para que las configuraciones de campus funcionen.



Cree estas bases de datos junto con los usuarios conectándose a RDS.

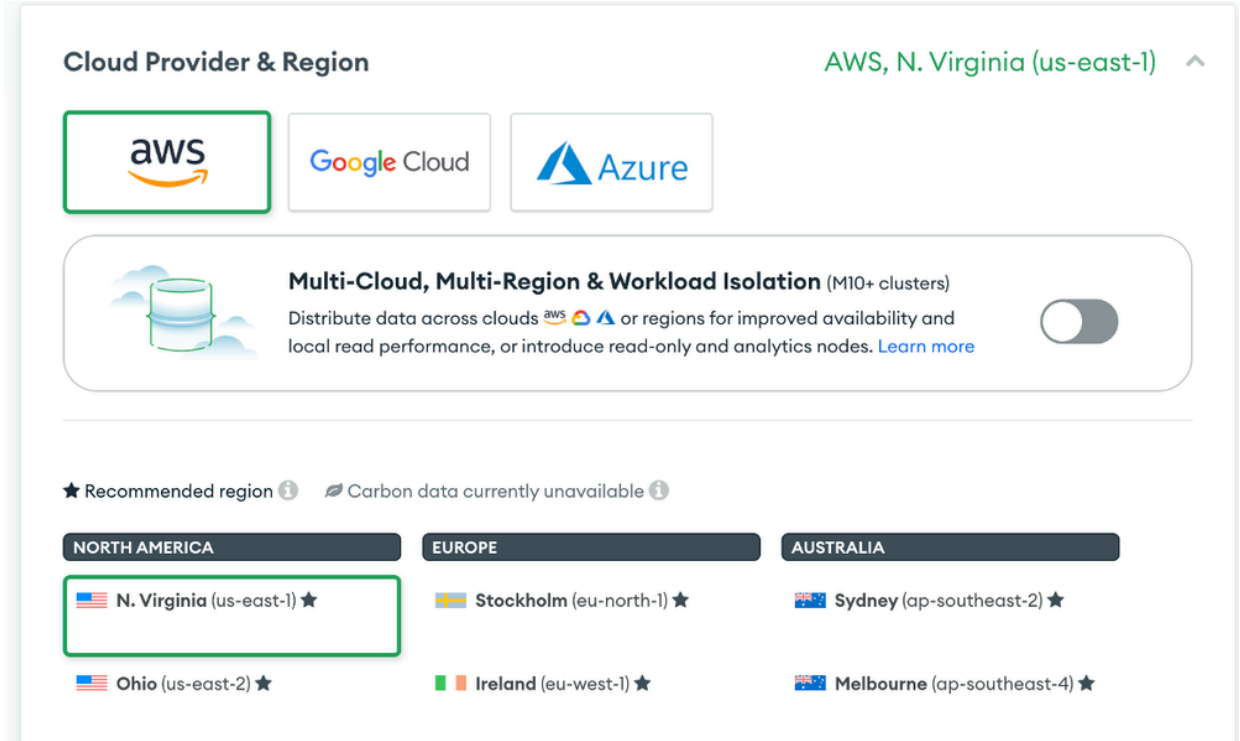
```
PG_ROOT_DATABASE=admin
PG_INITDB_ROOT_USERNAME=admin
PG_INITDB_ROOT_PASSWORD=Bp@Chang3d!
AUTH_DB_NAME=kong
AUTH_DB_USER=kong
AUTH_DB_PASSWORD=K@ngPwdCha*g3
WFE_DB_USER=camunda
WFE_DB_PASSWORD=WOrkFlo#ChangeNow
WFE_DB_NAME=process-engine
```

