Amazon EKS에서 비즈니스 프로세스 자동화 애플리케이션 배포 및 관리: 실용적 가이드

목차

요약

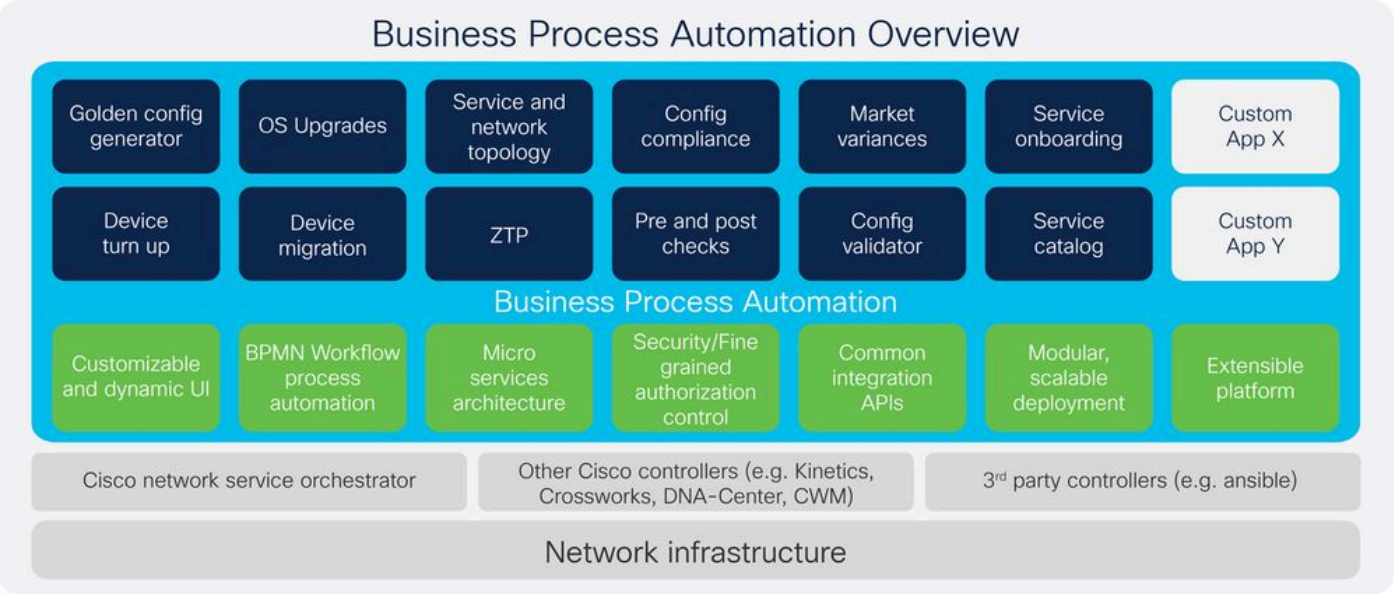
이 백서에서는 Amazon Elastic Kubernetes Service(EKS)를 사용한 BPA(Business Process Automation) 애플리케이션 배포 및 관리에 대한 포괄적인 가이드를 제공합니다. 사전 요구 사항에 대해 간략하게 설명하고 EKS 활용의 이점을 강조하며 EKS 클러스터, Amazon RDS 데이터베이스 및 MongoDB Atlas 설정에 대한 단계별 지침을 제공합니다. 또한 이 백서는 구축 아키텍처를 자세히 살펴보고 환경 요구 사항을 명시하여 컨테이너화된 BPA 애플리케이션에 EKS를 활용하려는 조직을 위한 완벽한 리소스를 제공합니다.

키워드

Amazon EKS, Kubernetes, AWS, RDS, MongoDB Atlas, DevOps, 클라우드 컴퓨팅, 비즈니스 프로세스 자동화.

소개

BPA



오늘날의 디지털 시대에 기업은 다양한 IT 환경에서 복잡한 비즈니스 프로세스를 간소화하고 자동화하고자 합니다. BPA(Business Process Automation)는 조직의 운영 효율성 향상, 오류 감소, 서비스 제공 개선을 가능하게 하는 중추적 기술로 부상했습니다. BPA는 워크플로 자동화, 서비스 프로비저닝 및 기성 자동화 애플리케이션을 발전시키기 위한 몇 가지 주요 혁신 및 개선 사항을 소개합니다.

BPA 플랫폼은 OS 업그레이드, 서비스 프로비저닝, 오케스트레이션 엔진으로의 통합과 같은 비즈니스 및 IT/운영 활용 사례와 애플리케이션을 호스팅합니다. 고객은 Cisco 전문가를 통해 제공되는 자문, 구현, 비즈니스 크리티컬 서비스, 솔루션 지원을 포함한 서비스 및 BPA 기능 라이프사이클에 액세스하고, 모범 사례, 검증된 기법 및 방법론을 통해 비즈니스 프로세스를 자동화하고 시스템의 위험을 제거할 수 있습니다.

이러한 라이프사이클 기능은 서브스크립션 기반이거나 개별 요구에 맞게 맞춤화할 수 있습니다. 구현 서비스는 자동화 가속화를 위해 툴과 프로세스를 정의, 통합 및 배포하는 데 도움이 됩니다. Cisco 전문가는 민첩한 프로세스 및 CICD(Continuous Integration and Continuous Delivery) 툴을 기반으로 사용자 스토리를 수집하고 설계하고 개발하기 위한 공식적인 프로세스를 수행하고, 신규 또는 기존 워크플로, 장치 및 서비스의 자동화된 테스트를 통해 유연한 서비스를 구현합니다. Solution Support를 통해 고객은 24시간 연중무휴 중앙 집중식 지원을 이용할 수 있으며, 소프트웨어 중심 문제에 초점을 맞추고 다중 공급업체 및 Cisco의 계층형 소프트웨어 모델을 통해 제공되는 오픈 소스 지원을 이용할 수 있습니다. Cisco 솔루션 지원 전문가는 최초 통화에서 최종 해결까지 사례를 관리하고 여러 공급업체와 동시에 업무를 수행하는 주요 연락 창구의 역할을 수행할 수 있도록 도와줍니다. 솔루션 수준의 전문가와 함께 작업하면 최대 44% 적은 수의 문제를 경험하여 비즈니스 연속성을 유지하고 BPA 투자 수익을 빠르게 얻을 수 있습니다.

FMC 및 Ansible 관리 디바이스 지원, AQF(Advanced Queuing Framework)를 사용한 병렬 실행, NDFC 및 FMC 디바이스에 대한 확장된 구성 규정 준수 등의 주요 기술 기능은 BPA를 대규모 엔터프라이즈 자동화를 위한 포괄적인 솔루션으로 포지셔닝합니다. SD-WAN 관리, 장치 온보딩 및 방화벽 정책 거버넌스의 기능이 추가된 이 릴리스는 대규모 멀티벤더 환경의 요구 사항을 충족하면서 네트워크 보안 및 자동화의 중요한 측면을 다룹니다.

EKS

Amazon Elastic Kubernetes Service(EKS)는 Amazon Web Services(AWS)에서 제공하는 완전히 관리되는 Kubernetes 서비스입니다. 2018년에 출시된 EKS는 오픈 소스 컨테이너 오케스트레이션 플랫폼인 Kubernetes를 사용하여 컨테이너화된 애플리케이션의 구축, 관리 및 확장 프로세스를 간소화합니다. EKS는 Kubernetes 클러스터 관리의 복잡성을 추상화하므로 개발자는 기본 인프라를 처리할 필요 없이 애플리케이션을 구축하고 실행하는 데 집중할 수 있습니다.

애플리케이션 구축에 Amazon EKS를 사용할 때의 이점

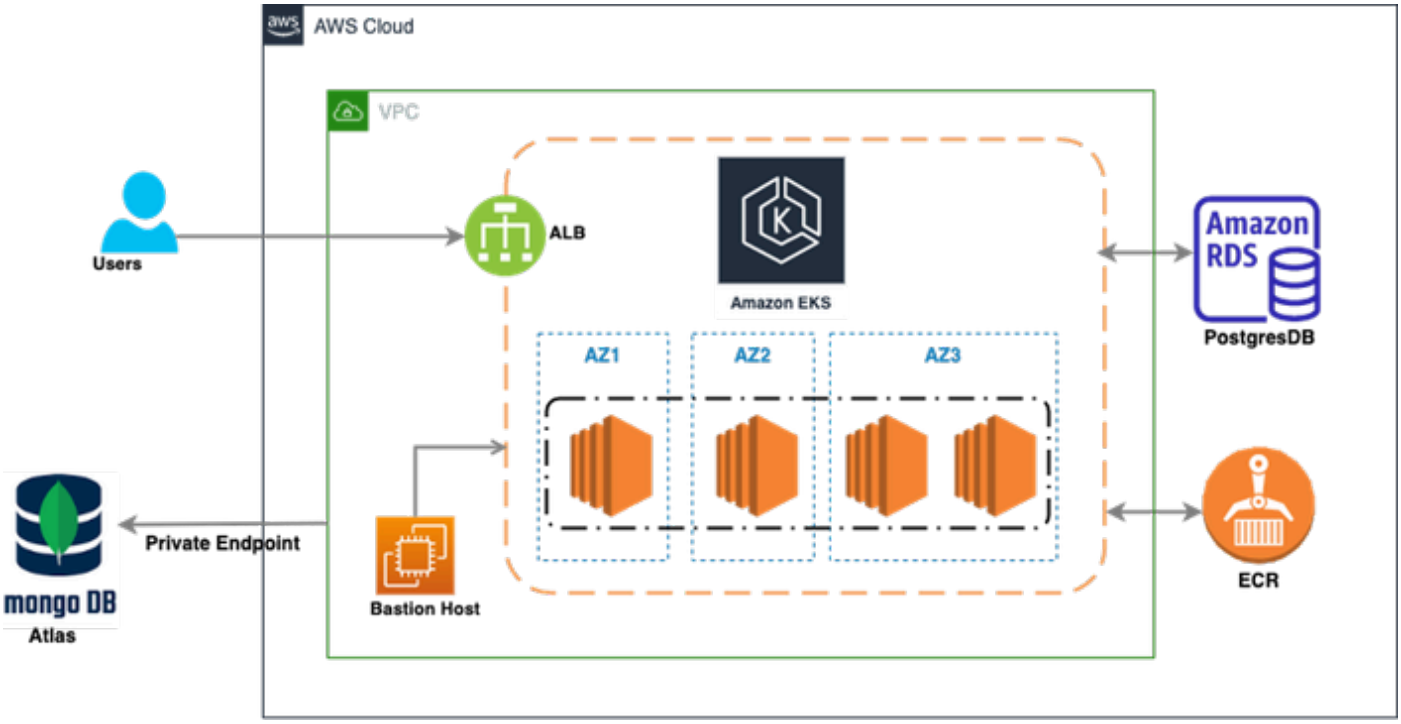
Amazon EKS는 애플리케이션 구축에 여러 가지 이점을 제공하므로 컨테이너화된 애플리케이션 및 마이크로서비스를 활용하는 조직에서 많이 사용하는 선택입니다.

