# Amazon EKSでのビジネスプロセス自動化アプリケーションのデプロイと管理：実用的なガイド

## 内容


### 概要

このホワイトペーパーでは、Amazon Elastic Kubernetesサービス(EKS)を使用したビジネスプロセス自動化(BPA)アプリケーションのデプロイと管理に関する包括的なガイドを紹介します。前提条件の概要を説明し、EKSを利用する利点を示し、EKSクラスター、Amazon RDSデータベース、およびMongoDB Atlasをセットアップする手順を説明します。さらに、導入アーキテクチャを詳しく調べ、環境要件を特定し、コンテナ化されたBPAアプリケーションにEKSを活用することを目指す組織のための徹底したリソースを提供します。



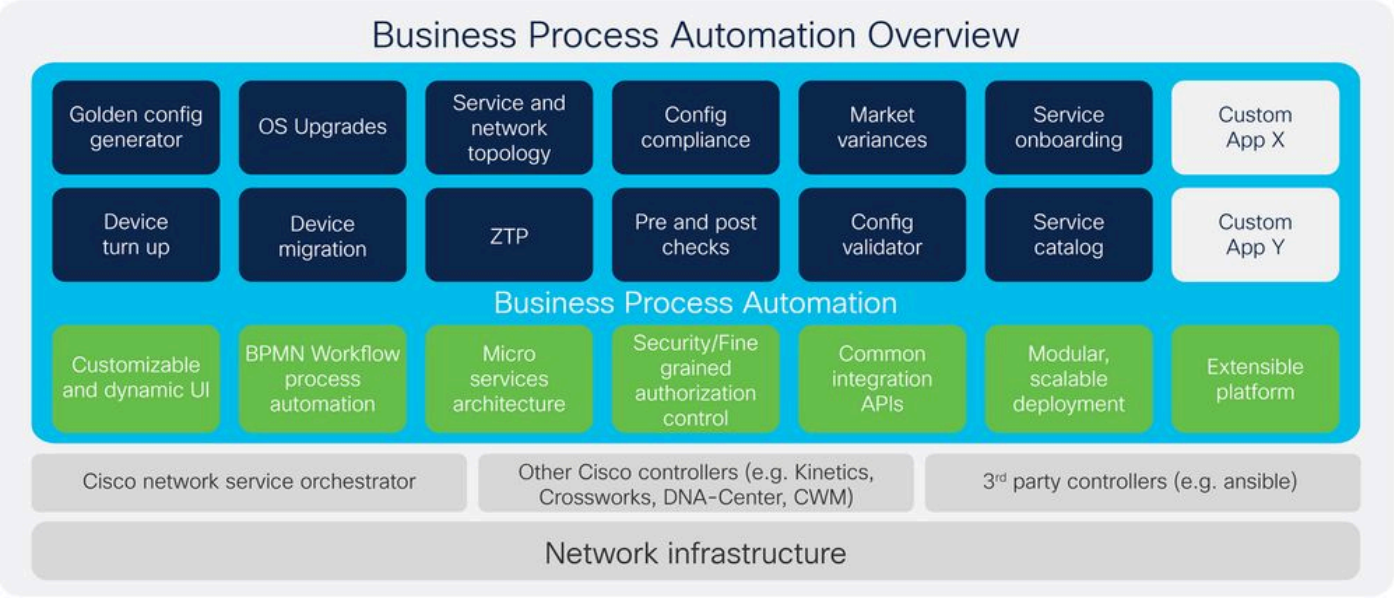
### キーワード

Amazon EKS、Kubernetes、AWS、RDS、MongoDB Atlas、DevOps、クラウドコンピューティング、ビジネスプロセス自動化。


---

## はじめに

## BPA




今日のデジタル時代において、企業は多様なIT環境にわたって複雑なビジネスプロセスを合理化および自動化することを求めています。業務プロセスの自動化(BPA)は、業務効率の向上、エラーの削減、サービス提供の改善を可能にする重要なテクノロジーとして登場しました。BPAは、ワークフロー自動化、サービスプロビジョニング、および市販の自動化アプリケーションの向上を目的とした、いくつかの重要な技術革新と機能拡張を導入しています。

BPAプラットフォームは、OSのアップグレード、サービスプロビジョニング、オーケストレーションエンジンへの統合など、ビジネスおよびIT/運用のユースケースとアプリケーションをホストします。お客様は、サービスのライフサイクルとBPA機能（アドバイザリ、実装、ビジネスクリティカルなサービス、ソリューションサポートなど）にアクセスできます。これらのサービスや機能は、シスコのエキスパート、ベストプラクティス、実証済みの手法によって提供され、ビジネスプロセスの自動化やシステムのリスク軽減に役立ちます。

これらのライフサイクル機能は、サブスクリプションベースにすることも、個々のニーズに合わせてカスタマイズすることもできます。実装サービスは、自動化を促進するツールとプロセスの定義、統合、導入を支援します。シスコのエキスパートは、要件を収集するための正式なプロセスを実施し、俊敏なプロセスとContinuous Integration and Continuous Delivery(CICD)ツールに基づいてユーザストーリーを設計および開発し、新規または既存のワークフロー、デバイス、およびサービスの自動テストによって柔軟なサービスを実装します。ソリューションサポートを利用すると、お客様はソフトウェア中心の問題に重点を置いた24時間365日の一元的なサポートを受けることができます。また、シスコの階層型ソフトウェアモデルを通じて提供されるマルチベンダーおよびオープンソースのサポートも利用できます。シスコのソリューションサポートのエキスパートは、最初の問い合わせから最終的な解決までサポートを提供し、複数のベンダーと同時に作業する主要な窓口として機能します。ソリューションレベルのエキスパートとの連携により、最大44 %の問題削減が可能となり、ビジネスの継続性を維持し、BPAへの投資を短期間で回収できます。

BPAは、FMCおよびAnsible管理対象デバイスのサポート、Advanced Queuing Framework(AQF)を使用した並行実行、NDFCおよびFMCデバイスの拡張された設定コンプライアンスなどの主要な技術的機能により、大規模なエンタープライズ自動化の包括的なソリューションとして位置付けら

れます。このリリースでは、SD-WAN管理、デバイスオンボーディング、およびファイアウォールポリシーガバナンスの機能が追加され、ネットワークセキュリティと自動化の重要な側面に対応し、大規模なマルチベンダー環境の要求に対応します。

**EKS（必須）**

Amazon Elastic Kubernetes Service (EKS)は、アマゾンウェブサービス(AWS)が提供する完全マネージド型のKubernetesサービスです。2018年に開始されたEKSは、オープンソースコンテナオーケストレーションプラットフォームであるKubernetesを使用して、コンテナ化アプリケーションの導入、管理、拡張のプロセスを簡素化します。EKSはKubernetesの複雑なクラスタ管理を抽象化し、開発者は基盤となるインフラストラクチャを処理する必要なく、アプリケーションの構築と実行に集中できます。

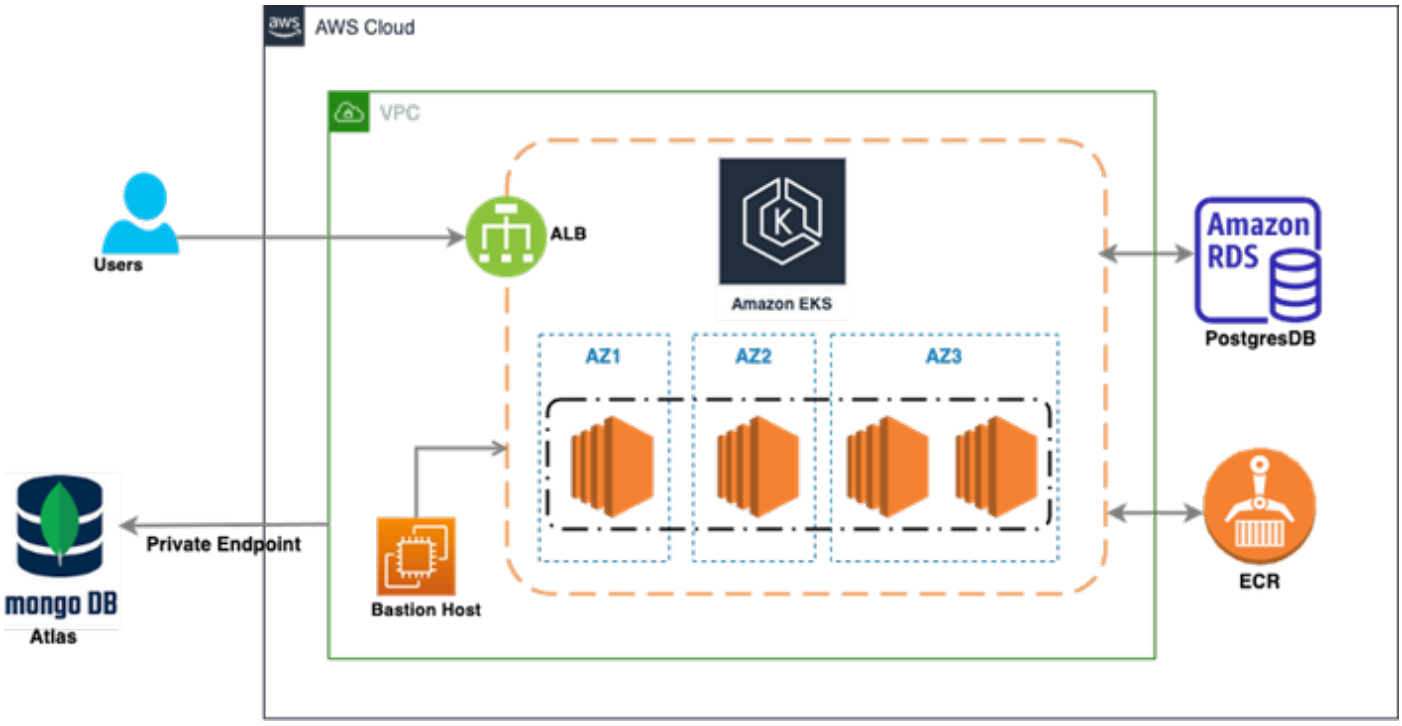
**アプリケーションのデプロイにAmazon EKSを使用する利点**

Amazon EKSはアプリケーションのデプロイに複数のメリットを提供するため、コンテナ化されたアプリケーションやマイクロサービスを利用する組織にとって一般的な選択肢となっています。