- Autenticación de contraseña

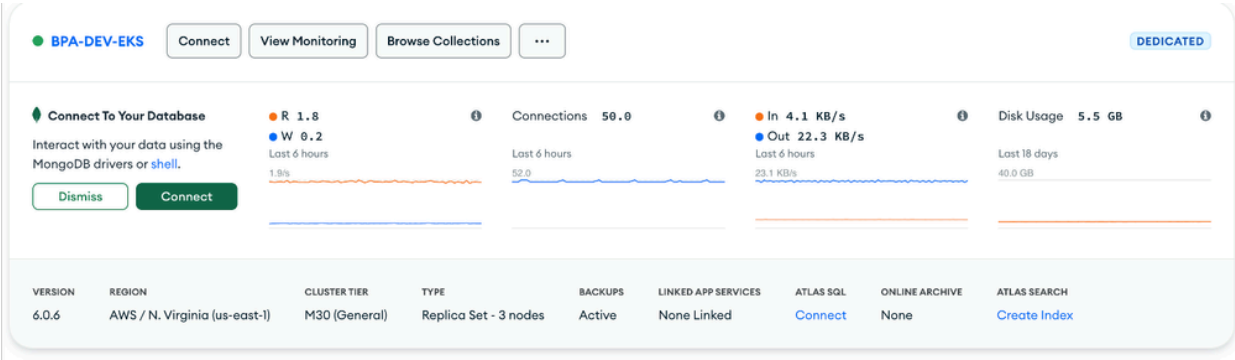
  Autentica mediante contraseñas de base de datos.

- **Configuración de Atlas MongoDB**
  La configuración de Atlas MongoDB implica:
  - **Inicio de sesión en Atlas MongoDB.**
  - **Seleccionar la organización y el proyecto.**
  - **Crear un clúster dedicado con las especificaciones adecuadas.**

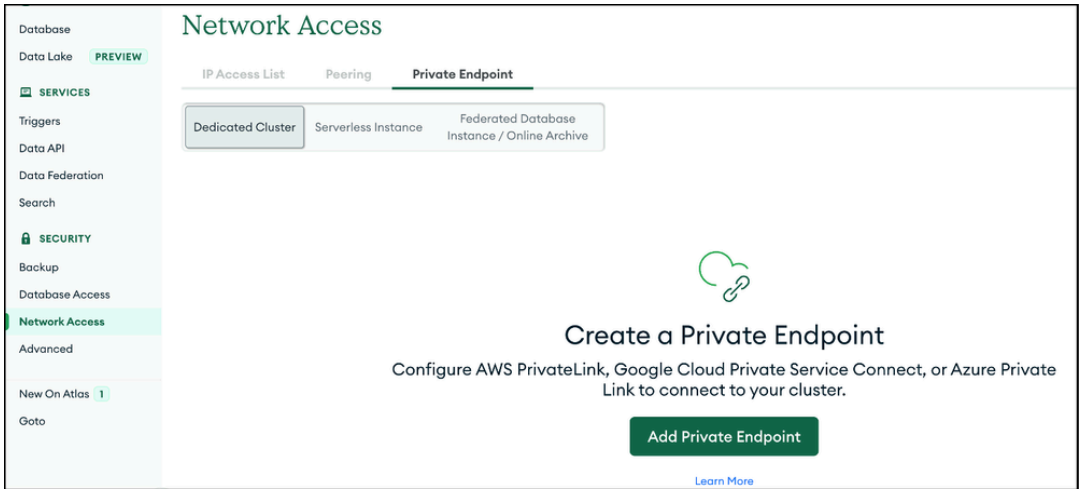
  - **Seleccione el nivel dedicado, el proveedor de nube y la región.**



