

## はじめての自然言語処理（NLP : Neuro Linguistic Programing）ハンズオン

アマゾン ウェブ サービス ジャパン

シニアエバンジェリスト 亀田 治伸

はじめに：

このハンズオンでは AWS が提供している、自然言語処理関連の 4 つのサービスを用い

て、日本語の処理を体験するハンズオンです。以下のサービスを用います。

- ・ Amazon Translate : 高速で高品質なニューラル機械翻訳サービス
- ・ Amazon Polly: 深層学習技術を使用したテキスト読み上げサービス
- ・ Amazon Transcribe : 音声をテキストに変換する自動音声認識サービス
- ・ Amazon Comprehend : テキストから洞察を見つけるサービス
- ・ Amazon Elasticsearch Service : Elasticsearch, Kibana のマネージドサービス
- ・ AWS Cloud9 : クラウドベースの統合開発環境 (IDE)
- ・ AWS Lambda: サーバレスコンピューティングサービス

作業は特に指示されない限り、東京リージョンで行いますが、途中手順によってオハイオ

リージョンを使いますので、文中の記載に従ってください。

### 1. IAM ロールの設定

ハンズオンで作成する Lambda 関数が他のサービスの実行に必要な権限を作成しま

す。

- 1-1. IAM のトップ画面にいき、「ロール」をクリックします。

## Identity and Access Management (IAM)

### ダッシュボード

#### ▼ アクセス管理

グループ

ユーザー

ロール

ポリシー

ID プロバイダー

アカウント設定

- 1-2. 「ロールの作成」を押します



- 1-3. 「Lambda」を選択し、「次のステップ : アクセス権限」を押します

## ロールの作成

1 2 3 4

信頼されたエンティティの種類を選択

**AWS サービス**  
EC2、Lambda、およびその他

**別の AWS アカウント**  
お客様またはサードパーティーに属しています

**ウェブ ID**  
Cognito または任意の OpenID プロバイダ

**SAML 2.0 フェデレーション**  
企業ディレクトリ

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	ElasticLoadBalancing	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

\* 必須

キャンセル

次のステップ: アクセス権限

1-4.検索項目に[translate]と入力し、出てきた「TranslateFullAccess」を選びます。 **右**

**下のボタンはまだ押しません。**

## ロールの作成



1 2 3 4

▼ 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** ▼  4 件の結果を表示中

	ポリシー名 ▼	次として使用
<input type="checkbox"/>	▶ PollyAndTranslateForWordPressPolicy-0308	なし
<input type="checkbox"/>	▶ PollyAndTranslateForWordPressPolicy-0819	Permissions policy (1)
<input checked="" type="checkbox"/>	▶  TranslateFullAccess	Permissions policy (1)
<input type="checkbox"/>	▶  TranslateReadOnly	なし

1-4. 同様に「Polly」「Transcribe」「Comprehend」も検索し FullAccess にチェックを

入れ最後に、「次のステップ : タグ」を押します。

## ロールの作成

1 2 3 4

### ▼ 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 polly

4 件の結果を表示中

	ポリシー名	次として使用
<input checked="" type="checkbox"/>	▶  AmazonPollyFullAccess	Permissions policy (1)
<input type="checkbox"/>	▶  AmazonPollyReadOnlyAccess	Permissions policy (2)
<input type="checkbox"/>	▶ PollyAndTranslateForWordPressPolicy-0308	なし
<input type="checkbox"/>	▶ PollyAndTranslateForWordPressPolicy-0819	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

1-5. 何も入力せず「次のステップ: 確認」を押します。

キャンセル

戻る

次のステップ: 確認

1-6. ロール名に「YYYYMMDDnlpbandson」と入力します。(YYYYMMDD は本日の日付)

## ロールの作成

1

2

3

4

### 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\*

20200117

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

-  [TranslateFullAccess](#)
-  [AmazonPollyFullAccess](#)
-  [AmazonTranscribeFullAccess](#)
-  [ComprehendFullAccess](#)

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

1-7. 4 つのサービスへの FullAccess がついていることを確認し、「ロールの作成」を押します。以下のような緑で成功が表示されたら作業完了です。皆さんが作成したロールの名前が表示されているはずです。

✔ ロール **20200117** が作成されました。

## 2. Amazon Translate

# この手順はオプションです。作業を飛ばしても問題ありません

2-1. 何か適当な英語のニュースを探してコピペしておきます。あまり長い文字だと作業

が不必要に大変になるので、だいたい 150 文字以内にしましょう。

2-2.AWS Lambda の管理者画面にいき、画面右上の「関数の作成」を押します。



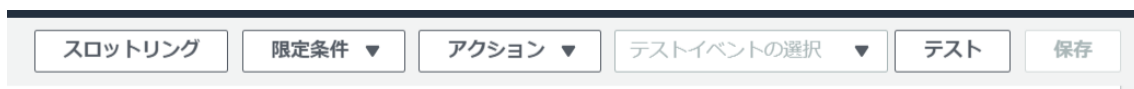
2-3.関数名に「YYYYMMDDtranslate」(YYYYMMDD は本日の日付)を入力し、ランタイムは「Python3.8」を選択します。

2-4.「実行ロールの選択または作成」を伸ばして、「既存のロールを選択する」を選び先ほど作成した IAM ロールを選択します。選択がおわったら「関数の作成」を押します。

2-5.サンプルで挿入されているコードをすべて消し「translate lambda.txt」の内容をコピーします。その後右上の「保存」を押します。



2-6. 右上の「テスト」を押します。



2-7. イベント名に適当な名前を入れあらかじめ挿入されている文字列を削除し、

「translatelambdatest.txt」の中身に置き換えます。

イベントテンプレート

Hello World ▼

イベント名

testtranslate

```

1 {
2   "text": "hello"
3 }

```

2-8. [hello]の部分をコピーしておいた英語の記事に入れ替えます。この際改行は全て抜

いてください。その後右下の「作成」を押します。

イベント名

testtranslate

```
1 {  
2   "text": "A mystery pneumonia outbreak in mainland China has sparked fears across eastern Asia"  
3 }
```

キャンセル

作成

2-9. 右のブラウザのスクロールバーを一番上に持っていき、「テスト」を押してください。以下が表示されれば成功です。

✔ 実行結果: 成功 (ログ)

▶ 詳細

2-10. 「詳細」を伸ばすと翻訳された日本語がでできます。





### 3. Amazon Polly

先ほどは Lambda で Translate を操作しましたが、実は、Translate、Polly、Transcribe、Comprehend は全てマネージメントコンソールからテストすることができるようになっています。このハンズオンではこの後何度も Polly を使用しますので、マネージメントコンソールでの手順を紹介します。

#### 3-1. Polly の管理者画面にアクセスします。



#### 3-2. Translate で翻訳した文字列をコピーします。Translate の手順を飛ばした方は、な

んでもいいので 150 字以下程度の文字列をコピーします。

### テキスト読み上げ機能

音声の聴き取り、カスタマイズ、ダウンロード。準備が整ったら統合。

ウインドウにテキストを入力するか貼り付けて、言語と地域を選択し、音声を選択します。次に、[音声を聴く] を選択し、アプリケーションとサービスに統合します。

3000 文字以内の場合、すぐにリスン、ダウンロード、または保存できます。100,000 文字までの場合、タスクは S3 バケットに保存する必要があります。

プレーンテキスト SSML ⓘ

Lambda を使用すれば、実質どのようなタイプのアプリケーションやバックエンドサービスでも管理を必要とせずに実行できます。コードさえアップロードすれば、高可用性を実現しながらコードを実行およびスケールアップするために必要なことは、すべて Lambda により行われます。コードは、他の [AWS](#) サービスから自動的にトリガーされるよう設定することも、ウェブまたはモバイルアプリから直接呼び出すよう設定することもできます。

209 文字を使用

デフォルトのテキストを表示 テキストを消去

言語とリージョン

日本語 ▼

音声

☒ Mizuki, 女性

☐ Takumi, 男性

▶ 音声を聴く

📄 ダウンロード MP3

サンプルレート: 22050Hz

[ファイル形式を変更](#)

3-3. 男性か女性の音声を選び、「ダウンロード MP3」を押すと、しばらく待てば mp3

ファイルが生成されます。人名や地名など難読文字のカスタマイズもこの画面から可能で

すが、このハンズオンでは割愛します。興味がある方は講師に聞いてください。

3-4. ダウンロードされた mp3 を再生し、正しく変換されていることを確認します。

## 4. Transcribe

先ほど生成した mp3 から文字起こしを行う環境を作ります。

4-1. S3 バケットを東京リージョンで作るため、S3 の管理者画面に行きます。

4-2. 「バケットを作成する」を押して YYYYMMDDName (YYYYMMDD は本日の日

付。Name はご自身のアルファベットの名前) を入れます。リージョンが東京となっていることを確認し、「次へ」を押します。



バケットの作成

① 名前とリージョン    ② オプションの設定    ③ アクセス許可の設定    ④ 確認

名前とリージョン

バケット名 ⓘ

20200117kameda

リージョン

アジアパシフィック (東京)