**主な利点は次のとおりです。**

- マネージド**Kubernetes**コントロールプレーン:EKSは、Kubernetesコントロールプレーンの導入、スケーリング、およびメンテナンスを処理し、運用負担を軽減します。
- **簡素化されたクラスタ管理**:EKSは、Kubernetesクラスタの設定と管理の複雑さを抽象化します。
- 拡張性:EKSを使用すると、クラスタを簡単に拡張して、増大するワークロードに対応できます。
- ハイアベイラビリティ:EKSはマルチアベイラビリティゾーンの導入をサポートし、アベイラビリティと耐障害性を強化します。
- **AWSサービスとの統合**: EKSは、さまざまなAWSサービスとシームレスに統合されます。
- **DevOpsの自動化**:EKSは、コンテナ化されたアプリケーションの継続的な統合と継続的な導入(CI/CD)をサポートします。

**BPA導入アーキテクチャ**




この図は、いくつかの主要コンポーネントを使用して**AWS** にデプロイされたクラウドベースのインフラストラクチャの高レベルアーキテクチャを表しています。図の内訳を次に示します。

1. **Amazon EKS(Elastic Kubernetes Service)**：図のコアでは、Amazon EKSは3つのアベイラビリティーゾーン(AZ1、AZ2、AZ3)にデプロイされ、各ゾーン内にKubernetesワーカーノードがあります。これは、ワークロードが複数の可用性ゾーンに分散しているため、可用性が高く、フォールトトレラントな設定であることを示しています。

2. **ALB(Application Load Balancer)**：先頭に位置し、ユーザからトラフィックを受信し、アプリケーションワークロードを処理するためにEKSクラスタ全体にトラフィックを分散します。ロードバランサによって、要求が均等に分散され、トラフィックの需要に基づいた拡張を処理できるようになります。

3. **Amazon RDS (Relational Database Service)** - PostgreSQL：図の右側には、PostgreSQLを実行するAmazon RDSインスタンスが存在します。このデータベースには、EKSクラスタ内で実行されているアプリケーションからアクセスできます。

4. **ECR(Elastic Container Registry)**：これはDockerコンテナイメージが格納および管理される場所で、ワークロードを実行するためにAmazon EKSにデプロイされます。

5. **MongoDB Atlas**：左側のMongoDB Atlasは、プライベートエンドポイントを通じてアーキテクチャに統合されています。MongoDB Atlasは、クラウドホスト型のNoSQLデータベースサービスで、ここではドキュメントベースのデータベース要件を処理するために使用されます。プライベートエンドポイントは、MongoDB Atlasインスタンスと他のAWSコンポーネント間の安全なプライベート通信を保証します。

6. **Bastion Host**:VPC(Virtual Private Cloud)内に配置されたBastion Hostは、管理者がインターネットに直接公開されることなくVPC内のリソースにアクセスするための安全なエントリポイントを提供します。

全体として、このアーキテクチャは、リレーショナル(PostgreSQL)データベースとNoSQL(MongoDB)データベースの両方をサポートし、Amazon EKSを使用してコンテナー化されたアプリケーションをデプロイおよび管理するための、可用性、拡張性、および安全性に優れたソリューションを提供します。

- **EKSクラスタセットアップ**
  AWS CLIを使用してAmazon EKSクラスターを作成するには、`eksctl`コマンドラインユーティリティを使用できます。次にコマンドの例を示します。

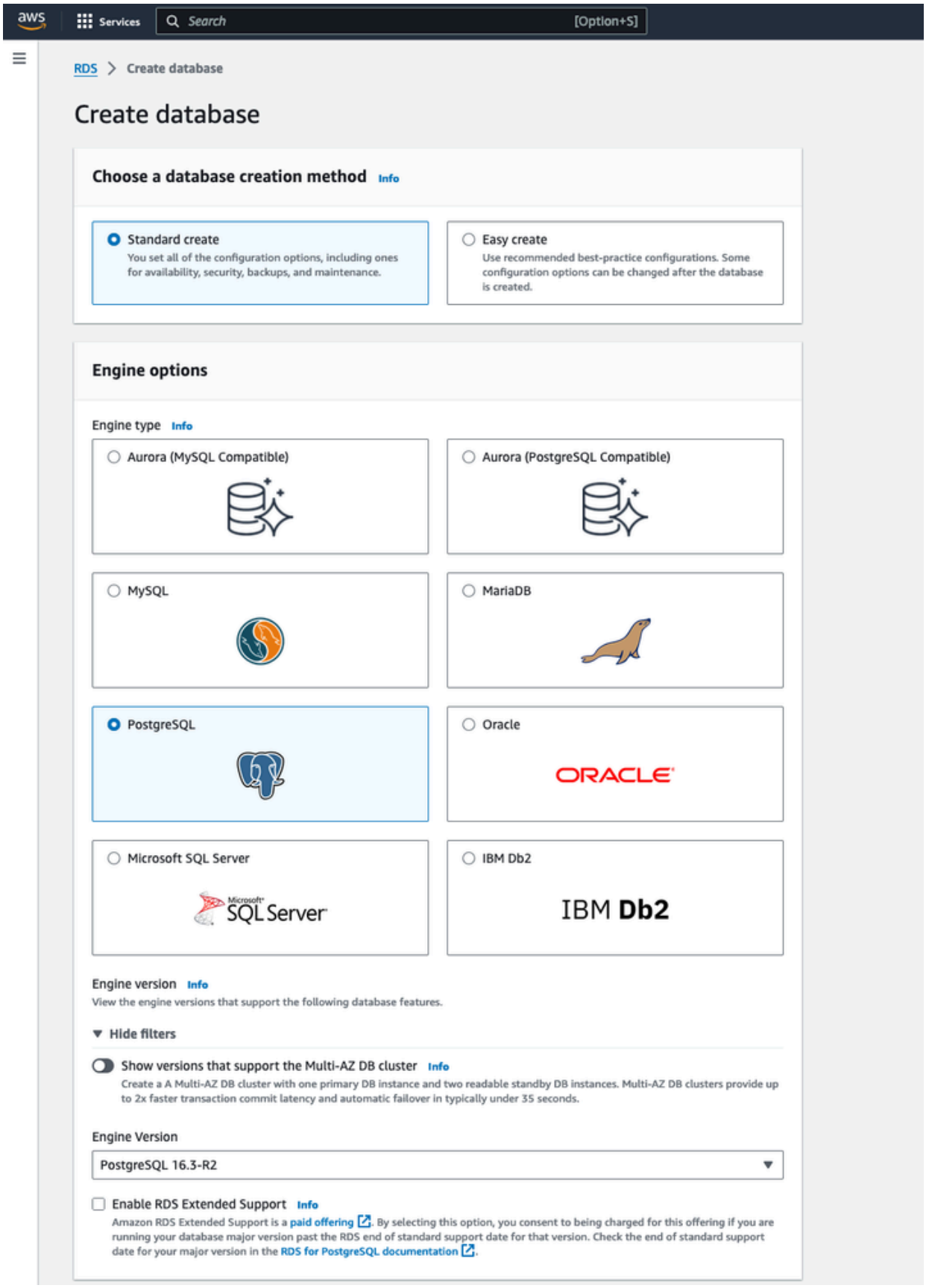
```
eksctl create cluster \
  --name
```