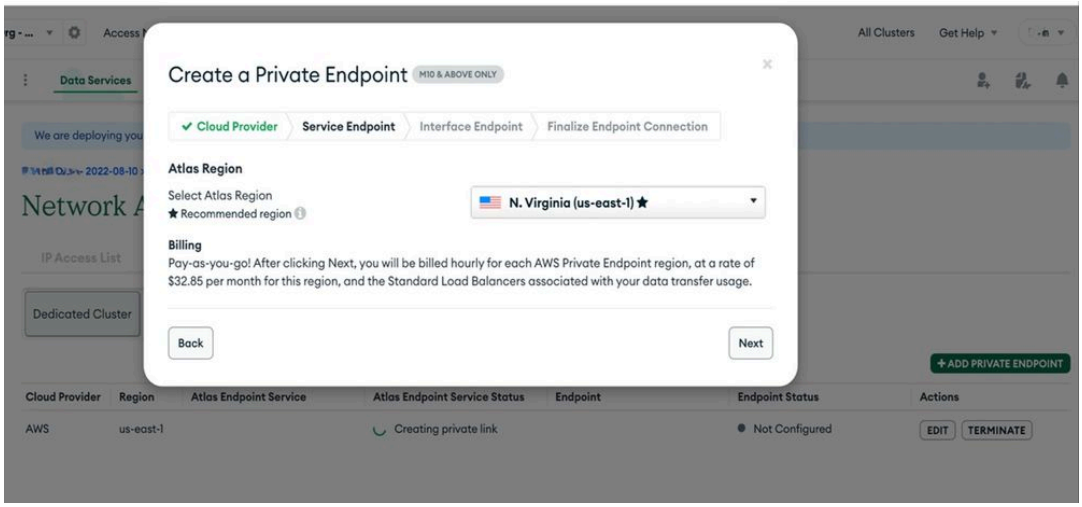
- **Seleccione el clúster dedicado del nivel adecuado (hemos utilizado M30 como nivel), proporcione el nombre del clúster adecuado y haga clic en Create Cluster (Crear clúster). Inicializará el grupo monogodb del sistema Atlas.**



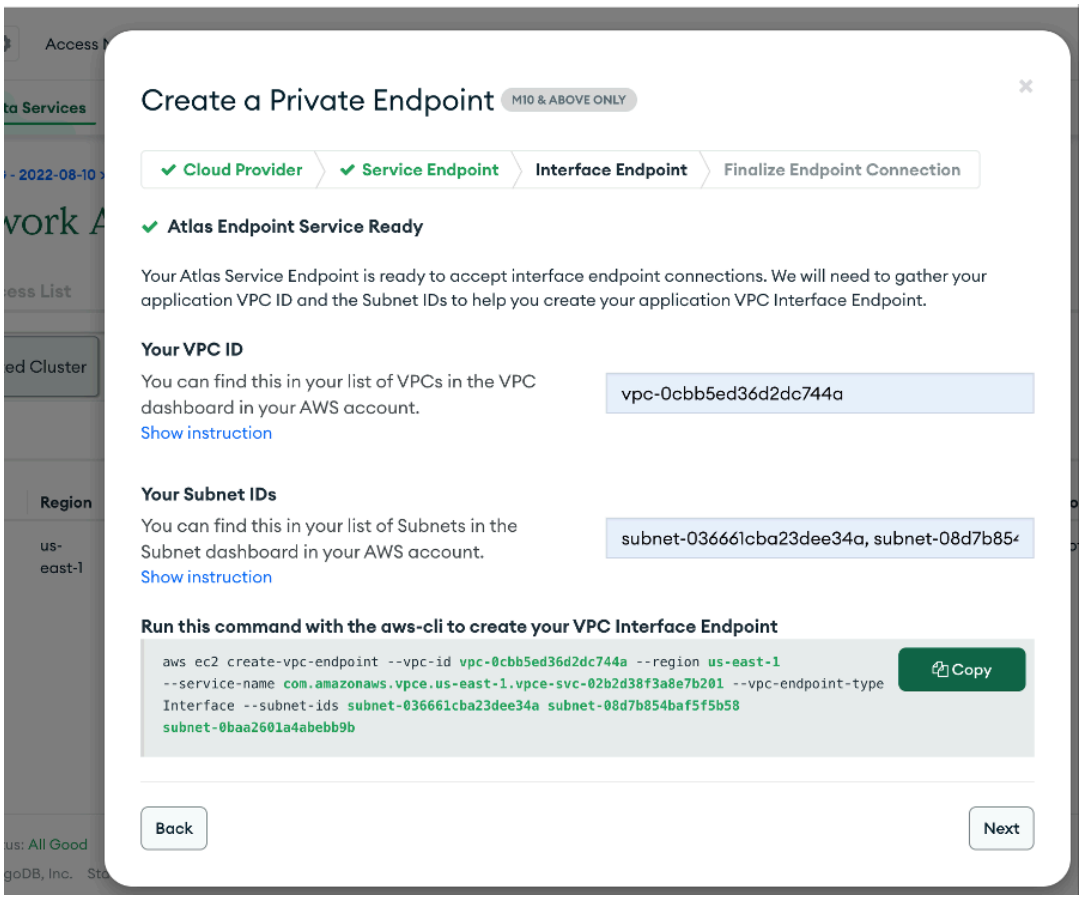
- **Configuración del terminal privado VPC para el clúster Atlas y K8S.**
  - **Haga clic en Network Access Select Private Endpoint y haga clic en Add Private Endpoint.**



