Deploy and Manage Business Process Automation Application on Amazon EKS: A Practical Guide

Contents

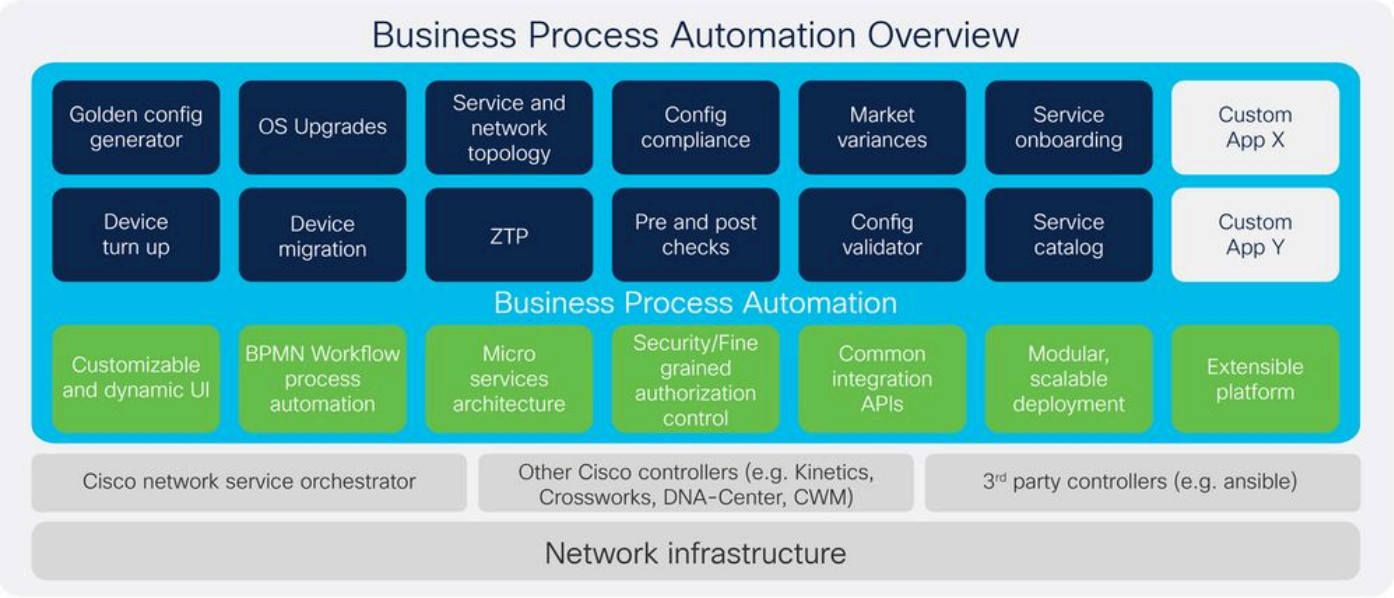
ABSTRACT

This paper presents a comprehensive guide on deploying and managing Business Process Automation (BPA) applications using Amazon Elastic Kubernetes Service (EKS). It outlines the prerequisites, highlights the benefits of utilizing EKS, and provides step-by-step instructions for setting up an EKS cluster, Amazon RDS database, and MongoDB Atlas. Additionally, the paper delves into the deployment architecture and specifies the environment requirements, offering a thorough resource for organizations aiming to leverage EKS for their containerized BPA applications.

Keywords: Amazon EKS, Kubernetes, AWS, RDS, MongoDB Atlas, DevOps, Cloud Computing, Business Process Automation.

INTRODUCTION

BUSINESS PROCESS AUTOMATION (BPA)



Cisco Business Process Automation (BPA) Services offer an end-to-end consulting and support services portfolio designed for process and workflow automation and orchestration. The BPA platform is scalable and microservices-based, featuring an embedded workflow engine, digital user interface, and common integration middleware. This platform helps automate complex network configuration changes and associated processes, making it suitable for both service provider customers and large global enterprises.

Key benefits of Cisco BPA Services include:

- Automating complicated methods and operating procedures.
- Enhancing team expertise to speed up automation initiatives.
- Accelerating the rollout of new services with an improved user interface/portal.
- Integrating legacy networks with new automation capabilities.

The BPA platform supports various business and IT/operational use cases such as OS upgrades, service provisioning, and integration with orchestration engines. Customers can access a lifecycle of services and BPA capabilities, including advisory, implementation, business-critical services, and solution support, Cisco BPA Services aim to increase operational efficiencies, reduce costly errors, improve business agility, and deliver faster returns on automation investments.

AMAZON ELASTIC KUBERNETES SERVICE (EKS)

Amazon Elastic Kubernetes Service (EKS) is a fully managed Kubernetes service provided by Amazon Web Services (AWS). Launched in 2018, EKS simplifies the process of deploying, managing, and scaling containerized applications using Kubernetes, an open-source container orchestration platform. EKS abstracts the complexities of Kubernetes cluster management, allowing developers to focus on building and running applications without the need to handle the underlying infrastructure.

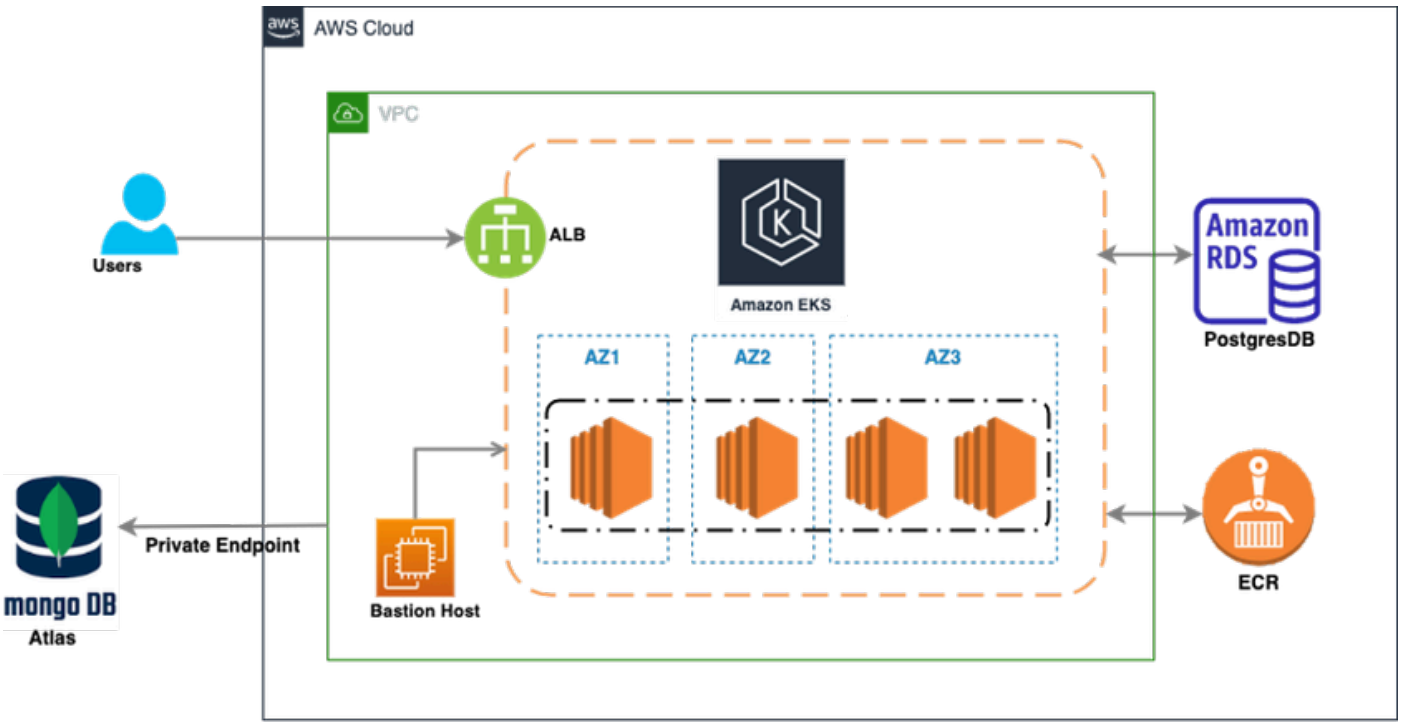
Benefits of Using Amazon EKS for Application Deployment

Amazon EKS offers several benefits for application deployment, making it a popular choice for organizations leveraging containerized applications and microservices.

Key advantages include:

- Managed Kubernetes Control Plane: EKS handles the deployment, scaling, and maintenance of the Kubernetes control plane, reducing operational burden.
- Simplified Cluster Management: EKS abstracts the complexities of setting up and managing Kubernetes clusters.
- Scalability: EKS allows for easy scaling of clusters to accommodate growing workloads.
- High Availability: EKS supports multi-availability Zone deployments, enhancing availability and fault tolerance.
- Integration with AWS Services: EKS integrates seamlessly with various AWS services.
- DevOps Automation: EKS supports continuous integration and continuous deployment (CI/CD) for containerized applications.

BPA DEPLOYMENT ARCHITECTURE



This image represents a high-level architecture of a cloud-based infrastructure deployed on AWS , using several key components. Here's a breakdown of the diagram:

1. Amazon EKS (Elastic Kubernetes Service): At the core of the diagram, Amazon EKS is deployed across three availability zones (AZ1, AZ2, AZ3), with Kubernetes worker nodes inside each zone. This indicates a highly available and fault-tolerant setup, as the workloads are spread across multiple availability zones.
2. ALB (Application Load Balancer): This is positioned at the front, receiving traffic from users and distributing it across the EKS cluster for handling application workloads. The load balancer ensures that the requests are evenly distributed and can handle scaling based on traffic demand.
3. Amazon RDS (Relational Database Service) - PostgreSQL: On the right side of the diagram, an Amazon RDS instance running PostgreSQL is present. This database can be accessed by applications

running within the EKS cluster.