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

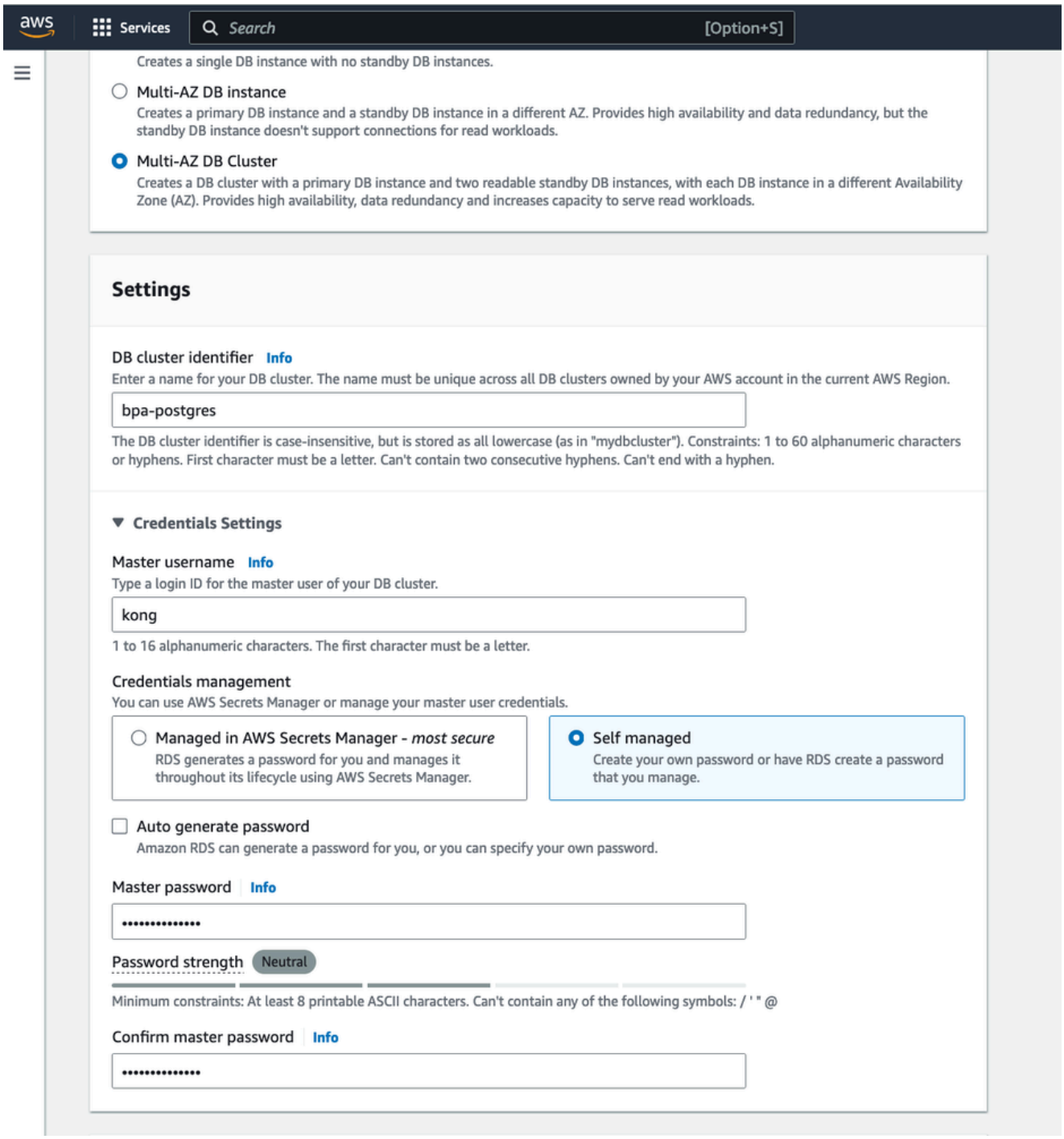
- **RDSデータベースセットアップ**

  Amazon RDSでリレーショナルデータベースをデプロイするには、次の手順を実行します。

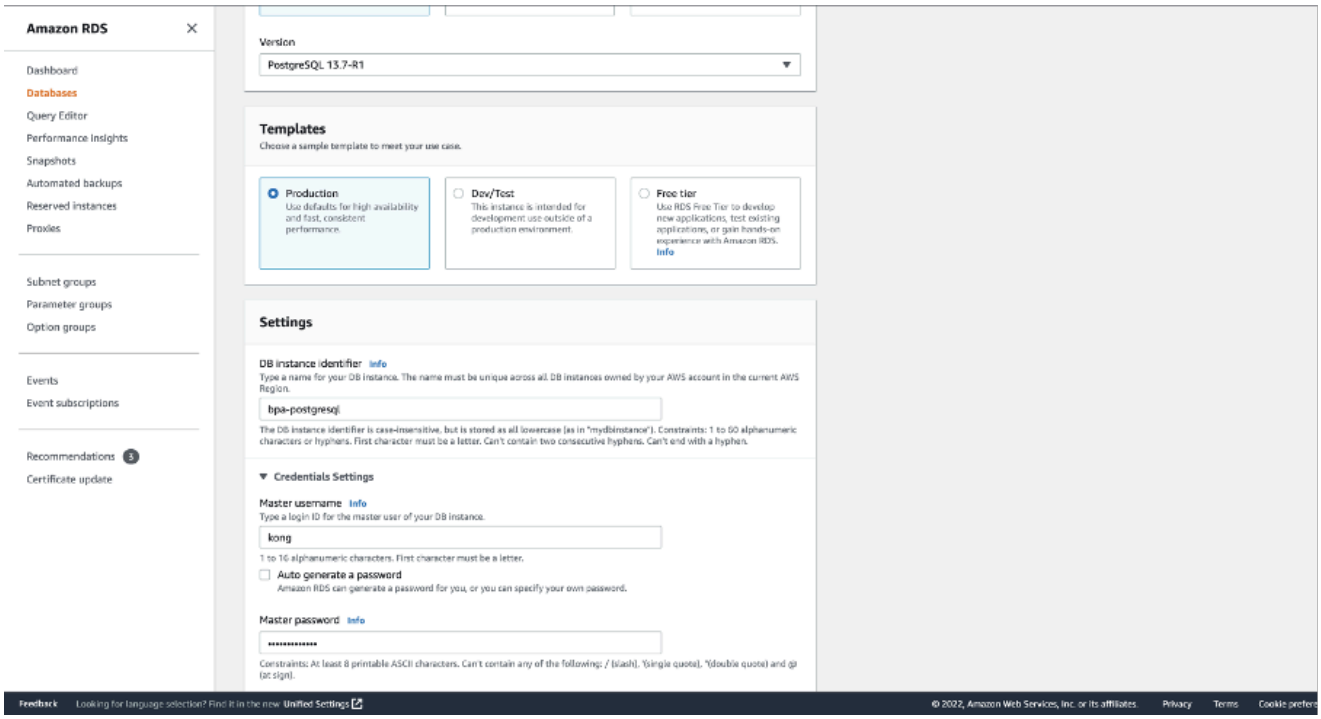
  - AWSマネジメントコンソールにアクセスし、Amazon RDSサービスに移動します。
  - 必要な仕様を使用して新しいデータベース・インスタンスを作成します。
  - Amazon EKSクラスターからの着信接続を許可するようにセキュリティグループを構成します。