주요 이점은 다음과 같습니다.

- **관리형 Kubernetes 컨트롤 플레인**: EKS는 Kubernetes 컨트롤 플레인의 구축, 확장 및 유지 보수를 처리하여 운영 부담을 줄입니다.

- **간소화된 클러스터 관리**: EKS는 Kubernetes 클러스터 설정 및 관리의 복잡성을 추상화합니다.
- **확장성**: EKS는 증가하는 워크로드를 수용하기 위해 손쉽게 클러스터를 확장할 수 있도록 지원합니다.
- **고가용성**: EKS는 다중 가용 영역 구축을 지원하여 가용성과 내결함성을 향상시킵니다.
- **AWS 서비스와의 통합**: EKS는 다양한 AWS 서비스와 원활하게 통합됩니다.
- **DevOps Automation**: EKS는 컨테이너화된 애플리케이션을 위해 지속적인 통합 및 지속적인 구축(CI/CD)을 지원합니다.

BPA 구축 아키텍처



이 이미지는 몇 가지 주요 구성 요소를 사용하여 **AWS**에 배포된 클라우드 기반 인프라의 상위 레벨 아키텍처를 나타냅니다. 여기 다이어그램의 분류가 있다:

1. **Amazon EKS(Elastic Kubernetes Service)**: 다이어그램의 핵심은 3개의 가용 영역(AZ1, AZ2, AZ3)에 걸쳐 Amazon EKS가 구축되며, 각 영역 내에는 Kubernetes 작업자 노드가 있습니다. 이는 워크로드가 여러 가용 영역에 분산되어 있기 때문에 가용성과 내결함성이 뛰어난 설정을 나타냅니다.

2. **ALB(Application Load Balancer)**: 애플리케이션 워크로드를 처리하기 위해 EKS 클러스터 전체에 트래픽을 분배하고 사용자로부터 트래픽을 수신하는 전면에 배치됩니다. 로드 밸런서는 요청이 균등하게 분배되도록 하며 트래픽 수요에 따라 확장을 처리할 수 있습니다.

3. **Amazon RDS(Relational Database Service)** - PostgreSQL: 다이어그램의 오른쪽에 PostgreSQL을 실행하는 Amazon RDS 인스턴스가 있습니다. 이 데이터베이스는 EKS 클러스터

내에서 실행 중인 응용 프로그램에서 액세스할 수 있습니다.

4. **ECR(Elastic Container Registry)**: 여기서 Docker 컨테이너 이미지가 저장 및 관리되며, 이는 워크로드를 실행하기 위해 Amazon EKS에 구축됩니다.

5. **MongoDB Atlas**: 왼쪽의 MongoDB Atlas는 프라이빗 엔드포인트를 통해 아키텍처에 통합됩니다. MongoDB Atlas는 클라우드 호스팅 NoSQL 데이터베이스 서비스로, 여기에서 문서 기반 데이터베이스 요구 사항을 처리하는 데 사용됩니다. 프라이빗 엔드포인트는 MongoDB Atlas 인스턴스와 다른 AWS 구성 요소 간의 안전한 프라이빗 통신을 보장합니다.

6. **Bastion Host**: VPC(Virtual Private Cloud) 내에 있는 Bastion Host는 관리자가 인터넷에 직접 노출하지 않고도 VPC 내의 리소스에 액세스할 수 있는 보안 진입점을 제공합니다.

전반적으로 이 아키텍처는 관계형(PostgreSQL) 및 NoSQL(MongoDB) 데이터베이스를 모두 지원하는 동시에 Amazon EKS를 사용하여 컨테이너화된 애플리케이션을 배포하고 관리할 수 있는 가용성과 확장성이 뛰어나며 안전한 솔루션을 제공합니다.

- **EKS 클러스터 설정**
  AWS CLI를 사용하여 Amazon EKS 클러스터를 생성하려면 `eksctl` 명령줄 유틸리티를 사용할 수 있습니다. 다음은 예제 명령입니다.

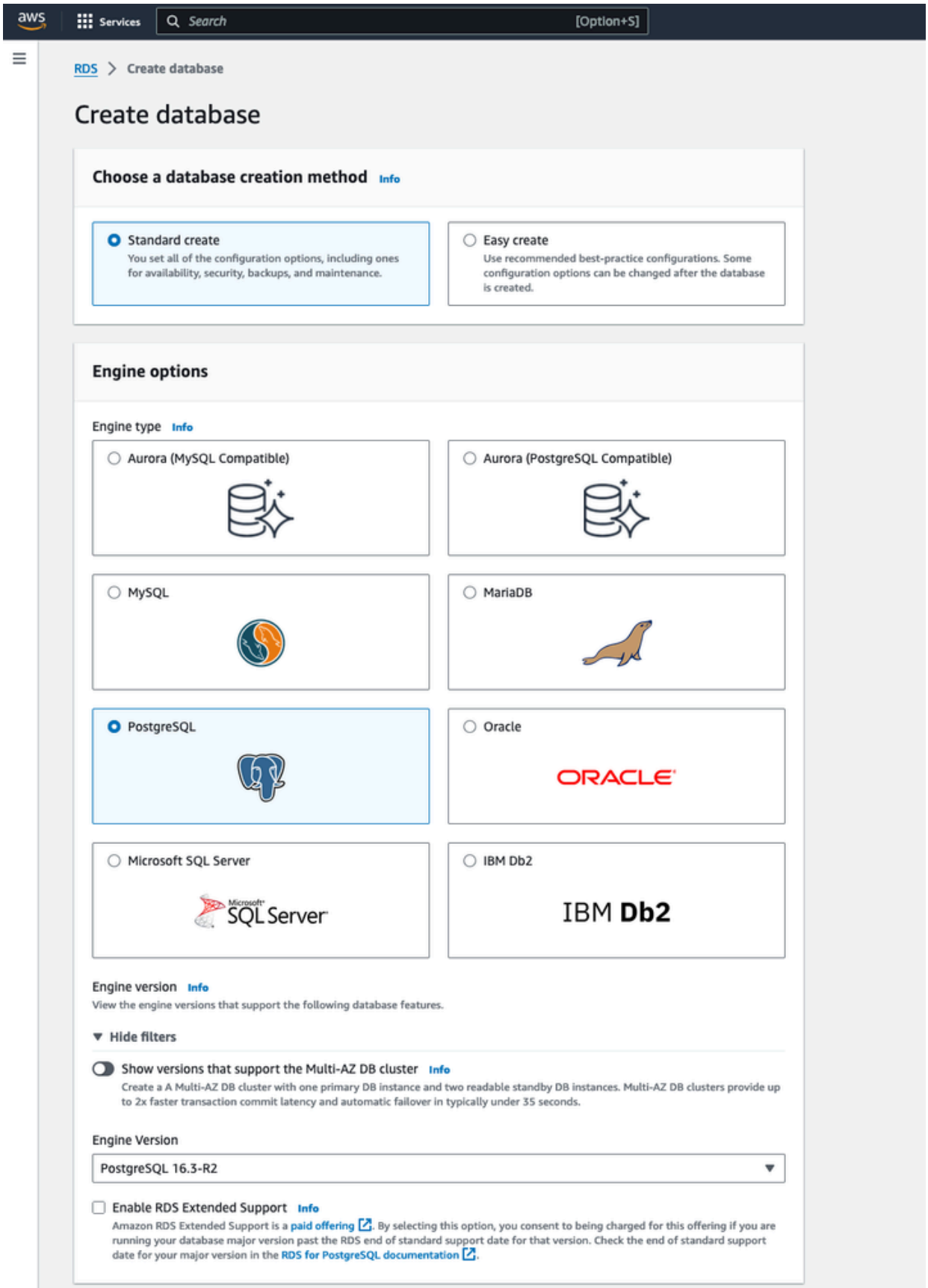
```
eksctl create cluster \
  --name
```