4. ECR (Elastic Container Registry): This is where Docker container images are stored and managed, which are then deployed to Amazon EKS for running the workloads.
5. MongoDB Atlas: On the left side, MongoDB Atlas is integrated into the architecture through a private endpoint. MongoDB Atlas is a cloud-hosted NoSQL database service, used here to handle document-based database requirements. The private endpoint ensures secure, private communication between the MongoDB Atlas instance and other AWS components.
6. Bastion Host: Positioned within the VPC (Virtual Private Cloud), a Bastion Host provides a secure entry point for administrators to access resources inside the VPC without directly exposing them to the internet.
7. Overall, this architecture provides a highly available, scalable, and secure solution for deploying and managing containerized applications using Amazon EKS, with support for both relational (PostgreSQL) and NoSQL (MongoDB) databases.

EKS CLUSTER SETUP

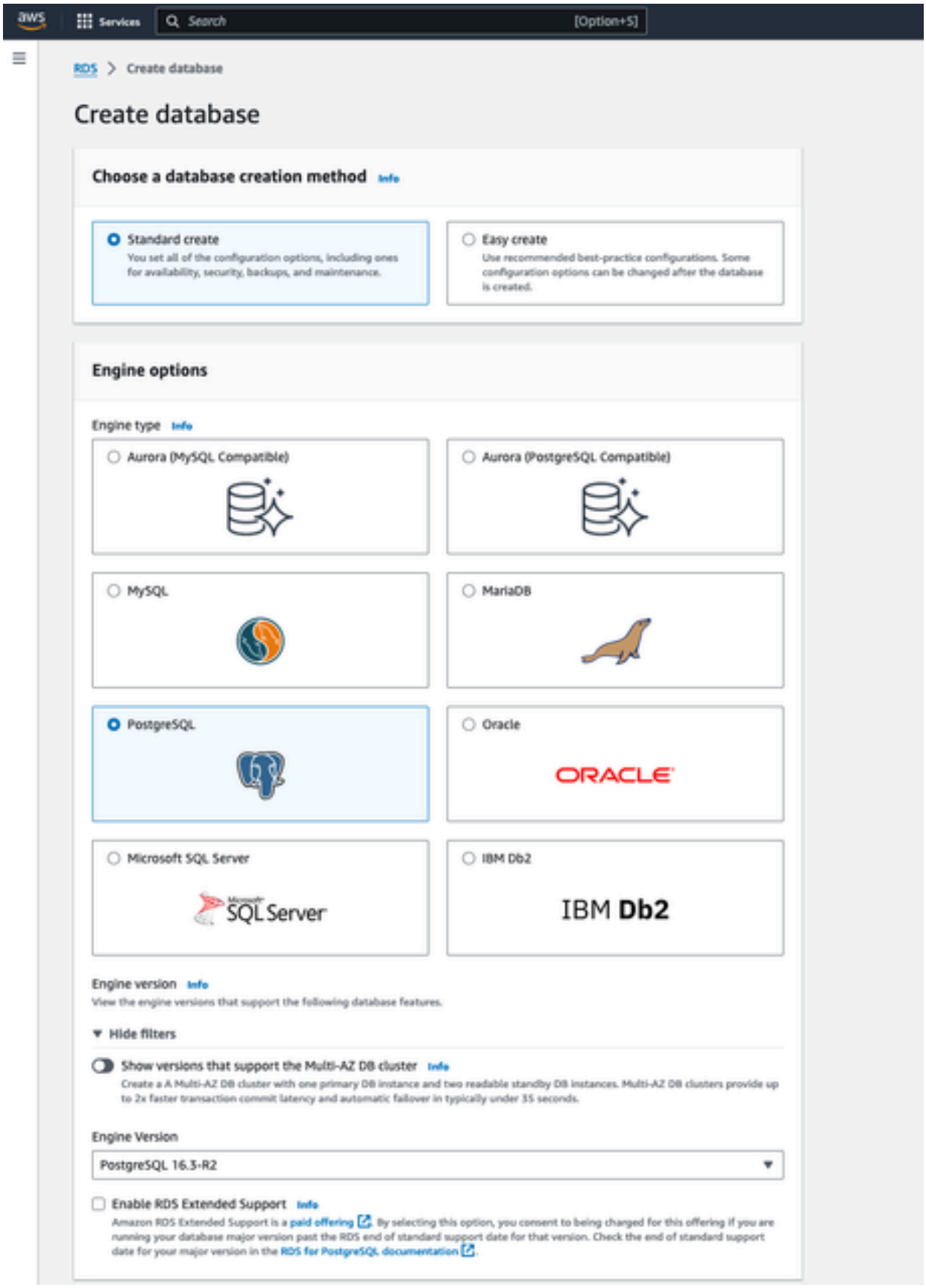
- To create an Amazon EKS cluster using the AWS CLI, the `eksctl` command-line utility can be used. This is an example command:

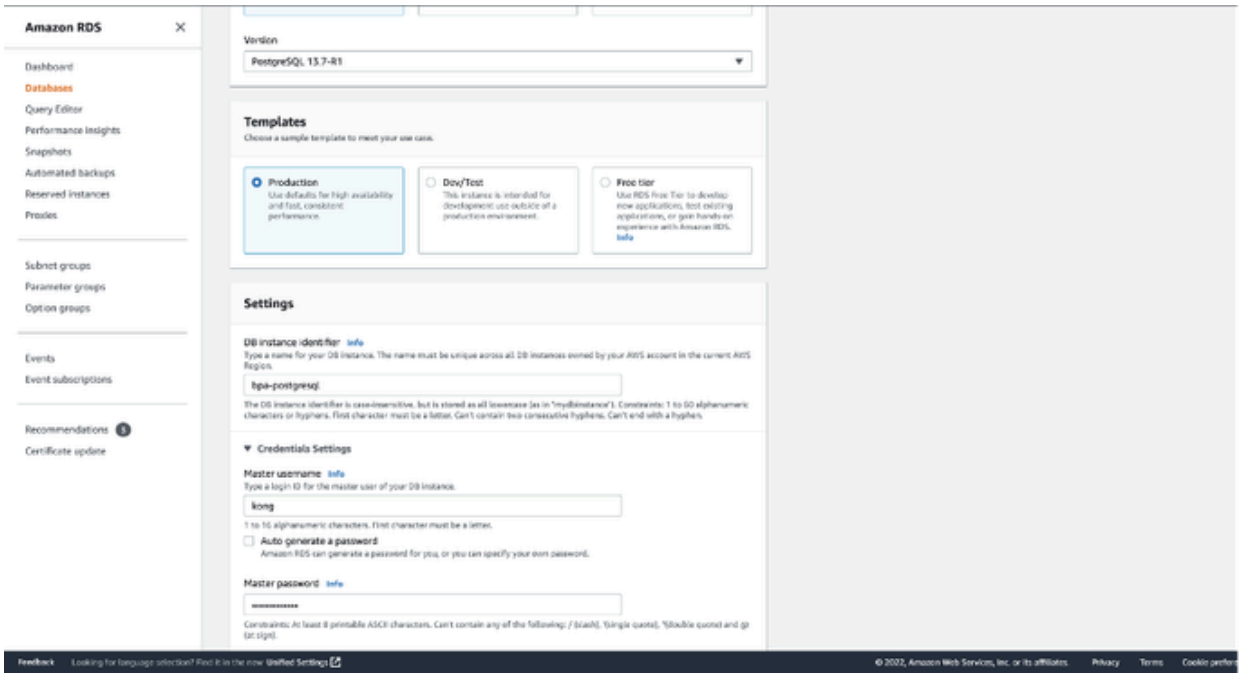
```
eksctl create cluster \
--name <my-eks-cluster> \
--region us-west-2 \
--nodegroup-name standard-workers \
--node-type t3.medium \
--nodes 4 \
--nodes-min 4 \
--nodes-max 6
```

