**Knowledge Accelerator - Plant GPT**

**By**

Table of Contents

[1. Introduction 3](#_Toc186466990)

[2. Prerequisites 8](#_Toc186466991)

[3. Architecture Diagram 16](#_Toc186466992)

[4. Deployment Steps 17](#_Toc186466993)

[5. Cost and Resource Management 20](#_Toc186466994)

[6. Testing and Troubleshooting 27](#_Toc186466995)

[7. HealthCheck 31](#_Toc186466996)

[8. Backup and Recovery 35](#_Toc186466997)

[9. Maintenance and Monitoring 40](#_Toc186466998)

[10. Emergency Maintenance 44](#_Toc186466999)

[11. Support 45](#_Toc186467000)

# IAM Role and Policy Details

**Knowledge Accelerator – Plant GPT solution Deployment Role**

* Purpose: This IAM role is assumed by the Knowledge Accelerator – Plant GPTsolution components during the deployment process to provision and configure the necessary AWS resources.
* Permissions:

|  |
| --- |
| { "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "cloudformation:CreateStack", "cloudformation:UpdateStack", "cloudformation:DeleteStack", "cloudformation:DescribeStacks", "cloudformation:DescribeStackResources", "cloudformation:DescribeStackEvents", "cloudformation:ListStacks" ], "Resource": "\*" }, { "Effect": "Allow", "Action": [ "ec2:RunInstances", "ec2:CreateSecurityGroup", "ec2:AuthorizeSecurityGroupIngress", "ec2:AuthorizeSecurityGroupEgress", "ec2:DescribeInstances", "ec2:DescribeSecurityGroups", "ec2:DescribeVpcs", "ec2:DescribeSubnets", "ec2:DescribeRouteTables", "ec2:AssociateRouteTable", "ec2:CreateRoute", "ec2:CreateVpcEndpoint", "ec2:ModifyInstanceAttribute" ], "Resource": "\*" }, { "Effect": "Allow", "Action": [ "ecs:CreateCluster", "ecs:DeleteCluster", "ecs:RegisterTaskDefinition", "ecs:DeregisterTaskDefinition", "ecs:CreateService", "ecs:UpdateService", "ecs:DeleteService", "ecs:RunTask", "ecs:StopTask" ], "Resource": "\*" }, { "Effect": "Allow", "Action": [ "rds:CreateDBInstance", "rds:DeleteDBInstance", "rds:ModifyDBInstance", "rds:DescribeDBInstances", "rds:CreateDBSubnetGroup", "rds:ModifyDBSubnetGroup", "rds:DeleteDBSubnetGroup" ], "Resource": "\*" }, { "Effect": "Allow", "Action": [ "neptune:CreateDBCluster", "neptune:DeleteDBCluster", "neptune:ModifyDBCluster", "neptune:DescribeDBClusters" ], "Resource": "\*" }, { "Effect": "Allow", "Action": [ "s3:CreateBucket", "s3:DeleteBucket", "s3:PutBucketPolicy", "s3:GetBucketPolicy", "s3:PutObject", "s3:GetObject", "s3:DeleteObject" ], "Resource": [ "arn:aws:s3:::<your-s3-bucket>", "arn:aws:s3:::<your-s3-bucket>/\*" ] }, { "Effect": "Allow", "Action": [ "lambda:CreateFunction", "lambda:DeleteFunction", "lambda:UpdateFunctionConfiguration", "lambda:InvokeFunction", "lambda:AddPermission", "lambda:RemovePermission" ], "Resource": "\*" }, { "Effect": "Allow", "Action": [ "apigateway:CreateRestApi", "apigateway:DeleteRestApi", "apigateway:UpdateRestApi", "apigateway:Deploy", "apigateway:PutMethod", "apigateway:DeleteMethod", "apigateway:PutIntegration", "apigateway:DeleteIntegration" ], "Resource": "\*" }, { "Effect": "Allow", "Action": [ "amplify:CreateApp", "amplify:DeleteApp", "amplify:UpdateApp", "amplify:CreateBranch", "amplify:DeleteBranch", "amplify:StartDeployment" ], "Resource": "\*" }, { "Effect": "Allow", "Action": [ "cognito-idp:CreateUserPool", "cognito-idp:DeleteUserPool", "cognito-idp:UpdateUserPool", "cognito-idp:CreateUserPoolClient", "cognito-idp:DeleteUserPoolClient", "cognito-idp:UpdateUserPoolClient" ], "Resource": "\*" }, { "Effect": "Allow", "Action": [ "secretsmanager:CreateSecret", "secretsmanager:DeleteSecret", "secretsmanager:UpdateSecret", "secretsmanager:GetSecretValue" ], "Resource": "\*" }, { "Effect": "Allow", "Action": [ "guardduty:CreateDetector", "guardduty:DeleteDetector", "guardduty:ListFindings", "guardduty:GetFindings" ], "Resource": "\*" }, { "Effect": "Allow", "Action": [ "cloudwatch:PutMetricData", "cloudwatch:GetMetricData", "cloudwatch:DescribeAlarms", "cloudwatch:PutDashboard", "cloudwatch:GetDashboard" ], "Resource": "\*" }, { "Effect": "Allow", "Action": [ "securityhub:EnableSecurityHub", "securityhub:DisableSecurityHub", "securityhub:ListFindings", "securityhub:GetFindings" ], "Resource": "\*" }, { "Effect": "Allow", "Action": [ "iam:PassRole" ], "Resource": "\*", "Condition": { "StringEquals": { "iam:PassedToService": [ "ec2.amazonaws.com", "ecs-tasks.amazonaws.com", "lambda.amazonaws.com", "rds.amazonaws.com", "apigateway.amazonaws.com" ] } } } ]} |

