



Amazon Neptune

グラフデータ可視化ハンズオン

確認

本ワークショップでは
実際にNeptuneを使ったアプリケーションを体験していただく
ためAWSアカウントが必要です



Agenda

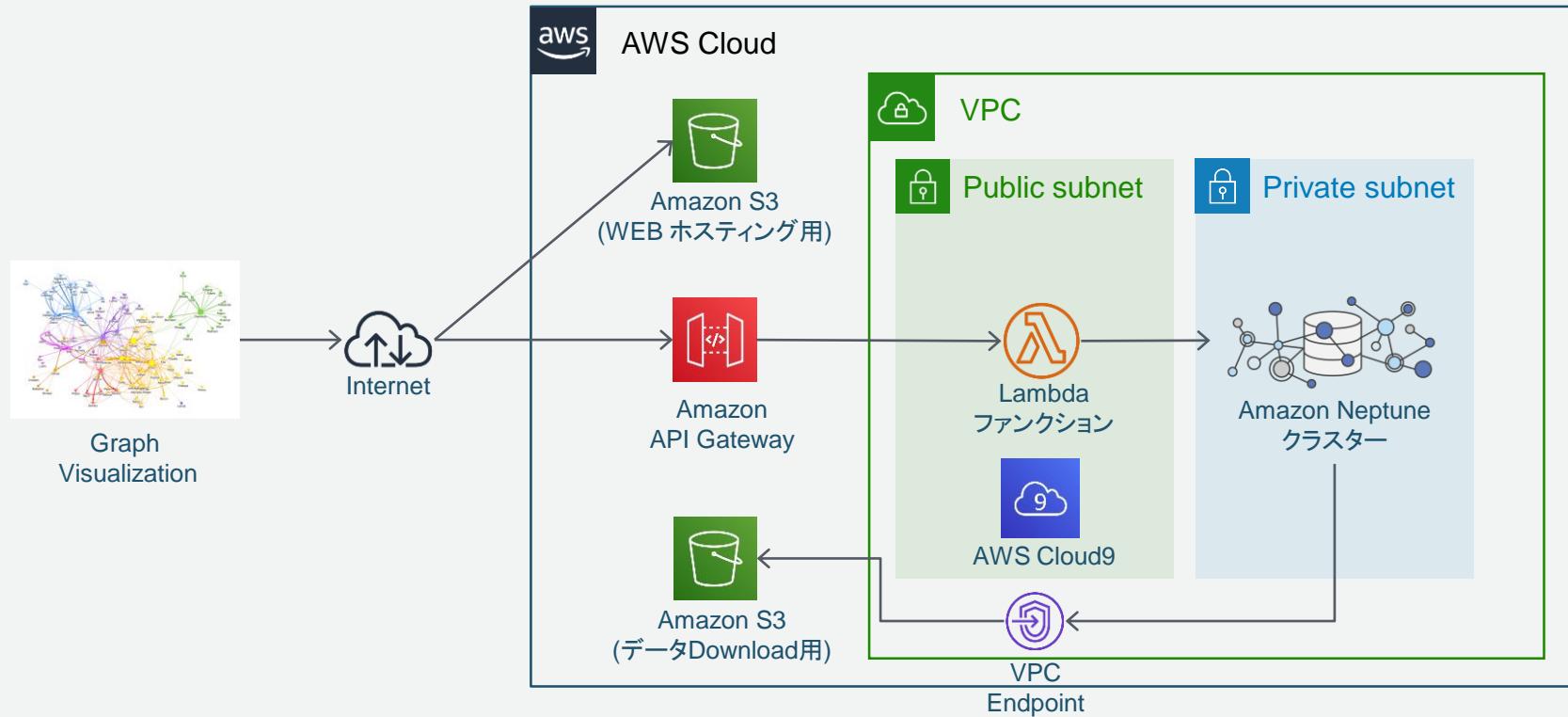
- ハンズオン環境作成
 - VPC及びサブネットの作成
 - AWS Cloud9の構築
- Amazon Neptune Cluster作成
- Amazon Neptuneにサンプルデータをロード
- AWS Lambda functionを作成・設定
- Amazon API Gateway - Proxy APIを作成・設定
- 静的WEBホスティング用のS3バケット設定
- S3バケットに静的ファイルアップロード



ハンズオン



これから作るもの

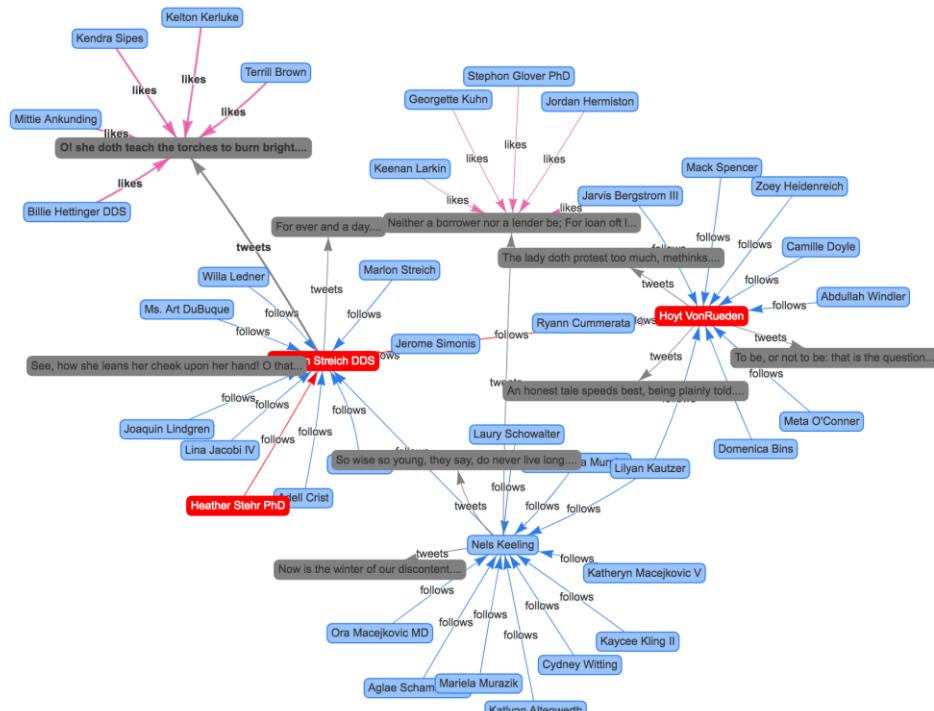


これから作るもの

Visualize Twitter data in Amazon Neptune using VIS.js library

Find Users: H

Go



ハンズオン環境作成

VPC及びサブネットの作成



AWSマネージメントコンソールにログインする

アカウント情報

The screenshot shows the AWS Management Console homepage. At the top left is the AWS logo. To its right are navigation links for 'サービス' (Services), 'リソースグループ' (Resource Groups), and a star icon. On the far right are icons for a bell (Notifications), a user profile (Profile), the region '東京' (Tokyo), and 'サポート' (Support). A red box highlights the AWS logo.

[ホームに戻る] ボタン [Home] button

AWS マネジメントコンソール AWS Management Console

リージョン Region

AWSのサービス AWS Services

サービスを検索する Search services

例 :Relational Database Service、データベース、RDS Example: Relational Database Service, Database, RDS

AWSサービスの検索ボックス AWS service search box

AWSの各サービス All AWS services

すべてのサービス All services

コンピューティング Computing

- EC2
- Lightsail
- ECR
- ECS
- EKS
- Lambda
- Batch
- Elastic Beanstalk
- Serverless Application Repository

開発者用ツール Developer Tools

- CodeStar
- CodeCommit
- CodeBuild
- CodeDeploy
- CodePipeline
- Cloud9
- X-Ray

機械学習 Machine Learning

- Amazon SageMaker
- Amazon Comprehend
- AWS DeepLens
- Amazon Lex
- Machine Learning
- Amazon Polly
- Rekognition
- Amazon Transcribe
- Amazon Translate
- Amazon Personalize
- Amazon Forecast
- Amazon Textract

モバイル Mobile

- AWS Amplify
- Mobile Hub
- AWS AppSync
- Device Farm

拡張現実 (AR) とバーチャルリアリティ (VR) AR and VR

- Amazon Sumerian

ストレージ Storage

ロボット工学 Robotics

- AWS RoboMaker

アプリケーション統合 Application Integration

外出先でリソースにアクセスする Access resources from anywhere

AWS コンソールモバイルアプリ Use the AWS Management Console mobile app to access the management console from your mobile device.

AWS を試す Try AWS

Amazon SageMaker Amazon SageMaker

すべての開発者とデータサイエンティストのための機械学習

Elasticsearch のディストリビューシ Elasticsearch distribution

フィードバック

日本語

言語

CloudFormation を開く

1. [東京]リージョン
になっていること確認

サービス リソースグループ 東京 サポート

2. [サービス] をクリック

CloudFormation

CloudFormation テンプレートによるリソースの作成と管理

EC2 AWS RoboMaker Athena Alexa for Business

Lightsail Amazon Managed Blockchain EMR Amazon Chime

ECR CloudSearch WorkMail

ECS Kinesis WorkMail

EKS QuickSight WorkSpaces

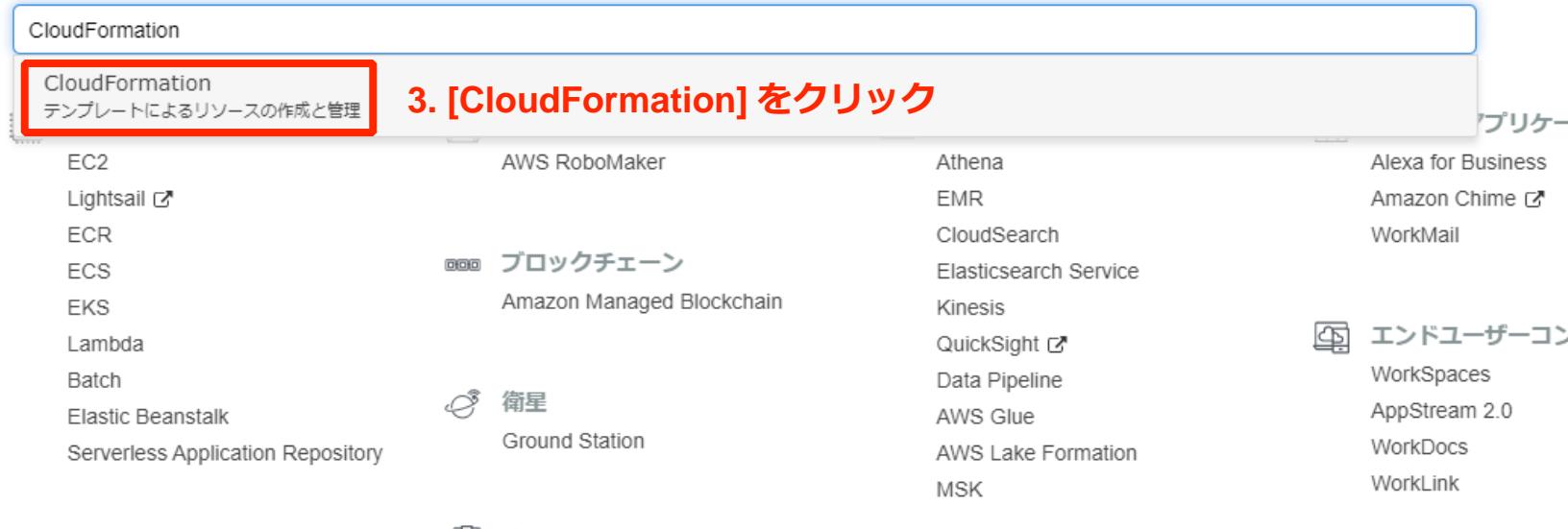
Lambda Data Pipeline AppStream 2.0

Batch AWS Glue WorkDocs

Elastic Beanstalk Ground Station AWS Lake Formation

Serverless Application Repository衛星 MSK WorkLink

3. [CloudFormation] をクリック



スタックの作成

[Stack creation] →
新しいリソースを使用（標準）をクリック

CloudFormation > スタック

Stack (0)