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

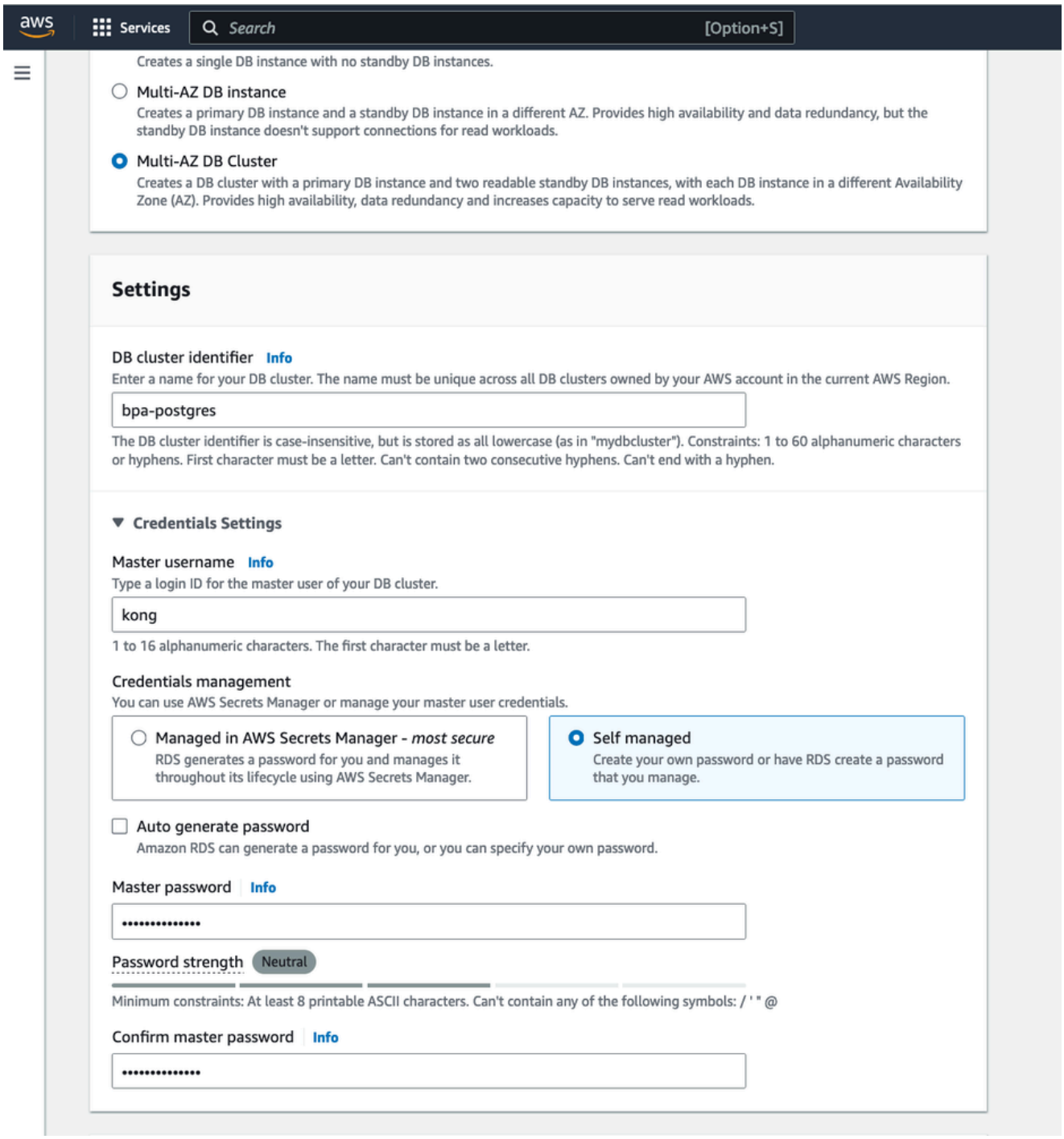
- **RDS 데이터베이스 설치**

  Amazon RDS에 관계형 데이터베이스를 배포하려면 다음 단계를 수행해야 합니다.

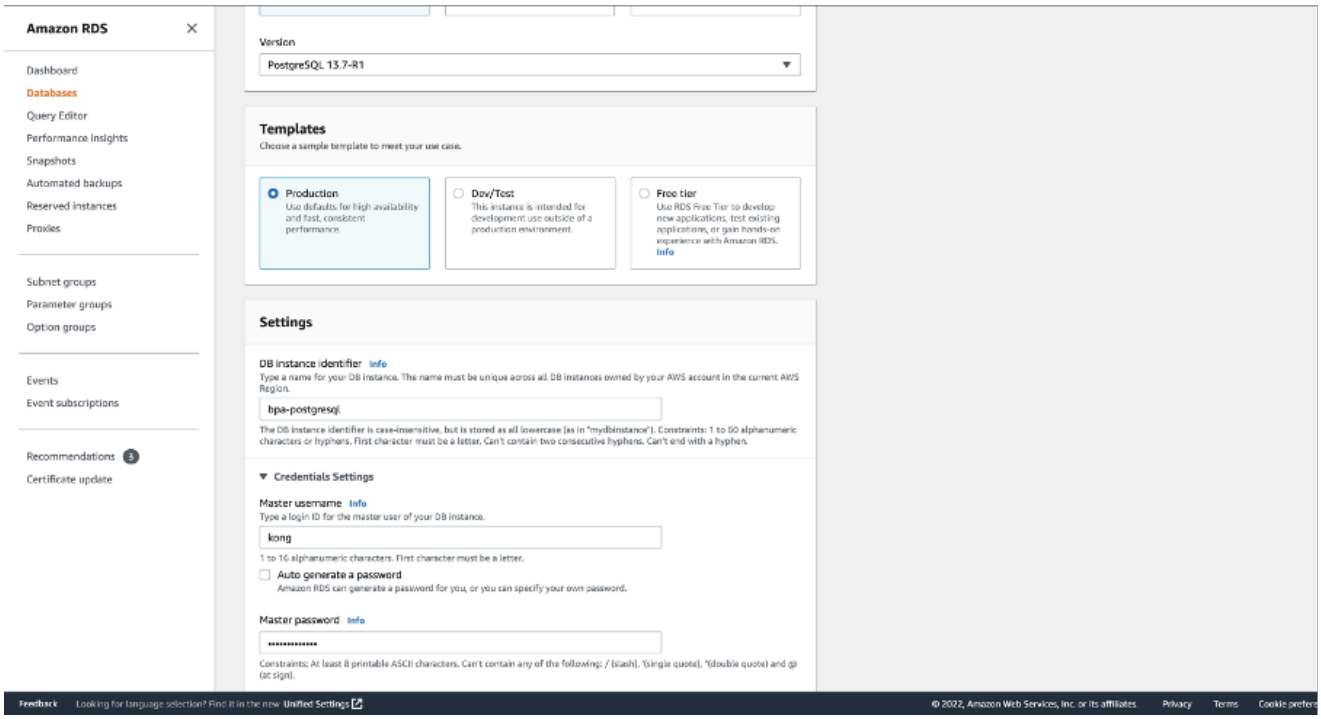
  - AWS Management Console에 액세스하여 Amazon RDS 서비스로 이동합니다.
  - 원하는 사양으로 새 데이터베이스 인스턴스를 생성합니다.
  - Amazon EKS 클러스터에서 들어오는 연결을 허용하도록 보안 그룹을 구성합니다.



드롭다운 메뉴를 사용하여 최신 버전의 PostgreSQL을 선택합니다. 여기서는 "PostgreSQL 16.3-R1"입니다.



이를 위해 데이터베이스 인스턴스에 이름을 지정하고 사용자 이름 및 비밀번호를 생성합니다.

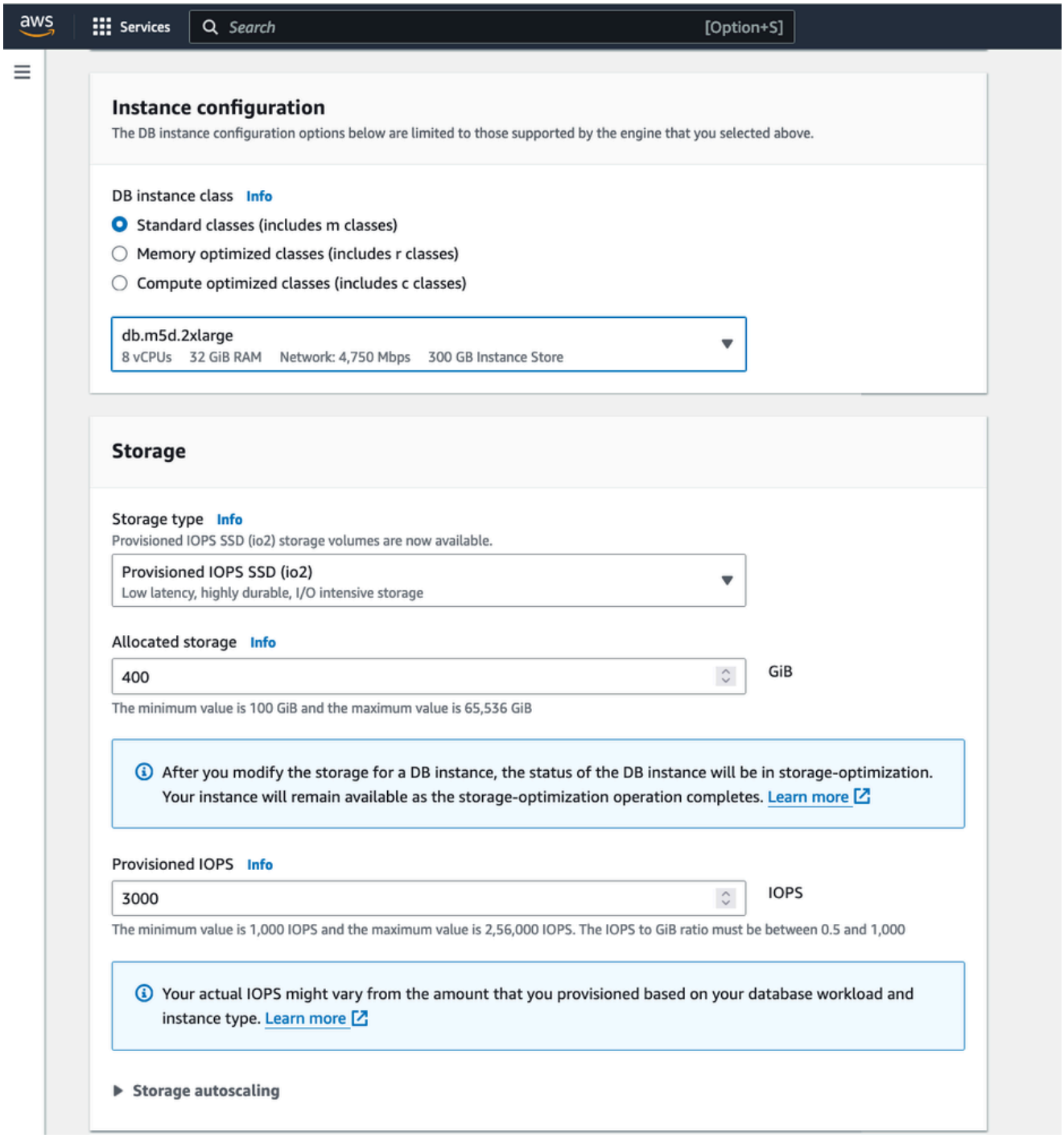


"DB 인스턴스 크기" 및 "스토리지"에 대한 기본 설정이 선택되어 있는지 확인합니다.