**Knowledge Accelerator – Plant GPT solution Operations Role**

* Purpose: This IAM role is assumed by the Knowledge Accelerator – Plant GPT solution components during normal operations to access and manage the deployed resources.
* Permissions:

 **Lambda Role**:

|  |
| --- |
| { "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "bedrock:InvokeModel", "bedrock:InvokeModelWithResponseStream" ], "Resource": "arn:aws:bedrock:region:account-id:model/\*" }, { "Effect": "Allow", "Action": [ "rds:DescribeDBInstances", "rds-db:connect" ], "Resource": "arn:aws:rds:region:account-id:db:your-db-instance-id" }, { "Effect": "Allow", "Action": [ "ecs:RunTask", "ecs:DescribeTasks", "ecs:StopTask" ], "Resource": "arn:aws:ecs:region:account-id:task/your-cluster-name/\*", "Condition": { "StringEquals": { "ecs:cluster": "arn:aws:ecs:region:account-id:cluster/your-cluster-name" } } }, { "Effect": "Allow", "Action": [ "neptune-db:connect" ], "Resource": "arn:aws:neptune-db:region:account-id:cluster/your-neptune-cluster-id" }, { "Effect": "Allow", "Action": [ "s3:GetObject", "s3:PutObject" ], "Resource": "arn:aws:s3:::your-bucket-name/\*" }, { "Effect": "Allow", "Action": [ "secretsmanager:GetSecretValue" ], "Resource": "arn:aws:secretsmanager:region:account-id:secret:your-secret-name" }, { "Effect": "Allow", "Action": [ "logs:CreateLogGroup", "logs:CreateLogStream", "logs:PutLogEvents" ], "Resource": "arn:aws:logs:region:account-id:log-group:/aws/lambda/your-lambda-function-name:\*" }, { "Effect": "Allow", "Action": [ "cognito-idp:AdminInitiateAuth", "cognito-idp:ListUsers" ], "Resource": "arn:aws:cognito-idp:region:account-id:userpool/your-user-pool-id" } ]} |

 **ECS Role:**

|  |
| --- |
| { "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "neptune-db:connect" ], "Resource": "arn:aws:neptune-db:region:account-id:cluster/your-neptune-cluster-id" }, { "Effect": "Allow", "Action": [ "s3:GetObject", "s3:PutObject", "s3:ListBucket" ], "Resource": [ "arn:aws:s3:::your-bucket-name", "arn:aws:s3:::your-bucket-name/\*" ] }, { "Effect": "Allow", "Action": [ "secretsmanager:GetSecretValue" ], "Resource": "arn:aws:secretsmanager:region:account-id:secret:your-secret-name" } ]} |

# Key Creation and Management

* **AWS KMS –**
	+ Purpose: The Knowledge Accelerator - Plant GPT uses an AWS managed key to encrypt sensitive data at rest.
	+ Location: The AWS managed key is created and managed within the customer's AWS KMS service.
	+ Considerations: The AWS KMS manages the lifecycle of AWS managed keys, including key rotation and backup. The Knowledge Accelerator - Plant GPT does not have the ability to delete or modify the CMK.
* **Database Credentials**:
	+ Store securely in AWS Secrets Manager, avoiding hardcoding in application code.