RDS DATABASE SETUP

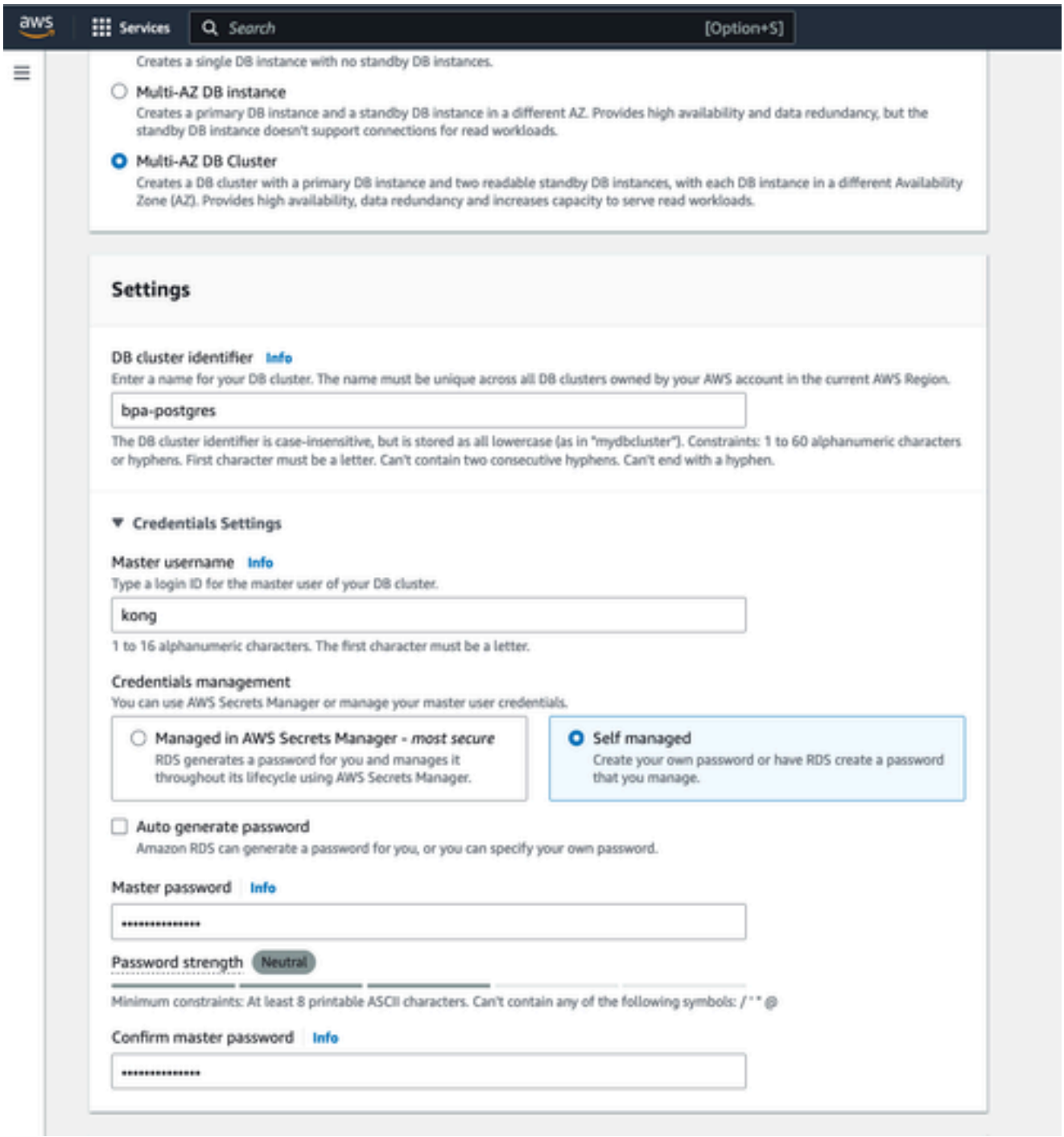
Deploying a relational database on Amazon RDS involves these steps:

1. Access the AWS Management Console and navigate to the Amazon RDS service.
2. Create a new database instance with the desired specifications.
3. Configure the security group to allow incoming connections from your Amazon EKS cluster.
    1. Using the drop-down menu, select the most recent version of PostgreSQL. In our case, it is “PostgreSQL 16.3-R1”.

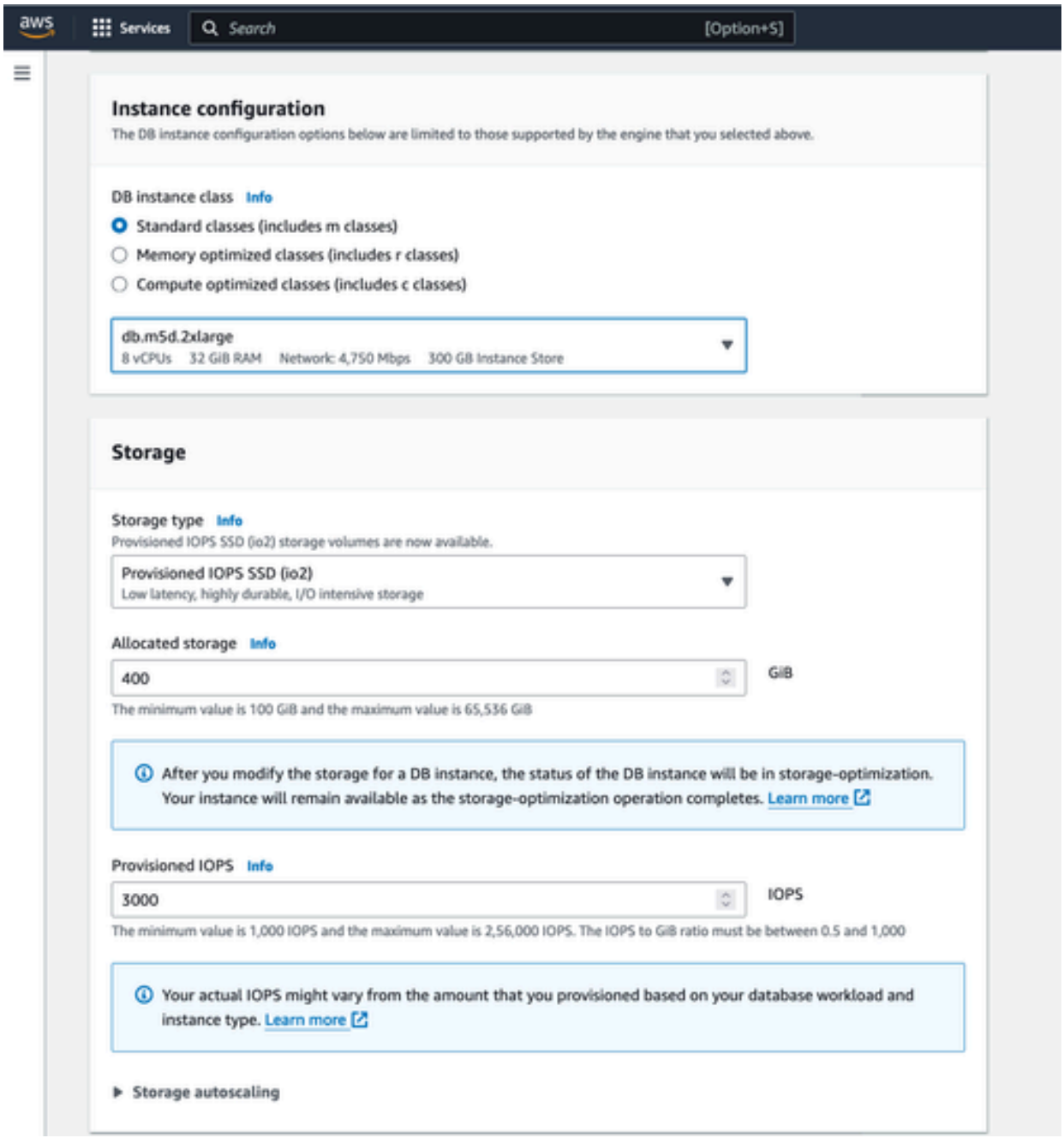

aws
Services
Search
[Option+S]
RDS > Create database
Create database
Choose a database creation method Info
Standard create
You set all of the configuration options, including ones for availability, security, backups, and maintenance.
Easy create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.
Engine options
Engine type Info
Aurora (MySQL Compatible)
Aurora (PostgreSQL Compatible)
MySQL
MariaDB
PostgreSQL
Oracle
ORACLE
Microsoft SQL Server
SQL Server
IBM Db2
IBM Db2
Engine version Info
View the engine versions that support the following database features.
Hide filters
Show versions that support the Multi-AZ DB cluster Info
Create a A Multi-AZ DB cluster with one primary DB instance and two readable standby DB instances. Multi-AZ DB clusters provide up to 2x faster transaction commit latency and automatic failover in typically under 35 seconds.
Engine Version
PostgreSQL 16.3-R2
Enable RDS Extended Support Info
Amazon RDS Extended Support is a paid offering. By selecting this option, you consent to being charged for this offering if you are running your database major version past the RDS end of standard support date for that version. Check the end of standard support date for your major version in the RDS for PostgreSQL documentation.



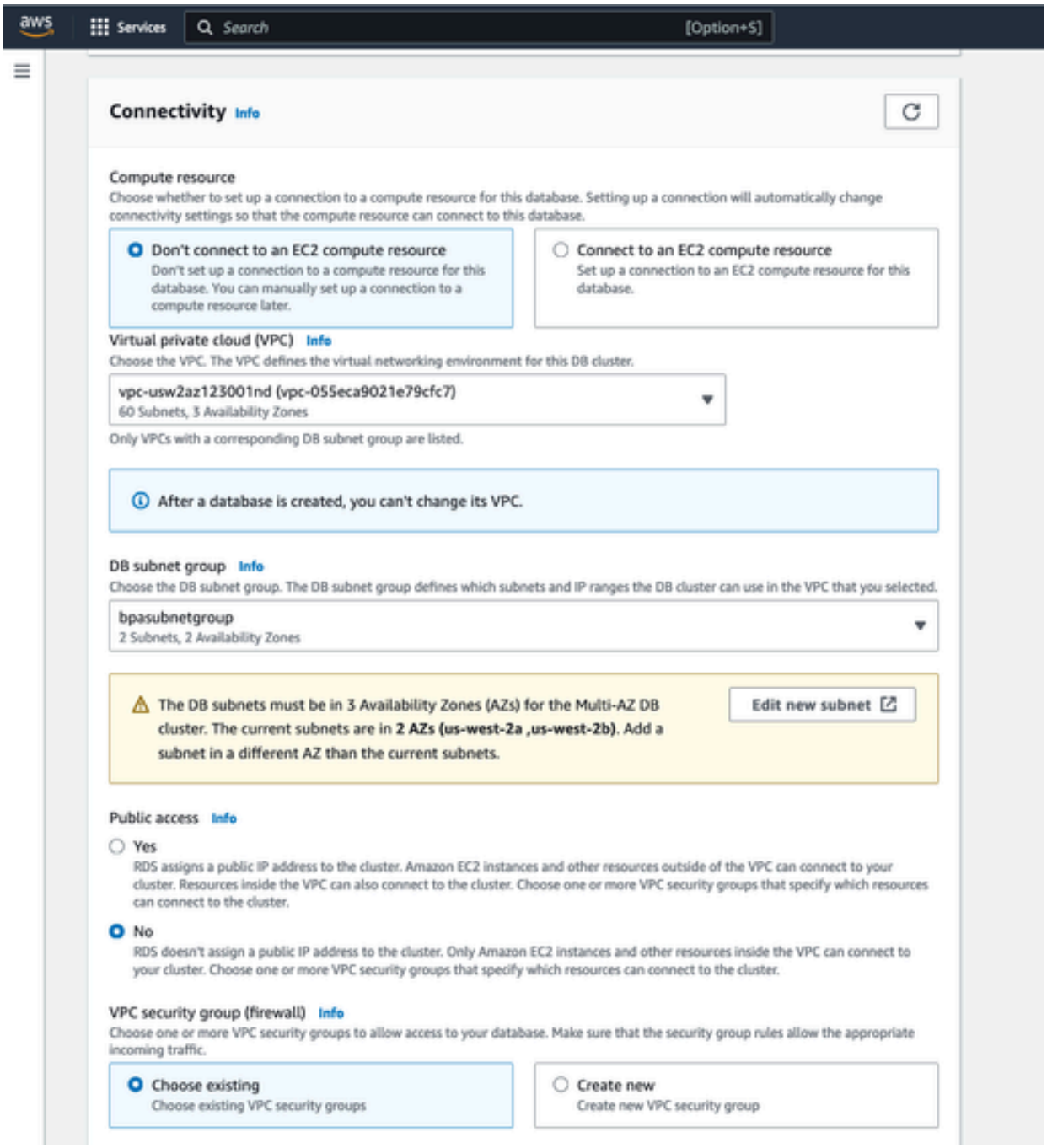
4. For this give the database instance a name and create a username and password.