ドロップダウンメニューを使用して、PostgreSQLの最新バージョンを選択します。この例では、「PostgreSQL 16.3-R1」です。




このため、データベース・インスタンスに名前を付け、ユーザー名とパスワードを作成します。

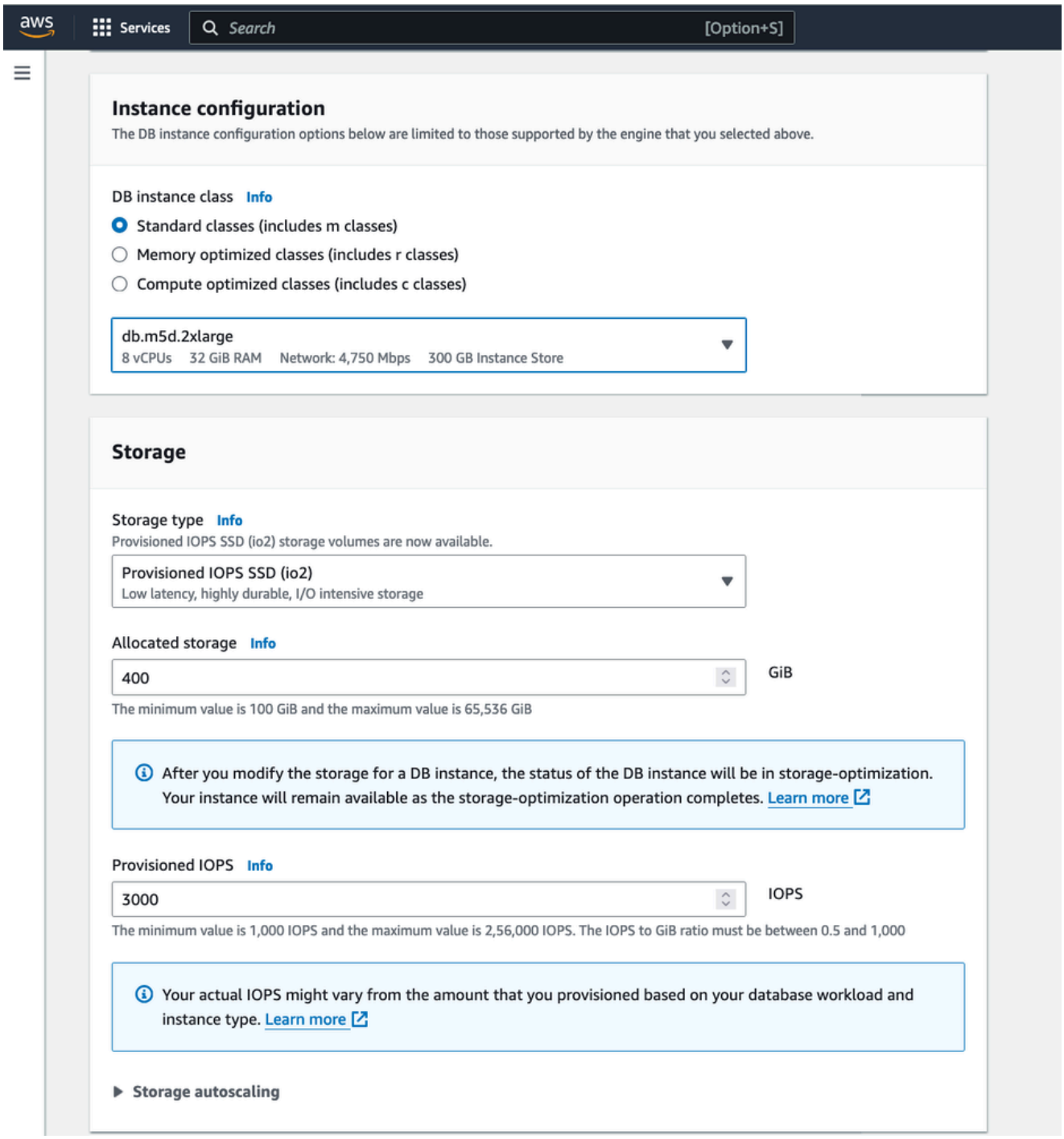



「DBインスタンスサイズ」と「ストレージ」のデフォルト設定が選択されていることを確認します。

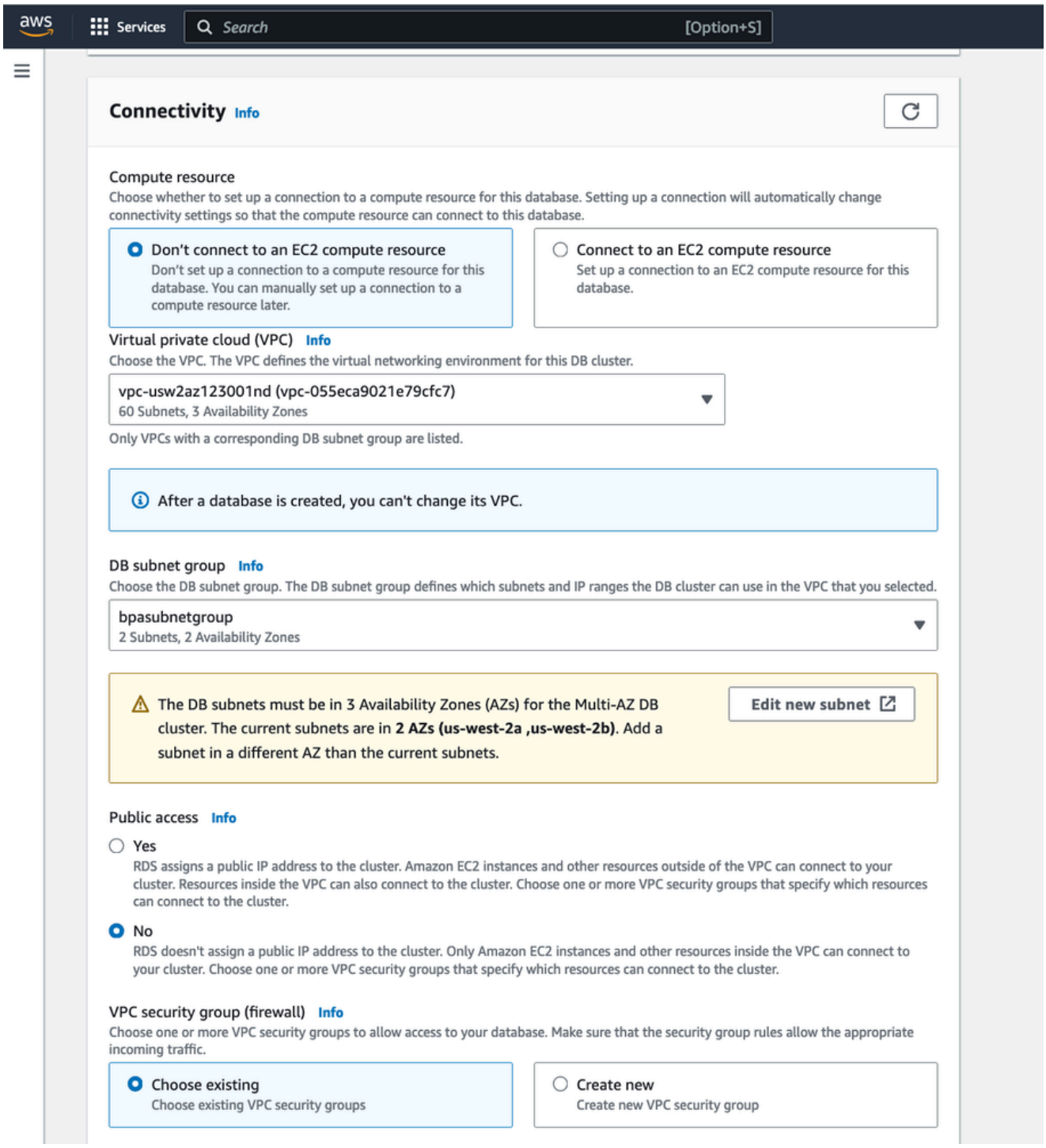
クラスタサイズとデータ要件に応じて、適切なDBインスタンスのサイズとストレージタイプを選択します。

この使用例に基づいて、次の設定を選択しました。

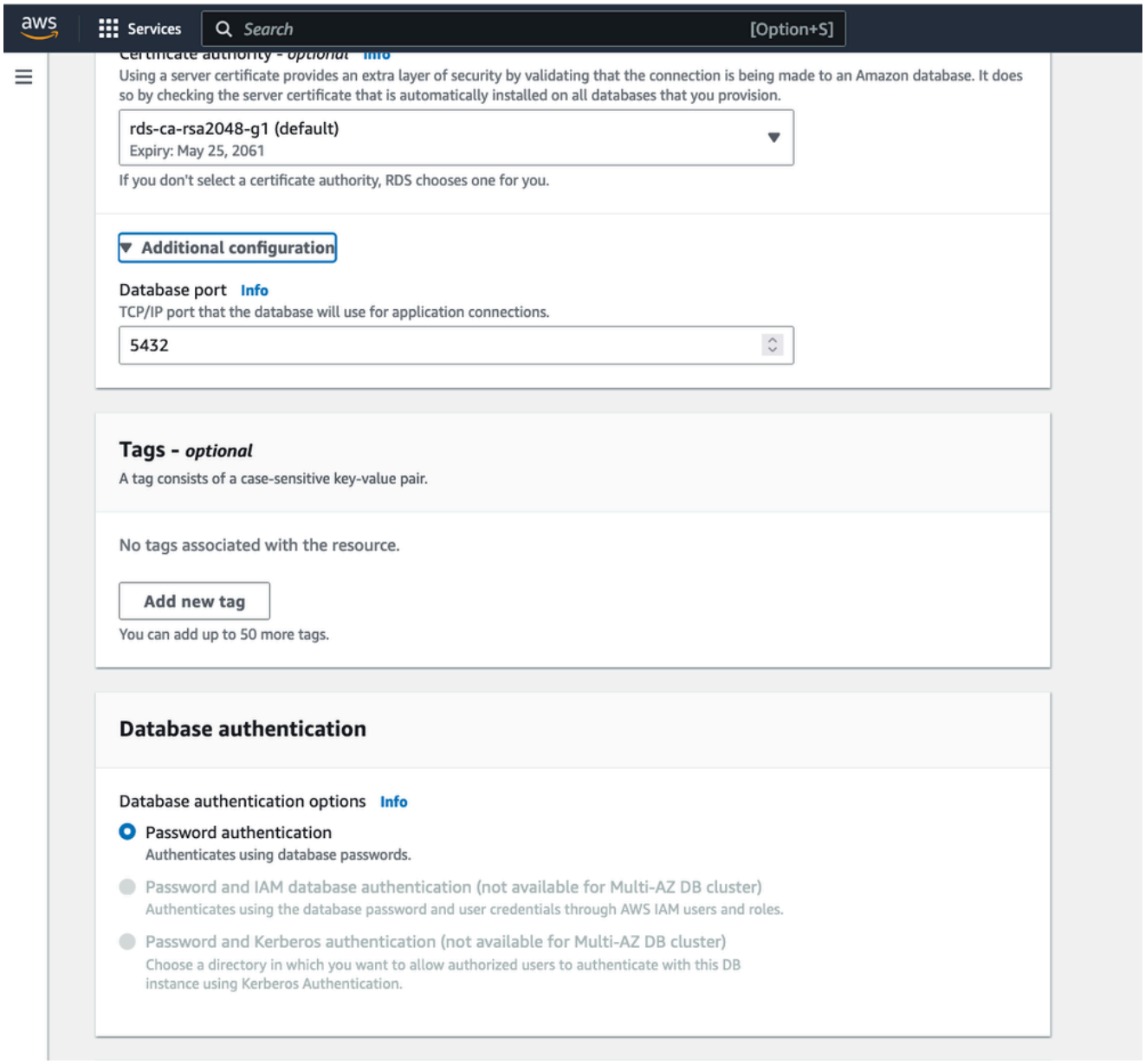
- **DBインスタンスサイズ**: db.m5d.2xlarge
    - 8 つの vCPU
    - 32 GiBメモリ
    - ネットワーク：4,750 Mbps
    - 300 GBインスタンスストア