既存のバケットから設定をコピー

バケットを選択する (省略可) 62 バケット

作成    キャンセル    次へ

4-3.   すべてデフォルトで作成しますので、もう2回「次へ」を押し、最後に「バケットを作成」を押します。

バケットの作成

名前とリージョン

オプションの設定

アクセス許可の設定

4 確認

名前とリージョン

バケット名

20200117kameda

リージョン

アジアパシフィック (東京)

編集

オプション

バージョンing

無効

サーバーアクセスのログ記録

無効

タグ

0 タグ

オブジェクトレベルのログ記録

無効

デフォルト暗号化

なし

CloudWatch リクエストメトリクス

無効

オブジェクトのロック

無効

編集

アクセス権限

パブリックアクセスをすべてブロック

オン

新しいアクセスコントロールリスト (ACL) を介して許可されたバケットとオブジェクトへのパブリックアクセスをブロックする

オン

任意のアクセスコントロールリスト (ACL) を介して許可されたバケットとオブジェクトへのパブリックアクセスをブロックする

編集

戻る

バケットを作成

4-4. 先ほど作成したバケットをクリックし「アップロード」を押します。

Amazon S3

>

20200117kameda

20200117kameda

概要

プロパティ

アクセス権限

管理

アクセスポイント

アップロード

フォルダの作成

ダウンロード

アクション

4-5. 先ほど作成した mp3 をアップロードします。



4-6. デフォルトのままアップロードしますので、「次へ」を 3 回、「アップロード」を 1 回押してください。完了すると以下のような画面になります。



4-6-1. 先ほど作成した IAM ロールは、Translate、Polly、Transcribe、Comprehend への権限が付与されていますが、このままでは S3 へのアクセスが許可されていません。

IAM ロールの画面に戻り、S3 へのアクセス権限を付与します。先ほど作成したロールを IAM ロール管理画面から特定し、名前をクリックします。

<div> <div>ロールの作成</div> <div>ロールの削除</div> </div>		<div> <div>🔄</div> <div>⚙️</div> <div>🔍</div> </div>
<div> <div>🔍</div> <div>20200117</div> </div>		1件の結果を表示中
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 ▼
<input type="checkbox"/> 20200117	AWS サービス: lambda	今日

#### 4-6-2. 「ポリシーをアタッチします」をおします

ロール

> 20200117

概要

ロールの削除

ロール ARN

arn:aws:iam::294963776963:role/20200117

ロールの説明

Allows Lambda functions to call AWS services on your behalf. | [編集](#)

インスタンスプロファイル ARN

パス /

作成時刻

2020-01-17 14:25 UTC+0900

最後のアクティビティ

2020-01-17 14:41 UTC+0900 (今日)

最大 CLI/API セッション期間

1 時間 [編集](#)

アクセス権限

信頼関係

タグ

アクセスアドバイザー

セッションの無効化

Permissions policies (5 適用済みポリシー)

ポリシーをアタッチします

インラインポリシーの追加

ポリシー名 ▼	ポリシータイプ ▼
<div>🔍</div> <div>TranslateFullAccess</div>	<div>AWS 管理ポリシー</div> <div>✕</div>

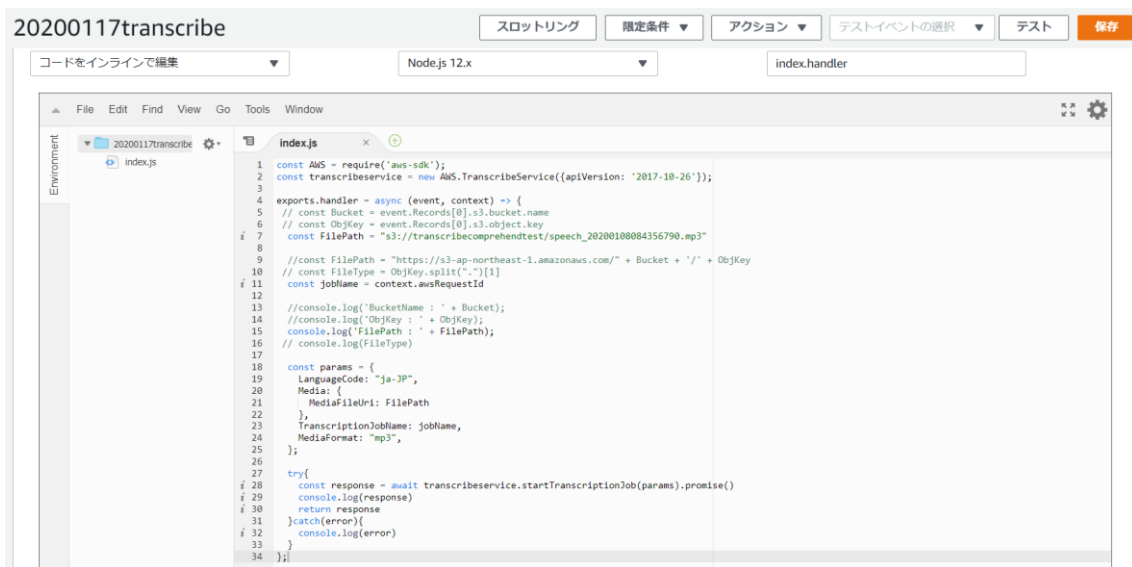
#### 4-6-3. 「AmazonS3FullAccess」を選び「ポリシーのアタッチ」を押します。

#### 4-7. 別のタブで Lambda の管理者画面を開き、Translate と同じ要領で

YYYYMMDDtranscribe という関数を作ります。今度はランタイムに Node12.x

を選択します。**先ほど作成したロールを指定することを忘れずに行ってください。**

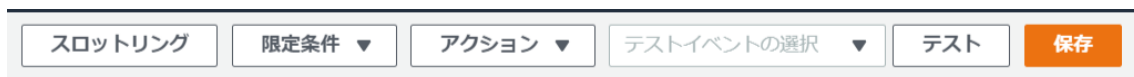
#### 4-8. 「transcribelambda.txt」の中身をコピーします。



4-9. [const FilePath]の部分を、[先ほど作成したバケット名/アップした mp3 ファイル名]に変換します。以下のような文字列になるはずです。

“s3://20200117kameda/speech\_20200117061325085.mp3”

4-10. 右上の「保存」を押します。



4-11. 「テスト」を押します。イベント名に適当な名前を入れて、「作成」を押します。

# この関数では外部インプットは必要ありませんので、文字列はデフォルトのままダミーのテストイベントを作成する必要があります。

イベントテンプレート

Hello World

イベント名

test

```
1 {  
2   "key1": "value1",  
3   "key2": "value2",  
4   "key3": "value3"  
5 }
```

キャンセル

作成

4-12. ブラウザのスクロールバーを一番に持っていき「テスト」を押します。以下のような表示[ IN\_PROGRESS]が戻れば成功です。



4-13. Transcribe の管理者画面に行きます。プロジェクトが開始されていますので、[Complete]になるまでしばらく待ちます。



Amazon Transcribe > Transcription jobs

### Transcription jobs

Download
Delete
Copy
Create job

Status: All

< 1 > ⚙


	Name	Language	Created	Expires in	Status
<input type="radio"/>	33d6ce0d-a034-4508-a3c3-9ea97846efb1	Japanese (Japan)	2020/1/17 15:35:04	-	In progress

[Amazon Transcribe](#)
[>](#)
[Transcription jobs](#)
[>](#)
33d6ce0d-a034-4508-a3c3-9ea97846efb1

33d6ce0d-a034-4508-a3c3-9ea97846efb1

[Download full transcript](#)
[Delete](#)
[Copy](#)