# Network Configuration Details

* **Network Configuration**:

**Virtual Private Cloud (VPC):**

The Knowledge Accelerator - Plant GPT deployment requires a dedicated Amazon VPC with the following configuration:

* + VPC CIDR block: 10.0.0.0/20
	+ 2 private subnets in different availability zones (10.0.8.0/22 and 10.0.12.0/22)

**Security Groups**:

The following security groups are required for the Knowledge Accelerator - Plant GPT deployment:

* + RDS\_Security\_Group: Allows inbound traffic on port 3306.
	+ Neptune\_Security\_Group: Allows inbound traffic on port 8182.
	+ Neo4j\_Security\_Group: Allows inbound traffic on ports 7474 and 7687.
	+ ChromaDB\_Security\_Group: Allows inbound traffic on port 8000.

**Network ACLs and Routing**:

* Network ACLs are configured to allow the necessary ingress and egress traffic for the Knowledge Accelerator - Plant GPT components.
* Route tables are set up to ensure proper routing between the public and private subnets, as well as to the internet gateway.

**Public Subnet Route table**

|  |  |
| --- | --- |
| Destination | Target |
| 0.0.0.0/0 | Internet Gateway |

 **Private Subnet Route table**

|  |  |
| --- | --- |
| Destination | Target |
| 0.0.0.0/0 | NAT Gateway |
| 10.0.0.0/20 | local |

* **Secrets Management**:
	+ Store sensitive information, such as database credentials, in **AWS Secrets Manager**.
	+ **Best Practices**:
		1. Enable automatic rotation for better security.
		2. Restrict access to secrets by assigning appropriate IAM policies to the application role.
	+ Examples:
		1. Encrypt sensitive data in Amazon S3 using server-side encryption (SSE-S3 or SSE-KMS).
		2. Enable encryption for RDS or DynamoDB data using AWS KMS keys.

**4.4 Customer Sensitive Data**

* The Knowledge Accelerator - Plant GPT solution processes and stores the following type of sensitive customer data:
	+ Personally Identifiable Information (PII): Customer names, email addresses, and phone numbers
* The PII data is stored in an Amazon RDS PostgreSQL database, with encryption-at-rest enabled using an AWS KMS customer managed key (CMK).
* The financial data is stored in an Amazon S3 bucket, also encrypted-at-rest
* Access to the sensitive data is restricted using IAM roles and policies, following the principle of least privilege.

**4.5 Disable IMDSv1**

* Go to EC2 dashboard.
* Select your instance, click on Actions, then select Modify instance metadata options.
* Enable instance metadata service option and select required option for IMDSv2.
* Click on save.

# Cost and Resource Management

**Billable Services**

This section outlines the **billable AWS services** involved in the deployment, along with guidance on whether each service is **mandatory** or **optional**. Additionally, the guide provides details on the **cost model** and **licensing costs** to help customers understand the financial implications of the deployment.

**1. Billable Services Overview**

The following services are utilized in the deployment, each with associated costs based on usage. A detailed breakdown is provided below:

**Mandatory Billable Services**

These services are essential for the deployment and must be provisioned for the solution to function correctly:

* **Amazon EC2 (Elastic Compute Cloud)**:
	+ **Purpose**: Virtual machines for running the application and processing workloads.
	+ **Cost**: Billed based on instance type, region, and usage time (per second or per hour).
	+ **Licensing**: EC2 instances are typically billed under an **on-demand** model, but **reserved instances** or **spot instances** can offer cost savings.
* **AWS Lambda**:
	+ **Purpose**: Serverless computing for executing code without managing servers (useful for event-driven tasks).
	+ **Cost**: Billed based on **invocations** and **compute time** (duration of execution).
	+ **Licensing**: Pay-per-use based on the **execution time** of the Lambda function.
* **AWS ECS Fargate**:
	+ **Purpose**: Serverless computing for executing code without managing servers (useful for event-driven tasks).
	+ **Cost**: Billed based on **vCPU, memory** and **storage** resources consumed by your containerized applications running on AWS ECS Fargate (duration of execution).
	+ **Licensing**: Pay-per-use based on the **execution time** of the container application.
* **AWS Amplify**:
	+ **Purpose**: Serverless web application and mobile application hosting platform, easy to build, deploy and scale.
	+ **Cost**: Billed based on number of build minutes, monthly GB server and monthly GB stored.
	+ **Licensing**: Pay-per-use based on the **build, deploy and hosting** time of the application.
* **AWS Neptune**:
	+ **Purpose**: Serverless graph database for creating and storing Knowledge Graph.
	+ **Cost**: Billed based on **Neptune capacity units (NCU’s)** and **I/O rate**.
	+ **Licensing**: Pay-per-use based on the **I/O** operations.
* **Amazon VPC (Virtual Private Cloud)**:
	+ **Purpose**: Networking service to isolate resources in a private network.
	+ **Cost**: While VPC itself is **free**, there may be charges for components like **NAT gateways**, **VPN connections**, or **Data Transfer** across Availability Zones.
* **Amazon S3 (Simple Storage Service)**:
	+ **Purpose**: Used for storing application data, logs, and other assets.
	+ **Cost**: Billed based on storage size, retrieval operations, and data transfer. Charges vary by storage class (e.g., **S3 Standard**, **S3 Glacier**).
	+ **Licensing**: S3 is billed based on **storage** and **data transfer**, with pricing tiers available.
* **Amazon RDS (Relational Database Service)**:
	+ **Purpose**: Database service for hosting relational data (PostgreSQL).
	+ **Cost**: Billed based on instance type, storage, and data transfer.
	+ **Licensing**: RDS pricing depends on the database engine, instance type, and **provisioned storage**.
* **AWS IAM (Identity and Access Management)**:
	+ **Purpose**: Used to manage access control and define permissions for users and roles.
	+ **Cost**: IAM itself is **free**, but additional costs may apply for services such as **AWS Secrets Manager** or **AWS KMS** if used with IAM roles for encryption.
* **AWS Secrets Manager**:
	+ **Purpose**: Service for managing sensitive data, such as database credentials and API keys.
	+ **Cost**: Billed based on **secrets stored** and **API calls**.
	+ **Licensing**: The first **three secrets** are free, after which there are charges based on the number of secrets and API usage.
* **AWS KMS (Key Management Service)**:
	+ **Purpose**: Manages encryption keys for securing data at rest (e.g., S3, RDS).
	+ **Cost**: Billed based on **key usage** and **API requests**.
	+ **Licensing**: KMS is **pay-as-you-go**, with charges based on the number of keys and operations performed (e.g., encryption, decryption).

**Optional Billable Services**

These services are not strictly required but may be utilized to enhance the deployment, depending on customer preferences or specific use cases:

* **Amazon CloudWatch**:
	+ **Purpose**: Monitoring service for logs, metrics, and alarms.
	+ **Cost**: Billed based on the number of custom metrics, log data ingested, and the number of CloudWatch Alarms.
	+ **Licensing**: Charges for logs and custom metrics. **Free tier** available for basic monitoring.

**Cost Model and Licensing**

**Cost Model**

The deployment leverages a **pay-as-you-go** cost model, meaning customers are billed based on actual usage of AWS services. Key factors influencing costs include:

* **Compute resources**: Costs depend on the instance types, storage, and number of hours the instances are running.
* **Storage**: Charges are based on the volume of data stored (e.g., in S3 or EBS), along with data retrieval and transfer costs.
* **Data transfer**: Data transferred between AWS services within the same region is usually free, but data transferred across regions or out to the internet incurs charges.
* **Additional services**: Optional services like **CloudWatch** or **Lambda** are billed based on usage.

**Licensing Costs**

* **AWS License Included Options**: Some AWS services, such as Amazon RDS and Amazon EC2, offer **license-included** options, which include the software license as part of the EC2 or RDS instance cost.

**Cost Estimate Example:**

For a typical deployment, assuming the use of the following services:

* **2 EC2 instances** (r6g.2xlarge, r6g.large running 24/7)
* **100 GB storage** in **S3**
* **1 TB transfer** in/out of S3 per month
* **1 RDS instance** (db.t3.medium) for a database

The estimated monthly cost breakdown would look like:

| **Service** | **Monthly Estimate** |
| --- | --- |
| **Amazon EC2** | $450 |
| **Amazon S3** | $10 |
| **Amazon RDS** | $70 |
| **Data Transfer** | $20 |
| **AWS Neptune** | $465 |
| **AWS Fargate** | $580 |
| **AWS Lambda**  | $250 |
| **AWS CloudWatch (optional)** | $5 |

**Total Estimated Monthly Cost**: $1800 (subject to actual usage).

**TCO Link** - https://calculator.aws/#/estimate?id=2b9cee6af03e47603f50f11713bf2d9880cda5ea

# Provisioning Resources and Guidance for Type and Size Selection

In this section, we provide both **scripts** and **guidance** on selecting the appropriate resource types and sizes for the deployment. The below section provides guidance on selecting the appropriate AWS resources based on the deployment's requirements. Scripts are provided for provisioning resources quickly, while guidance on selecting the right EC2 instance types, RDS database sizes, and IAM roles helps ensure optimal performance and cost-efficiency.