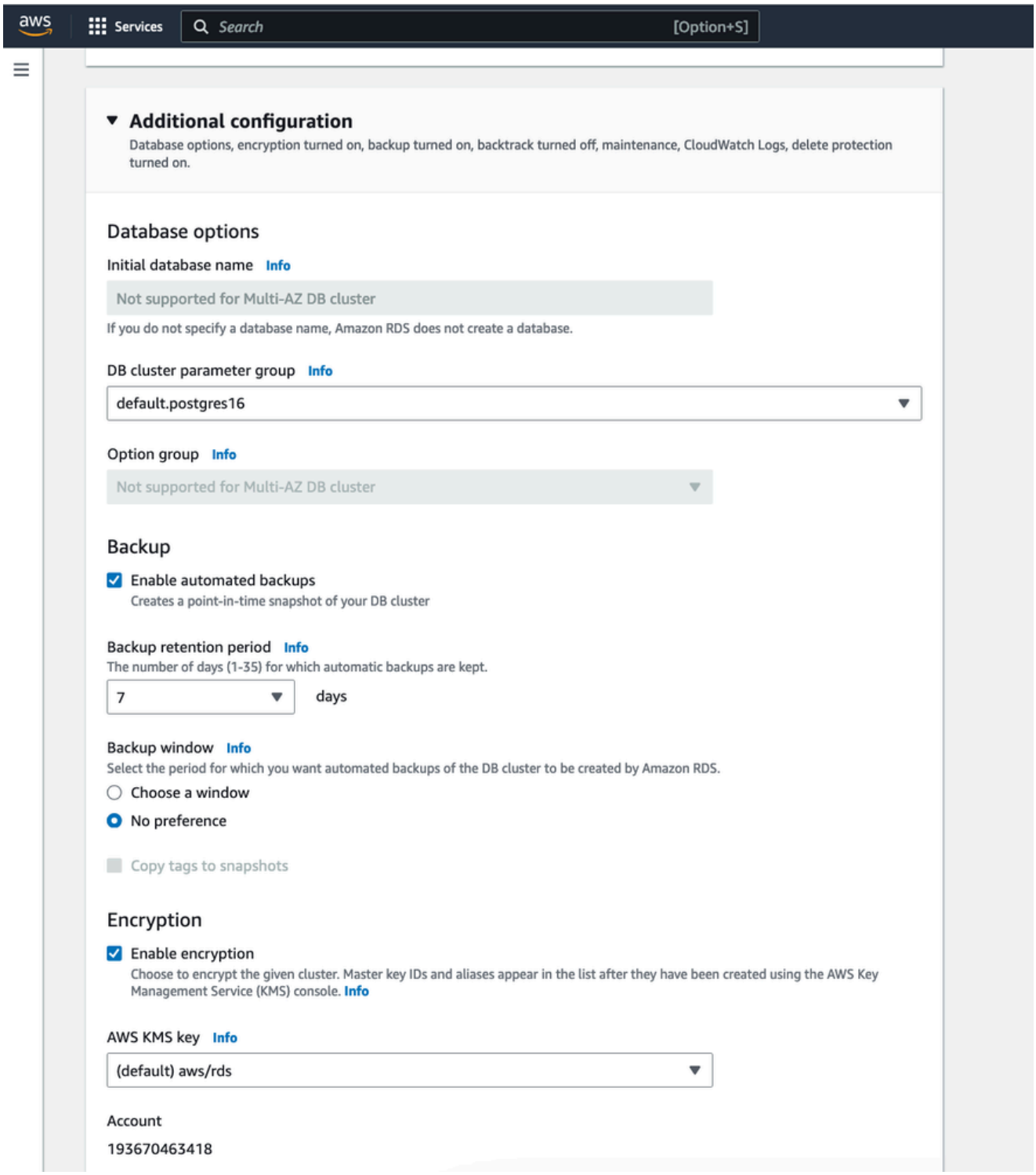


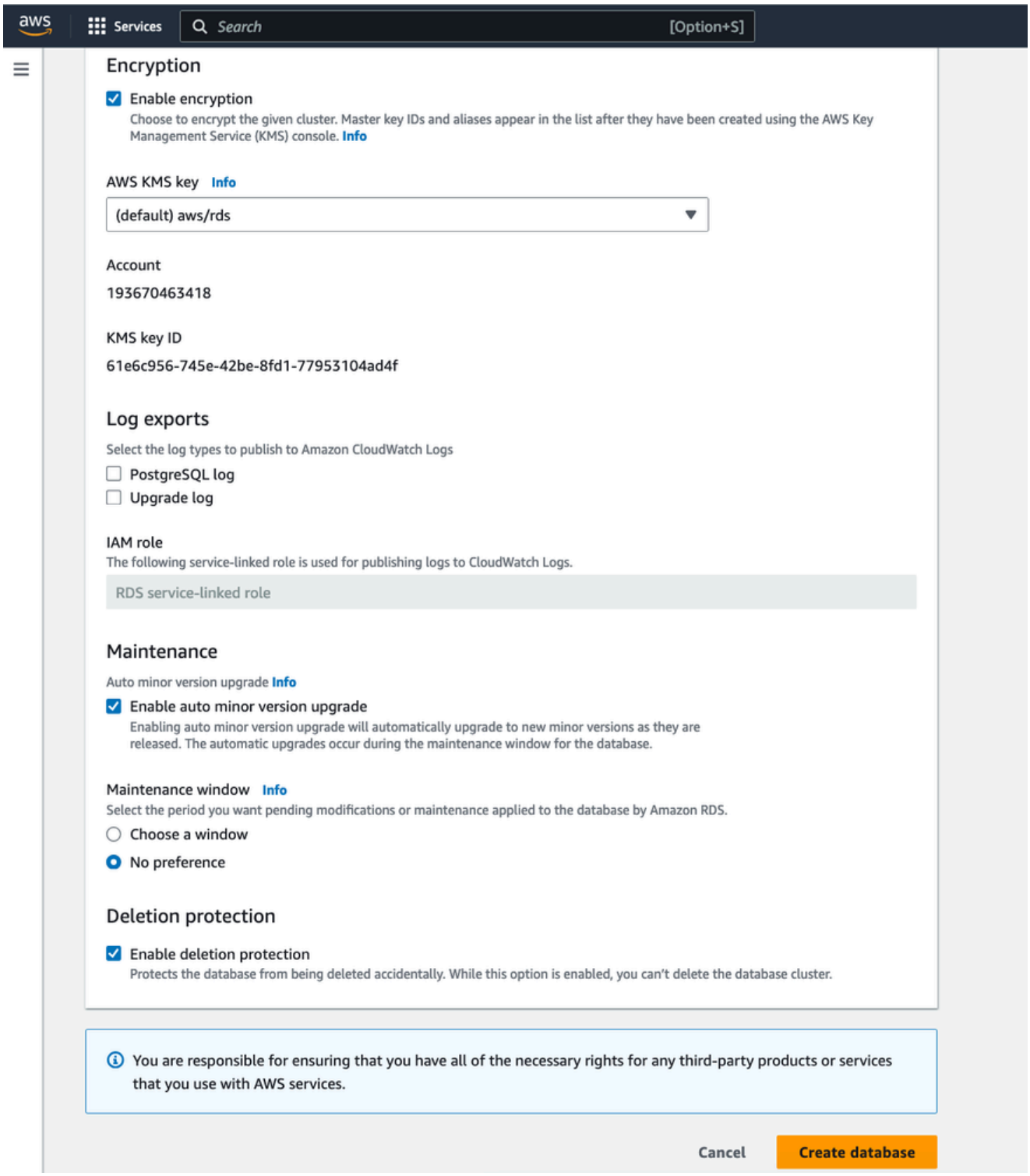

ユースケースに応じて適切な値を選択します。デフォルト値を選択しました。




「データベース認証」で、パスワード認証を選択したことを確認します。データベースパスワードを使用して認証します。


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.

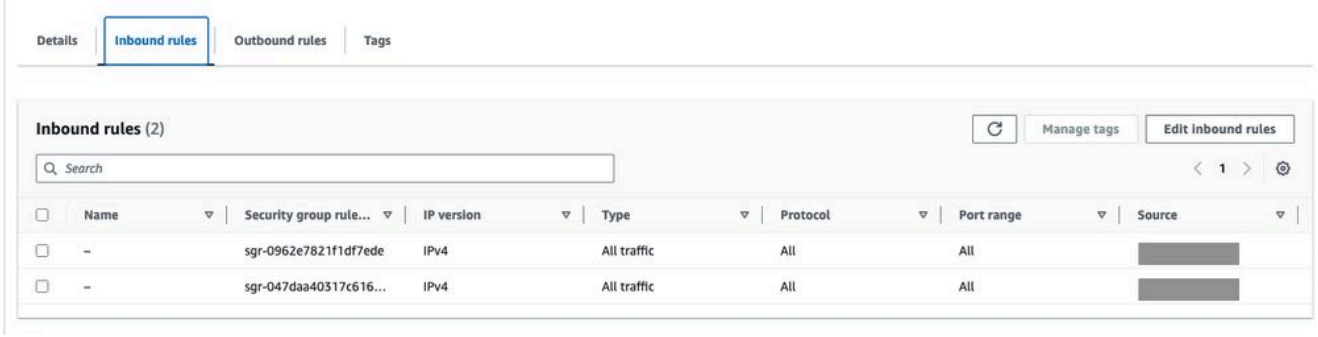

aws
Services
Search
[Option+S]
Additional configuration
Database options, encryption turned on, backup turned on, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned on.
Database options
Initial database name Info
Not supported for Multi-AZ DB cluster
If you do not specify a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