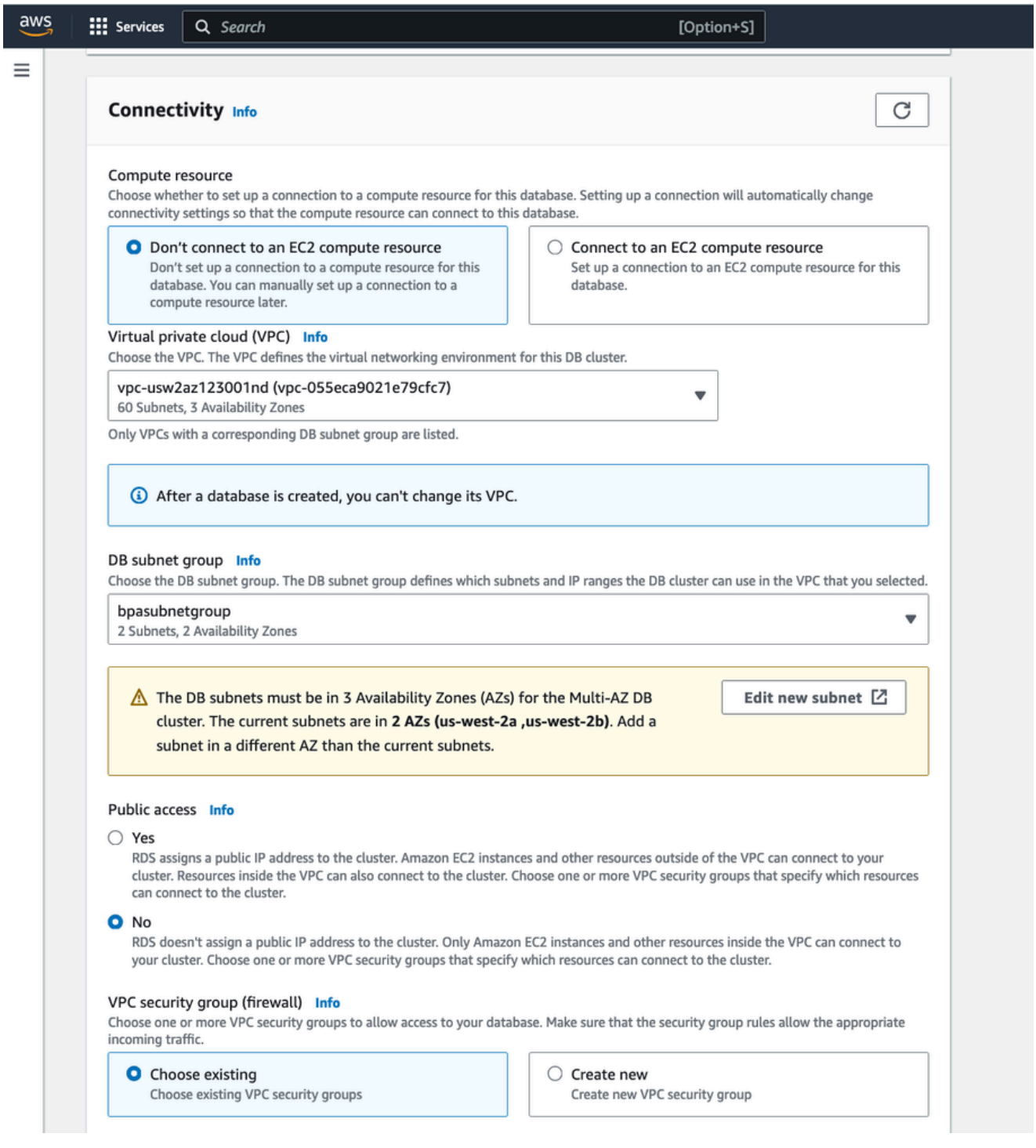
클러스터 크기 및 데이터 요구 사항에 따라 적절한 DB 인스턴스 크기 및 스토리지 유형을 선택합니다.

사용 사례에 따라 다음 컨피그레이션을 선택했습니다.

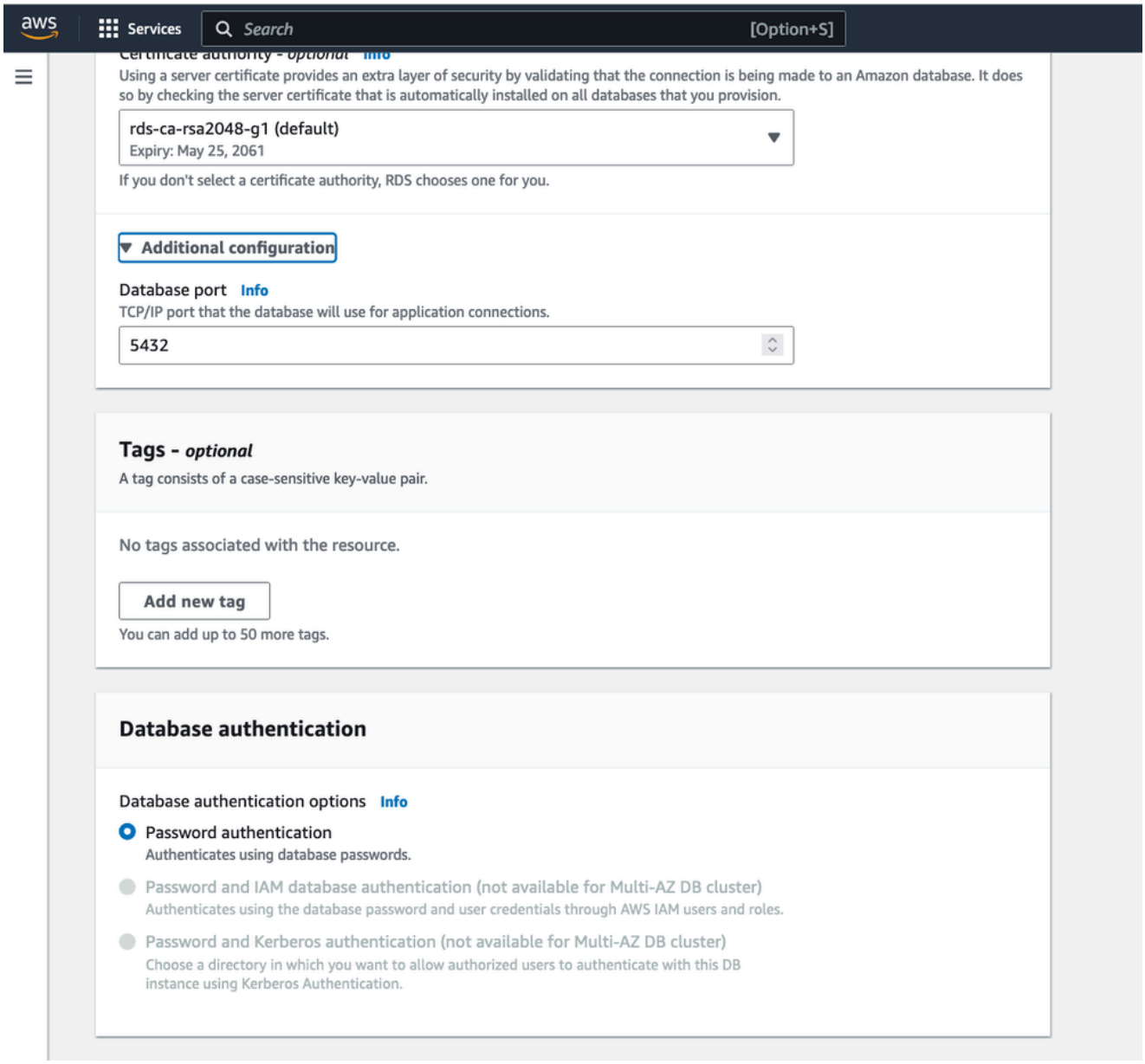
- **DB 인스턴스 크기**: db.m5d.2xlarge
    - vCPU 8개
    - 32Gb RAM
    - 네트워크: 4,750Mbps
    - 300GB 인스턴스 저장소