### Job details

<b>Name</b> 33d6ce0d-a034-4508-a3c3-9ea97846efb1	<b>Ended</b> 2020/1/17 15:36:20	<b>Vocabulary filter</b> -
<b>Status</b>  Complete	<b>Input file format</b> mp3	<b>Vocabulary filtering method</b> -
<b>Language</b> Japanese (Japan)	<b>Audio sampling rate</b> 22050 Hz	<b>Input data location</b> s3://20200117kameda/speech_20200117061325085.mp3 <a href="#">🔗</a>
<b>Expiration</b> <a href="#">Info</a> The transcription is available for 89 more days.	<b>Audio identification</b> None	<b>Output data location</b> <a href="https://s3.ap-northeast-1.amazonaws.com/aws-transcribe-ap-northeast-1-prod/294963776963/33d6ce0d-a034-4508-a3c3-9ea97846efb1/acf001d9-8a73-47bc-8a4a-2bcf8ddcf912/asrOutput.json">https://s3.ap-northeast-1.amazonaws.com/aws-transcribe-ap-northeast-1-prod/294963776963/33d6ce0d-a034-4508-a3c3-9ea97846efb1/acf001d9-8a73-47bc-8a4a-2bcf8ddcf912/asrOutput.json</a> <a href="#">🔗</a>
<b>Created</b> 2020/1/17 15:35:04	<b>Alternative results</b> Disable	
<b>Started</b> 2020/1/17 15:35:04	<b>Custom vocabulary</b> None	

```
[{"id": "編集(書式) へリ上げ", "jobName": "33d6c0cd-a034-4508-ac3c-9ea79da4febfb", "accountId": "294963776963", "results": {"transcripts": [{"transcript": "俗村を使用すれば実質どのようなタイプの\n「リクエスト」や「サービス」でも管理を可能にします。実行できます。コード上のセキュリティは効果を実現しながらコードを実行す\nるために必要となる脆弱性を修正し、より安全なシステムを実現します。また、このシステムのセキュリティが自動的にトリガされるよう設定\nされています。", "confidence": 0.4108, "content": "俗村", "type": "pronunciation"}, {"start_time": 0.5, "end_time": 0.59, "alternatives": [{"confidence": 1.0, "content": "使用"}], "type": "pronunciation"}, {"start_time": 0.59, "end_time": 0.93, "alternatives": [{"confidence": 1.0, "content": "使用"}], "type": "pronunciation"}, {"start_time": 1.24, "end_time": 1.4, "alternatives": [{"confidence": 1.0, "content": "と"}], "type": "pronunciation"}, {"start_time": 1.4, "end_time": 1.42, "alternatives": [{"confidence": 1.0, "content": "の"}], "type": "pronunciation"}, {"start_time": 2.16, "end_time": 2.42, "alternatives": [{"confidence": 0.9987, "content": "とどの"}], "type": "pronunciation"}, {"start_time": 2.42, "end_time": 2.61, "alternatives": [{"confidence": 0.9987, "content": "よう"}], "type": "pronunciation"}, {"start_time": 2.61, "end_time": 2.73, "alternatives": [{"confidence": 1.0, "content": "な"}], "type": "pronunciation"}, {"start_time": 2.73, "end_time": 3.03, "alternatives": [{"confidence": 1.0, "content": "タ"}], "type": "pronunciation"}, {"start_time": 3.03, "end_time": 3.23, "alternatives": [{"confidence": 1.0, "content": "の"}], "type": "pronunciation"}, {"start_time": 3.23, "end_time": 4.12, "alternatives": [{"confidence": 1.0, "content": "アプリケーション"}], "type": "pronunciation"}, {"start_time": 4.12, "end_time": 4.32, "alternatives": [{"confidence": 1.0, "content": "や"}], "type": "pronunciation"}, {"start_time": 4.32, "end_time": 5.03, "alternatives": [{"confidence": 0.9538, "content": "バック"}], "type": "pronunciation"}, {"start_time": 5.03, "end_time": 5.37, "alternatives": [{"confidence": 0.9538, "content": "サービス"}], "type": "pronunciation"}, {"start_time": 5.37, "end_time": 5.84, "alternatives": [{"confidence": 1.0, "content": "サ"}], "type": "pronunciation"}, {"start_time": 5.84, "end_time": 5.94, "alternatives": [{"confidence": 0.8933, "content": "で"}], "type": "pronunciation"}, {"start_time": 5.94, "end_time": 6.1, "alternatives": [{"confidence": 0.8933, "content": "も"}], "type": "pronunciation"}, {"start_time": 6.1, "end_time": 6.47, "alternatives": [{"confidence": 0.8933, "content": "管理"}], "type": "pronunciation"}, {"start_time": 6.47, "end_time": 6.57, "alternatives": [{"confidence": 1.0, "content": "の"}], "type": "pronunciation"}, {"start_time": 6.57, "end_time": 7.02, "alternatives": [{"confidence": 1.0, "content": "必要"}], "type": "pronunciation"}]}
```

## 5. Comprehend

先ほどの Transcribe が書き起こした mp3 の内容を Comprehend で分析します。

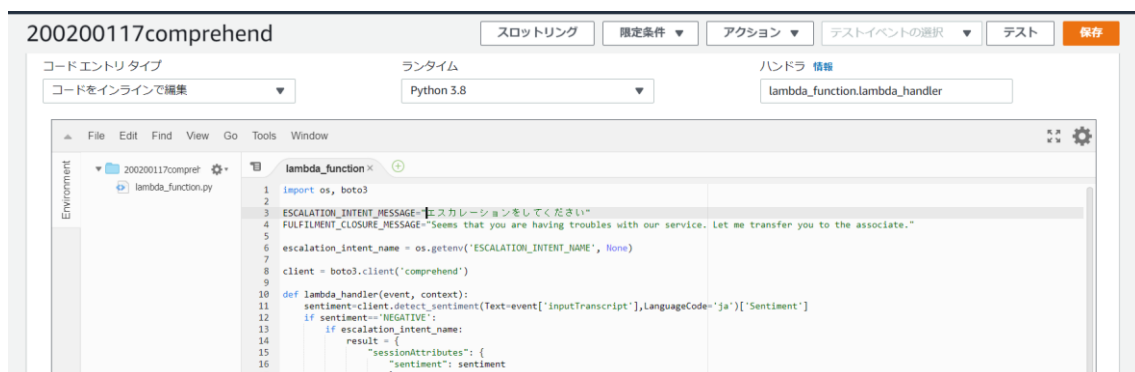
5-1. **2020 年 1 月現在、Comprehend は東京リージョンにはないため、オハイオリージョンで関数を作ります。画面上のリージョンをオレゴンに切り替えます。**

Lambda 関数の作成画面に再度いき、YYYYMMDDComprehend で関数を作成します。

ランタイムは Python3.8 を指定し、先ほどと同様に今日作成した IAM ロールを設定することを忘れずに行います。

5-2. [comprehendlambda.txt]の中身をコピーします。

このサンプルコードは、Comprehend が Negative なメッセージであると判断した場合、"エスカレーションをしてください"というメッセージが出力されます。



5-3. 画面右上の「保存」を押します。その後「テスト」を押し、イベント名に適当な名前を付けた後、[comprehendlambdatest.txt]をコピーします。

5-4. テスト文字列には「あなたを攻撃します」と記載されていますが、一度そのままテストを実行します。以下のような画面が出力されます。

✔ 実行結果: 成功 (ログ)

▼ 詳細

関数の実行から返された結果が以下のエリアに表示されます。関数から結果を返す方法の詳細については、[こちら](#)を参照してください。

```
{
  "sessionAttributes": {
    "sentiment": "NEGATIVE"
  },
  "dialogAction": {
    "type": "Close",
    "fulfillmentState": "Failed",
    "message": {
      "contentType": "PlainText",
      "content": "エスカレーションをしてください"
    }
  }
}
```

5-5. 先ほどのテストで使った文字列を「あなたが好きです」に変えて再度テストを実行してください。修正するためには、以下の箇所から「テストイベントの設定」を選択することで、先ほど設定したテストを上書きすることができます。

5-6. テストを実行すると分析結果が「Positive」に代わり、エスカレーションがされなくなりました。

✔ 実行結果: 成功 (ログ)

▼ 詳細

関数の実行から返された結果が以下のエリアに表示されます。関数から結果を返す方法の詳細については、[こちら](#)を参照してください。

```
{
  "sessionAttributes": {
    "sentiment": "POSITIVE"
  },
  "dialogAction": {
    "type": "Delegate",
    "slots": {
      "slot1": "None",
      "slot2": "None"
    }
  }
}
```

5-7. では先ほど Transcribe で文字お越しした文字列に入れ替えて再度テストをしてみてください。ほとんどのニュース記事の場合、例え内容が悲惨な事件などであっても、その論調は感情を持たせず客観的に記載されていますので、[Neutral]と出力されるケースが多いようです。

🟢 実行結果: 成功 (ログ)

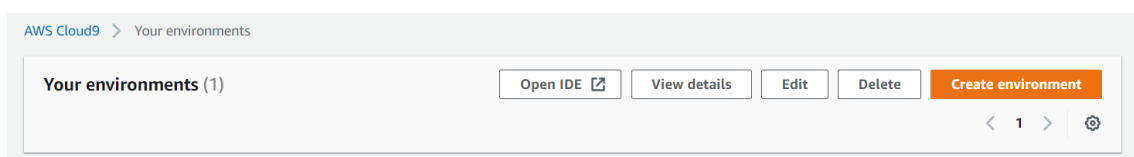
▼ 詳細

関数の実行から返された結果が以下のエリアに表示されます。関数から結果を返す方法の詳細については、[こちら](#)を参照してください。

```
{
  "sessionAttributes": {
    "sentiment": "NEUTRAL"
  },
  "dialogAction": {
    "type": "Delegate",
    "slots": {
      "slot1": "None",
      "slot2": "None"
    }
  }
}
```

6. Transcribe と Comprehend を連携させ、音声の分析結果を Elasticsearch service の kibana で可視化する。**以下の作業はすべてオレゴンで行います。Cloud9 は起動した後、個別の IAM ロール設定が可能です。デフォルトでは、AWS の管理者画面にログインしている IAM の権限を引き継ぎます。この手順ではデフォルト権限設定を利用するため、Comprehend、Transcribe、ElasticSearch Service、S3、Cloud9 へのフル権限を持っている IAM ユーザーで管理者画面にログインして作業を行ってください。**