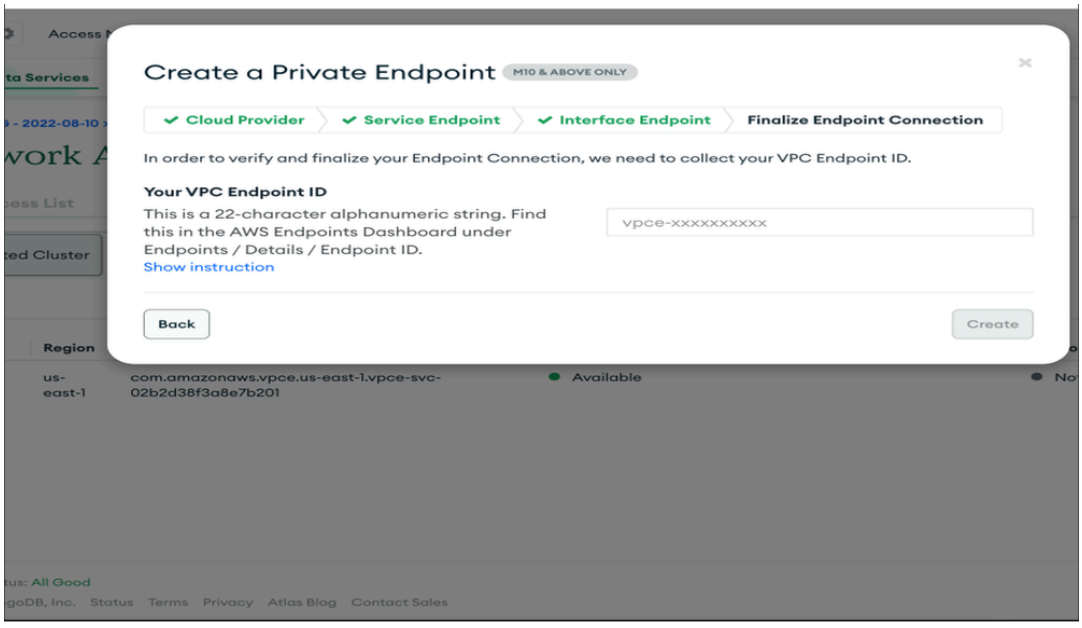
  - **Seleccione Cloud Provider como AWS, seleccione la región correspondiente y haga clic en Next (Siguiente).**