Stack name filter: 検索

アクティブ ネスト表示

Stack creation button (highlighted with a red box)

Stack details table headers: 名前, 状態, 作成時間, 説明

No stacks found message: Stack does not exist.
表示するStackがありません

Stack creation button (highlighted with a red box)

入门ガイドを見る

テンプレートを利用する

CloudFormation > スタック > スタックの作成

三
ステップ 1
テンプレートの指定

ステップ 2
スタックの詳細を指定

ステップ 3
スタックオプションの設定

ステップ 4
レビュー

スタックの作成

前提条件 - テンプレートの準備

テンプレートの準備
各スタックはテンプレートに基づきます。テンプレートとは、スタックに含む AWS リソースに関する設定情報を含む JSON または YAML ファイルです。

テンプレートの準備完了 サンプルテンプレートを使用 デザイナーでテンプレートを作成

テンプレートの指定

テンプレートは、スタックのリソースおよびプロパティを表す JSON または YAML ファイルです。

テンプレートソース
テンプレートを選択すると、保存先となる Amazon S3 URL が生成されます。

Amazon S3 URL テンプレートファイルのアップロード

Amazon S3 URL
`https://hamashi-sharing.s3-ap-northeast-1.amazonaws.com/step1_vpc.yaml`

S3 URL: URL を指定すると生成されます。

キャンセル

[Amazon S3 URL] に以下URLを入力して、次へ
https://hamashi-sharing.s3-ap-northeast-1.amazonaws.com/step1_vpc.yaml

スタックの詳細を指定し作成する

CloudFormation > スタック > スタックの作成

ステップ 1
テンプレートの指定

ステップ 2
スタックの詳細を指定

ステップ 3
スタックオプションの設定

ステップ 4
レビュー

[HandsOnName-UserName-vpc] が作成されるので
既に同名のVPCが存在している場合、パラメータを調整し
て同名にならないよう気をつける（日付等）

スタックの名前

スタックの名前

DevDayWSStep1

1. 任意のスタック名を入力

スタック名では、大文字および小文字 (A-Z～a-z)、数字 (0-9)、ダッシュ (-) を使用することができます。

パラメータ

パラメータは、テンプレートで定義されます。また、パラメータを使用すると、スタックを作成または更新する際にカスタム値を入力できます。

HandsOnName

handson

UserName

user1

2. 任意のユーザー名を入力

3. [次へ] を選択しこの後の画面でも同様に[次へ]を選択
最後の画面で[スタックの作成]を選択した後、スタックのステータスが
[CREATE_COMPLETE]になるのを確認する

キャンセル

戻る

次へ



CloudFormationのスタックの出力値をメモっておく

出力タブで出力されたStackの実行結果の値をメモ帳か何かに
メモしておいてください（後ほど使います）

キー	値	説明	エクスポート名
HandsOnVPC	vpc-06d0bf3427890d001	-	handson-user1-vpc
HandsOnVPCDefaultSG	sg-0108cce49513ab064	-	handson-user1-vpc-default-sg
PrivateSubnet1c	subnet-01009cb4ed4813285	-	handson-user1-private-1c
PrivateSubnet1d	subnet-0ad427a43ca7ebd5f	-	handson-user1-private-1d
PubSubnet1c	subnet-04a8186baf52cfb0c	-	handson-user1-public-1c

サブネットの確認

[サービス] から [VPC]を選択
左のナビゲーションから[サブネット]を選択し
3つのサブネットが作成されている事を確認

The screenshot shows the AWS VPC Subnet creation interface. On the left, a navigation sidebar lists VPC-related services: VPC ダッシュボード, VPC でフィルタリング:, Virtual Private Cloud, VPC, サブネット (which is selected and highlighted in orange), ルートテーブル, インターネットゲートウェイ, Egress Only インターネットゲートウェイ, and DHCP オプションセット. The main content area has tabs for "サブネットの作成" and "アクション". A search bar contains the text "vpc-06d0bf...". Below it, a table displays three subnets:

Name	サブネット ID	状態	VPC	IPv4 CIDR	アベイ
handson-user1-private-1c	subnet-01009cb4ed4813285	available	vpc-06d0bf3427890d001 ...	10.0.1.0/24	ap-no
handson-user1-private-1d	subnet-0ad427a43ca7ebd5f	available	vpc-06d0bf3427890d001 ...	10.0.2.0/24	ap-no
handson-user1-public-1c	subnet-04a8186baf52cfb0c	available	vpc-06d0bf3427890d001 ...	10.0.0.0/24	ap-no

A red box highlights the first row of the table, specifically the "Name" column and its corresponding subnet details.

ハンズオン環境作成

AWS Cloud9の構築



AWS Cloud9環境の構築

The screenshot shows the AWS Management Console interface. At the top left is the AWS logo, which is highlighted with a red box and has a red arrow pointing towards it from the top left corner. The top navigation bar includes 'サービス' (Services), 'リソースグループ' (Resource Groups), and user information 'hama @ awsj-hama' and '東京' (Tokyo). A search bar at the top center contains the placeholder text '名前または機能（EC2、S3 または VM、ストレージなど）でサービスを検索します。' (Search services by name or function (EC2, S3, or VM, Storage, etc.)). Below the search bar is a grid of service icons. A yellow callout box highlights the 'AWSマネジメントコンソールのメニューに移動' (Move to the AWS Management Console menu) link. In the bottom right corner of the main content area, another yellow callout box contains the text 'AWSサービス検索で"cloud9"と入力し、候補からCloud9を選択' (Type "cloud9" into the AWS Service search and select Cloud9 from the suggestions).

履歴

コンソールのホーム

VPC

ECS

EC2

CloudWatch

ECR

aws

サービス

リソースグループ

名前または機能（EC2、S3 または VM、ストレージなど）でサービスを検索します。

グループ A~Z

AWSマネジメントコンソールのメニューに移動

EC2 AWS RoboMaker Athena プリケーション

Lightsail EMR Alexa for Business

ECR CloudSearch Amazon Chime

ECS ブロックチェーン Elasticsearch Service WorkMail

AWS サービス

cloud

Cloud9
コードの記述、実行、デバッグのためのクラウド IDE

CloudFormation
テンプレートによるリソースの作成と管理

CloudWatch Metrics CloudTrail

Storage Gateway Config

AWS Backup

CloudWatch Events

Amazon Lex

CloudWatch Metrics CloudTrail

Storage Gateway Config

AWS Backup

CloudWatch Events

Amazon Lex

AWSサービス検索で"cloud9"と入力し、候補からCloud9を選択

aws

AWS Cloud9環境の構築

The screenshot shows the AWS Cloud9 console interface. On the left, there's a sidebar with 'Your environments' (Shared with you, Account environments), 'How-to guide', and 'Developer Tools' (AWS Cloud9: a cloud IDE for writing, running, and debugging code). The main area is titled 'Your environments (2)' and shows two environment cards. A large red arrow points to the 'Create environment' button at the top right of this section. A yellow callout box contains Japanese text: 'Cloud9のコンソールで[Create environment]をクリック'. Another yellow callout box at the bottom right contains Japanese text: '初めてのCloud9環境を構築する場合は、このような画面が表示される'. The bottom right corner features the AWS logo.

AWS Cloud9

Your environments

Shared with you

Account environments

How-to guide

Developer Tools

AWS Cloud9
a cloud IDE for writing,
running, and debugging
code

AWS Cloud9 allows you to write, run, and debug your code with just a browser. With AWS Cloud9, you have immediate access to a rich code editor, integrated debugger, and built-in terminal with preconfigured AWS CLI. You can get started in minutes and no longer have to spend the time to install local applications or configure your development machine.

AWS Cloud9 > Your environments

Your environments (2)

Open IDE View details Edit Delete Create environment

Cloud9のコンソールで[Create environment]をクリック

Type EC2 Permissions Owner

Type EC2 Permissions Owner

Description これは漢字のテストです。

New AWS Cloud9 environment

Create environment

初めてのCloud9環境を構築する場合は、このような画面が表示される

Open IDE

aws

AWS Cloud9環境の構築

Name environment

Environment name and description

Name
The name needs to be unique per user. You can update it at any time in your environment settings.

Limit: 60 characters

Description - *Optional*
This will appear on your environment's card in your dashboard. You can update it at any time in your environment settings.

Limit: 200 characters

Cancel **Next step**

Environment Nameに“{ユーザ名}-handson-env”と
入力し[Next step]をクリック

AWS Cloud9環境の構築

≡

Step 1
Name environment

Configure settings

Step 2
Configure settings

Step 3
Review

Environment settings

Environment type Info

Choose between creating a new EC2 instance for your new environment.

- Create a new instance for environment (EC2)
Launch a new instance in this region to run your new environment.
- Connect and run in remote server (SSH)
Display instructions to connect remotely over SSH and run your environment.

この後設定するネットワーク設定以外はDefaultの設定で問題なし

Instance type

- t2.micro (1 GiB RAM + 1 vCPU)
Free-tier eligible. Ideal for educational users and exploration.
- t3.small (2 GiB RAM + 2 vCPU)
Recommended for small-sized web projects.
- m5.large (8 GiB RAM + 2 vCPU)
Recommended for production and general-purpose development.
- Other instance type
Select an instance type.

t3.nano



Platform

- Amazon Linux
- Ubuntu Server 18.04 LTS

Cost-saving setting

Choose a predetermined amount of time to auto-hibernate your environment and prevent unnecessary charges. We recommend a hibernation settings of half an hour of no activity to maximize savings.