6-1. Cloud9 を起動するため、Cloud9 の管理者画面にいきます。



6-2. 「Create Environment」を押します。[Name]には YYYYMMDDhandson (YYYYMMDD は本日の日付) と入力し、[Next Step]を押します。。

## 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](#)

- 6-3. 画面下の[Network Settings]を以下の様に伸ばし、パブリックサブネットを持つ VPC とパブリックサブネットを選択します。不明な場合は、チューターに聞いて下さい。
- 別タブで VPC の管理者画面を開くと確認しやすくなります。


**▼ Network settings (advanced)**

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



[Create new VPC](#)

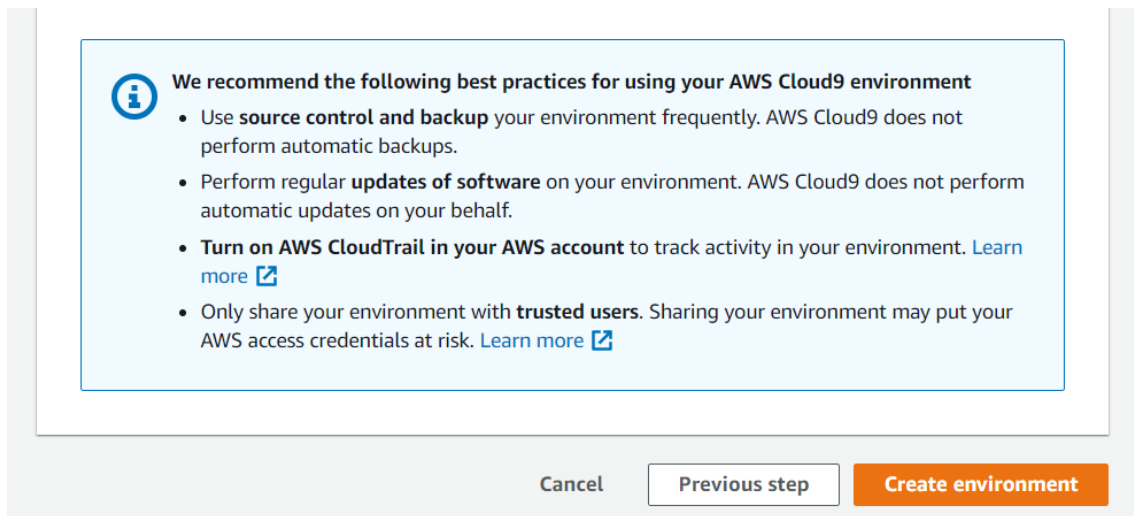
**Subnet**  
Select a range of IP addresses in your VPC to isolate EC2 resources from each other.



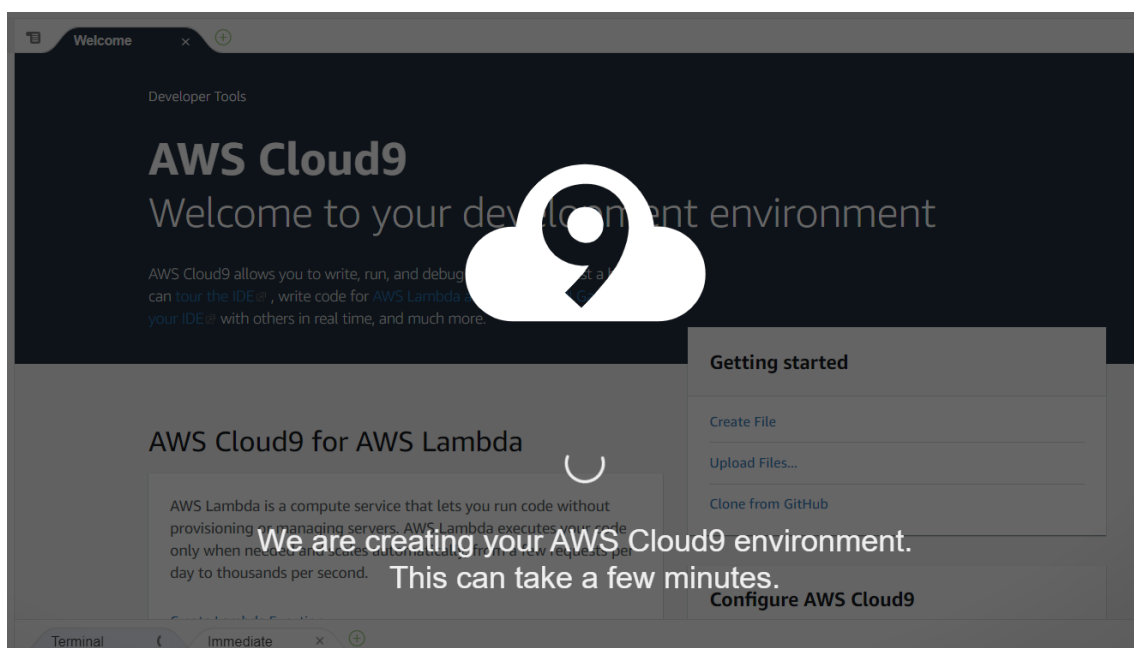
[Create new subnet](#)

[Cancel](#) [Previous step](#) [Next step](#)

- 6-4. 「Next Step」を押します。



6-5. 「Create environment」を押すと以下の画面に切り替わります。



6-6. 起動は数分間かかりますが、完了するとコンソールが使えるようになります。

6-7. オレゴンに S3 バケットを作成するために、以下のコマンドをタイプして下さい。

```
aws s3 mb s3://<yyyymmddtranscribename> --region us-west-2
```

(YYYYMMDD は本日の日付、name はお名前)

```
ec2-user:~/environment $ aws s3 mb s3://20200117transcribekameda --region us-west-2
make_bucket: 20200117transcribekameda
```

6-8. 念のため S3 の管理者画面で S3 バケットが正しく作られているか確認します。表示されない場合、再読み込みをしてみましょう。

6-9. オレゴンに Elasticsearch service を起動するため Elasticsearch service の管理者画面に遷移します。

## Amazon Elasticsearch Service ダッシュボード

新しいドメインの作成

Elasticsearch ドメイン

6-10. 「新しいドメインの作成」を押します。

### デプロイタイプの選択

デプロイタイプはユースケースの一般的な設定を指定します。これらの設定は、ドメインの作成後に、いつでも変更できます。

- デプロイタイプ
- ☐ 本番稼働用  
安定したパフォーマンスを確保する場合、複数のアベイラビリティゾーンと専用マスターノード。
  - ☒ 開発およびテスト  
Elasticsearch エンドポイントが 1 つのみ必要な場合、1 つのアベイラビリティゾーン。
  - ☐ カスタム  
すべての利用可能なオプションから設定を選択します。
  - ☐ UltraWarm プレビュー  
選択して UltraWarm をプレビューします。

6-11. 「開発およびテスト」を選びます。

## バージョン

ドメインの Elasticsearch バージョンを選択します。

Elasticsearch のバージョン 6.2

キャンセル 次へ

6-12. 「6.2」を選んで「次へ」を押します。

6-13. ドメイン名に[transcribetestYYYYMMDD](YYYYMMDD は今日の日付)と入力します。

## ドメインの設定

ドメインは、Elasticsearch の実行に必要なリソースのコレクションです。ドメイン名はドメインエンドポイントの一部になります。

Elasticsearch ドメイン名 transcribe20200117

名前は小文字で始まり、3～28 文字で構成される必要があります。有効な文字は、a～z (小文字のみ)、0～9、- (ハイフン) です。

6-14. 「次へ」を押します。

6-15. ネットワーク構成で「パブリックアクセス」を選びます。

## ネットワーク構成

インターネットまたは VPC アクセスを選択します。VPC アクセスを有効にするため、お客様の VPC のプライベート IP アドレスが使用されます。これにより、デフォルトでセキュリティが提供されます。お客様は、セキュリティグループを使用して VPC 内でネットワークアクセスを制御します。オプションで、制限付きアクセスポリシーを適用してセキュリティのレイヤーを追加できます。インターネットエンドポイントはパブリックアクセス可能です。パブリックアドレスを選択する場合、ドメインにアクセスする特定のユーザーまたは IP アドレスのみを許可するアクセスポリシーで、ドメインをセキュリティ保護する必要があります。

☐ VPC アクセス (推奨)  
☒ パブリックアクセス

6-16. 「アクセスポリシー」で「ドメインへのオープンアクセスを許可」を選びます。

## アクセスポリシー

ドメインへのアクセスを許可またはブロックするには、テンプレートのセクタからポリシーテンプレートを選択するか、[アクセスポリシーの追加または編集] ボックスで 1 つ以上の Identity and Access Management (IAM) ポリシーステートメントを追加します。