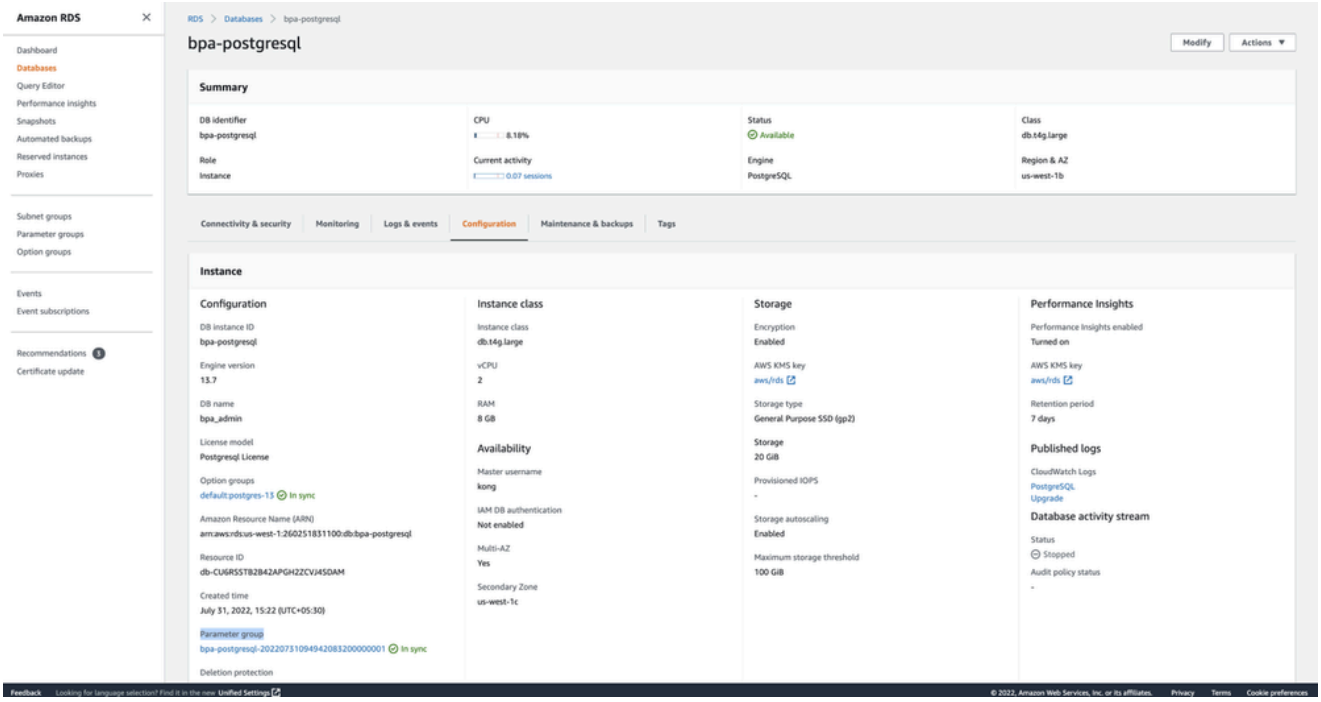

これが確認されたら、データベースを作成する準備が整います。 Amazon RDSダッシュボードに戻ります。インスタンスが使用可能であることを確認します。

## セキュリティグループルール

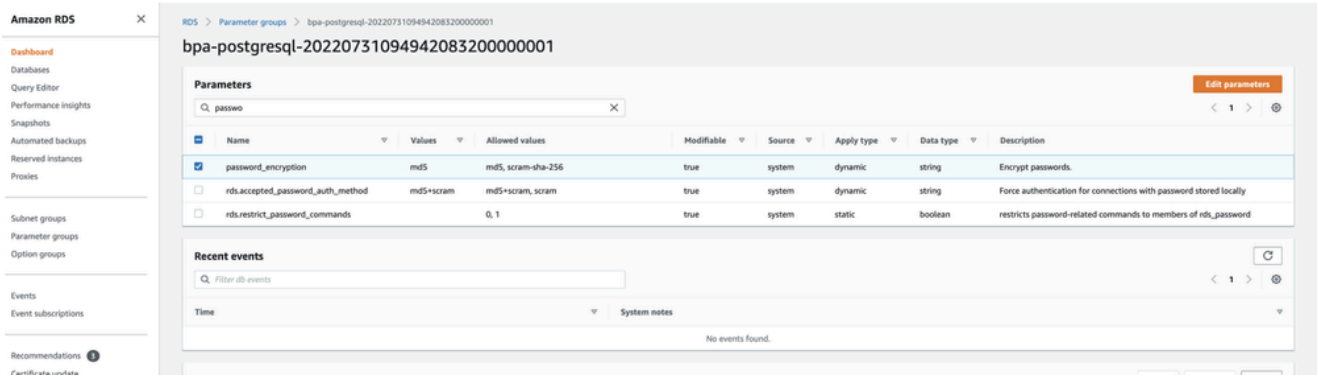
ポッドCIDRとノードCIDRブロックを使用して着信セキュリティグループを更新します。




**RDS -> Databases -> DB-NAMEで、configurationをクリックしてParameter Groupセクションを参照し、表示するパラメータグループをクリックします。**




「password_encryption」を検索し、値をブランクまたは他の値からmd5に変更します。これは、camunda設定を機能させるために必要です。




**RDSに接続して、これらのデータベースをユーザーと共に作成します。**

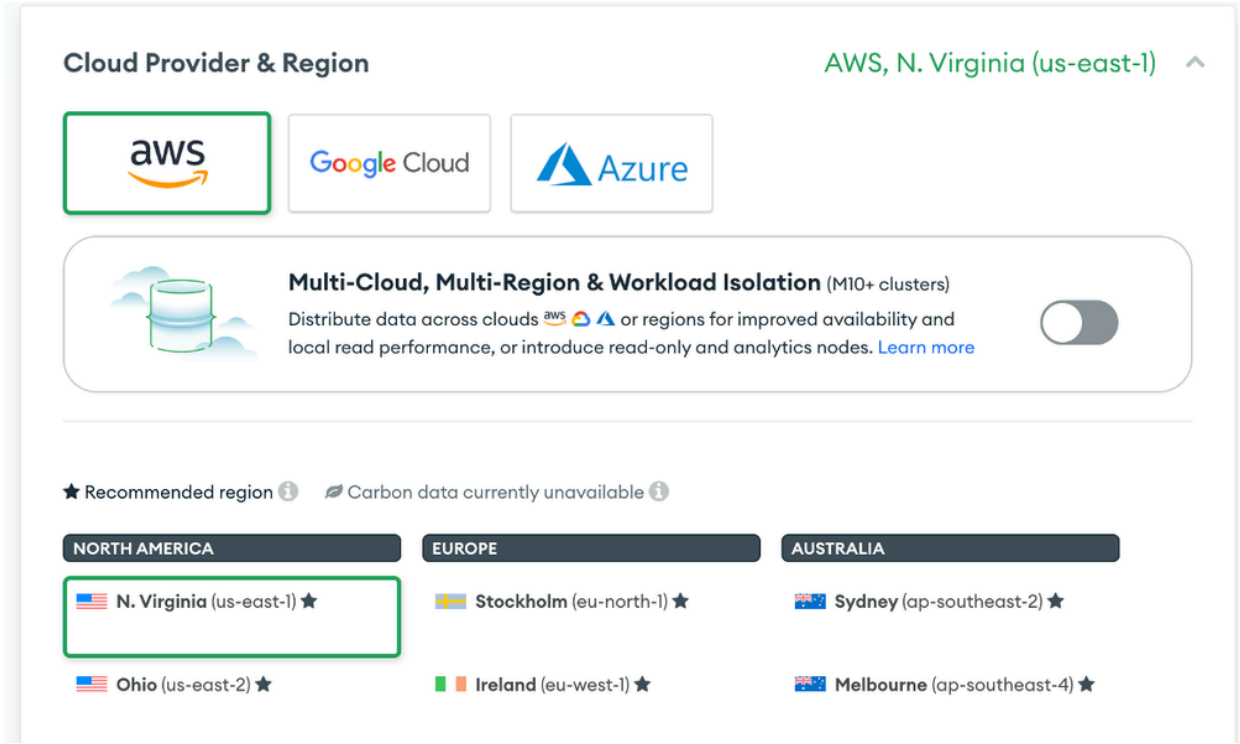
```
PG_ROOT_DATABASE=admin
PG_INITDB_ROOT_USERNAME=admin
PG_INITDB_ROOT_PASSWORD=Bp@Chang3d!
AUTH_DB_NAME=kong
AUTH_DB_USER=kong
AUTH_DB_PASSWORD=K@ngPwdCha*g3
WFE_DB_USER=camunda
WFE_DB_PASSWORD=WOrkFlo#ChangeNow
WFE_DB_NAME=process-engine
```

- パスワード認証

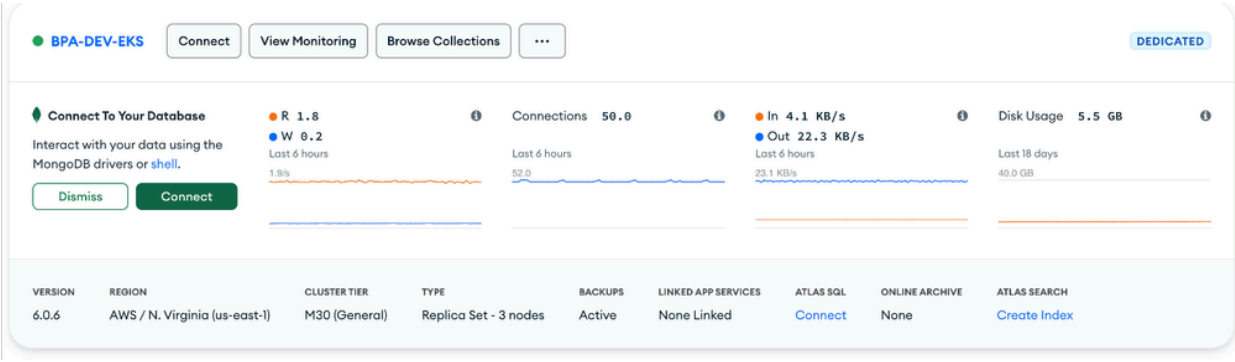
  データベースパスワードを使用して認証します。

- **Atlas MongoDBセットアップ**
  **Atlas MongoDBのセットアップには次の作業が含まれます。**
  - **Atlas MongoDBにログインします。**
  - **組織とプロジェクトを選択します。**
  - **適切な仕様を持つ専用クラスタの作成**