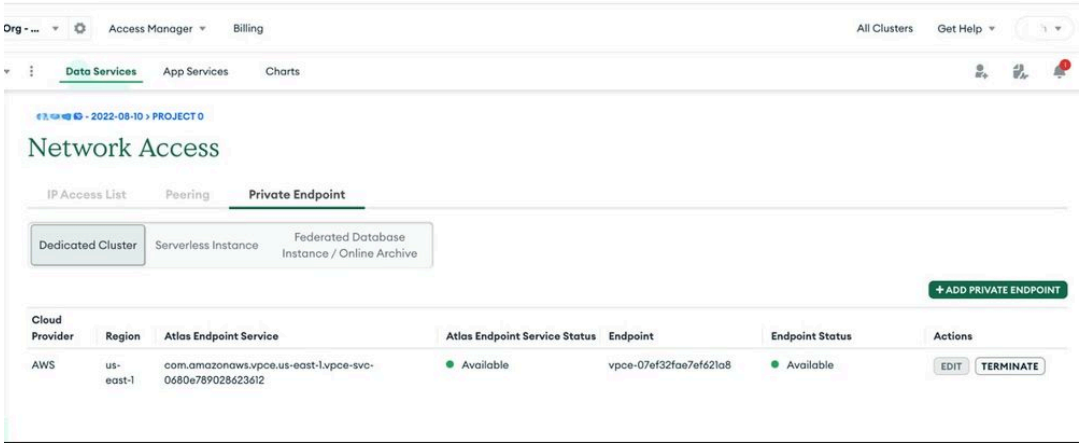
- **Proporcione el ID de PVC y los ID de subred respectivos. Una vez que ingrese los detalles, copie el comando vpc end point creation y ejecútelo en la consola de AWS. Obtendrá el ID del terminal vpc como resultado.**



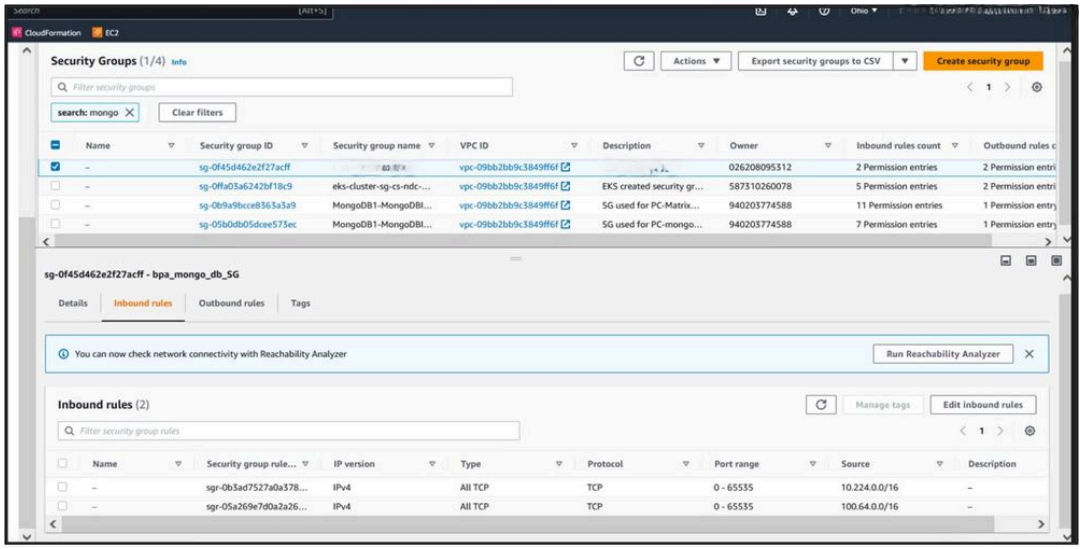
- **Haga clic en Next (Siguiente) para pegar la ID del terminal VPC y haga clic en Create (Crear).**



- **Una vez que se haya creado correctamente, el estado del terminal será Disponible, como se muestra en la siguiente imagen. Se debe crear un punto final de VPC para POD CIDR. En nuestro caso hemos utilizado "100.64.0.0/16" .**



- **Agregue reglas de entrada al terminal de vpc recién creado. El vpc-terminal estará en la cuenta padre y se debe asignar un grupo de seguridad al vpc-terminal recién creado.**