After 30 minutes (default)



AWS Cloud9環境の構築

Network settings(advanced)をクリックして詳細画面を開き、メモしておいたHandsOnVPCと、PubSubnet1cの値を指定し、[Next step]をクリック

▼ Network settings (advanced)

Network (VPC)
Launch your EC2 instances into an existing Amazon Private Cloud (VPC) or create a new one.

vpc-05c36055722adf30d ▼ C Create new VPC

Subnet
Select a subnet to launch your instances into. Subnets must be in the same VPC and Availability Zone as your instances.

subnet-00514330cb4a58c6a | Non-default in ap-northeast-1a ▼ C Create new subnet

Cancel Previous step Next step



AWS Cloud9環境の構築

AWS Cloud9 > Environments > Create environment

Step 1

Name environment

Step 2

Configure settings

Step 3

Review

Review

確認画面で[Create environment]をクリック

Environment name and settings

Name

handsoneenv

Description

hands on environment

We recommend the following best practices for using your AWS Cloud9 environment

- Use **source control** and **backup** your environment frequently. AWS Cloud9 does not perform automatic backups.
- Perform regular **updates of software** on your environment. AWS Cloud9 does not perform automatic updates on your behalf.
- Turn on **AWS CloudTrail** in your AWS account to track activity in your environment. [Learn more](#)
- Only share your environment with **trusted users**. Sharing your environment may put your AWS access credentials at risk. [Learn more](#)

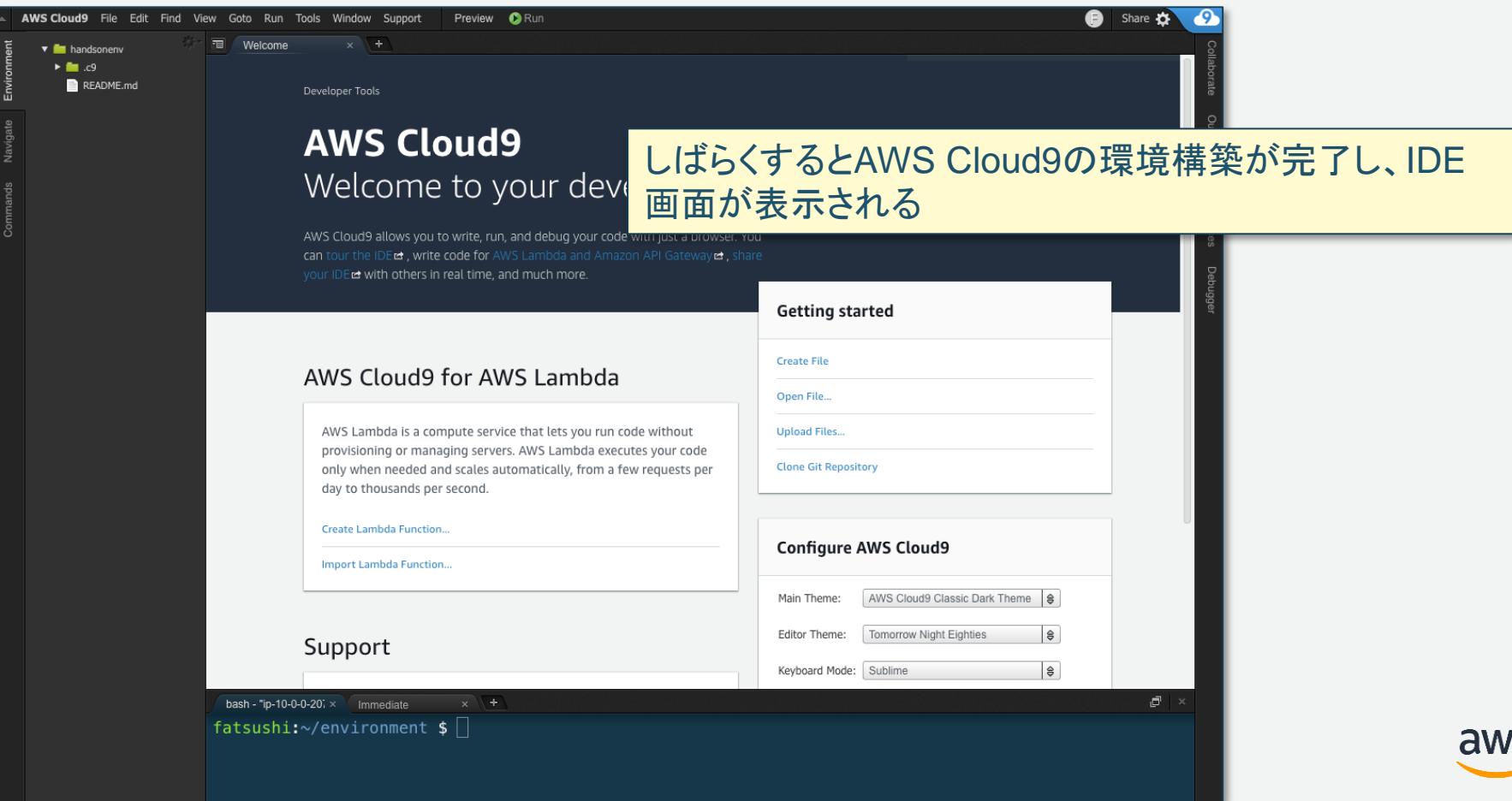
Cancel

Previous step

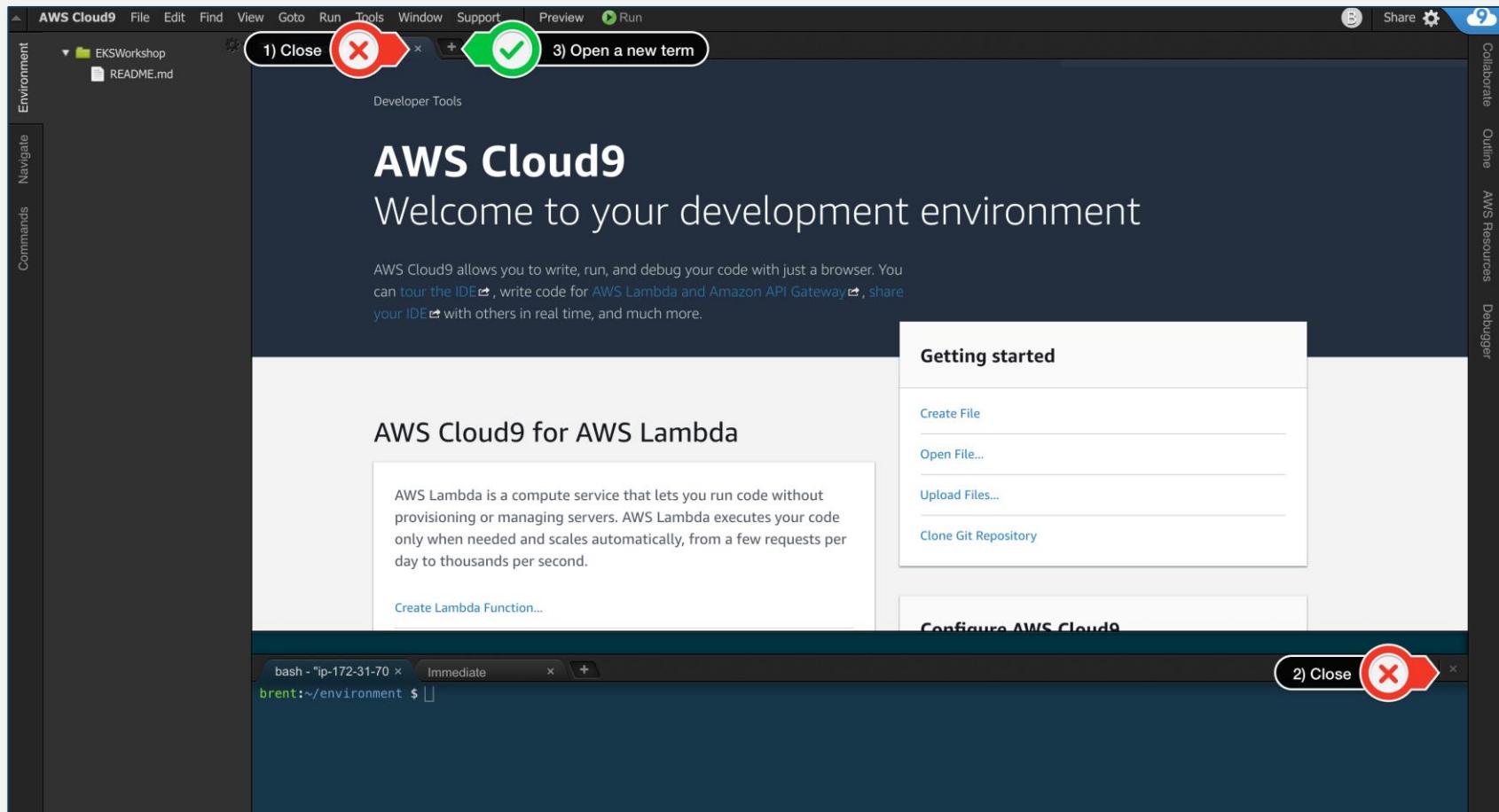
Create environment



AWS Cloud9環境の構築



新しくTerminalを開く



Amazon Neptune Cluster作成



IAMロール設定(1)

コンソール

Amazon NeptuneにアタッチするIAMロールを作成します。

Services -> IAM ^

The screenshot shows the AWS Identity and Access Management (IAM) service interface. On the left, a navigation pane lists various IAM management options like Access Management, Groups, Users, Policies, and Reports. The 'Roles' option is highlighted with a red box and has a red arrow pointing from the top-left towards it. At the top center, there's a blue button labeled 'ロールの作成' (Create Role) which is also highlighted with a red box. The main area displays a table of existing roles, with the first few rows visible:

ロール名	信頼されたエンティティ	最後のアクティビティ
AmazonSageMaker-ExecutionRole-2...	AWS サービス: sagemaker	なし
AmazonSageMaker-ExecutionRole-2...	AWS サービス: sagemaker	272 日間
amplifyconsole-backend-role	AWS サービス: amplify	200 日間
aws-codedstar-service-role	AWS サービス: codestar	なし
aws-elasticbeanstalk-ec2-role	AWS サービス: ec2	なし
aws-elasticbeanstalk-service-role	AWS サービス: elasticbeanstalk	なし
aws-quicksight-service-role-v0	AWS サービス: quicksight	なし
AWSServiceRoleForAccessAnalyzer	AWS サービス: access-analyzer (サービスにリンクされ...	今日
AWSServiceRoleForAmazonGuardDuty	AWS サービス: guardduty (サービスにリンクされ...	154 日間
AWSServiceRoleForAmazonMacie	AWS サービス: macie (サービスにリンクされ...	154 日間
AWS...	AWS サービス: ... (サービスにリンクされ...	201 日間



IAMロール設定(2)

ロールの作成

信頼されたエンティティの種類を選択

1 2 3 4

AWS サービス EC2, Lambda およびその他	別の AWS アカウント 新規登録またはサードパーティに登録されています	ウェブ ID Cognito または任意の OpenID プロバイダー	SAML 2.0 フェデレーション 企業アレクトリ
--------------------------------	---	--	------------------------------

AWS のサービスに対する操作の実行を許します。 詳細は[こちら](#)

このロールを使用するサービスを選択

EC2
Allows EC2 instances to call AWS services on your behalf.