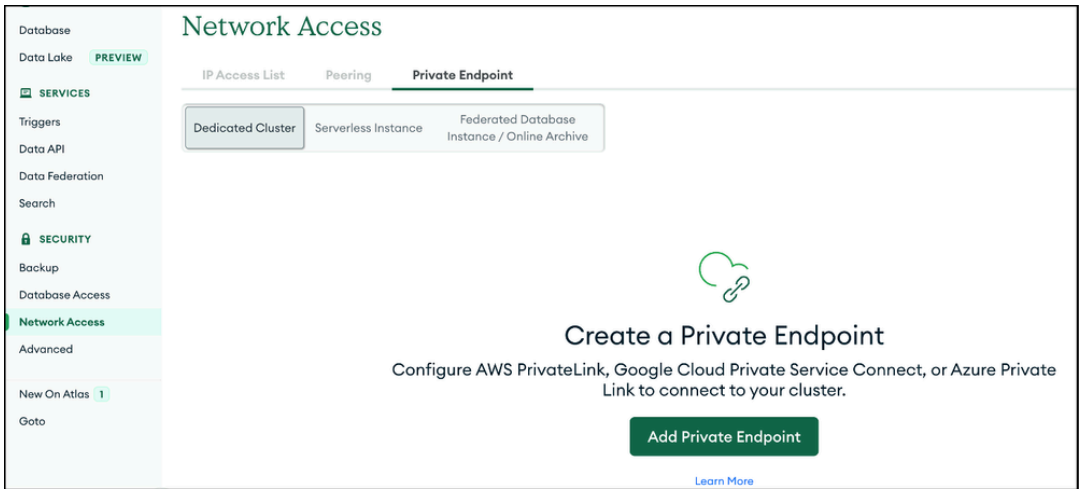

  - **Dedicated Tier、Cloud Provider & Regionを選択します。**



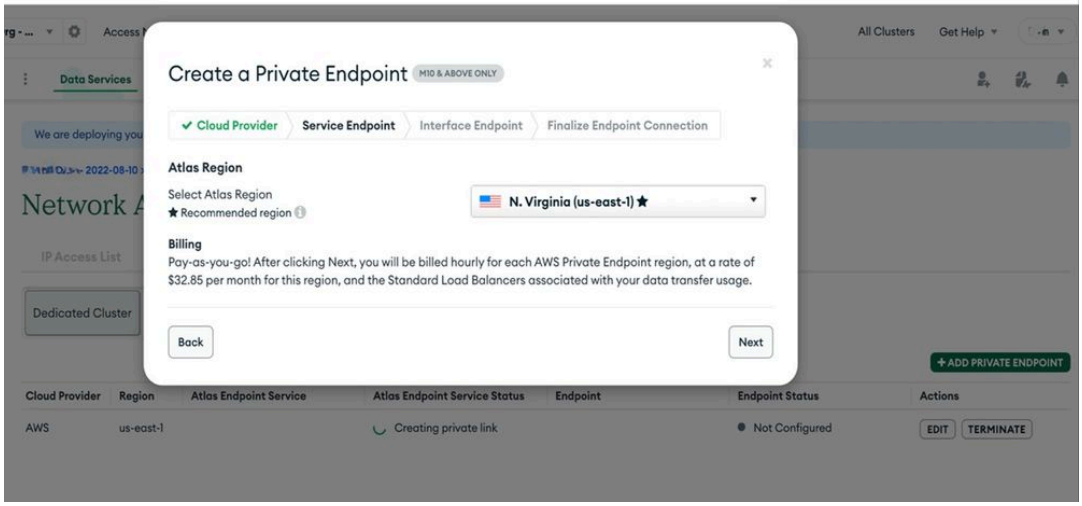
- 適切な階層（階層としてM30を使用）専用クラスタを選択し、適切なクラスタ名を指定して、[クラスタの作成]をクリックします。Atlas monogodbクラスタを初期化します。



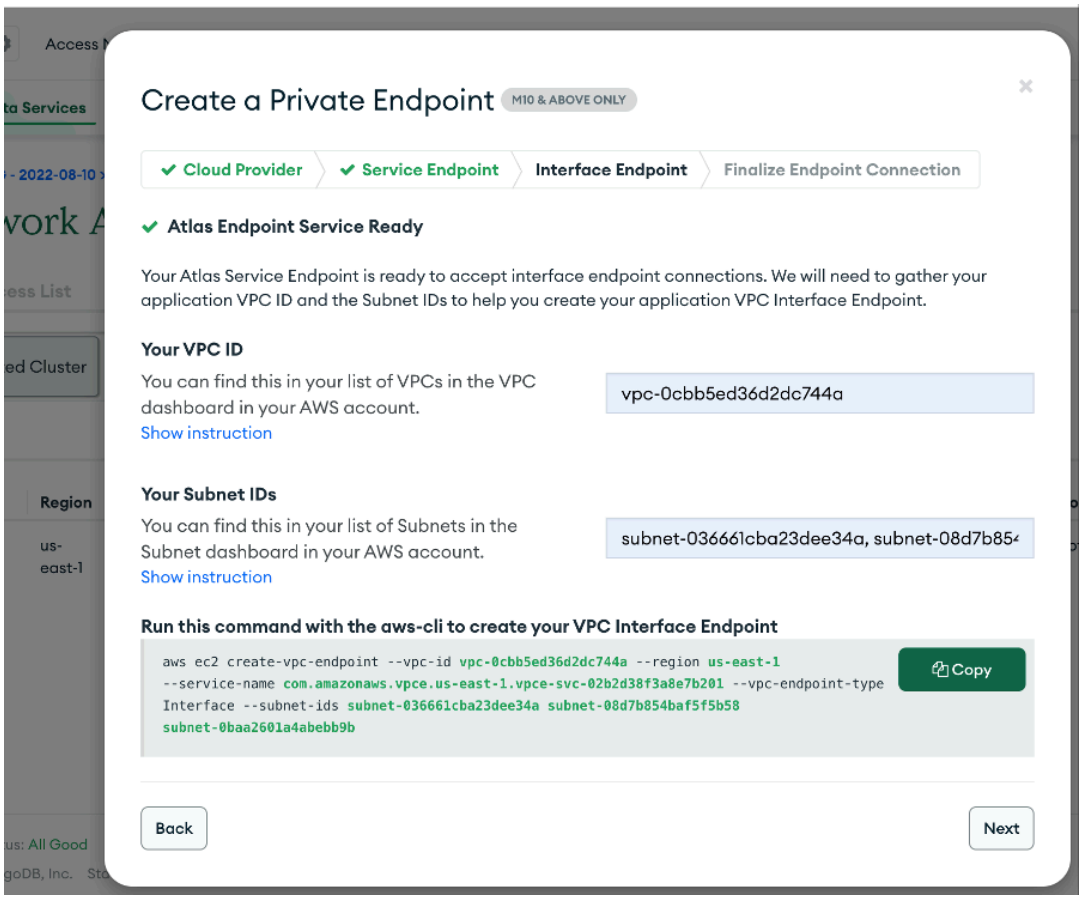

  - AtlasおよびK8SクラスタのVPCプライベートエンドポイントをセットアップします。
    - Network Access Select Private Endpointをクリックし、Add Private Endpointをクリックします。