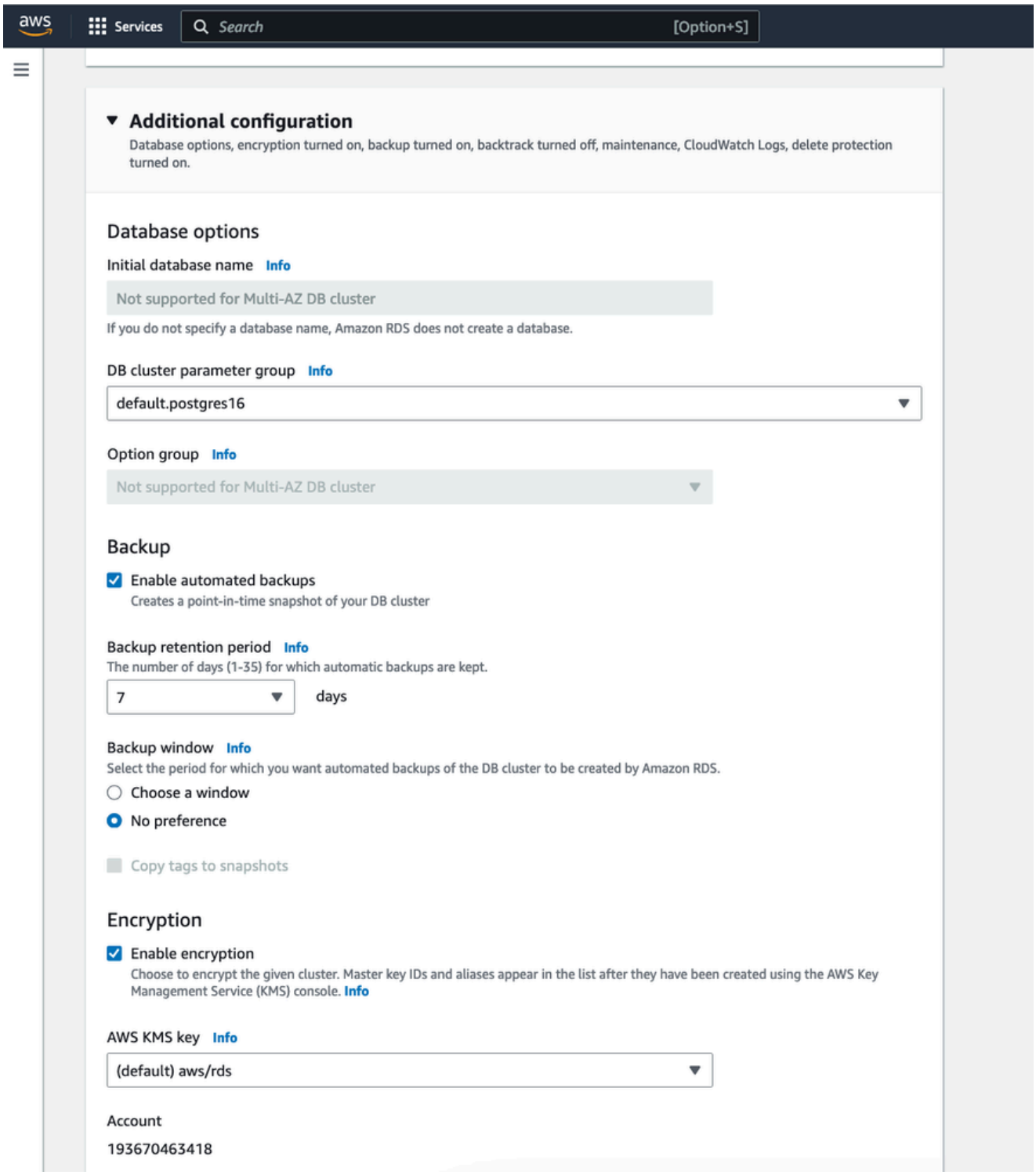


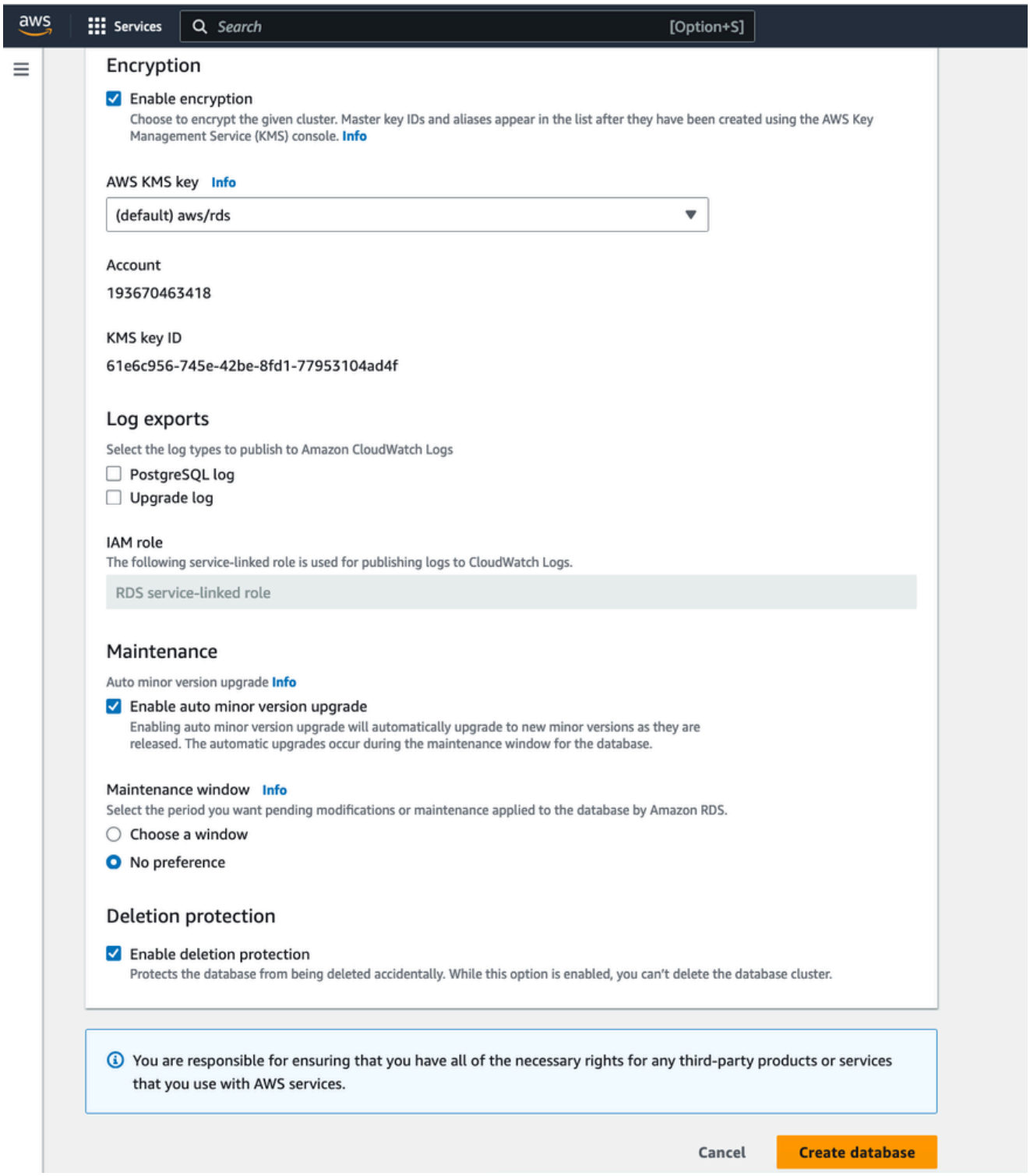
사용 사례에 따라 적절한 값을 선택합니다. 기본값을 선택했습니다.



"Database authentication(데이터베이스 인증)"에서 Password authentication(비밀번호 인증)을 선택했는지 확인합니다. 데이터베이스 비밀번호를 사용하여 인증합니다.


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.

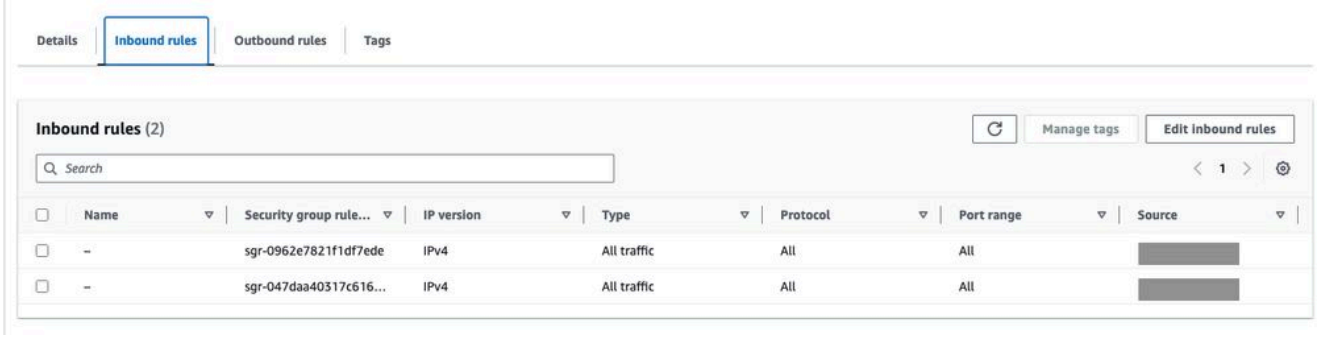

aws
Services
Search
[Option+S]
Additional configuration
Database options, encryption turned on, backup turned on, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned on.
Database options
Initial database name Info
Not supported for Multi-AZ DB cluster
If you do not specify a 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