Lambda
Allows Lambda functions to call AWS services on your behalf.

API Gateway	CodeBuild	EKS	KMS	RoboMaker
AWS Backup	CodeDeploy	EMR	Kinesis	S3
AWS Chatbot	CodeStar Notifications	ElasticCache	Lambda	SMS
AWS Support	Comprehend	Elastic Beanstalk	Lex	SNS
Amplify	Config	Elastic Container Service	License Manager	SWF
AppStream 2.0	Connect	Elastic Transcoder	Machine Learning	SageMaker
AppSync	DMS	Elastic Load Balancing	Macie	Security Hub
Application Auto Scaling	Data Lifecycle Manager	Forecast	MediaConvert	Service Catalog
Application Discovery Service	Data Pipeline	Global Accelerator	Migration Hub	Step Functions
Batch	DataSync	Glue	OpsWorks	Storage Gateway
Chime	DeepLens	Greengrass	Personalize	Textract
CloudFormation	Directory Service	GuardDuty	QLDB	Transfer
CloudHSM	DynamoDB	IAM Access Analyzer	RAM	Trusted Advisor
CloudTrail	EC2	Inspector	Rekognition	VPC
CloudWatch Application Insights	EC2 - Fleet	IoT	Rekognition	WorkLink
CloudWatch Events	EC2 Auto Scaling	IoT Things Graph	WorkMail	

ユースケースの選択

RDS
Allows RDS to perform operations using AWS resources on your behalf.

RDS - Add Role to Database
Allows you to grant RDS access to additional resources on your behalf.

RDS - Beta
Allows RDS to perform operations using AWS resources on your behalf in the Beta region.

RDS - CloudHSM
Allows RDS to manage CloudHSM resources on your behalf.

RDS - Directory Service
Allows RDS to manage Directory Service resources on your behalf.

* 必須

キャンセル 次のステップ: アクセス権限

Roleタイプで
AWS サービス>
RDS >
RDS - Add Role to Databaseを選択してから
次のステップ: アクセス権限を選択



IAMロール設定(3)

ロールの作成

▼ Attach アクセス権限ポリシー

新しいロールにアタッチするポリシーを 1 つ以上選択します。

ポリシーの作成

ポリシーのフィルタ ▾ Q s3 6 件の結果を表示中

ポリシー名	次として使用
<input type="checkbox"/> AmazonDMSRedshiftS3Role	なし
<input type="checkbox"/> AmazonS3FullAccess	Permissions policy (6)
<input checked="" type="checkbox"/> AmazonS3ReadOnlyAccess	なし
<input type="checkbox"/> AWSLambdaS3ExecutionRole-5615287b-f675-4884-8c12-e70ab57acc6d	なし
<input type="checkbox"/> AWSQuickSightS3Policy	Permissions policy (1)
<input type="checkbox"/> QuickSightAccessForS3StorageManagementAnalyticsReadOnly	Permissions policy (1)

▶ アクセス権限の境界の設定

* 必須 キャンセル 戻る 次のステップ: タグ

フィルタでS3と入力し、AmazonS3ReadOnlyAccess
ポリシーにチェックをつけ、「次のステップ：タグ」
へ

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

IAMロール設定(4)

Create role

Add tags (optional)

IAM tags are key-value pairs you can add to your role. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this role. [Learn more](#)

Key	Value (optional)	Remove
Add new key		

You can add 50 more tags.

Cancel Previous **Next: Review**

確認

以下に必要な情報を指定してこのロールを見直すから、作成してください。

ロール名* hamashi-neptune-role

ロールの説明 Allows you to grant RDS access to additional resources on your behalf.

最大 1000 文字。英数字と「+=, @_」を使用します。

信頼されたエンティティ AWS のサービス: rds.amazonaws.com

ポリシー AmazonS3ReadOnlyAccess

アクセス権限の境界 アクセス権限の境界が設定されていません

追加されたタグはありません。

* 必須 キャンセル 戻る **ロールの作成**

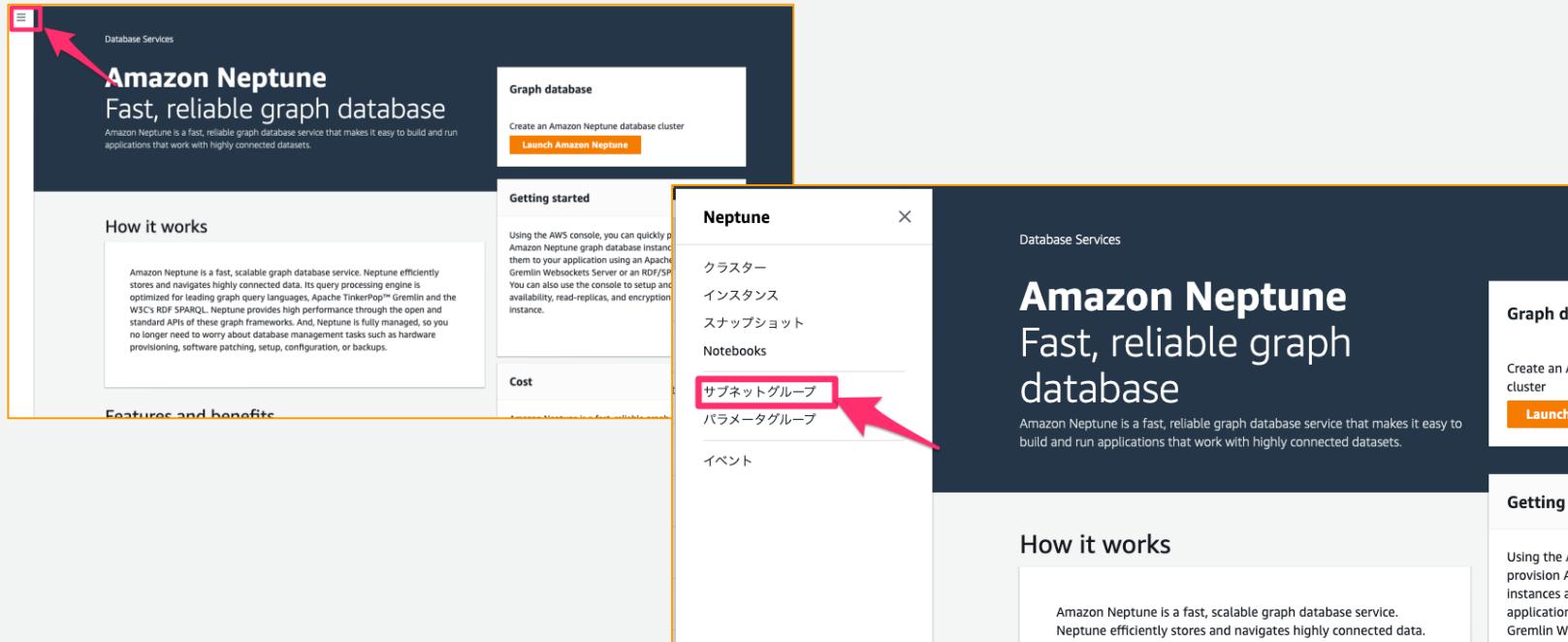


Neptune用サブネットグループ設定(1)

Amazon Neptuneのサブネットグループを作成する

Neptune
サブネットグループ

Services -> Neptune ^



Neptune用サブネットグループ設定(1)

Amazon Neptuneのサブネットグループを作成する

The screenshot shows the Amazon Neptune Subnet Groups management interface. On the left, a sidebar menu lists 'Neptune' (selected), 'クラスター', 'インスタンス', 'スナップショット', 'Notebooks', 'サブネットグループ' (highlighted in orange), 'パラメータグループ', and 'イベント'. The main content area is titled 'サブネットグループ (4)' and displays a table of existing subnet groups. The table columns are '名前' (Name), '説明' (Description), 'ステータス' (Status), and 'VPC'. The status column shows green checkmarks for all entries. A red box highlights the 'DB サブネットグループの作成' (Create DB Subnet Group) button at the top right of the table header. Below the table, there is a search bar labeled 'フィルタ サブネットグループ'.

名前	説明	ステータス	VPC
default	default	完了	vpc-c73508a2
nestle_rds_sgr	nestle_rds_sgr	完了	vpc-61c11e06
rdssubnetgroup	rdsSubnetGroup	完了	vpc-47e7fa22
wp-subnetgp	wp-subnetgp_20161118_075524	完了	vpc-022ae666



Neptune用サブネットグループ設定(2)

Amazon Neptuneのサブネットグループを作成する

DB サブネットグループの作成

新しいサブネットグループを作成するには、名前と説明を入力し、既存の VPC を選択します。次に、その VPC に関連するサブネットを追加できます。

サブネットグループの詳細

名前
サブネットグループの作成後に名前を変更することはできません。

説明

VPC
DB サブネットグループに使用するサブネットに対応する VPC 識別子を選択します。サブネットグループの作成後に別の VPC 識別子を選択することはできません。

1.任意のDBサブネットグループ名および説明を入力
例：neptune-sub-group

2.最初にメモっておいたCloudFormationで作成した
VPCのIDを選択

Neptune用サブネットグループ設定(3)

Amazon Neptuneのサブネットグループを作成する

■ サブネットの追加

サブネットをこのサブネットグループに追加します。サブネットを1つずつ追加することも、このVPCに関連するすべてのサブネットを追加することもできます。このグループの作成後、追加/編集ができます。最低で2つのサブネットが必要です。

このVPCに関連するすべてのサブネットを追加します

アベイラビリティーゾーン
ap-northeast-1d

サブネット
subnet-0ad427a43ca7ebd5f (10.0.2.0/24)

サブネットを追加します

このサブネットグループのサブネット (2)

アベイラビリティーゾーン	サブネット ID	CIDR ブロック	アクション
ap-northeast-1d	subnet-0ad427a43ca7ebd5f	10.0.2.0/24	削除
ap-northeast-1c	subnet-01009cb4ed4813285	10.0.1.0/24	削除

キャンセル 作成

3.最初にメモっておいたCloudFormationで作成したPrivateSubnetを2つサブネットグループに追加する

[注意]サブネットを追加するボタンをサブネットをグループに追加する毎、毎回押下する必要があります。

2.作成ボタンを押下して、サブネットグループを作成



Neptuneクラスター作成(1)

Amazon Neptuneのクラスターを作成する

The screenshot shows the Amazon Neptune console interface. On the left, there is a sidebar with the following menu items:

- データベース (highlighted with a red box)
- スナップショット
- Notebooks
- サブネットグループ
- パラメータグループ

The main area is titled "Neptune > データベース" and contains a table with the following columns:

DB 識別子	ロール	エンジン	リージョンと AZ	サイズ	ステータス	CPU

At the top right of the main area, there is a row of buttons: "クローンソース" (Clone Source), "変更" (Change), "アクション" (Action), and a large orange button labeled "データベースの作成" (Create Database) which is also highlighted with a red box.