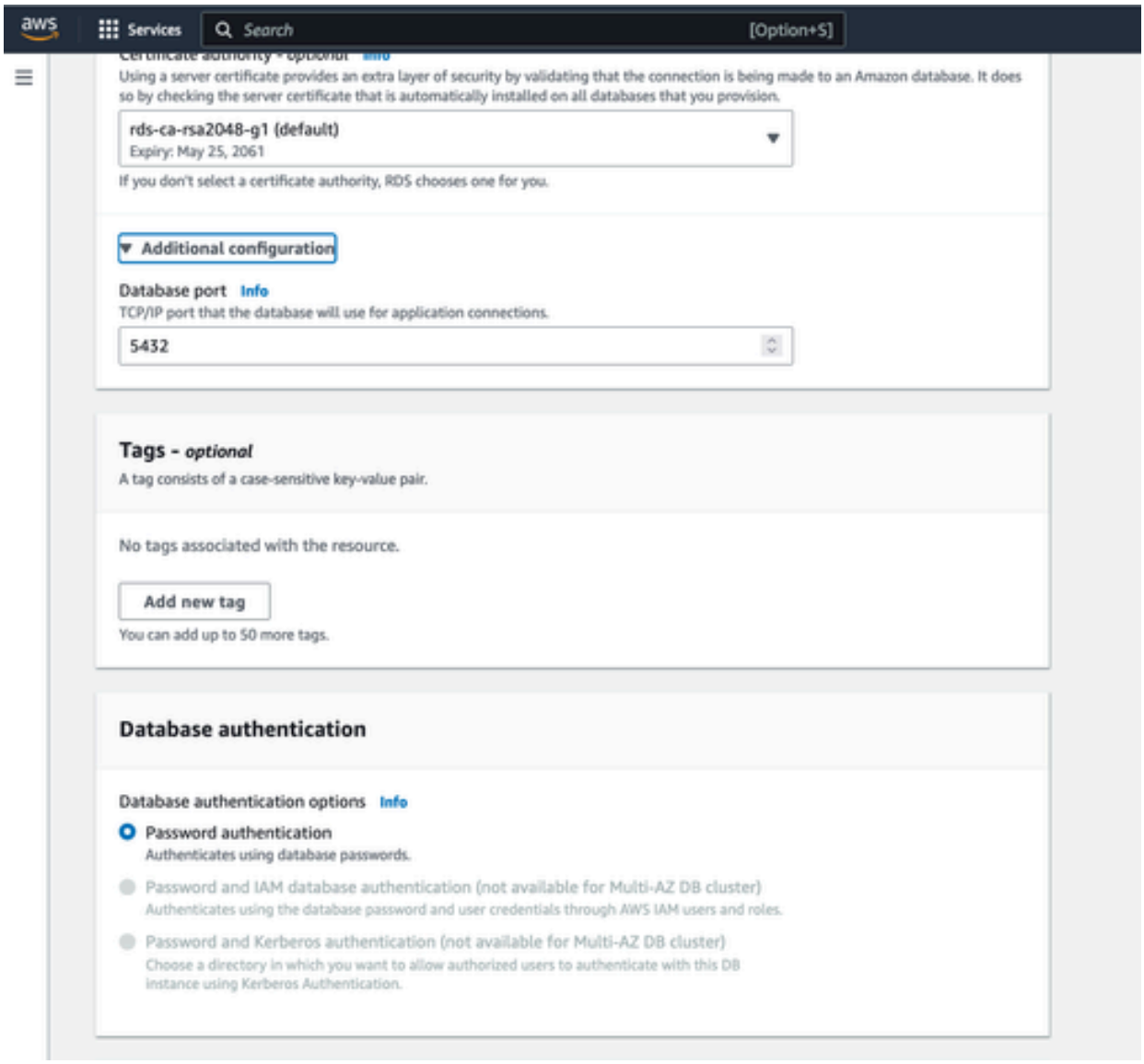
1. Ensure that the default settings for “DB instance size” and “Storage” are selected. Depending on the cluster size and data requirements, select the appropriate DB instance size and storage type.
2. Based on our use case, we have chosen the following configuration**:**
    - DB Instance Size: db.m5d.2xlarge
    - 8 vCPUs
    - 32 GiB RAM
    - Network: 4,750 Mbps
    - 300 GB Instance Store
3. Select appropriate values according to your use case. We have selected the default values.