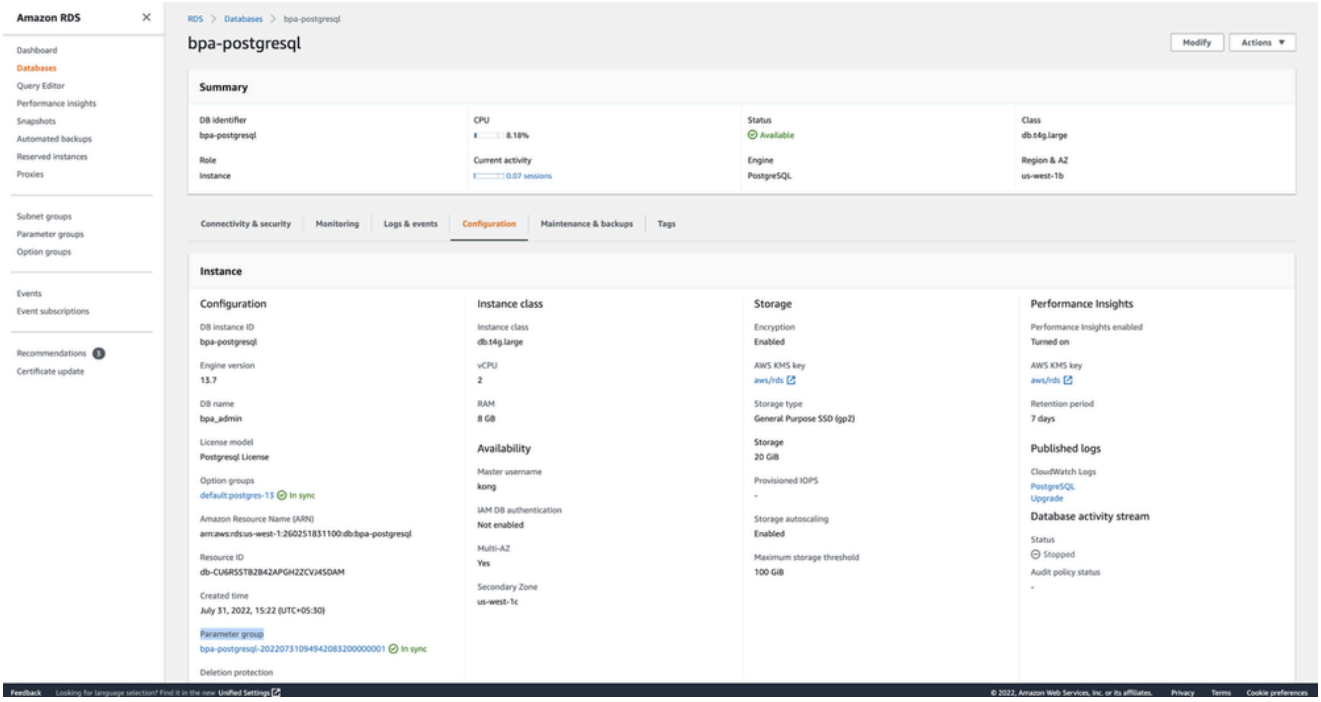
검증이 완료되면 데이터베이스를 생성할 준비가 됩니다. Amazon RDS 대시보드로 돌아갑니다. 인스턴스를 사용할 수 있는지 확인합니다.

보안 그룹 규칙

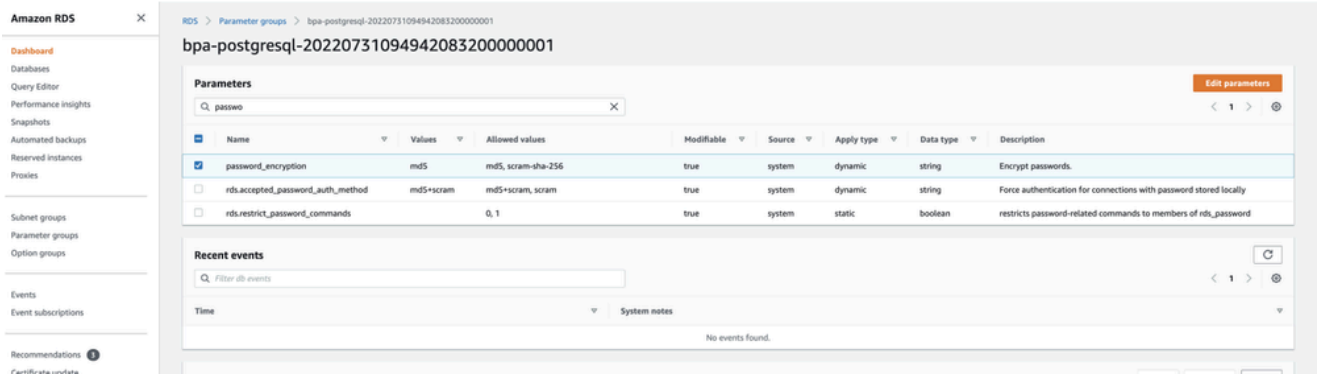
Pod CIDR 및 노드 CIDR 블록으로 인바운드 보안 그룹을 업데이트합니다.



RDS -> 데이터베이스 -> DB-NAME에서 구성을 클릭하고 매개 변수 그룹 섹션을 참조한 다음 표시할 매개 변수 그룹을 클릭합니다.



"password_encryption"을 검색하고 값을 blank/other 값에서 md5로 변경합니다. 이는 camunda 컨피그레이션이 작동하기 위해 필요합니다.



RDS에 연결하여 이러한 데이터베이스를 사용자와 함께 만듭니다.

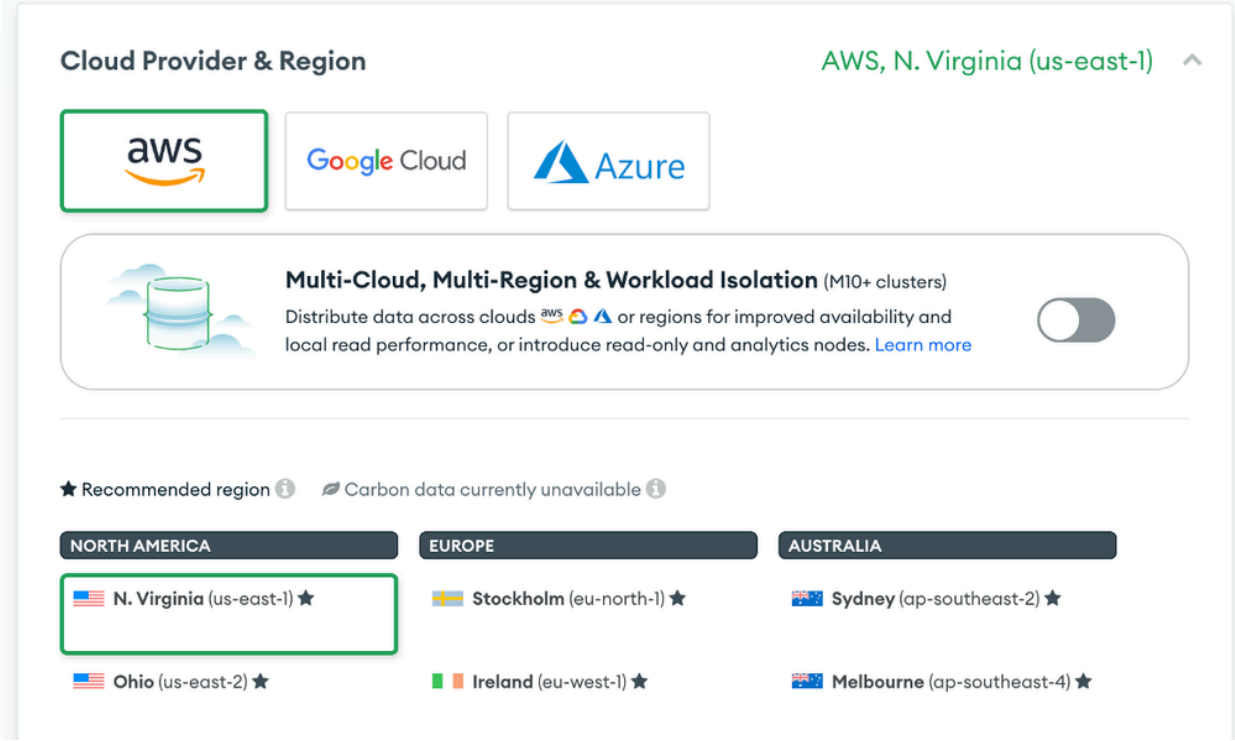
```
PG_ROOT_DATABASE=admin
PG_INITDB_ROOT_USERNAME=admin
PG_INITDB_ROOT_PASSWORD=Bp@Chang3d!
AUTH_DB_NAME=kong
AUTH_DB_USER=kong
AUTH_DB_PASSWORD=K@ngPwdCha*g3
WFE_DB_USER=camunda
WFE_DB_PASSWORD=WOrkFlo#ChangeNow
WFE_DB_NAME=process-engine
```

- 비밀번호 인증

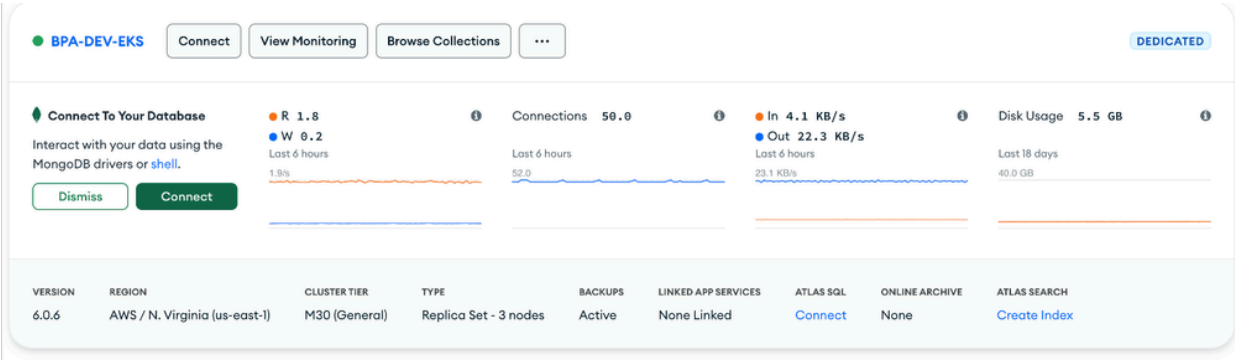
  데이터베이스 비밀번호를 사용하여 인증합니다.