ドメインアクセスポリシーの設定

テンプレートを選択

アクセスポリシーの追加または編集

1

1 つ以上の AWS アカウントまたは IAM ユーザーにアクセスを許可、または拒否

特定の IP からのドメインへのアクセスを許可

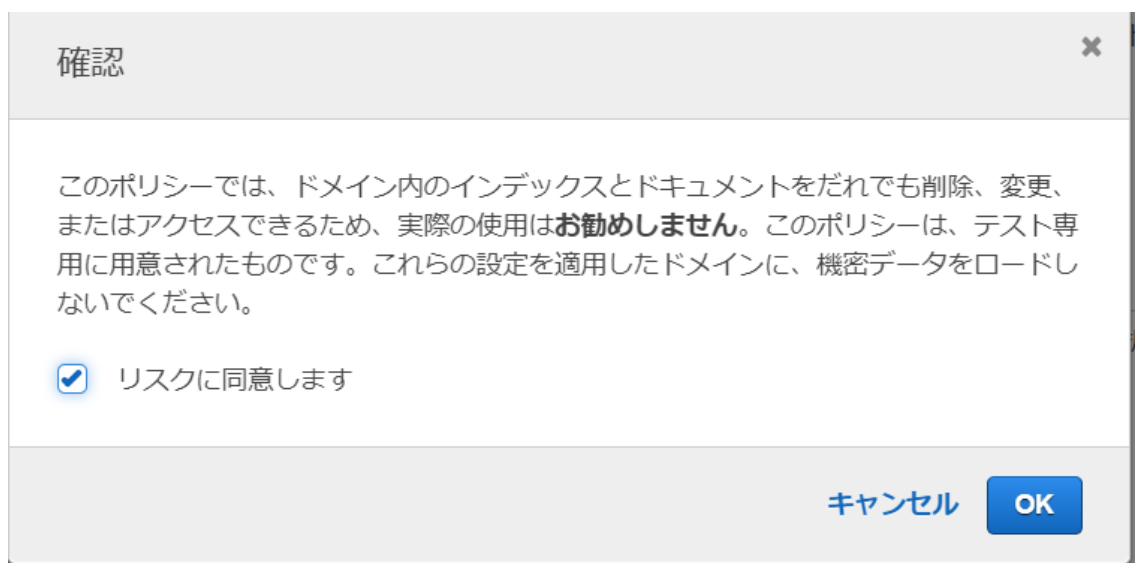
ドメインへのアクセスを拒否  
(このポリシーを選択すると、だれもお客様のドメインのエンドポイントにアクセスできなくなります)

別のドメインからアクセスポリシーをコピー

**ドメインへのオープンアクセスを許可**  
(このポリシーでは、ドメイン内のインデックスとドキュメントをだれでも削除、変更、またはアクセスできるため、実際の使用はお勧めしません)



6-17. 確認ダイアログでチェックをつけて「OK」を押します。



6-18. 「次へ」を押します。最後に確認画面が表示されるので「確認」を押します。

6-19. 現在作成中です。作成が完了したらエンドポイント及び bKibana の URL が表示されますので、しばらくまち、コピーしておきます。(別のタブで Cloud9 の画面に戻って次の Step に進んでも問題ありません)



6-20. 以下 2 つのコマンドを Cloud9 で実行します。コピペすると正常に動作しないケースが

ありますので。手で入力します。

```
python --version
```

```
pip --version
```

ハンズオンを行う時期によって出力が異なりますが、以下のように Python はバージョン 3、pip がバージョン 2 を指している場合、以下のコマンドを実行して、pip の向き先を変更します。

```
sudo update-alternatives --config python
```

```
Selection    Command
-----
*+ 1         /usr/bin/python2.7
   2         /usr/bin/python3.6
Enter to keep the current selection[+], or type selection number: █
```

2 を選びます。これで Python3 環境の整備が終わりました。

6-21. 必要なライブラリをインストールするため、以下の 3 つのコマンドを実行します。赤

字で「Permission denied」と表示された場合、先頭に sudo をつけて実行してください。

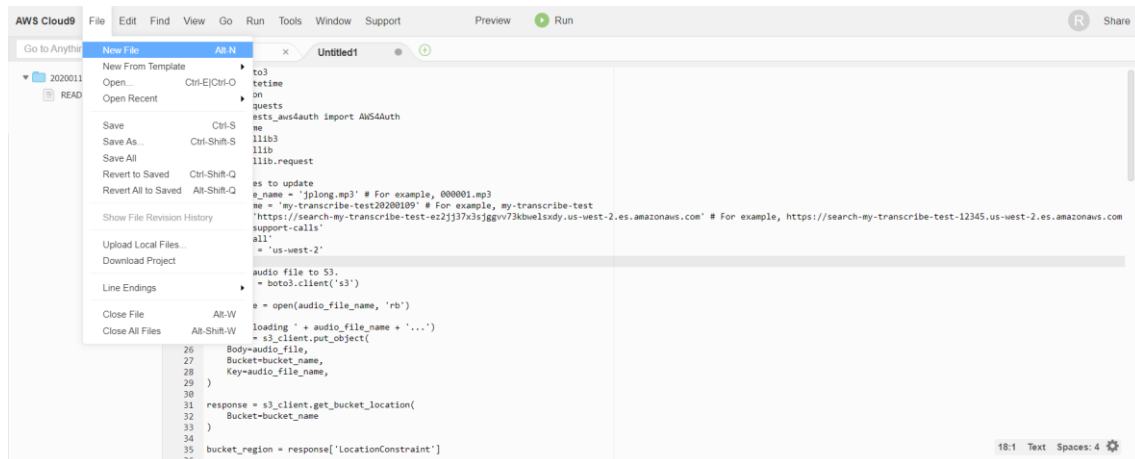
```
pip install boto3
```

```
pip install requests
```