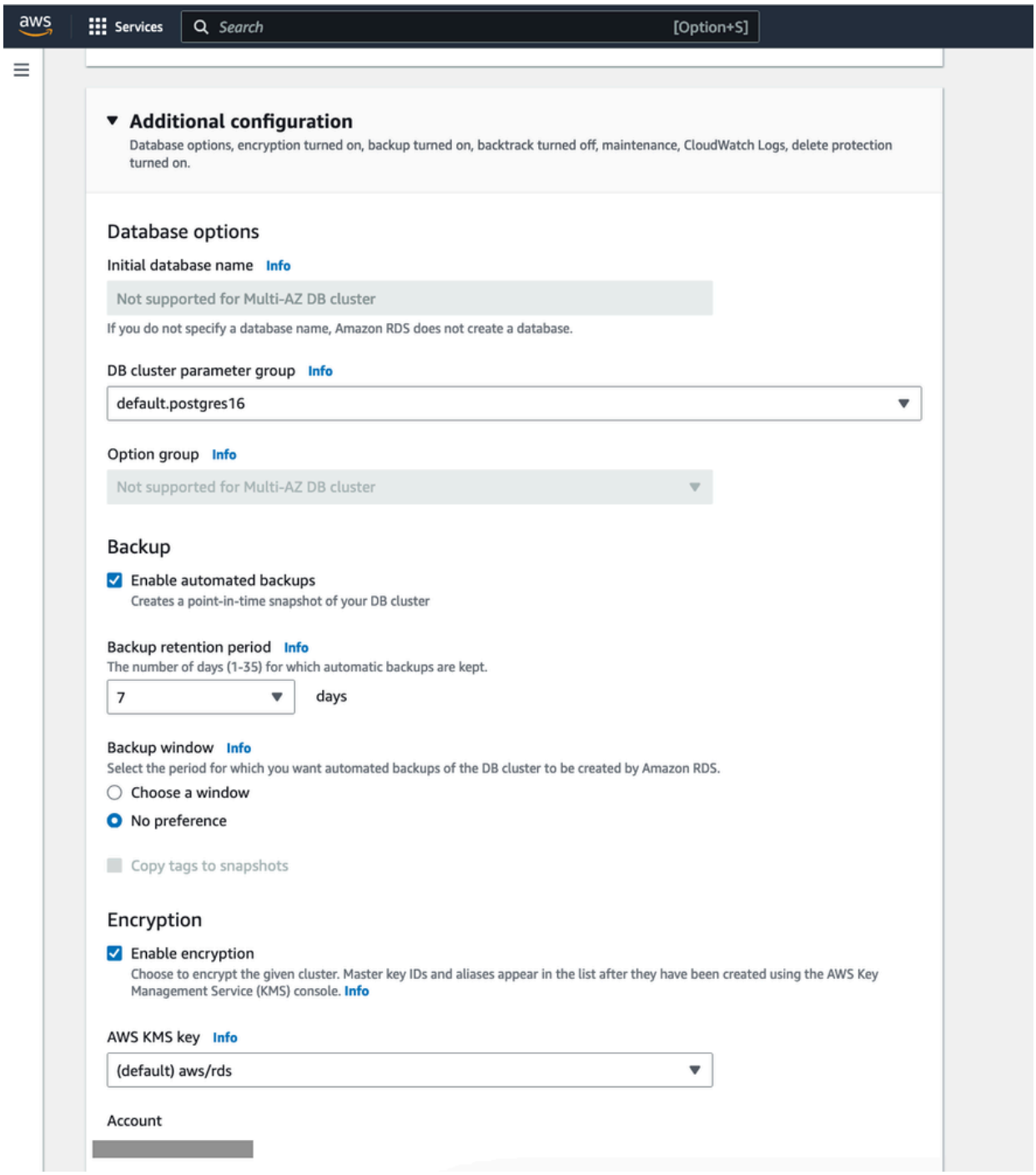


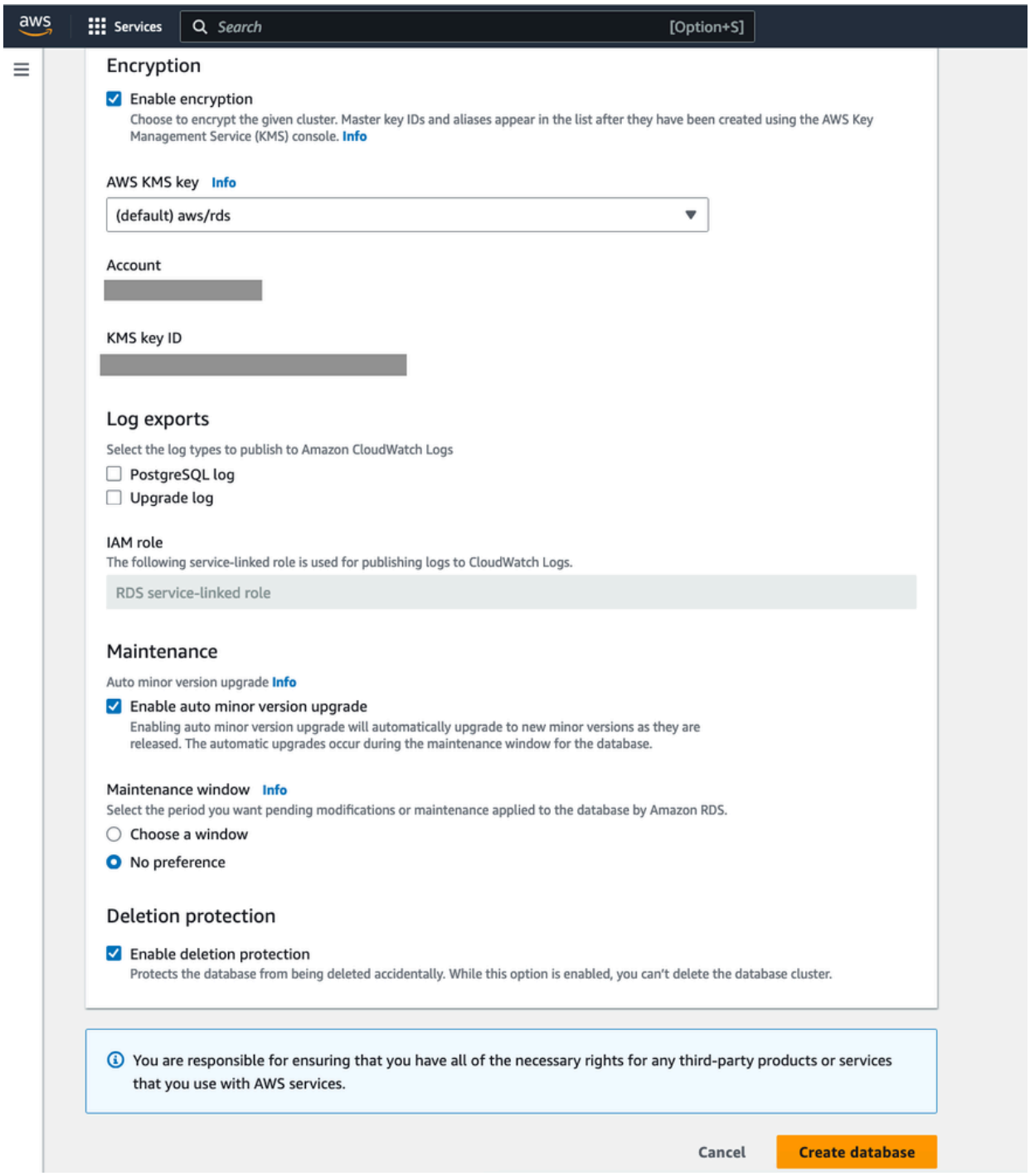
4. Make sure in “Database authentication” we have selected Password authentication. Authenticates using database passwords.



5. Once that is verified, we are ready to create the database. Return to the Amazon RDS dashboard. Confirm that the instance is available for use.

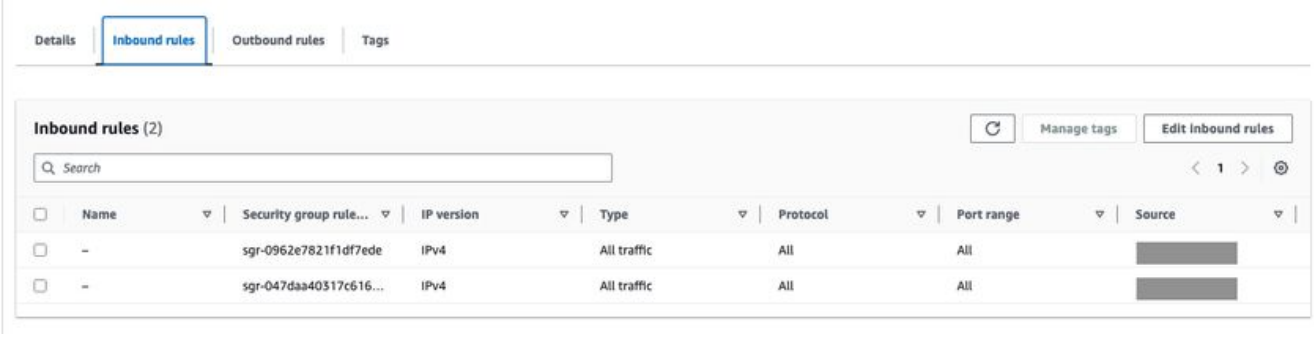

aws
Services
Search
[Option+S]
Using a server certificate provides an extra layer of security by validating that the connection is being made to an Amazon database. It does so by checking the server certificate that is automatically installed on all databases that you provision.
rds-ca-rsa2048-g1 (default)
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

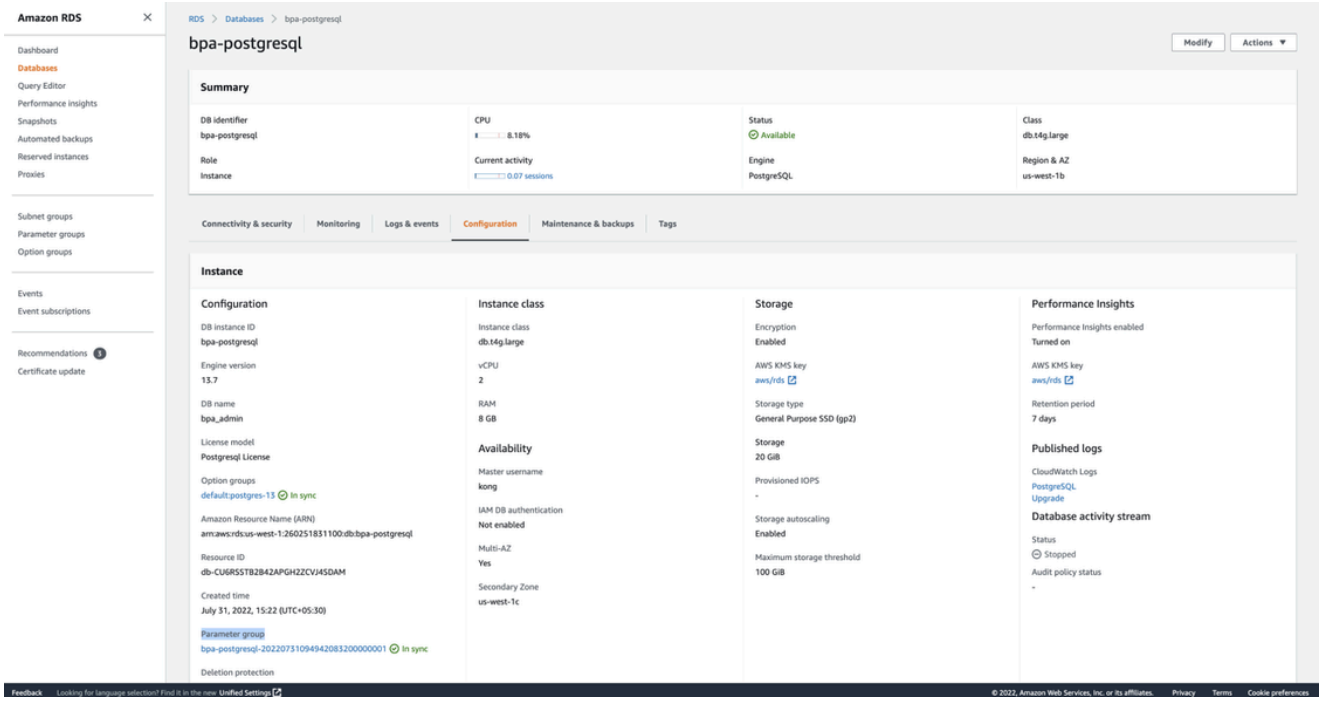


Security Group Rules

1. Update the inbound security group with the pod CIDR and node CIDR block.
2. In RDS -> Databases -> DB-NAME, click configuration and refer the Parameter Group section and click the parameter group to view.



3. Search for “password_encryption” and change the value to md5 from blank / other value. This is needed for camunda configurations to work.