- **Atlas MongoDB 설정**
  Atlas MongoDB 설정은 다음과 같습니다.
  - **Atlas MongoDB에 로그인합니다.**
  - **조직 및 프로젝트 선택**
  - **적절한 사양으로 전용 클러스터 생성**

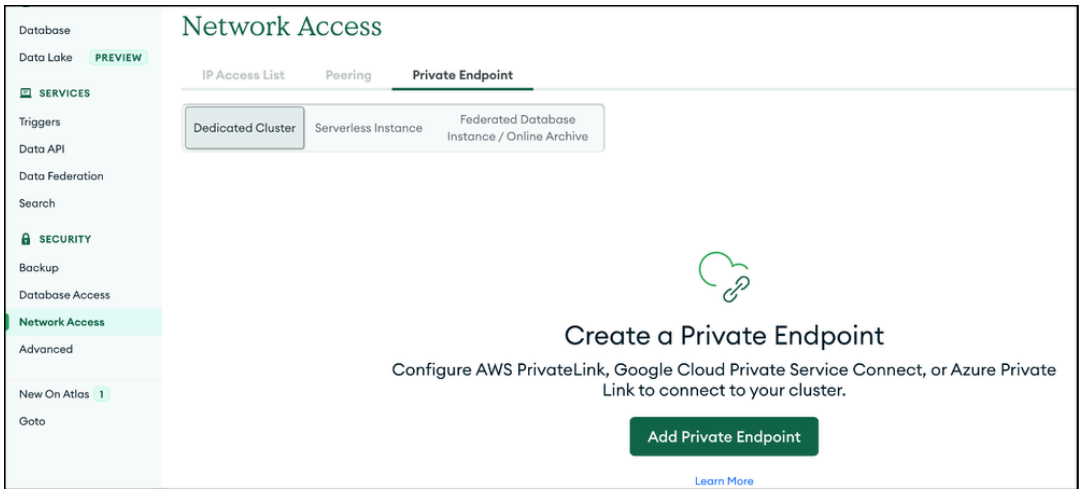
  - **Dedicated tier(전용 계층), Cloud Provider & Region(클라우드 제공자 및 지역)을 선택합니다.**



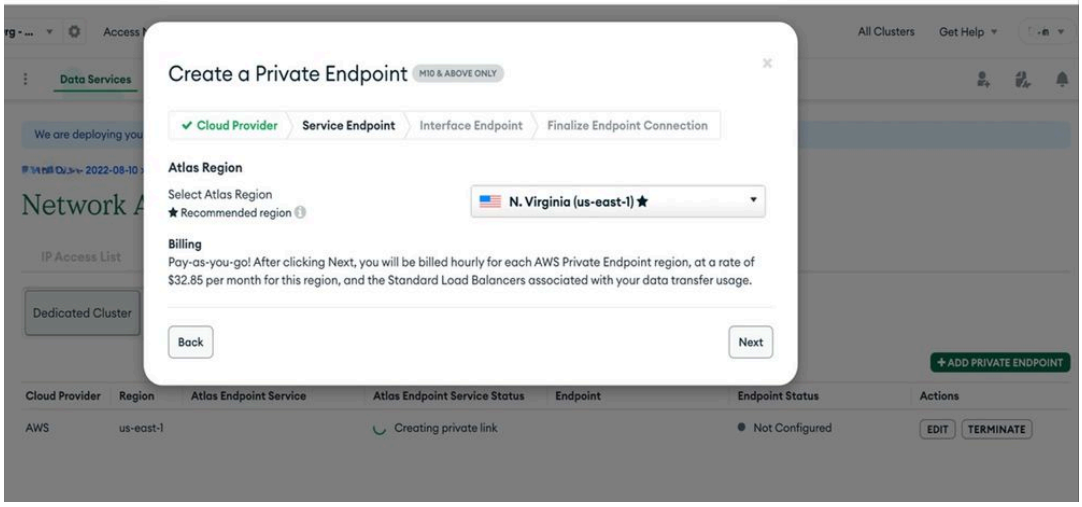
- 적절한 계층(M30을 계층으로 사용함)을 선택하고 적절한 클러스터 이름을 제공한 다음 Create Cluster(클러스터 생성)를 클릭합니다. Atlas monogodb 클러스터를 초기화합니다.



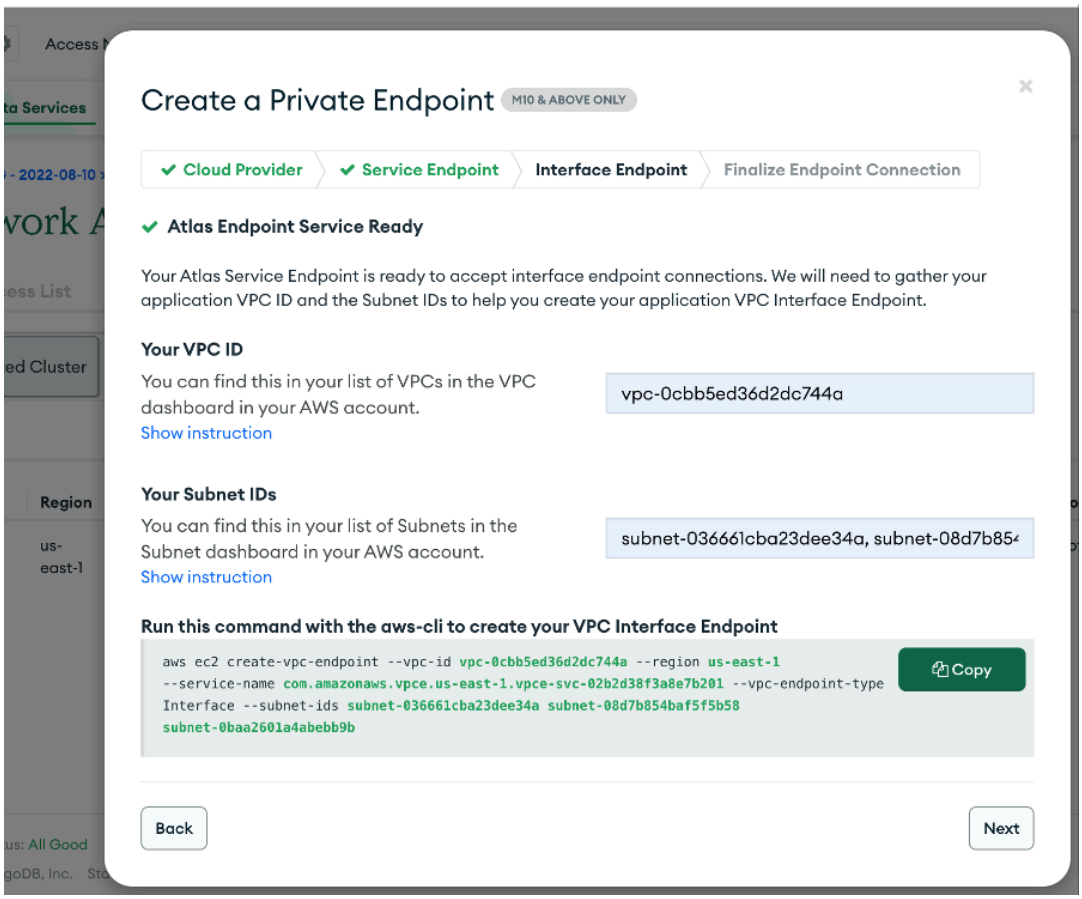
- Atlas 및 K8S 클러스터에 대한 VPC 프라이빗 엔드포인트 설정
  - Network Access(네트워크 액세스) Select Private Endpoint(개인 엔드포인트 선택) > Add Private Endpoint(개인 엔드포인트 추가)를 클릭합니다.



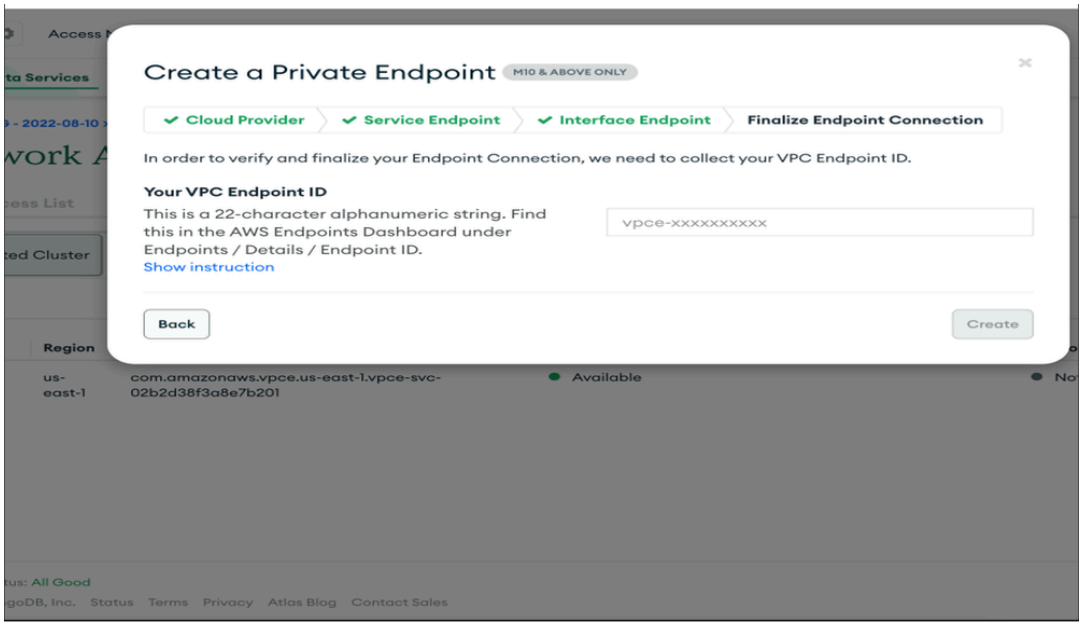
- **Cloud Provider as AWS를 선택하고 각 리전을 선택한 후 Next(다음)를 클릭합니다.**