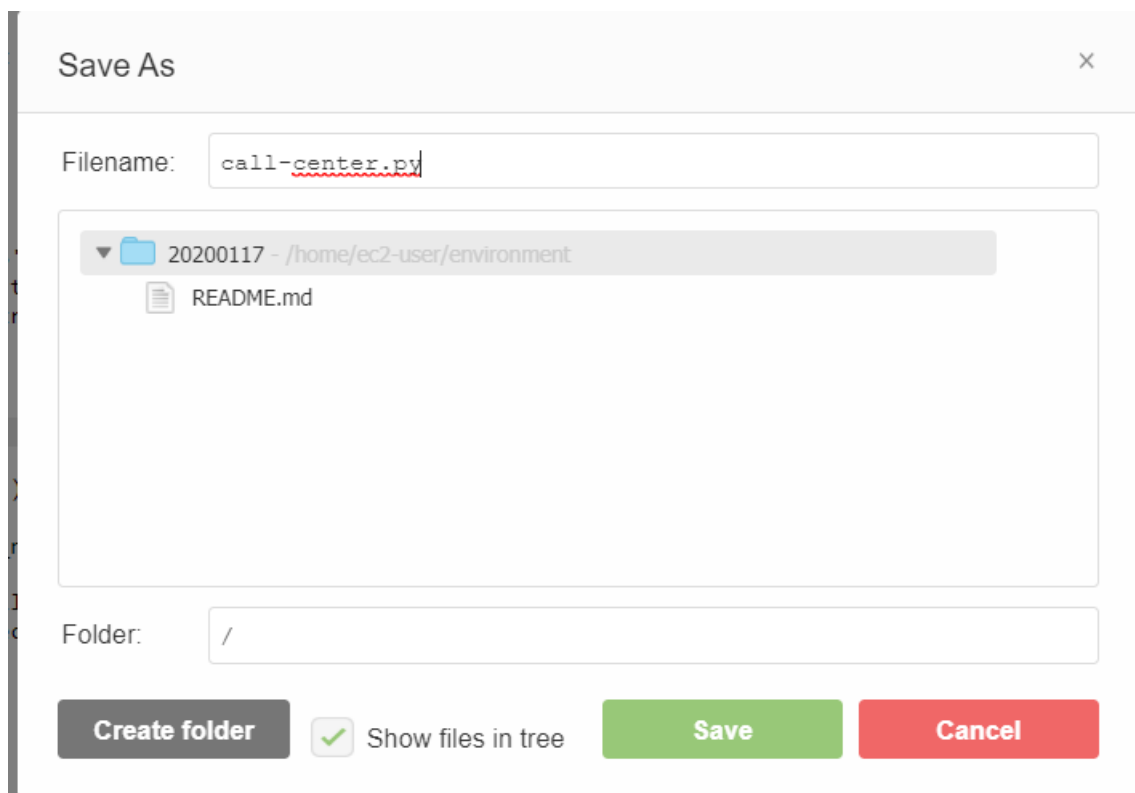
```
pip install requests_aws4auth
```

6-22. 必要なライブラリのインストールが完了しましたので、File→New File と選んで、新

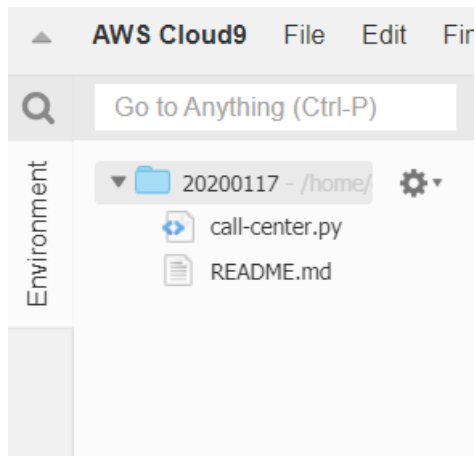
しく表れたタブに call-center.py の中身をコピペします。



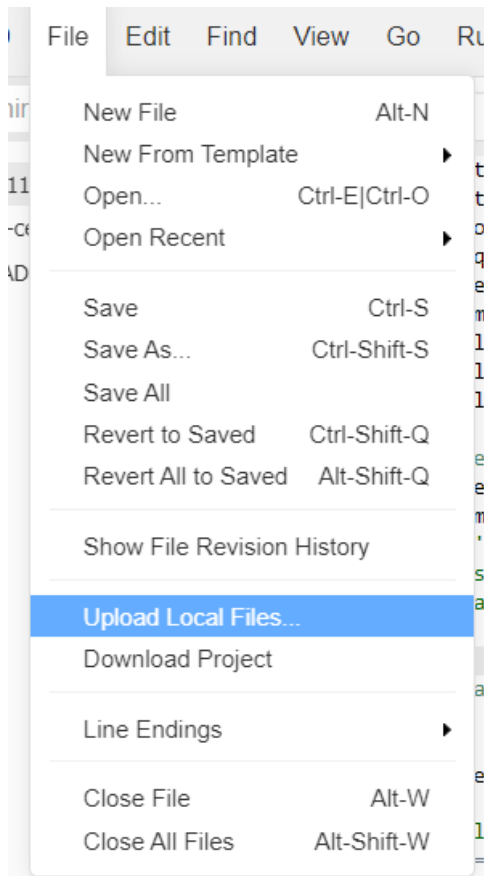
6-23. File→Save 通して、Filename に[call-center.py]と入力し「Save」を押します。



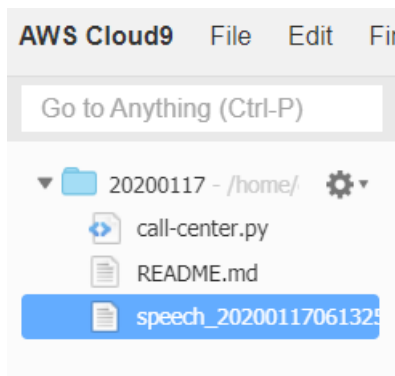
6-24. 以下のフォルダ構成となります。



6-25. File→Upload Local Files を選び、先ほど作成した mp3 をアップロードします。



6-26. Upload 画面は自動で閉じませんので、右上のバツを押して閉じます。以下のフォルダ構成となります。



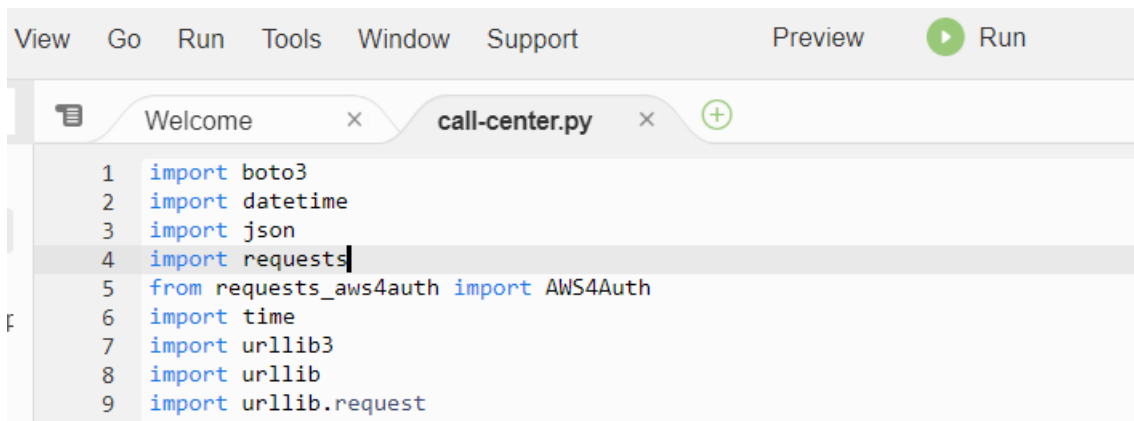
6-27. ソースコードを修正します。12 行目、13 行目、14 行目を修正します。

12 行目 : アップロードした mp3 ファイル名

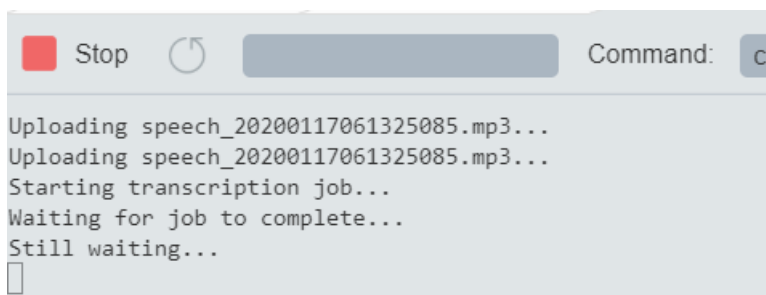
13 行目 : オレゴンに作成した S3 バケット名

14 行目 : Elasticsearch service のエンドポイント

6-28. ファイルを File→Save したらいいよ実行です。画面上の Run ボタンを押します。



6-29. 以下の画面が表示されれば正しく動作しています。



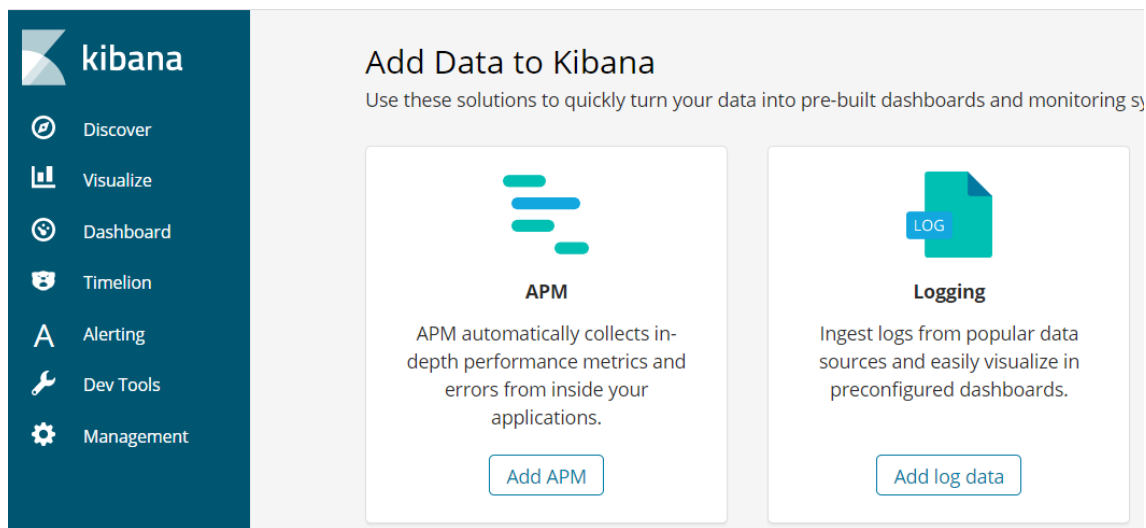
す。

重複エラーが発生する他、再度実行する場合、ファイル名を変更するか、管理者画面から

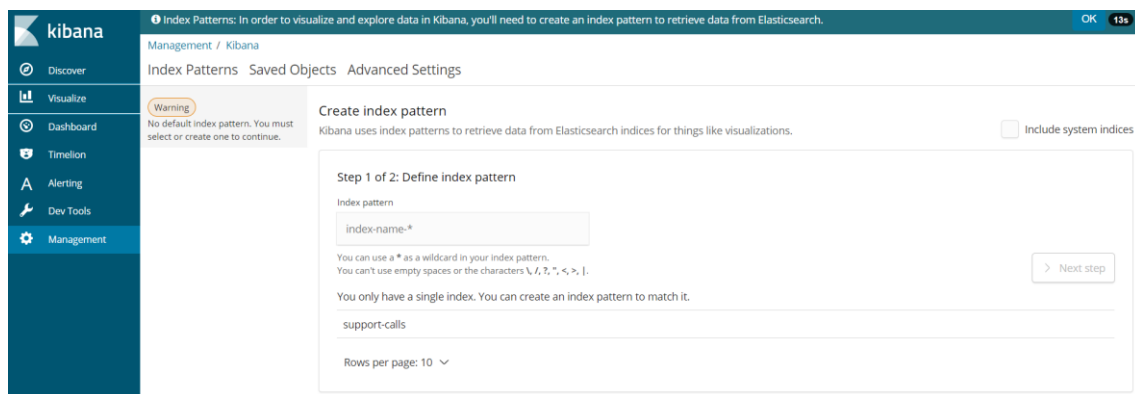
Comprehend が Keyword 分析を行い Elasticsearch Service にデータが投入されています。

```
Process exited with code: 0
```