1. Create these Databases along with users by connecting to the RDS.

```
PG_ROOT_DATABASE=admin
PG_INITDB_ROOT_USERNAME=admin
PG_INITDB_ROOT_PASSWORD=xxxxxxx
AUTH_DB_NAME=kong
AUTH_DB_USER=kong
AUTH_DB_PASSWORD=xxxxxxxxx
WFE_DB_USER=camunda
WFE_DB_PASSWORD=xxxxxxxx
WFE_DB_NAME=process-engine
```

2. To update database passwords, modify the values in the `bpa-helm-chart/bpa/env/environment.txt` file. This file is used for authenticating database connections.

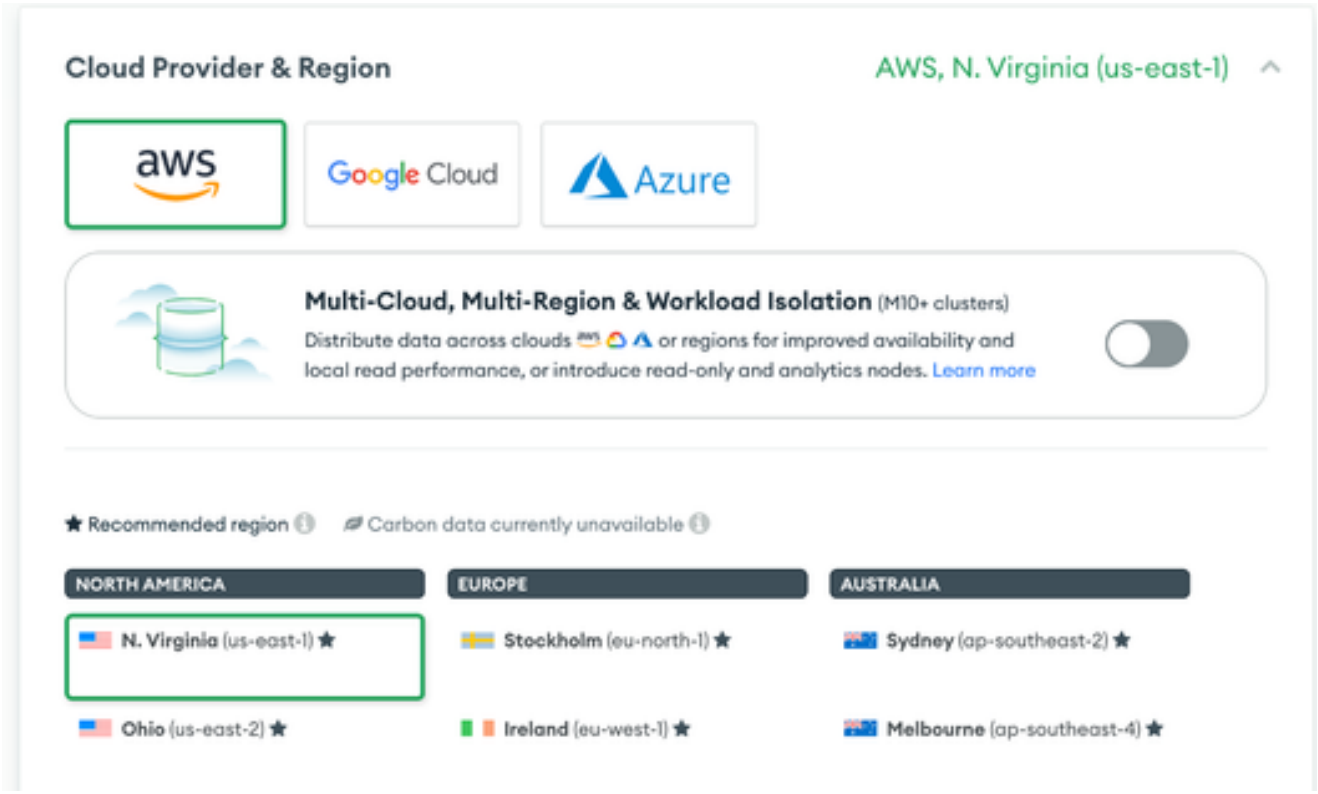
ATLAS MONGODB SETUP

Setting up Atlas MongoDB involves:

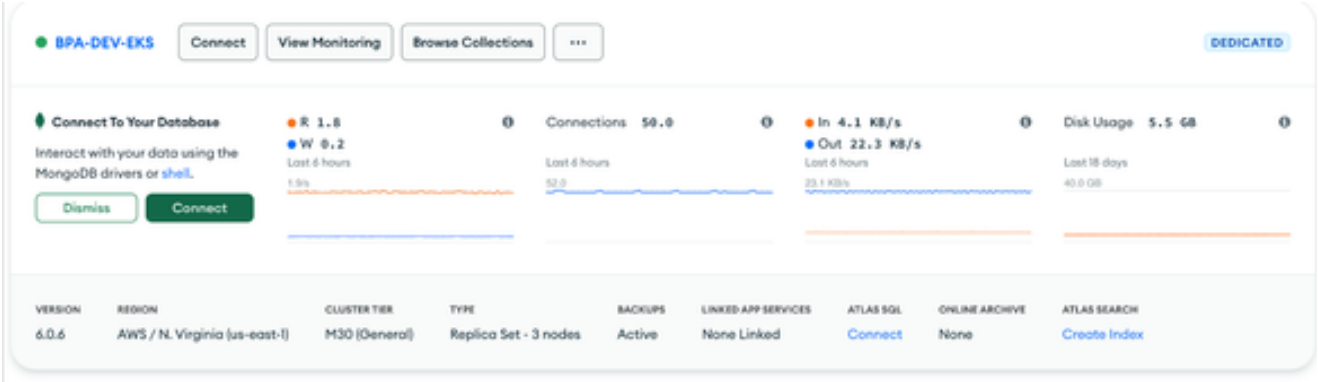
1. Logging into Atlas MongoDB.

2. Selecting the organization and project.
3. Creating a dedicated cluster with the appropriate specifications and version. In our case, it is “MongoDB Atlas v5.0.29”.

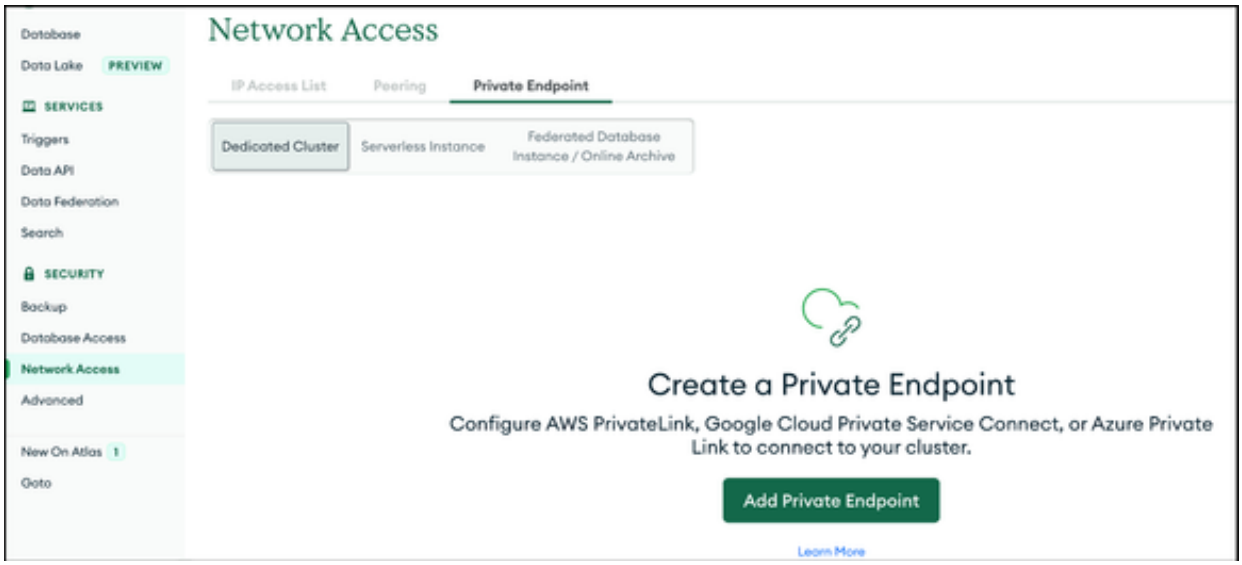
4. Select the Dedicated tier, Cloud Provider & Region.