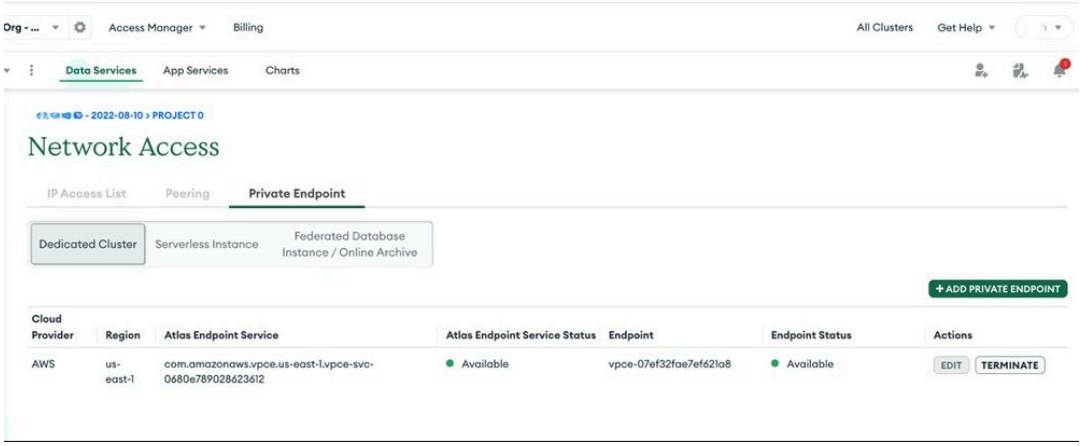
- **각 PVC ID 및 서브넷 ID를 제공합니다. 세부사항을 입력하면 vpc 엔드포인트 생성 명령을 복사하여 aws 콘솔에서 실행합니다. VPC 엔드포인트 ID를 출력으로 가져옵니다.**



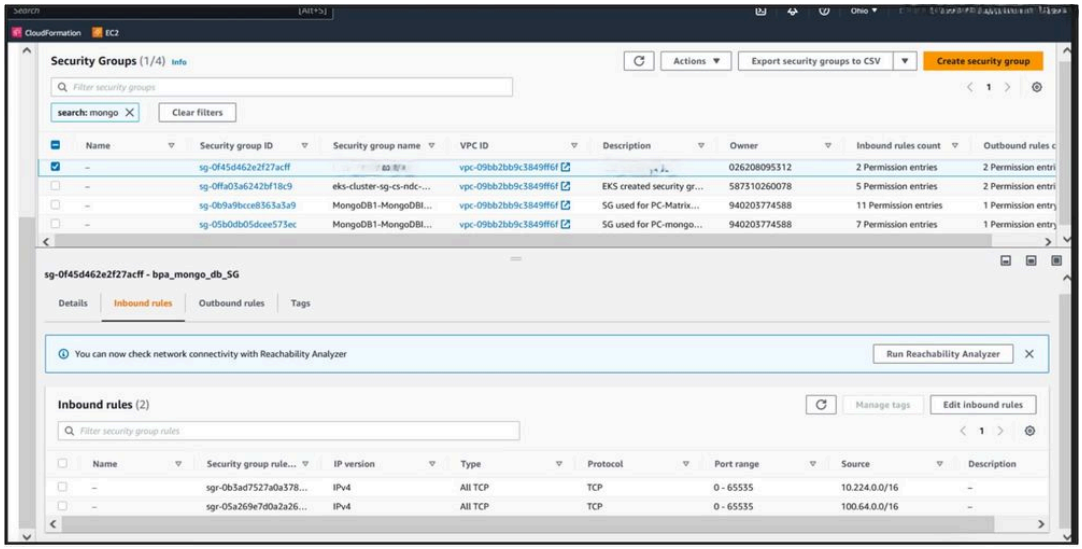
- **Next(다음)를 클릭하여 VPC 엔드포인트 ID를 붙여넣고 Create(생성)를 클릭합니다.**



- **성공적으로 생성되면 다음 그림과 같이 Endpoint(엔드포인트) 상태를 Available(사용 가능)로 설정합니다. 포드 cidr에 대해 VPC 엔드포인트를 생성해야 합니다. 여기서는 "100.64.0.0/16"을 사용했습니다.**



- **새로 생성된 vpc-endpoint에 인바운드 규칙을 추가합니다. vpc-endpoint는 상위 어카운트에 있으며 보안 그룹은 새로 생성된 vpc-endpoint에 할당되어야 합니다.**



이미지 레지스트리로서의 ECR

Amazon ECR 리포지토리를 생성하고 여기에 Docker 이미지를 푸시하려면 몇 가지 단계가 필요합니다. ECR 리포지토리를 생성하고 Docker 이미지에 태그를 지정한 다음 AWS CLI를 사용하여 리포지토리에 푸시하는 단계입니다.

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

바꾸기:

- **ecr 저장소에** 대해 원하는 이름을 가진 사용자 **이미지** 이름
- AWS 리전과의 **your**-regional

EKS 노드에 대한 IAM 역할 구성

EKS 작업자 노드(EC2 인스턴스)에 ECR에서 이미지를 가져올 수 있는 권한이 연결된 필요한 IAM 역할이 있는지 확인합니다. 필요한 IAM 정책은 다음과 같습니다.

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

EKS 작업자 노드와 연결된 IAM 역할에 이 정책을 연결합니다.

BPA 구축
BPA 구축에는 EKS 작업자 노드 레이블 지정, 노드에서 디렉토리 준비, BPA 패키지 복사, Helm을 사용하여 BPA 구축 등 여러 단계가 포함됩니다.

Cisco는 고객 배포를 위해 다음 버전의 소프트웨어 및 클라우드 서비스를 활용했습니다.

- **BPA:** 4.0.3-6
- **RDS(관계형 데이터베이스 서비스)**: 16.3-R2
- **MongoDB 지도**: v5.0.29
- **EKS(Elastic Kubernetes Service)**: v1.27

이러한 구성 요소는 Cisco의 구축이 견고하고 확장 가능하며 필요한 워크로드를 효율적으로 처리할 수 있도록 보장합니다.

- **EKS 작업자 노드 레이블 지정**

```
kubectl label node
```

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

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

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

```
 name=node-4
```

- **노드에서 디렉터리 준비**
  노드 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
```

노드 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
```

노드 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
```

노드 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
```

- BPA 패키지 복사

```
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/
```

- Helm을 사용하여 BPA 구축

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

인그레스 설정

- **인그레스 활성화**
  값 업데이트.Ymlto 인그레스 사용:

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

- **BPA 인증서를 사용하여 암호 생성**
  인증서 디렉토리로 이동하여 암호를 생성합니다.

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

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

- **인그레스 컨트롤러 업데이트**
  새로 만든 암호를 **`ingress-controller.yaml`** 파일:

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

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

- **인그레스 인증서 업데이트**
  키 삭제 및 설치를 수행하여 인그레스 인증서를 업데이트합니다.

환경 사양

환경 사양에는 EC2 인스턴스, 로드 밸런서, VPC 엔드포인트 및 RDS 인스턴스에 대한 요구 사항이 포함됩니다. 주요 사양은 다음과 같습니다.

EC2 요구 사항:

스토리지 요구 사항:노드당 2TB 공간 EBS 볼륨을 /opt에 마운트하고 모든 노드에 대해 /etc/fstab에 항목을 추가합니다.

보안 그룹 인바운드: 30101, 443, 0 - 65535 TCP, 22(ssh).

보안 그룹 아웃바운드: 모든 트래픽을 활성화해야 합니다.

DNS 확인자: EC2에는 /etc/resolve.conf에 온-프레미스 확인자가 있어야 합니다.

로드 밸런서 요구 사항:

- 리스너 포트는 443, 30101이어야 합니다.
- VPC 엔드포인트 요구 사항(Atlas MongoDB).
- 아틀라스 연결을 위해 생성된 VPC 엔드포인트는 상위 계정(aws-5g-ndc-prod)에서 사용할 수 있습니다. VPC 엔드포인트에는 모든 인바운드 액세스를 허용하는 보안 그룹이 있어야 합니다

(0 - 65535).

RDS 요구 사항:

RDS 유형: db.r5b.2xlarge

Postgres 엔진 버전: 13.7

보안 그룹: Inboud는 POD CIDR 소스의 트래픽을 허용해야 합니다.

주요 개념 및 구성 요소
Amazon EKS를 사용하여 애플리케이션을 효과적으로 구축하고 관리하려면 Kubernetes의 기본 사항을 이해하는 것이 중요합니다.

결론
이 백서에서는 Amazon EKS를 사용하여 BPA(Business Process Automation) 애플리케이션을 배포하고 관리하는 데 필요한 자세한 가이드를 제공합니다. 요약된 단계에 따라 주요 개념을 이해하면 컨테이너화된 BPA 애플리케이션에서 EKS의 이점을 활용할 수 있습니다.

참조

- **Amazon Web Services, "Amazon EKS 설명서" [온라인]. 사용 가능 :https://docs.aws.amazon.com/eks/**
- **Kubernetes, "Kubernetes 문서" [온라인]. 사용 가능:https://kubernetes.io/docs/home/**
- **Cisco BPA 살펴보기 https://www.cisco.com/c/en/us/solutions/collateral/service-provider/at-a-glance-c45-742579.html**
- **BPA 운영 가이드 https://www.cisco.com/c/dam/en/us/support/docs/bpa/v403/cisco-bpa-operations-guide-v403.pdf**
- **BPA 개발자 가이드 https://www.cisco.com/c/dam/en/us/support/docs/bpa/v403/cisco-bpa-developer-guide-v403.pdf**

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.