Neptuneクラスター作成(2)

Amazon Neptuneのクラスターを作成する

ステップ1
DB 詳細の指定

ステップ2
[詳細設定] の設定

Neptune > データベース > データベースの作成

DB 詳細の指定

インスタンスの仕様

Neptune does not have a free tier. On-demand instances let you pay for your database by the hour with no long-term commitments or upfront fees. See the [Neptune pricing page](#) for complete details.

DB エンジン
neptune

DB エンジンのバージョン [info](#)
Neptune-1.0.1.0.200502.0

DB インスタンスのクラス [info](#)
db.r5.large — 2 vCPU, 16 GiB RAM

Enable high availability (Multi-AZ) [info](#)

Create read replica in different zone
 いいえ

設定

DB インスタンス識別子 [info](#)
AWS アカウントが現在のリージョンで所有しているすべての DB インスタンスにおいて一意な名前を指定します。

hama-neptune-hanson

DB instance identifier is case insensitive, but stored as all lower-case, as in "myinstance".
Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens.
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.

キャンセル 次へ

1.DBインスタンスのバージョンは
200502.0、クラスはdb.r5.largeを選択

2.マルチAZ構成を有効化

3.任意のDBインスタンス識別子を入力
例：{ユーザ名}-neptune-hanson

Neptuneクラスター作成(3)

Amazon Neptuneのクラスターを作成する

ステップ1
DB 詳細の指定

ステップ2
[詳細設定] の設定

Neptune > データベース > データベースの作成

[詳細設定] の設定

ネットワーク & セキュリティ

Virtual Private Cloud (VPC) info

VPC は、この DB インスタンスの仮想ネットワーク環境を定義します。

hands-on-user1-vpc (vpc-06d0bf3427890d001)



対応する DB サブネットグループがある VPC のみが表示されます。

サブネットグループ info

選択した VPC で DB インスタンスが使用できるサブネットと IP 範囲を定義する DB サブネットグループ。

neptune-sub-group

アベイラビリティーゾーン info

指定なし

VPC セキュリティグループ

セキュリティグループには、DB インスタンスにアクセスする必要があるすべての EC2 インスタンスとデバイスからの接続を許可するルールがあります。

- 新規の VPC セキュリティグループを作成
- 既存の VPC セキュリティグループの選択

CloudFormationで作成したVPCを選択し、先ほど作成したサブネットグループを選択する。
残りはデフォルトの値のままで、「データベースを作成」ボタンを押下。
※作成完了まで10分ほどかかります。

削除保護

削除保護の有効化

データベースが誤って削除されるのを防ぎます。このオプションが有効になっていない場合、データベースを削除することはできません。

キャンセル

戻る

データベースの作成

Neptuneクラスター作成(4)

Amazon Neptuneのクラスターを作成する

The screenshot shows two pages from the Amazon Neptune console:

- Cluster Page:** Shows one cluster named "hama-neptune-hanson-cluster". The status column for this cluster is highlighted with a red box and contains the text "利用可能" (Available).
- Instances Page:** Shows two instances: "hama-neptune-hanson" and "hama-neptune-hanson-ap-northeast-1c". Both instances have their status listed as "利用可能" (Available) in the status column.

クラスター、インスタンスそれぞれで利用可能ステータスになっているかを確認してください。



IAM Role付与(1)

The screenshot shows the AWS Neptune console. On the left, a sidebar menu lists 'Neptune' (selected), 'データベース' (highlighted with a red box), 'スナップショット', 'Notebooks', 'サブネットグループ', 'パラメータグループ', and 'イベント'. In the main area, the 'データベース' tab is selected, showing a list of databases. A red box highlights the first database entry, 'hkamed20200305', which is a cluster. Another red box highlights the 'アクション' button at the top right of the table header. The table columns include 'DB 識別子', 'ロール', 'エンジン', 'リージョンと AZ', 'サイズ', 'ステータス', and 'CPU'.

DB 識別子	ロール	エンジン	リージョンと AZ	サイズ	ステータス	CPU
hkamed20200305	クラスター	Neptune	ap-northeast-1	-	利用可能	-
hkamed20200305	書き込み	Neptune	ap-northeast-1d	db.r5.large	利用可能	4.00
hkamed20200305-ap-northeast-1a	読み込み	Neptune	ap-northeast-1a	db.r5.large	利用可能	4.00

作成したデータベースにIAM Roleを付与するため、
クラスターにチェックをつけて、アクションより、
「IAMロールの管理」を押下



IAM Role付与(2)

Neptune > データベース > IAM ロールの管理

IAM ロールの管理

IAM ロールの管理

更新

クラスターに IAM ロールを追加 [info](#)

AWS Service Role For RDS

ロールの追加

クラスターの現在の IAM ロール (1)

ロール	状況
hamashi-neptune-role	適用されています

完了

クラスターにIAMロールを追加よりあらかじめ作成しておいたIAMロールをNeptuneクラスターに追加する。
最後に完了ボタンを押下（削除ボタンが灰色で回っているのは無視してOKです）

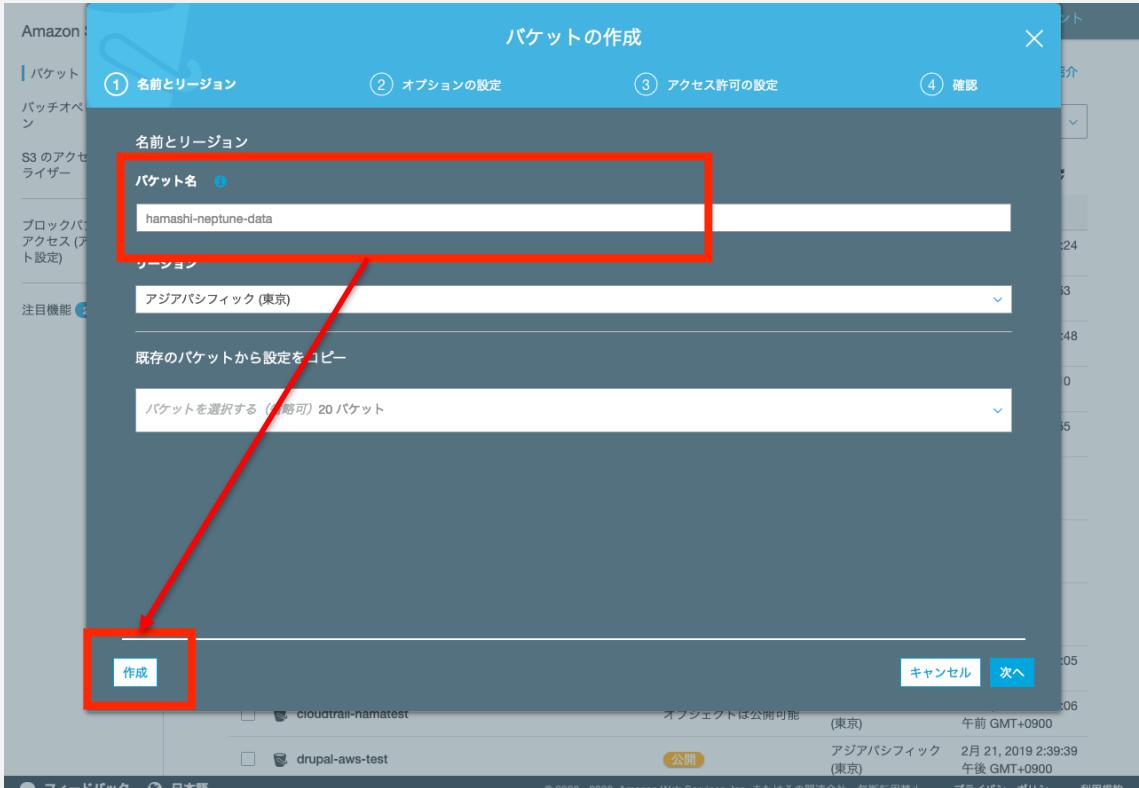


Amazon Neptuneに サンプルデータをロード



サンプルデータ用にS3バケットを新規作成

Services -> S3 へ



バケットを作成より、
バケット名を以下のように入力し
て、作成ボタンを押下する。

{ユーザ名}-neptune-data

設定は全てデフォルト

NeptuneからS3への接続のため VPCエンドポイントを作成(1)

Services -> VPC へ

VPC ダッシュボード

VPC でフィルタリング:

VPC の選択

Virtual Private Cloud

VPC

サブネット

ルートテーブル

インターネットゲートウェイ

Egress Only インターネットゲートウェイ

DHCP オプションセット

Elastic IP

エンドポイント

エンドポイントのサービス

NAT ゲートウェイ

アラートとメトリクス

エンドポイントの作成

アクション ▾

タグ/属性によるフィルター、またはキーワードによる検索

Name	エンドポイント ID	VPC ID	サービス名	エンドポイントタイプ	ステータス
	vpce-49b04520	vpc-c73508a2 d...	com.amazonaws.ap-northeast-1.s3	Gateway	使用可能
	vpce-7eb04517	vpc-47e7fa22 te...	com.amazonaws.ap-northeast-1.s3	Gateway	使用可能

NeptuneからS3への接続のため VPCエンドポイントを作成(2)

エンドポイントの作成

VPC エンドポイントを使用して、ご使用の VPC を他のサービスへ安全に接続できます。
インターフェイスエンドポイントには、[PrivateLink](#) が搭載されており、Elastic Network Interface (ENI) をサービス宛てのトラフィックのエントリーポイントとして使用します。
ゲートウェイエンドポイントは、サービスに対するトラフィックのルートテーブル内のルートのターゲットとして機能します。