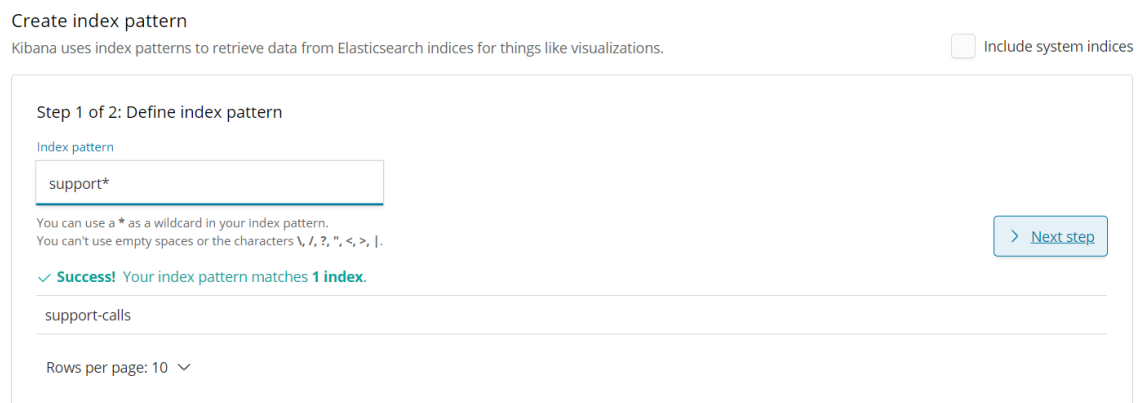
6-31. 先ほどコピーした Kibana の URL にブラウザでアクセスします。



6-32. 「Visualize」を押します。



6-32. [Index Pattern]に[support\*]と入力し[Next Step]を押します。



6-33. [Time Filter field name]に timestamp を選び[Create Index Pattern]を押します。

## Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations.

### Step 2 of 2: Configure settings

You've defined **support\*** as your index pattern. Now you can specify some settings before we create it.

Time Filter field name Refresh

timestamp

The Time Filter will use this field to filter your data by time.

You can choose not to have a time field, but you will not be able to narrow down your data by a time range.

> Show advanced options

< Back

Create index pattern

6-34. 正しくデータが投入され、設定も正しければ以下の画面が表示されます。

## ★ support\*

Time Filter field name: timestamp

This page lists every field in the **support\*** index and the field's associated core type as recorded by Elasticsearch. While this list allows you to view the core type of each field, changing field types must be done using Elasticsearch's [Mapping API](#).

fields (12)

scripted fields (0)

source filters (0)

Q Filter

All field types

name	type	format	searchable	aggregatable	excluded	controls
_id	string		✓	✓		
_index	string		✓	✓		
_score	number					
_source	_source					
_type	string		✓	✓		
keywords	string		✓			
keywords.keyword	string		✓	✓		
sentiment	string		✓			
sentiment.keyword	string		✓	✓		
timestamp	date		✓	✓		

6-35. もう一度 Visualization を押します。

+

0-0 of 0

Looks like you don't have any visualizations. Let's create some!

+

Create a visualization

0 items selected

0-0 of 0

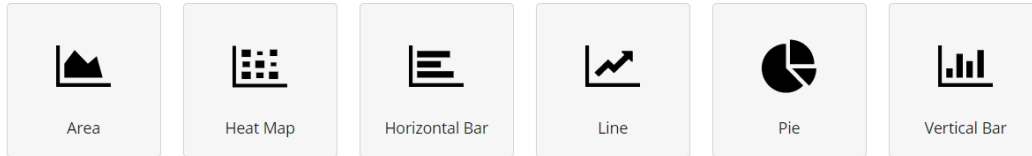
6-36. [Create a visualization]を押します。



## Select visualization type

Search visualization types...

### Basic Charts



6-37. 円グラフを選びます。

### From a New Search, Select Index

Filter... 1 of 1

Name ▲

support\*

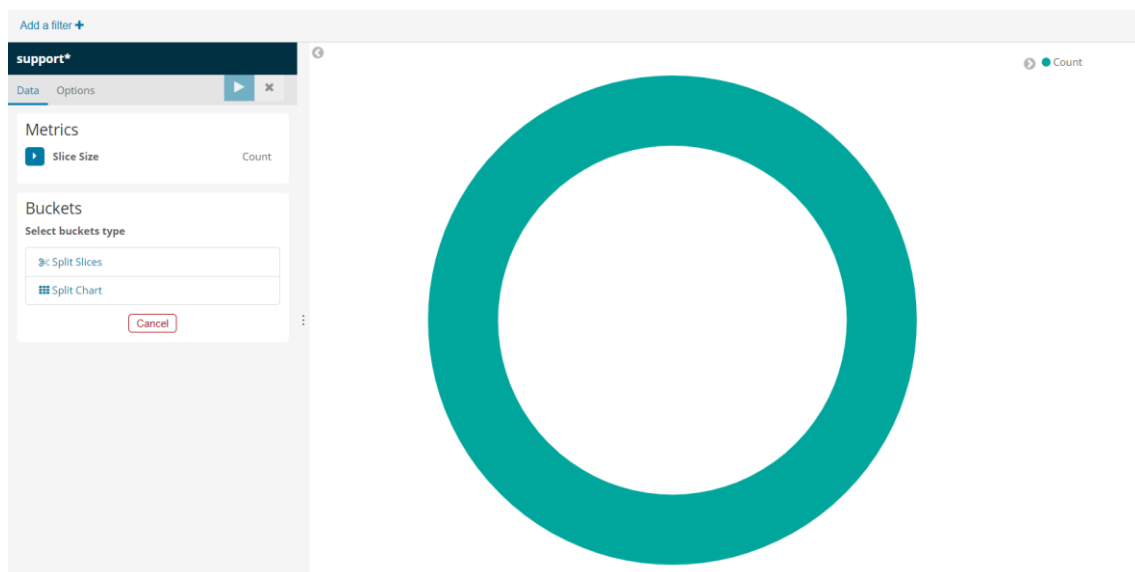
### Or, From a Saved Search

Saved Searches Filter... 0-0 of 0 [Manage saved searches](#)

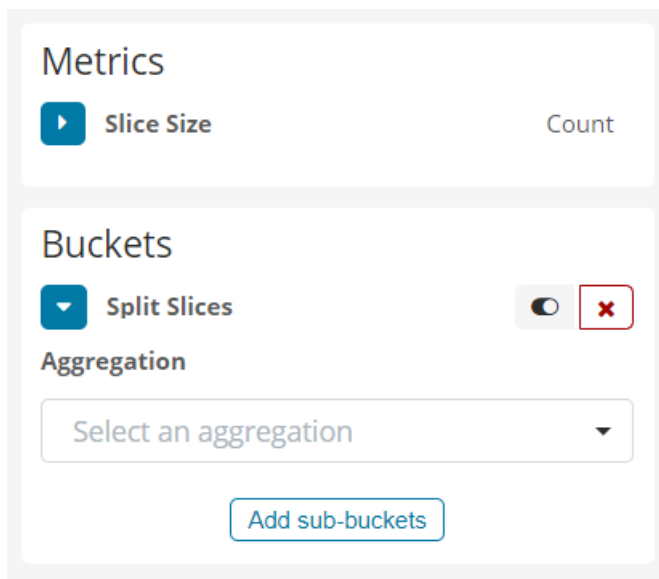
Name ▲

No matching saved searches found.

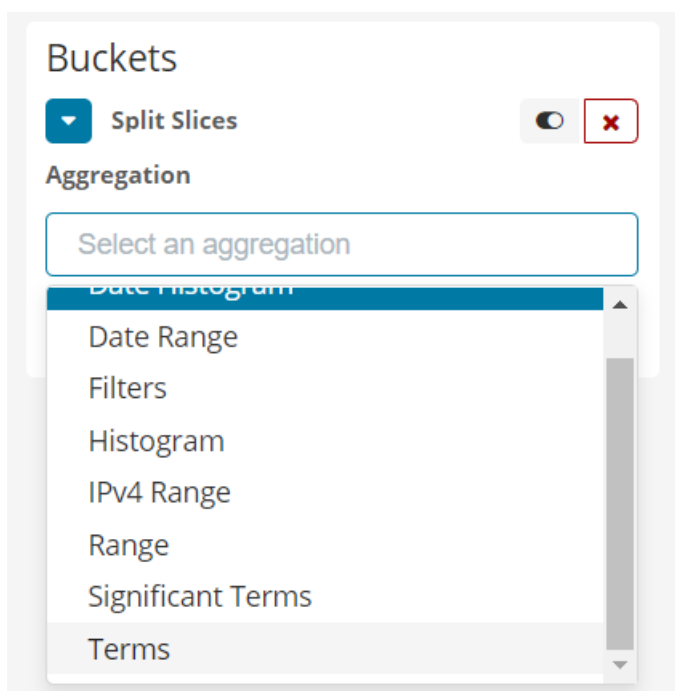
6-38. 先ほど作成した[support\*]を選びます。



6-39. [Split Slices]を選びます。



6-40. [Aggregation]で[Term]を選びます。



6-41. [Filed]で[sentiment.keyword]を選びます。

**Buckets**

☒ Split Slices ☐

**Aggregation**

Terms ▼

**Field**

Select a field

- string
- \_id
- \_index
- \_type
- keywords.keyword
- sentiment.keyword
- transcript.keyword