ECR como registro de imágenes

La creación de repositorios de Amazon ECR y la inserción de imágenes Docker en ellos implica varios pasos. Estos son los pasos para crear un repositorio ECR, etiquetar una imagen Docker y enviarla al repositorio mediante la CLI de AWS.

```
aws ecr create-repository --repository-name your-image-name --region your-region
```

Sustituir:

- **your-image-**name con el nombre deseado para el repositorio de ECR.
- **su región con** su región de AWS

Configuración de la función IAM para nodos EKS

Asegúrese de que los nodos de trabajo EKS (instancias EC2) tengan la función IAM necesaria asociada con permisos para extraer imágenes de ECR. La política IAM necesaria es:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "ecr:BatchCheckLayerAvailability"
      ],
      "Resource": "*"
    }
  ]
}
```

Adjunte esta política al rol IAM asociado con sus nodos de trabajo EKS.

Implementación de BPA
La implementación de BPA implica varios pasos, incluidos el etiquetado de nodos de trabajo EKS, la preparación de directorios en nodos, la copia de paquetes BPA y la implementación de BPA mediante Helm.

Para la implementación de nuestros clientes, hemos utilizado las siguientes versiones de software y servicios en la nube:

- **BPA:** 4.0.3-6
- **RDS (servicio de base de datos relacional)**: 16.3-R2
- **MongoDB Atlas**: v5.0.29
- **EKS (Elastic Kubernetes Service)**: v1.27

Estos componentes garantizan que nuestra implementación sea sólida, escalable y capaz de gestionar las cargas de trabajo necesarias de forma eficaz.

- **Etiquetado de nodos de trabajadores EKS**

```
kubectl label node
```

```
name=node-1 kubectl label node
```

```
name=node-2 kubectl label node
```

```
name=node-3 kubectl label node
```

```
name=node-4
```

- **Preparación de Directorios en Nodos**
  Nodo 1:

```
rm -rf /opt/bpa/data/
mkdir -p /opt/bpa/data/zookeeper1
mkdir -p /opt/bpa/data/zookeeper4
mkdir -p /opt/bpa/data/zookeeper5
chmod 777 /opt/bpa/data/zookeeper1
chmod 777 /opt/bpa/data/zookeeper4
chmod 777 /opt/bpa/data/zookeeper5
mkdir -p /opt/bpa/data/kafka1
chmod 777 /opt/bpa/data/kafka1
sysctl -w vm.max_map_count=262144
```

Nodo 2:

```
rm -rf /opt/bpa/data
sysctl -w vm.max_map_count=262144
mkdir -p /opt/bpa/data/kafka2
```

```
mkdir -p /opt/bpa/data/zookeeper2
mkdir -p /opt/bpa/data/zookeeper4
mkdir -p /opt/bpa/data/zookeeper5
chmod 777 /opt/bpa/data/kafka2
chmod 777 /opt/bpa/data/zookeeper2
chmod 777 /opt/bpa/data/zookeeper4
chmod 777 /opt/bpa/data/zookeeper5
```

Nodo 3:

```
rm -rf /opt/bpa/data
sysctl -w vm.max_map_count=262144
mkdir -p /opt/bpa/data/kafka3
mkdir -p /opt/bpa/data/zookeeper3
mkdir -p /opt/bpa/data/zookeeper4
mkdir -p /opt/bpa/data/zookeeper5
chmod 777 /opt/bpa/data/kafka3
chmod 777 /opt/bpa/data/zookeeper3
chmod 777 /opt/bpa/data/zookeeper4
chmod 777 /opt/bpa/data/zookeeper5
```

Nodo 4:

```
mkdir -p /opt/bpa/data/elk
mkdir -p /opt/bpa/data/metrices/prometheus
mkdir -p /opt/bpa/data/metrices/grafana
chmod 777 /opt/bpa/data/metrices
chmod 777 /opt/bpa/data/metrices/prometheus
chmod 777 /opt/bpa/data/metrices/grafana
sysctl -w vm.max_map_count=262144
```

- Copia de paquetes BPA