- サービスカテゴリ AWS サービス
 サービスを名前で検索
 ご使用の AWS Marketplace サービス

サービス名 com.amazonaws.ap-northeast-1.s3 

サービス名	所有者	タイプ
com.amazonaws.ap-northeast-1.logs	amazon	Interface
com.amazonaws.ap-northeast-1.monitoring	amazon	Interface
com.amazonaws.ap-northeast-1.qldb.sess...	amazon	Interface
com.amazonaws.ap-northeast-1.rds.gateway	amazon	Interface
com.amazonaws.ap-northeast-1.s3	amazon	Gateway
com.amazonaws.ap-northeast-1.sagemak...	amazon	Interface
com.amazonaws.ap-northeast-1.sagemak...	amazon	Interface
com.amazonaws.ap-northeast-1.secretsm...	amazon	Interface
com.amazonaws.ap-northeast-1.serviceca...	amazon	Interface
com.amazonaws.ap-northeast-1 sns	amazon	Interface
com.amazonaws.ap-northeast-1.sqs	amazon	Interface
com.amazonaws.ap-northeast-1.ssm	amazon	Interface
com.amazonaws.ap-northeast-1.ssmmess...	amazon	Interface
com.amazonaws.ap-northeast-1.states	amazon	Interface
com.amazonaws.ap-northeast-1.vpc	amazon	Interface

1. [Service Name] (サービス名)
com.amazonaws.ap-northeast-1.s3 を選択します。

NeptuneからS3への接続のため VPCエンドポイントを作成(3)



2. Neptune DB インスタンスを含む VPC を選択し、Netuneクラスターに関連するサブネットに関連付けられているルートテーブルの横にあるチェックボックスをオンにします。

3. エンドポイントの作成ボタンを押下します。



Cloud9からNeptuneへの接続許可(1)

Services -> EC2 へ

1. メモってあるVPCのIDでフィルタリングし、Cloud9と名前に含まれているインスタンスをチェック

Name	グループ ID	グループ名	VPC ID
aws-cloud9-hama-handson-env-5963284c955a47a7b3559da578f65...	sg-095278ecad95a2f1f	aws-cloud9-hama-handson-...	vpc-06d0bf3427890d001
	sg-0a632012fb7e74eab	rds-launch-wizard	vpc-06d0bf3427890d001

セキュリティグループ: sg-095278ecad95a2f1f

説明 インバウンド アウトバウンド タグ

グループ名: aws-cloud9-hama-handson-env-5963284c955a47a7b3559da578f65845-InstanceSecurityGroup-FDWETU14UEKVY
グループの説明: Security group for AWS Cloud9 environment aws-cloud9-hama-handson-env-5963284c955a47a7b3559da578f65845
グループ ID: sg-095278ecad95a2f1f
VPC ID: vpc-06d0bf3427890d001

2. Cloud9のインスタンスのセキュリティグループIDをメモしておく

Cloud9からNeptuneへの接続許可(2)

The screenshot shows two parts of the AWS VPC console. The top part is a list of security groups within a VPC, with one group selected and highlighted by a red box. The bottom part is a detailed view of the selected security group, showing its inbound rules.

セキュリティグループ: sg-0a632012fb7e74eab

説明 インバウンド アウトバウンド タグ

編集

タイプ (i) プロトコル (i) ポート範囲 (i) ソース (i) 説明 (i)

カスタム TCP ルール TCP 8182 101.143.41.115/32

3. rds-launchで始まるグループ名のセキュリティグループのチェックをOnにして、インバウンドを編集する

Cloud9からNeptuneへの接続許可(3)

インバウンドのルールの編集

タイプ	プロトコル	ポート範囲	ソース	説明
カスタム TCP	TCP	8182	カスタム 101.143.41.115/32	例: SSH for Admin Desktop
カスタム TCP	TCP	8182	カスタム sg-095278ecad95a2f1f	from cloud9

注意: 既存のルールを編集すると、編集したルールが削除されて、新しい詳細を含む新しいルールが作成されます。これにより、そのルールに依存するトラフィックは、新しいルールが作成されるまで非常に短時間切断されます。

4. ルールを追加し、Cloud9のセキュリティグループから、カスタムTCP 8182ポートでアクセスできるように設定しておく

保存

タイプ	プロトコル	ポート範囲	ソース	説明
カスタム TCP ルール	TCP	8182	101.143.41.115/32	



作成したS3バケットにサンプルデータのCSVをアップロード

Cloud9 terminal ^

```
# ワークディレクトリに移動
$ cd ~/environment
# Neptune用のサンプルコードをclone
$ git clone https://github.com/aws-samples/amazon-neptune-samples.git
# Neptune用のサンプルデータのディレクトリに移動
$ cd amazon-neptune-samples/gremlin/visjs-neptune/sampleddata/
# S3バケットにサンプルデータのCSVをアップロード（先ほど作ったデータ用のS3バケット名で置き換える）
$ aws s3 cp . s3://bucket-name --recursive
```

S3からNeptuneにデータをロードする(1)

Cloud9 terminal ^

```
# ワークディレクトリに移動
$ cd ~/environment
# Neptune ローダーを実行 (エンドポイント、Amazon S3 パス、リージョン、iamRoleArnの値を置き換え)
$ curl -X POST \
-H 'Content-Type: application/json' \
https://{{your-neptune-endpoint}}:8182/loader -d '
{
  "source" : "s3://{{bucket-name}}/",
  "format" : "csv",
  "iamRoleArn" : "{{NeptuneクラスターにアタッチしたIAM ロールのARN}}",
  "region" : "{{リージョンコード ※東京ならap-northeast-1}}",
  "failOnError" : "FALSE",
  "parallelism" : "MEDIUM",
  "updateSingleCardinalityProperties" : "FALSE"
}'
```



S3からNeptuneにデータをロードする(2)

```
## 先ほどのcURLの結果
{
  "status" : "200 OK",
  "payload" : {
    "loadId" : "2a90dce6-4337-4105-b34a-4577861e091d"
  }
}
## 非同期実行のステータス確認("LOAD_COMPLETED" : 7になっていればOK)
$ curl -G 'https://:{your-neptune-endpoint}:8182/loader/{上記のloadId}'
{
  "status" : "200 OK",
  "payload" : {
    "feedCount" : [
      {
        "LOAD_COMPLETED" : 7
      }
    ],
  }
},
```



AWS Lambda functionを作成・ 設定



IAMロール設定(1)

Lambda ファンクションにアタッチするIAMロールを作成します。

コンソール

Services -> IAM ^

The screenshot shows the AWS IAM Roles page. On the left, there is a navigation sidebar with various options like Access Management, Groups, Users, Policies, and Reports. The 'Roles' option is highlighted with a red box and has a red arrow pointing from the top-left towards it. At the top center, there is a blue button labeled 'Roleの作成' (Create Role) which is also highlighted with a red box. Below the button is a search bar with the placeholder '検索' (Search). The main area displays a table with 41 results, showing columns for 'Role Name', 'Trusted Entity', and 'Last Activity'. Each row lists a role name, its trusted entity (e.g., AWS Service: sagemaker), and the last activity status (e.g., 'なし' for 'no activity').

Role Name	信頼されたエンティティ	最後のアクティビティ
AmazonSageMaker-ExecutionRole-2...	AWS サービス: sagemaker	なし
AmazonSageMaker-ExecutionRole-2...	AWS サービス: sagemaker	272 日間
amplifyconsole-backend-role	AWS サービス: amplify	200 日間
aws-codedstar-service-role	AWS サービス: codestar	なし
aws-elasticbeanstalk-ec2-role	AWS サービス: ec2	なし
aws-elasticbeanstalk-service-role	AWS サービス: elasticbeanstalk	なし
aws-quicksight-service-role-v0	AWS サービス: quicksight	なし
AWSRoleForAccessAnalyzer	AWS サービス: access-analyzer (サービスにリンクされ...	今日
AWSRoleForAmazonGuardDuty	AWS サービス: guardduty (サービスにリンクされ...	154 日間
AWSRoleForAmazonMacie	AWS サービス: macie (サービスにリンクされ...	154 日間
AWSRoleForAWSLambda	AWS サービス: lambda (サービスにリンクされ...	204 日間



IAMロール設定(2)

ロールの作成

信頼されたエンティティの種類を選択



AWS のサービスによるアクションの代行を許可します。 [詳細はこちら](#)

このロールを使用するサービスを選択

EC2

Allows EC2 instances to call AWS services on your behalf.

Lambda

Allows Lambda functions to call AWS services on your behalf.

API Gateway	CodeBuild	EKS	KMS	RoboMaker
AWS Backup	CodeDeploy	EMR	Kinesis	S3
AWS Chatbot	CodeStar Notifications	ElastiCache	Lambda	SMS
AWS Support	Comprehend	Elastic Beanstalk	Lex	SNS
Amplify	Config	Elastic Container Service	License Manager	SWF
AppStream 2.0	Connect	Elastic Transcoder	Machine Learning	SageMaker
AppSync	DMS	Elastic Load Balancing	Macie	Security Hub
Application Auto Scaling	Data Lifecycle Manager	Forecast	MediaConvert	Service Catalog
Application Discovery Service	Data Pipeline	Global Accelerator	Migration Hub	Step Functions
Batch	DataSync	Glue	OpsWorks	Storage Gateway
Chime	DeepLens	Greengrass	Personalize	Textract
CloudFormation	Directory Service	GuardDuty	QLDB	Transfer
CloudHSM	DynamoDB	IAM Access Analyzer	RAM	Troubled Advisor
* 必須	EC2	Inspector	Rekognition	

キャンセル

次のステップ: アクセス権限

Roleタイプで
AWS サービス>
Lambda を選択してから
次のステップ: アクセス権限を選択



IAMロール設定(3)

ロールの作成

▼ Attach アクセス権限ポリシー

新しいロールにアタッチするポリシーを1つ以上選択します。

ポリシーの作成

ポリシーのフィルタ

Lambda

ポリシー名	次として使用
<input type="checkbox"/> AWSApplicationAutoscalingLambdaConcurrencyPolicy	なし
<input type="checkbox"/> AWSCodeDeployRoleForLambda	なし
<input type="checkbox"/> AWSDeepLensLambdaFunctionAccessPolicy	なし
<input checked="" type="checkbox"/> AWSLambdaBasicExecutionRole	Permissions policy (1)
<input type="checkbox"/> AWSLambdaBasicExecutionRole-df04b326-d250-484e-898f-5af462a848c2	Permissions policy (1)
<input type="checkbox"/> AWSLambdaDynamoDBExecutionRole	なし
<input checked="" type="checkbox"/> AWSLambdaENIManagementAccess	なし
<input type="checkbox"/> AWSLambdaExecute	なし

▶ アクセス権限の境界の設定

1 2 3 4

フィルタでLambdaと入力し、
AWSLambdaBasicExecutionRoleポリシーと
AWSLambdaENIManagementAccessポリシーに
チェックをつけ、「次のステップ：タグ」へ

* 必須

日本語

キャンセル

戻る

次のステップ: タグ

IAMロール設定(4)

Create role

Add tags (optional)

IAM tags are key-value pairs you can add to your role. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this role. [Learn more](#)

Key	Value (optional)	Remove
Add new key		

You can add 50 more tags.