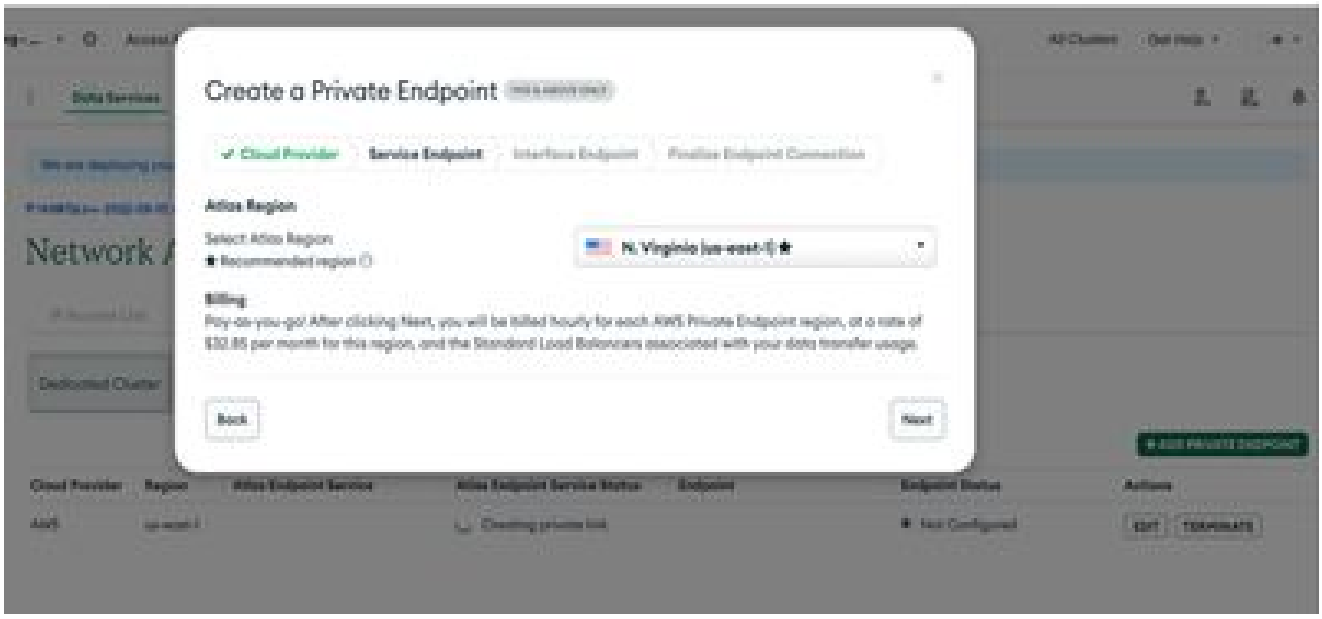
5. Select appropriate tier(we have used M30 as tier) dedicated cluster and provide appropriate cluster name and click on Create Cluster. It will initialize the Atlas monogodb cluster.



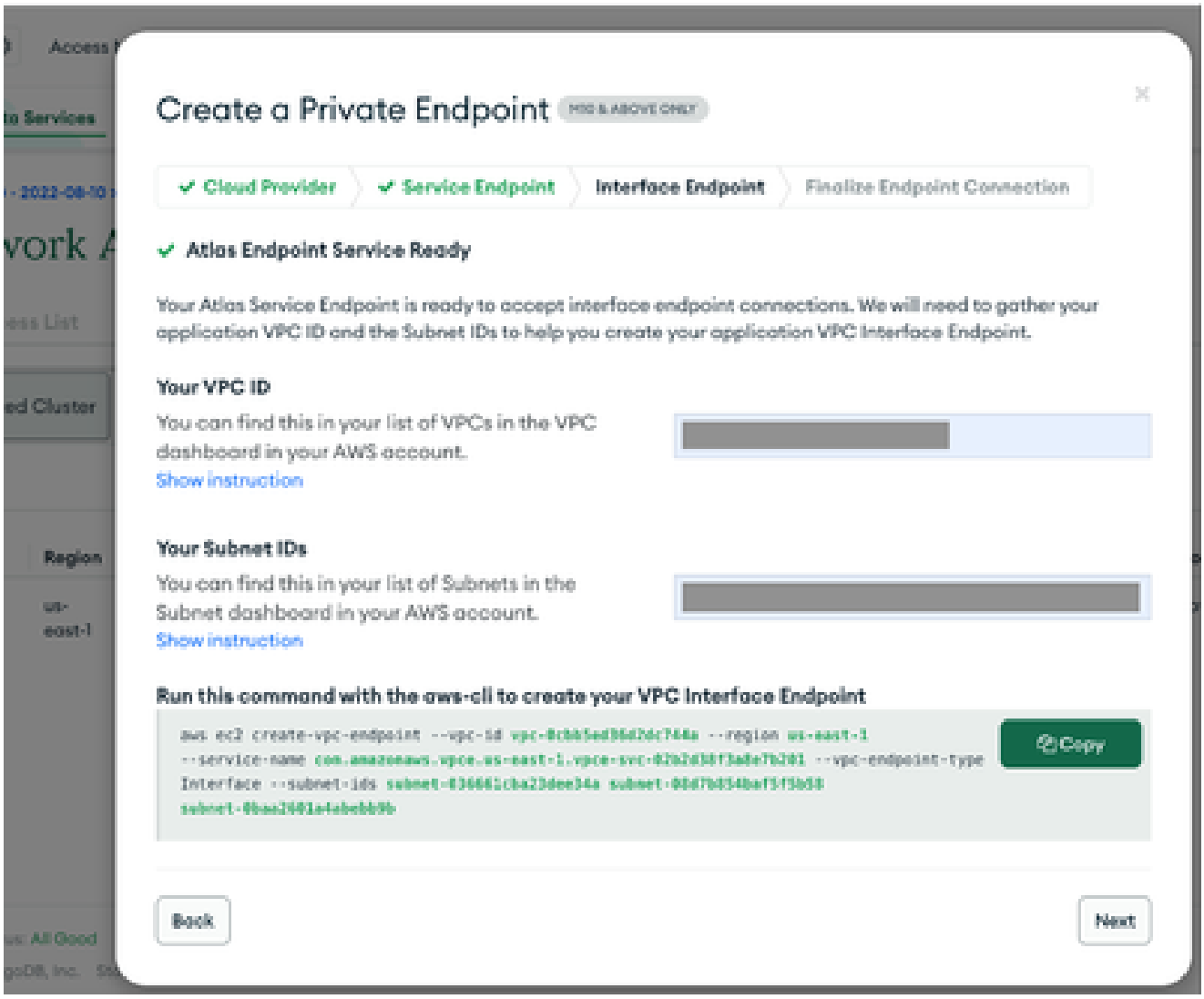
6. Setting up VPC private endpoint for the Atlas and K8S cluster.
    1. Click on the Network Access Select Private Endpoint à Click on Add Private Endpoint.



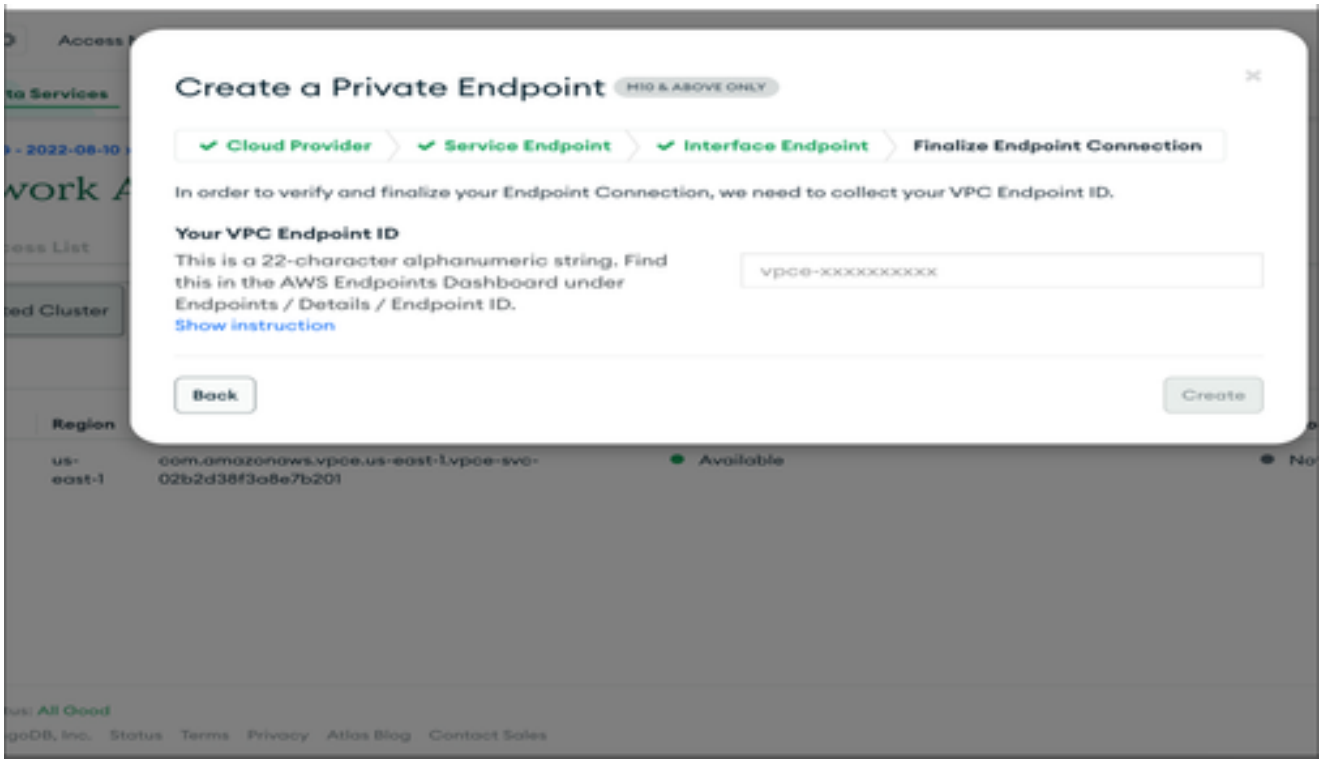
7. Select Cloud Provider as AWS, select respective Region and click on Next.