```
scp -r packages to node1:/opt/bpa/
scp -r packages to node2:/opt/bpa/
scp -r packages to node3:/opt/bpa/
scp -r packages to node4:/opt/bpa/
```

- Implementación de BPA mediante Helm

```
helm install bpa-rel --create-namespace --namespace bpa-ns /opt/EKS/bpa-helm-chart
```

Configuración de entrada

- **Habilitación del ingreso**
  Actualice **`values`**.yamlpara habilitar el ingreso:

```
ingress_controller: {create: true}
```

- **Creación de un secreto mediante un certificado BPA**
  Navegue hasta el directorio de certificados y cree un secreto:

```
cd /opt/bpa/
```

```
/bpa/conf/common/certs/ kubectl create secret tls bpa-certificate-ingress --cert=bap-cert
```

- **Actualización del controlador de ingreso**
  Agregue el secreto recién creado en el **`ingress-controller.yaml`** archivo:

```
cd /opt/bpa/
```

```
/templates/ vi ingress-controller.yaml "- --default-ssl-certificate=$(POD_NAMESPACE)/bpa-
```

- **Actualización del certificado de ingreso**
  Realice la eliminación e instalación del timón para actualizar el certificado de ingreso.

Especificaciones del entorno

Las especificaciones de entorno incluyen requisitos para instancias EC2, equilibradores de carga, terminales VPC e instancias RDS. Las especificaciones clave son:

Requisitos de EC2:

Requisitos de almacenamiento:2 TB de espacio por nodos. Monte el volumen EBS en /opt y agregue una entrada en /etc/fstab para todos los nodos.

Grupo de seguridad entrante: 30101, 443, 0 - 65535 TCP, 22 para ssh.

Grupo de seguridad saliente: todo el tráfico debe estar habilitado.

Resolución de DNS: EC2 debe tener resoluciones en las instalaciones en /etc/resolve.conf.

Requisitos del equilibrador de carga:

- Los puertos de receptores deben ser 443, 30101.
- Requisitos del terminal VPC (Atlas MongoDB).
- Los terminales VPC creados para la conectividad Atlas están disponibles en la cuenta principal (aws-5g-ndc-prod). El terminal VPC debe tener un grupo de seguridad que permita todo el acceso entrante (0 - 65535).

Requisitos de RDS:

Tipo de RDS: db.r5b.2xlarge

Versión del motor Postgres: 13.7

Grupo de seguridad: el tráfico entrante debe permitir el tráfico del origen de POD CIDR.

Conceptos y componentes clave
Comprender los fundamentos de Kubernetes es esencial para implementar y administrar eficazmente aplicaciones usando Amazon EKS.

Conclusión
Este documento proporciona una guía detallada para implementar y administrar aplicaciones de automatización de procesos empresariales (BPA) mediante Amazon EKS. Siguiendo los pasos descritos y comprendiendo los conceptos clave, las organizaciones pueden aprovechar las ventajas de EKS para sus aplicaciones BPA en contenedores.

Referencias

- **Servicios web de Amazon, "Documentación de Amazon EKS", [Online]. Disponible:https://docs.aws.amazon.com/eks/**
- **Kubernetes, "Documentación de Kubernetes" [Online]. Disponible:https://kubernetes.io/docs/home/**
- **Guía rápida de Cisco BPA https://www.cisco.com/c/en/us/solutions/collateral/service-provider/at-a-glance-c45-742579.html**
- **Guía de operaciones de BPA https://www.cisco.com/c/dam/en/us/support/docs/bpa/v403/cisco-bpa-operations-guide-v403.pdf**
- **Guía del desarrollador de BPA https://www.cisco.com/c/dam/en/us/support/docs/bpa/v403/cisco-bpa-developer-guide-v403.pdf**

Acerca de esta traducción
Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).