By tailoring the resource selection based on workload and scalability needs, customers can deploy an effective and cost-efficient infrastructure.

**Provisioning AWS Resources Using Scripts**

 The CloudFormation template will deploy the following services:

* 3 IAM Policies: Deployment policy and Service policy
* 3 IAM Roles: Deployment Role and Service Role
* 1 VPC with 2 Public subnets and 2 Private subnets, with Internet and NAT gateway
* 2 EC2 instances with Application load balancer and auto scaling
* 1 RDS MySQL instance and 1 Neptune
* 9 lambdas
* 1 S3 bucket
* 3 ECS tasks
* 1 Cognito userpool
* Secrets manager

Download the CloudFormation template here to automatically deploy resources. Follow the next few steps to deploy the CloudFormation template.

1. Login to AWS console.
2. Go to Cloudformation console
3. Click on create stack and select with “New resources (Standard)”
4. Specify template: choose from the following 1. Upload a template 2. S3 URL
5. Configure stack details: Stack name,
6. Enter values for below parameters.
	* InstanceType
	* AMIID
	* KeyName
	* MinInstanceCount
	* MaxInstanceCount
	* RDSInstanceType
	* RDSDatabaseName
	* RDSUsername
	* RDSPassword
	* NeptuneUsername
	* NeptunePassword
	* NeptuneInstanceType
	* NeptuneClusterName
	* LambdaFunction1Name
	* LambdaFunction2Name
	* LambdaFunction3Name
	* LambdaFunction4Name
	* LambdaFunction5Name
	* LambdaFunction6Name
	* LambdaFunction7Name
	* LambdaFunction8Name
	* LambdaFunction9Name
	* ECSTasksDefinition1
	* ECSTasksDefinition2
	* ECSTasksDefinition3
	* S3BucketName
	* CognitoUserpoolName
	* SecretsManagerSecretName
7. Configure rollback triggers to revert changes in case of errors.
8. Set stack options: Tag, Permissions, Stack policy
9. Review and deploy
10. Monitor stack creation

**Guidance for Resource Type and Size Selection**

**EC2 Instance Size Selection**

EC2 Instance Recommendations for Neo4j

* For the Knowledge Accelerator - Plant GPT Neo4j instance, the recommended EC2 instance type is r6g.xlarge.
* This instance type provides 8 vCPUs, 64 GB of memory, and adequate compute and network performance for the expected workload.

EC2 Instance Recommendations for ChromaDB

* For the Knowledge Accelerator - Plant GPT ChromaDB instance, the recommended EC2 instance type is r6g.large.
* This instance type provides 2 vCPUs, 16 GB of memory, and adequate compute and network performance for the expected workload.

Determining Optimal Sizing

* To determine the appropriate resource sizing for your deployment, consider the following factors:
	+ Expected number of users and concurrent connections
	+ Anticipated data volume and growth rate
	+ Performance requirements for response times and throughput

**RDS and Neptune database Instance Size Selection**

RDS Instance Recommendations:

* For the Knowledge Accelerator - Plant GPT RDS instance, the recommended EC2 instance type is db.t3.micro.
* This instance type provides 2 vCPUs, 1 GB of memory, and adequate compute and network performance for the expected workload.

Neptune Recommendations:

* For the Knowledge Accelerator - Plant GPT Neptune instance, the recommended NCUs are 4.
* This provides 8 GB of memory, along with corresponding CPU for the expected workload.

Determining Optimal Sizing

* To determine the appropriate resource sizing for your deployment, consider the following factors:
	+ Expected number of users and concurrent connections
	+ Anticipated data volume and growth rate
	+ Performance requirements for response times and throughput

**S3 Storage Size**

* **Standard Storage**: For frequently accessed data 1TB.
* **Glacier Storage**: For archival or infrequently accessed data 1TB.

# Encryption

* **Encryption:**
	+ Neo4j Enterprise – Enable native file store encryption in neo4j.conf:

dbms.security.file\_storage.encryption.level=basic

dbms.security.file\_storage.encryption.key=<your-key>

* + Encrypted EBS volume (AES-256 by default, KMS-managed key) are used for ChromaDB and Neo4j EC2 instance.
	+ For Neptune, by default customer’s data is encrypted with a key that AWS owns and manages for customer