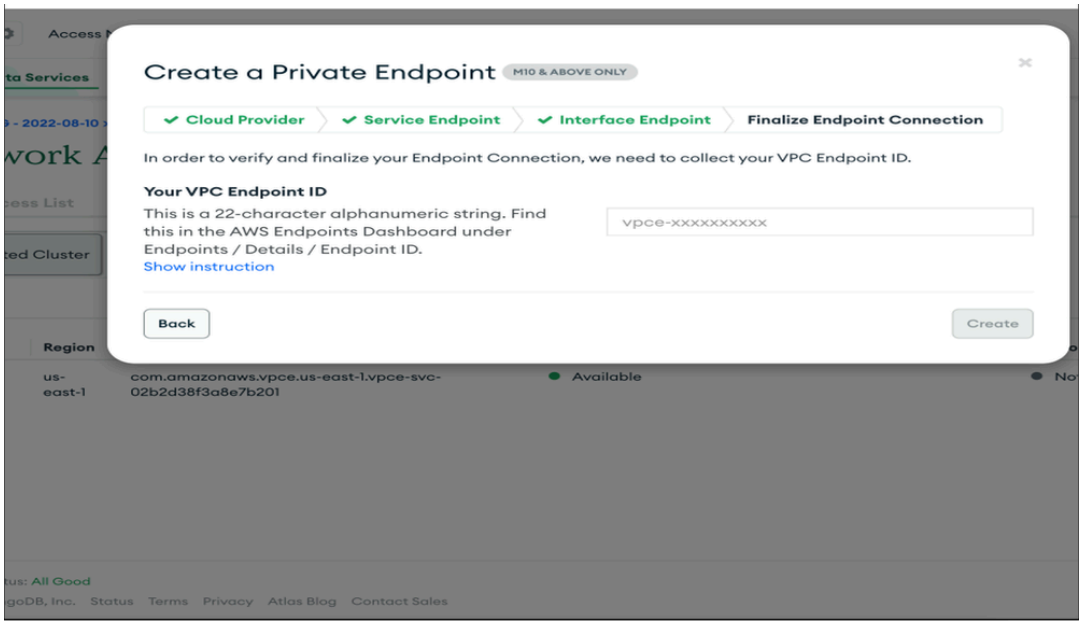

    - AWSとしてCloud Providerを選択し、それぞれのリージョンを選択してNextをクリックします。



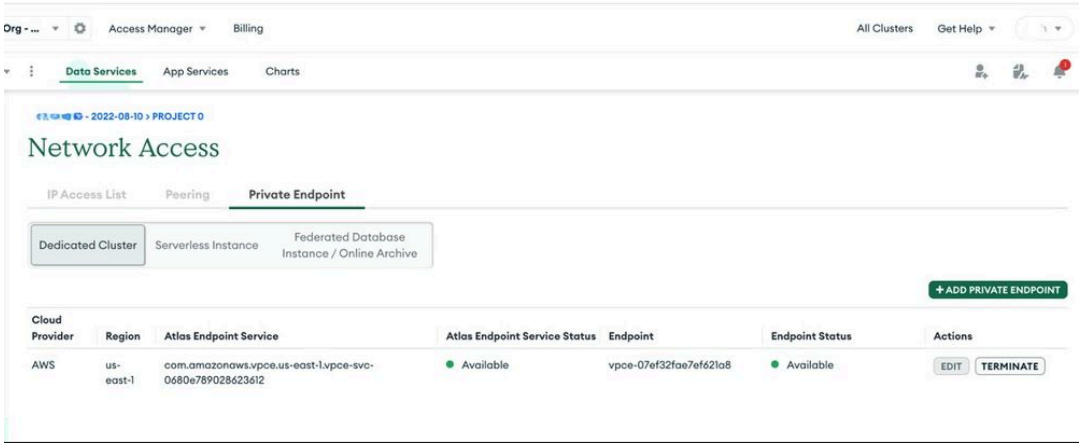

- それぞれのPVC IDとサブネットIDを入力します。詳細を入力したら、vpcエンドポイント作成コマンドをコピーし、awsコンソールで実行します。vpcエンドポイントidが出力として表示されます。



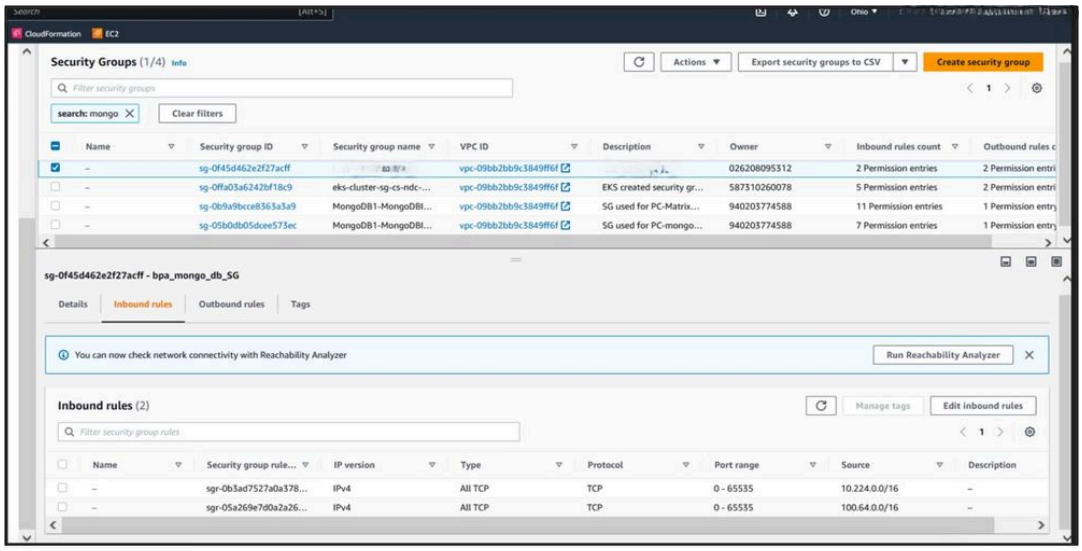

- Nextをクリックしてvpcエンドポイントidを貼り付け、Createをクリックします。




- 正常に作成されると、次の図に示すように、エンドポイントのステータスがAvailableになります。ポッドcidr用にVPCエンドポイントを作成する必要があるこの例では、「100.64.0.0/16」を使用しています。




- 新しく作成したvpc-endpointに受信ルールを追加します。vpc-endpointは親アカウントに存在し、新しく作成したvpc-endpointにセキュリティグループを割り当てる必要があります。




イメージレジストリとしての**ECR**

Amazon ECRリポジトリを作成し、Dockerイメージをそのリポジトリにプッシュするには、いくつかの手順を実行します。以下に、ECRリポジトリを作成し、Dockerイメージにタグを付け、AWS CLIを使用してリポジトリにプッシュする手順を示します。

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

置換：

- **your-image**-namewith**:**ECRリポジトリの適切な名前。
- AWSリージョンを使用したお客様のリージョン

**EKS**ノードの**IAM**ロールの構成

EKSワーカーノード（EC2インスタンス）に、ECRからイメージをプルするためのアクセス許可が関連付けられた必要なIAMロールがあることを確認します。必要なIAMポリシーは次のとおりです。

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

```
    }
  ]
}
```

このポリシーをEKSワーカーノードに関連付けられたIAMロールに関連付けます。

**BPAの導入**
BPAの展開には、EKSワーカーノードへのラベル付け、ノード上でのディレクトリの準備、BPAパッケージのコピー、Helmを使用したBPAの展開など、いくつかの手順があります。

お客様への導入には、次のバージョンのソフトウェアとクラウドサービスを使用しました。

- **BPA:**4.0.3-6
- **RDS（リレーショナルデータベースサービス）:** 16.3-R2
- **MongoDBアトラス:**v5.0.29
- **EKS（Elastic Kubernetesサービス）:**v1.27

これらのコンポーネントにより、堅牢でスケーラブルな導入が可能になり、必要なワークロードを効率的に処理できるようになります。

- **EKSワーカーノードのラベル付け**

```
kubectl label node
```

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

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

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

```
name=node-4
```

- ノード上のディレクトリの準備
  ノード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
```

  ノード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
```

  ノード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
```

  ノード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パッケージのコピー

```
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/
```

- Helmを使用したBPAの展開

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

## 入力セットアップ

- **入力の有効化**
  **values**.yamltoを更新して入力を有効にします。

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

- **BPA証明書を使用したシークレットの作成**
  証明書ディレクトリに移動し、シークレットを作成します。

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

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

- **入力コントローラの更新**
  新しく作成したシークレットを **入力コントローラ.yaml** DSN エントリの例：

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

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

- **入力証明書の更新**
  Helmの削除とインストールを実行して、入力証明書を更新します。

**環境仕様**
環境仕様には、EC2インスタンス、ロードバランサ、VPCエンドポイント、およびRDSインスタンスの要件が含まれます。主な仕様は次のとおりです。

**EC2要件：**

**ストレージ要件：**ノードあたり2 TBの容量EBSボリュームを/optにマウントし、すべてのノードのエントリを/etc/fstabに追加します。

**セキュリティグループインバウンド**:30101、443、0 ~ 65535 TCP、22（ssh用）

**アウトバウンドセキュリティグループ**：すべてのトラフィックを有効にする必要があります。

**DNSリゾルバ**:EC2には、/etc/resolve.confにオンプレミスのリゾルバが必要です。

**ロードバランサの要件：**

- リスナーポートは443、30101である必要があります。
- VPCエンドポイント要件(Atlas MongoDB)
- Atlas接続用に作成されたVPCエンドポイントは、親アカウント(aws-5g-ndc-prod)で使用できます。VPCエンドポイントには、すべての着信アクセスを許可するセキュリティグループが必要です(0 ~ 65535)。

**RDS要件：**

**RDSタイプ**: db.r5b.2xlarge

**Postgres Engine**バージョン:13.7

**セキュリティグループ**:Inboudは、POD CIDRソースからのトラフィックを許可する必要があります。

**主要な概念とコンポーネント**

Amazon EKSを使用してアプリケーションを効果的にデプロイおよび管理するには、Kubernetesの基礎を理解することが不可欠です。

---

**結論**
このホワイトペーパーでは、Amazon EKSを使用してビジネスプロセス自動化(BPA)アプリケーションをデプロイおよび管理するための詳細なガイドを提供します。概説された手順に従い、主要な概念を理解することで、組織はコンテナ化BPAアプリケーションに対してEKSの利点を活用できます。

---

参考資料


- **アマゾンウェブサービス、「Amazon EKSドキュメント」[オンライン]。入手可能：https://docs.aws.amazon.com/eks/**
- **Kubernetes, "Kubernetes Documentation", [オンライン].入手可能：https://kubernetes.io/docs/home/**
- **Cisco BPAの概要https://www.cisco.com/c/en/us/solutions/collateral/service-provider/at-a-glance-c45-742579.html**
- **BPA操作ガイドhttps://www.cisco.com/c/dam/en/us/support/docs/bpa/v403/cisco-bpa-operations-guide-v403.pdf**
- **BPA開発者ガイドhttps://www.cisco.com/c/dam/en/us/support/docs/bpa/v403/cisco-bpa-developer-guide-v403.pdf**

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。