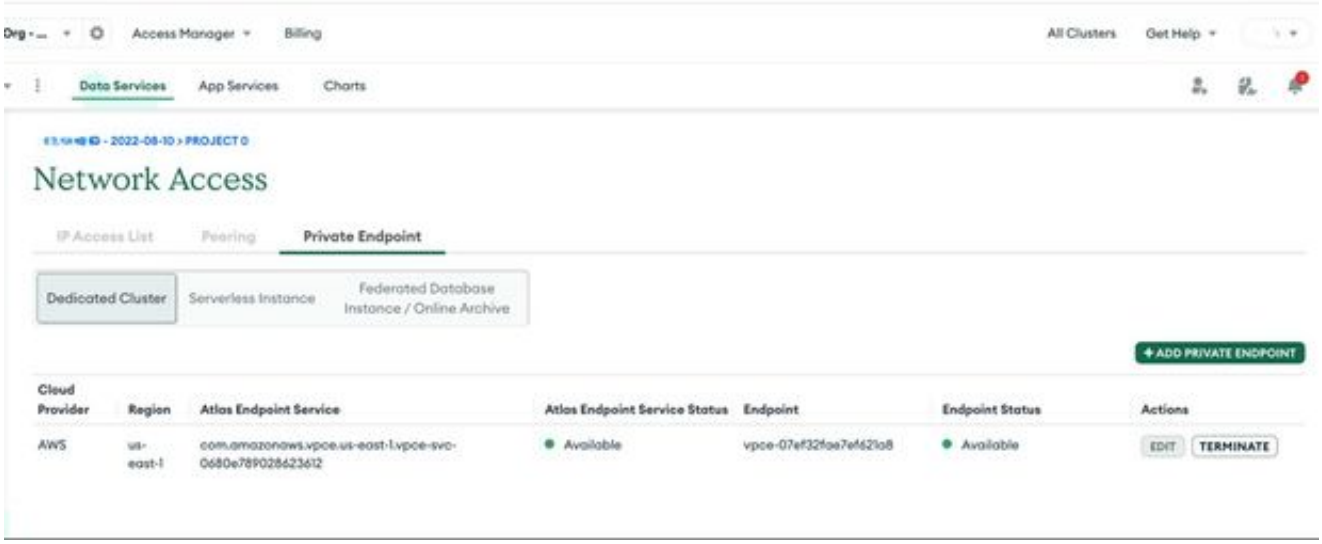
8. Provide Respective PVC id and subnet ids. Once you enter the details, Copy the vpc end point creation command and execute it in aws console. You will get the vpc endpoint id as output.



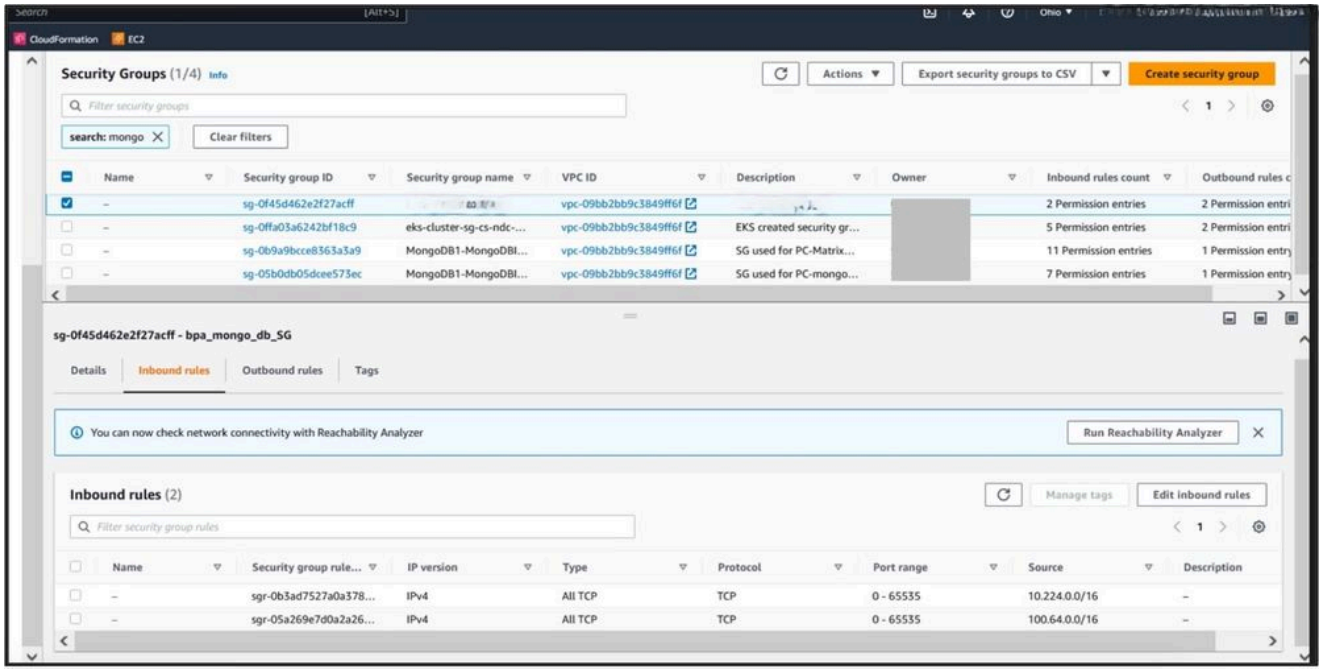
9. Click on Next to paste the VPC endpoint ID and click on Create.



10. Once it is successfully created, cndpoint status will be Available as shown in the next picture. VPC end-point must be created for pod cidr. In our case we have used "100.64.0.0/16" .



11. Add inbound rules to newly created vpc-endpoint. The vpc-endpoint will be in the parent account and a security group must be assigned to the newly created vpc-endpoint.



ECR AS IMAGE REGISTRY

1. Creating Amazon ECR repositories and pushing Docker images into them involves several steps. These are the steps to create an ECR repository, tag a Docker image, and push it to the repository using the AWS CLI.

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

2. Replace:

- your-image-name with the desired name for your ECR repository.
- your-region with your AWS region

3. Steps to tag and push the image.
   - Example**:**

```
docker tag containers.cisco.com/bpa/sase-service:4.0.3-522
<account number>.dkr.ecr.us-west-
2.amazonaws.com/<repository_path>/sase-service:4.0.3-522

docker push <account number>.dkr.ecr.us-west-
2.amazonaws.com/<repository_path>/sase-service:4.0.3-522
```

4. Configure IAM Role for EKS Nodes

5. Ensure that the EKS worker nodes (EC2 instances) have the necessary IAM role attached with permissions to pull images from ECR. The IAM policy required is:

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

6. Attach this policy to the IAM role associated with your EKS worker nodes.

BPA DEPLOYMENT

The deployment of BPA involves several steps, including labeling EKS worker nodes, preparing directories on nodes, copying BPA packages, and deploying BPA using Helm.

1. For our customer deployment, we have utilized the following versions of software and cloud services:
    1. BPA: **4.0.3-6**
    2. RDS (Relational Database Service): **PostgreSQL 16.3-R1**
    3. MongoDB Atlas: **v5.0.29**
    4. EKS (Elastic Kubernetes Service): **v1.27**
2. These components ensure that our deployment is robust, scalable, and capable of handling the required workloads efficiently.
3. Labeling EKS Worker Nodes

```
kubectl label node <worker_node_1> name=node-1
kubectl label node <worker_node_2> name=node-2
kubectl label node <worker_node_3> name=node-3
kubectl label node <worker_node_4> name=node-4
```

Preparing Directories on Nodes

- **NODE 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
```

- **NODE 2:**

```
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
```

- **NODE 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
```

- **NODE 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
```

Copying BPA Packages

```
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/
```

Deploying BPA Using Helm

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

Ingress Setup

Enabling Ingress

- Update `values.yaml`to enable ingress:

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

Creating a Secret Using BPA Certificate

- Navigate to the certificate directory and create a secret:

```
cd /opt/bpa/<BPA helm chart location>/bpa/conf/common/certs/
kubectl create secret tls bpa-certificate-ingress --cert=bap-cert.pem --key=bap-key.pem -n bpa-ns
```

Updating Ingress Controller

- Add the newly created secret in the `ingress-controller.yaml` file:

```
cd /opt/bpa/<BPA helm chart location>/templates/
vi ingress-controller.yaml
"- --default-ssl-certificate=$(POD_NAMESPACE)/bpa-certificate-ingress"
```

Updating Ingress Certificate

- Perform Helm delete and install to update the ingress certificate.

Environment Specifications

The environment specifications include requirements for EC2 instances, load balancers, VPC endpoints, and RDS instances. Key specifications are:

- EC2 Requirements:

    1. **Storage requirements:**2TB space per nodes. Mount EBS volume to /opt and add an entry in /etc/fstab for all the nodes.

    2. **Security group inbound**: 30101, 443, 0 – 65535 TCP, 22 for ssh.

    3. **Security group outbound**: All traffic must be enabled.

    4. **DNS Resolver**: EC2 must have on-prem resolvers in /etc/resolve.conf.

Load balancer requirements:

- Listeners ports must be 443, 30101.
- VPC End point Requirements (Atlas MongoDB).
- VPC end points created for Atlas connectivity is available in the parent account. VPC Endpoint must have security group which allows all inbound access(0 - 65535).

KEY CONCEPTS AND COMPONENTS

Understanding Kubernetes fundamentals is essential for effectively deploying and managing applications using Amazon EKS.

CONCLUSION

This paper provides a detailed guide for deploying and managing Business Process Automation (BPA) applications using Amazon EKS. By following the outlined steps and understanding the key concepts, organizations can leverage the benefits of EKS for their containerized BPA applications.

REFERENCES

Amazon Web Services, "Amazon EKS Documentation," [Online]. Available:https://docs.aws.amazon.com/eks/

Kubernetes, "Kubernetes Documentation," [Online]. Available:https://kubernetes.io/docs/home/

Cisco BPA at a Glance https://www.cisco.com/c/en/us/solutions/collateral/service-provider/at-a-glance-c45-742579.html

BPA Operations Guide **https://www.cisco.com/c/dam/en/us/support/docs/bpa/v403/cisco-bpa-operations-guide-v403.pdf**

BPA Developer Guide **https://www.cisco.com/c/dam/en/us/support/docs/bpa/v403/cisco-bpa-developer-guide-v403.pdf**