Cancel Previous **Next: Review**

確認

以下に必要な情報を指定してこのロールを見直してから、作成してください。

ロール名* **hamashi-lambda-role**

英数字と「+=,@_」を使用します。最大 64 文字。

ロールの説明

Allows Lambda functions to call AWS services on your behalf.

最大 1000 文字。英数字と「+=,@_」を使用します。

信頼されたエンティティ AWS のサービス: lambda.amazonaws.com

ポリシー AWSLambdaBasicExecutionRole AWSLambdaENIManagementAccess

アクセス権限の境界 アクセス権限の境界が設定されていません。

* 必須 キャンセル 戻る **ロールの作成**

日本語 © 2008 - 2020, Amazon Web Services, Inc. またはその関連会社。無断転用禁止。 プライバシーポリシー

ロール名は**(ユーザ名)-lambda-role**にしてロールの作成を押下



Lambda functionを作成(1)

Cloud9 terminal へ

```
# ワークディレクトリに移動
$ cd ~/environment

# Neptune用のサンプルコードのディレクトリに移動
$ cd amazon-neptune-samples/gremlin/visjs-neptune/
# npm installで対象のnode moduleをインストール
$ npm install

# Lambda functionに必要なファイルをzipで固めておく
$ zip lambdapackage.zip -r node_modules/ indexLambda.js
```



Lambda functionを作成(1)

```
# 用意したパッケージをLambdaにデプロイする
$ aws lambda create-function --function-name <lambda-function-name> ¥
--role "<lambda-iam-role>" ¥
--runtime nodejs10.x --handler indexLambda.handler ¥
--description "Lambda function to make gremlin calls to Amazon Neptune" ¥
--timeout 120 --memory-size 256 --publish ¥
--vpc-config SubnetIds=<subnet-id>,SecurityGroupIds=<sec-group-id> ¥
--zip-file fileb://lambdapackage.zip ¥
--environment Variables="{NEPTUNE_CLUSTER_ENDPOINT=<your-neptune-cluster-
endpoint>,NEPTUNE_PORT=<your-neptune-db-port>}"
```

<lambda-function-name>: 任意の名前
<lambda-iam-role> : Lambda用に作ったIAM RoleのARN
<subnet-id>: Cloud9のインスタンスと同じPublic Subnet
<sec-group-id>: Cloud9のインスタンスと同じSecurity Group



Amazon API Gateway - Proxy

APIを作成・設定



REST API作成(1)

Services -> API Gateway

The screenshot shows the Amazon API Gateway console with the following interface elements:

- Header:** Amazon API Gateway APIs
- Title:** Networking & Content Delivery
- Main Title:** Amazon API Gateway
- Slogan:** create, maintain, and secure APIs at any scale
- Description:** Amazon API Gateway は開発者が Amazon EC2、AWS Lambda、またはパブリックにアクセス可能なウェブサービスで動作しているバックエンドシステムへの API を作成し、管理するのに役立ちます。Amazon API Gateway を使用すると、API のカスタムクライアント SDK を生成して、バックエンドシステムをモバイル、ウェブ、およびサーバーアプリケーションまたはサービスに接続できます。
- Section: Choose an API type**
 - HTTP API Beta**: Build low-latency and cost-effective REST APIs with built-in features such as OAuth2, and native CORS support.
Works with the following: Lambda, HTTP バックエンド
[Import] [Create]
 - WebSocket API**: Build a WebSocket API using persistent connections for real-time use cases such as chat applications or dashboards.
Works with the following: Lambda, HTTP, AWS サービス
[Create]
 - REST API**: Develop a REST API where you gain complete control over the request and response along with API management capabilities.
Works with the following: Lambda, HTTP, AWS サービス
[Import] [Create] **(This button is highlighted with a red box and arrow)**
 - REST API Private**: Create a REST API that is only accessible from within a VPC.
Works with the following: Lambda, HTTP, AWS サービス
[Import] [Create]

1. REST APIの構築ボタンを押下

REST API作成(2)

Amazon API Gateway API > 作成 すべてのヒントを表示 ?

プロトコルを選択する

REST API と WebSocket API のどちらを作成するかを選択します。

REST WebSocket

新しいAPIの作成

Amazon API Gateway では、API とは、HTTPS エンドポイントを通じて呼び出し可能なリソースおよびメソッドの集合体のことを示します。

新しい API Swagger あるいは Open API 3 からインポート API の例

2. 新しいAPIを作成します

名前と説明

API のフレンドリー名と説明を選択します。

API 名* lambda-neptune-proxy-api
説明 function in VPC accessing Amazon Neptune

エンドポイントタイプ リージョン

* 必須

3.
API名:lambda-neptune-proxy-api
説明: API Proxy for AWS Lambda function in VPC
accessing Amazon Neptune

→ API の作成



REST API作成(3)

4. アクションよりリソースの作成を押下

API: lambda-neptune...
リソース
カスタムドメイン名
リソース
ステージ
オーソライザー
ゲートウェイのレスポンス
モデル

アクション▼ 新しい子!

リソースのアクション
メソッドの作成
リソースの作成

API アクション
API のデプロイ
API のインポート
API ドキュメントの編集
API の削除

このページを使用して、リソースの新しい子リソースを作成します。

プロキシリソースとして設定する

リソース名* proxy

リソースパス* / {proxy+}

角括弧を使用してパスパラメータを追加できます。たとえば、リソースパス `{username}` は、"username" という名前のパスパラメータを表します。プロキシリソースとして `/{proxy+}` を設定すると、そのサブリソースへのすべてのリクエストがキャッチされます。これは例えば、`/foo` への GET リクエストとして機能します。`/` へのリクエストを処理するには、/ リソースで新しい ANY メソッドを追加します。

API Gateway CORS を有効にする

* 必須

キャンセル リソースの作成

5. プロキシリソースにチェックをつける

6. API Gateway CORSを有効にチェック

7. リソースの作成を押下してリソース作成

REST API作成(4)

•/{proxy+}-ANY-セットアップ

API Gateway は ANY メソッドをプロキシ統合として設定します。プロキシ統合では、HTTP エンドポイントまたは Lambda 関数と通信できます。API Gateway は、リソースパス、ヘッダー、クエリ文字列パラメータ、および本文を含むリクエスト全体を HTTP エンドポイントに送信します。Lambda 統合では、API Gateway はデフォルトマッピングをすべてのリクエスト情報に適用し、レスポンスはデフォルトインターフェイスに従います。詳細については、[ドキュメント](#)をご参照ください。

統合タイプ Lambda 関数プロキシ [i](#)
 HTTP プロキシ [i](#)
 VPC リンク [i](#)

Lambda リージョン

Lambda 関数
neptune-demo-function [i](#)

デフォルトタイムアウトの使用 [i](#)

保存

8. プロキシ先のLambda関数（先ほど作ったもの）を指定して、保存ボタンを押下

REST API作成(5)

The screenshot shows the AWS Lambda-Neptune proxy API configuration. On the left, there's a sidebar with options like 'カスタムドメイン名' (Custom Domain Name), 'ステージ' (Stage), 'オーソライザー' (Authorizer), etc. The main area shows an API resource path '/{proxy+}' with an 'ANY' method. A context menu is open over this method, with the 'API のデプロイ' (Deploy API) option highlighted with a red box.

9. アクションよりAPIのデプロイを押下

This dialog box allows selecting a deployment stage. It has fields for 'デプロイされるステージ' (Deployment Stage) set to '新しいステージ' (New Stage) and 'ステージ名*' (Stage Name) set to 'test'. Below these are optional fields for 'ステージの説明' (Stage Description) and 'デプロイメントの説明' (Deployment Description). At the bottom right is a large blue 'デプロイ' (Deploy) button.

10. 新しいステージとしてtestを指定して、デプロイボタンを押し、APIが実行可能な状態となる

静的WEBホスティング用の S3バケット設定



静的WEBホスティング用のS3バケットを作成

Cloud9

Cloud9 terminal ^

```
# ワークディレクトリに移動  
$ cd ~/environment
```

```
# create Amazon S3 bucket with public read access  
$ aws s3api create-bucket --bucket <bucket-name> --acl public-read --region <aws-region-code> --create-bucket-configuration LocationConstraint=<aws-region-code>
```

```
# configure website hosting on S3 bucket  
$ aws s3api put-bucket-website --bucket <bucket-name> --website-configuration '{ "IndexDocument": { "Suffix": "visualize-graph.html" }, "ErrorDocument": { "Key": "visualization-error.html" } }'
```

S3をPublic権限で作っているので、会社の
ポリシーでBlockされてる方は注意

S3バケットに 静的ファイルアップロード



HTMLに記載されているAPIのエンドポイントを修正

Cloud9

Cloud9 terminal ^

```
# ソースディレクトリに移動  
$ cd ~/environment/amazon-neptune-samples/gremlin/visjs-neptune/  
  
# 左メニューより、 ~/environment/amazon-neptune-samples/gremlin/visjs-neptune/visualize-graph.html  
を編集する
```

9. PROXY_API_URLの エンドポイントを作成し たAPI Gatewayのエンド ポイントに置き換えて ファイルを保存する

```
49      <script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>  
50      <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>  
51      <link rel="stylesheet" href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">  
52  
53  
54      <script type="text/javascript" src="vis.js"></script>  
55      <link href="vis-network.min.css" rel="stylesheet" type="text/css" />  
56  
57      <script type="text/javascript">  
58          var PROXY_API_URL = "https://vr6cx4jz7i.execute-api.eu-west-1.amazonaws.com/test";  
59          // var nodes = null;  
60          var edges = null;  
61          var network = null;  
62          var resp =null;  
63          var options =null;  
64  
65          var searchfill=null;
```

S3バケットに静的ファイルアップロード

Cloud9 terminalへ

```
# ソースディレクトリに移動
$ cd ~/environment/amazon-neptune-samples/gremlin/visjs-neptune/

# S3バケットに静的ファイルアップロード
$ aws s3 cp ./ s3://<bucket-name> --recursive --exclude "*" --include "vis*" --acl public-read
```

これで作業は完了です！

最後に作成したページにアクセスしてみましょう！

<http://<bucket-name>.s3-website.<aws-region-code>.amazonaws.com>



動作確認

Visualize Twitter data in Amazon Neptune using VIS.js library

 Find Users:

1. フィールドに「A」とか入力すると、
ユーザのレコメンドがされますので、ま
ず1ユーザを表示します

Visualize Twitter data in Amazon Neptune using VIS.js library

Find Users: Avis Mosciski

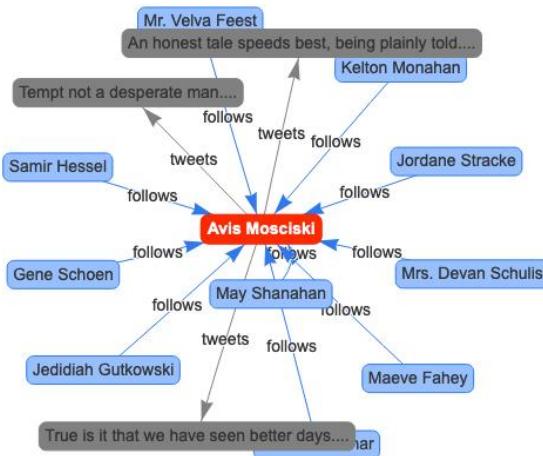
2. 表示されたユーザをクリックすると、
そのユーザから関係のあるツイートや
ユーザが引っ張られてきます

Avis Mosciski



動作確認

3. 今度は表示されたユーザやツイートをクリックすることでさらに、関連する情報がつながっていきます。Neptuneにグラフデータとして保存されたデータをLambdaから取得して可視化を繰り返しています



後始末



ハンズオンで利用したリソースを削除

- APIゲートウェイ
- Lambdaファンクション
- Cloud9
- Neptuneクラスター
- VPCエンドポイント
- S3バケット(データ、ホスト)
- CloudFormation stack
- CloudWatch ロググループ

これから順を追って削除していきます



API Gateway削除

The screenshot shows the AWS API Gateway console interface. On the left, there's a sidebar with various navigation options: API, カスタムドメイン名, API: lambda-neptune..., リソース (selected), ステージ, オーソライザ, ゲートウェイのレスポンス, モデル, リソースポリシー, ドキュメント, ダッシュボード, 設定, 使用量プラン, API キー, クライアント証明書, and VPC リンク. The main area has tabs for リソース (Resources) and アクション (Actions). Under Resources, the path is / /{prox} ANY OPTIO. A context menu is open over the ANY method, listing: リソースのアクション (Resource Actions), メソッドの作成 (Create Method), リソースの作成 (Create Resource), CORS の有効化 (Enable CORS), リソースドキュメントの編集 (Edit Resource Document), API アクション (API Actions), API のデプロイ (Deploy API), API のインポート (Import API), API ドキュメントの編集 (Edit API Document), and API の削除 (Delete API). The "API の削除" option is highlighted with a red box and a red arrow pointing to it from below.



Lambdaファンクション削除

Lambda > 関数

関数 (2)

関数名	説明	ランタイム	最終更新日時
neptune-demo-function	Lambda function to make gremlin calls to Amazon Neptune	Node.js 10.x	76.5 kB 1 時間前
lambda-http-get-hama	lambda-http-get-hama	nodejs4.3	639 bytes 3 年前

アクション ▲ 関数の作成

詳細を表示 テスト 削除



aws

Cloud9削除

AWS Cloud9 > Your environments

Your environments (1)

Open IDE View details Edit Delete Create environment

hama-handson-env

Type: EC2 Permissions: Owner

Description: No description available

削除前にNeptuneに紐づくセキュリティグループから、Cloud9のセキュリティグループの紐づけを削除してください（しないとコケます・・・）

Open IDE

Neptuneクラスター削除

Neptune < インスタンス

インスタンス (2)

DB インスタンス

変更
再起動
削除
フェイルオーバー
リードレプリカの作成
クローンの作成
スナップショットの取得
特定時点への復元

hama-neptune-handson

hama-neptune-handson-ap-northeast-1

CPU 4.00% な
CPU 3.00% な

Masterと Readerのインスタンスをそれぞれ削除する

インスタンスの削除が確認できれば、最後にクラスターも削除してください。（削除保護が効いている場合は、先にクラスタの変更で削除保護のチェックを外してください。）

Neptune < クラスター

クラスター (1)

クラスターの変更
IAM ロールの管理
今すぐアップグレード
次のウィンドウでアップグレード
クラスターの削除

hama-neptune-handson-cluster

利用可能



VPCエンドポイント削除

The screenshot shows the AWS VPC Endpoint Management console. On the left, there's a sidebar with navigation links like Virtual Private Cloud, VPC, Subnet, Route Table, Internet Gateway, Egress Only Internet Gateway, DHCP Options Set, Elastic IP, and Endpoints. The 'Endpoints' link is currently selected. The main area has tabs for 'Create Endpoint' and 'Actions'. A search bar and a filter for 'VPC Selection' are also present. The main table lists three endpoints:

Name	Endpoint Type	Status
vp...	Gateway	Available
vp...	Gateway	Available
vp...	Gateway	Available

An arrow points from the text 'エンドポイント削除' (Delete Endpoint) in the question to the 'Delete Endpoint' option in the context menu.

Actions ▾

- サブネットの管理
- ルートテーブルの管理
- プライベート DNS 名の変更
- エンドポイントの削除**
- ポリシーの編集
- セキュリティグループの管理
- タグの追加/編集

S3バケット(データ、ホスト)削除

+ バケットを作成する パブリックアクセス設定を編集する 空にする 削除 3 リージョン

バケット名	バケットとオブジェクトの権限	リージョン	最終更新日時
hamas3test-arena	非公開	アジアパシフィック (東京)	GMT+0900 4月 20, 2019 7:22:43 午後
hamas3bucket.test01	非公開	アジアパシフィック (東京)	GMT+0900 4月 20, 2019 7:22:43 午後
hamashi-neptune-data	非公開	アジアパシフィック (東京)	1月 23, 2020 7:23:24 午前 GMT+0900
hamashi-neptune-hosting	公開	アジアパシフィック (東京)	1月 23, 2020 12:39:05 午後 GMT+0900
hamashi-sharing	公開	アジアパシフィック (東京)	1月 23, 2020 5:38:59 午前 GMT+0900
hikitugi-hama	非公開	アジアパシフィック (東京)	3月 19, 2019 7:44:07 午後

データ用のバケットとホスト用のバケットそれぞれ削除する



CloudFormation Stack削除

CloudFormation > スタック

Stack (1)

削除

更新する

Stackアクション ▾

Stackの作成 ▾

Stack名によるフィルター

アクティブ ▾

ネスト表示

1

Stackの名前

hamashi-neptune-demo

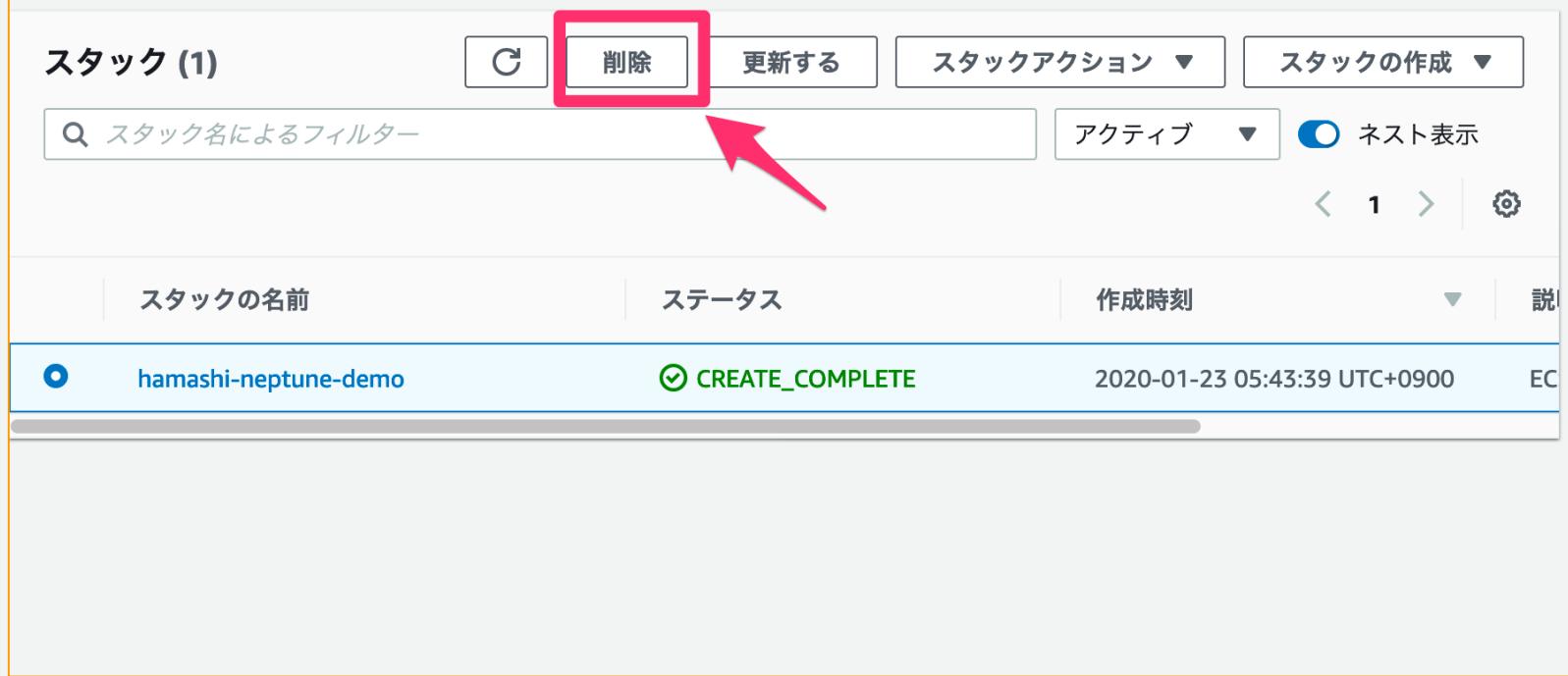
ステータス

CREATE_COMPLETE

作成時刻

2020-01-23 05:43:39 UTC+0900

説





CloudWatch ロググループ削除

The screenshot shows the AWS CloudWatch Log Groups console. On the left, there's a sidebar with a 'Metrics Filter' button and a 'Log Groups' section containing several log group names. A red arrow points from the text 'ロググループの削除' (Delete Log Group) in the dropdown menu to the same option in the sidebar. The main area displays a table of log groups with columns for 'インサイト' (Insights), '次の期間経過後にイベントを失効' (Events expire after), and 'メトリクス' (Metrics). One log group, 'RDSOSMetrics', has a status of '1か月(30日間)' (1 month (30 days)).

	インサイト	次の期間経過後にイベントを失効	メトリクス
RDSOSMetrics	調査	失効しない	0 フィード
QUOC0Z	調査	失効しない	0 フィード
	調査	失効しない	0 フィード
	調査	失効しない	0 フィード
	調査	失効しない	0 フィード
	調査	失効しない	0 フィード
	調査	失効しない	1 フィード
	調査	1か月(30日間)	0 フィード



お疲れ様でした！