6-42. [apply changes]を押します。

Add a filter +

**support\***

Data Options

**Buckets**

☒ Split Slices ☐

**Aggregation**

Terms ▼

**Field**

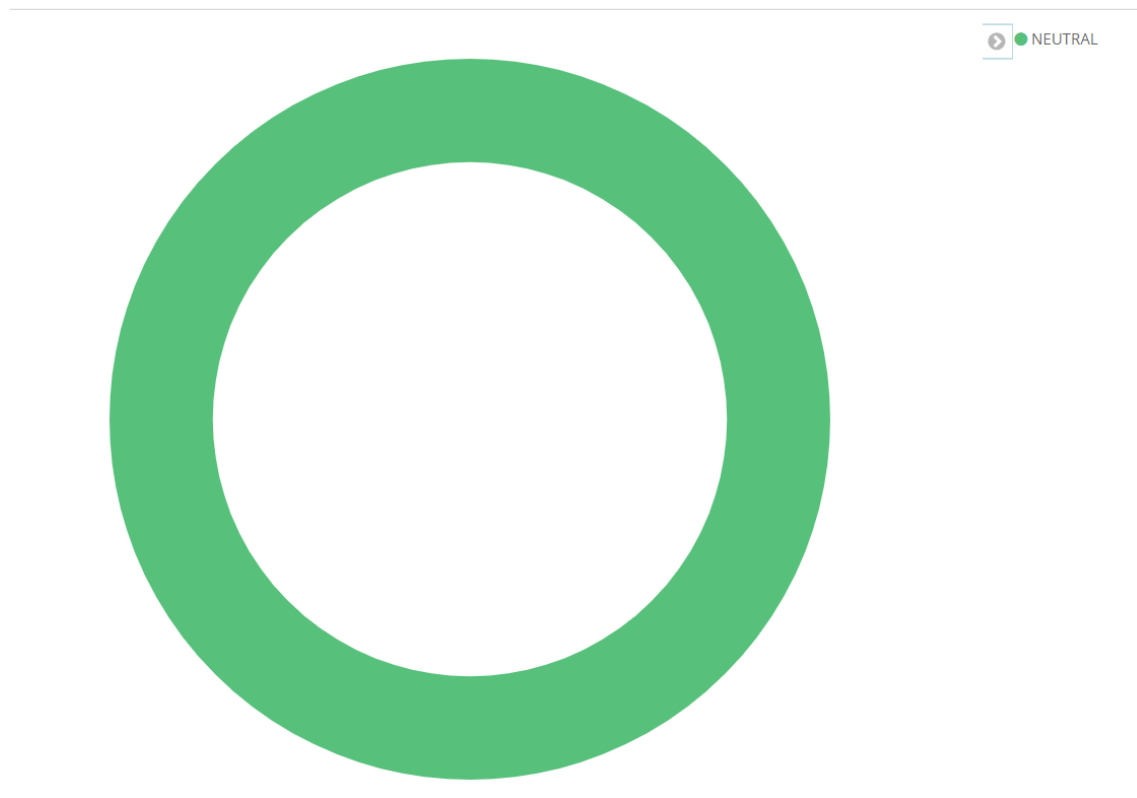
sentiment.keyword ▼

**Order By**

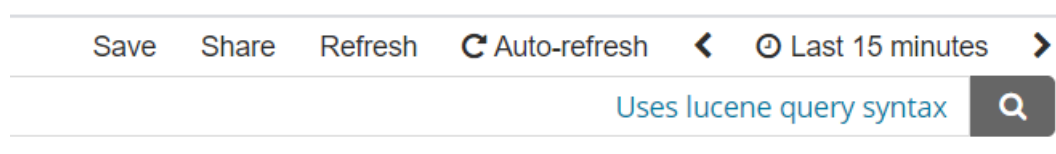
metric: Count ▼

6-43. 先ほど分析した文字列の結果（Positive,Neutral,Negative）が表示されます。まだ 1

件しか分析結果がありませんが、複数分析を行えば増えていきます。



6-44. 右上の[Save]で適当な名前で保存しておきます。



6-45. 再度[Visualization]を押します。

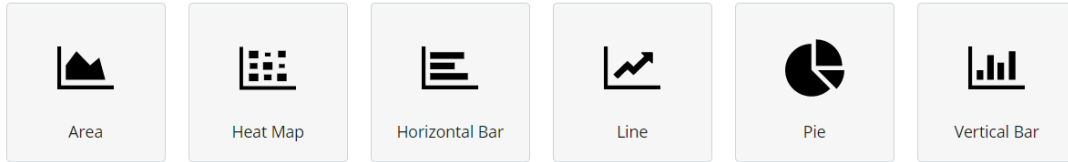
<input type="text" value="Search..."/>	<input type="button" value="+"/>	1-1 of 1
<input type="checkbox"/> Title ↑	Type	
<input type="checkbox"/> circle	Pie	
0 items selected		1-1 of 1

6-46. 先ほどのものが保存されていますが、もう 1 つ作成するので+ を押します。

## Select visualization type

Q Search visualization types...

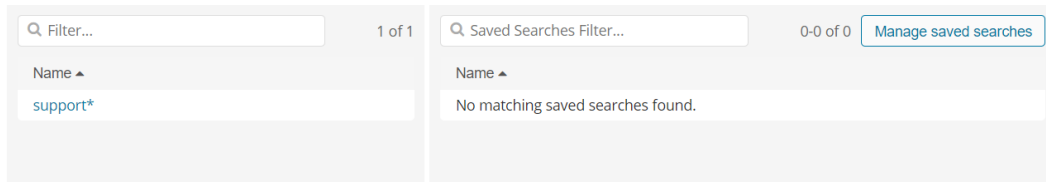
### Basic Charts



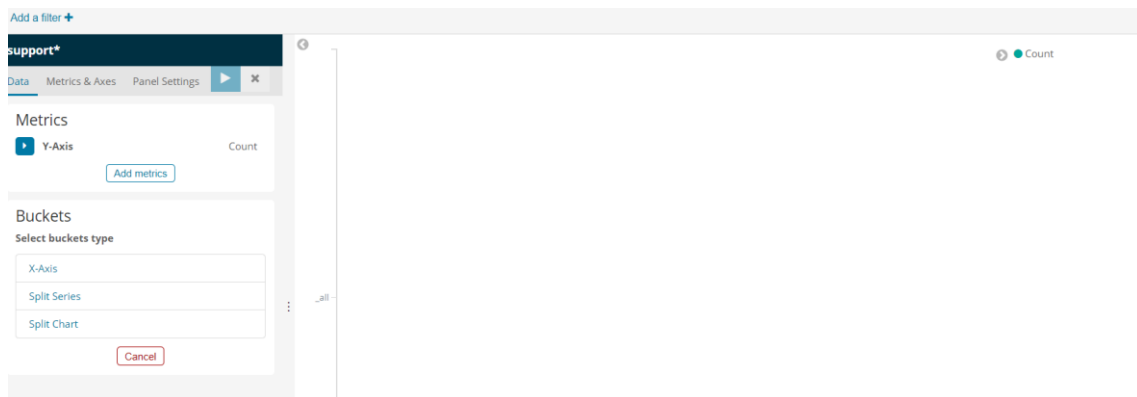
6-47. 今度は Horizontal Bar を選びます。

From a New Search, Select Index

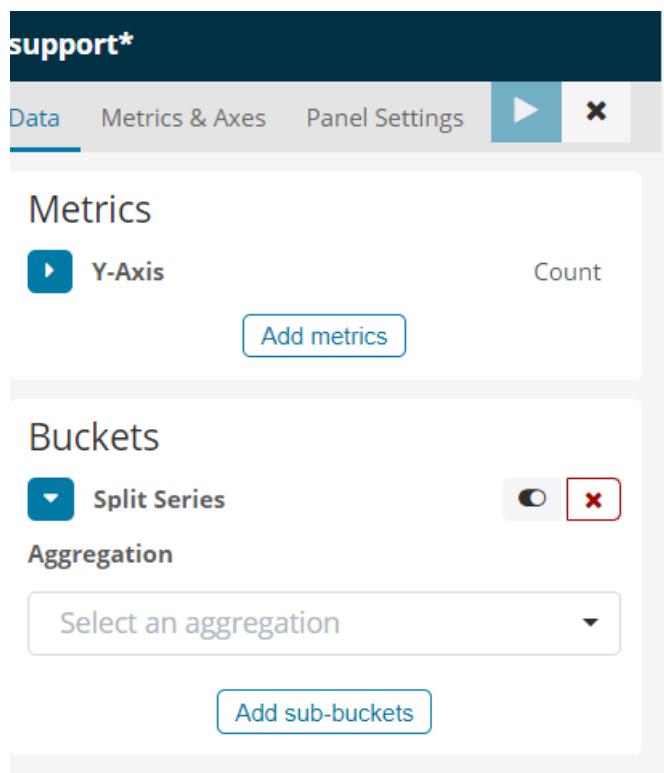
Or, From a Saved Search



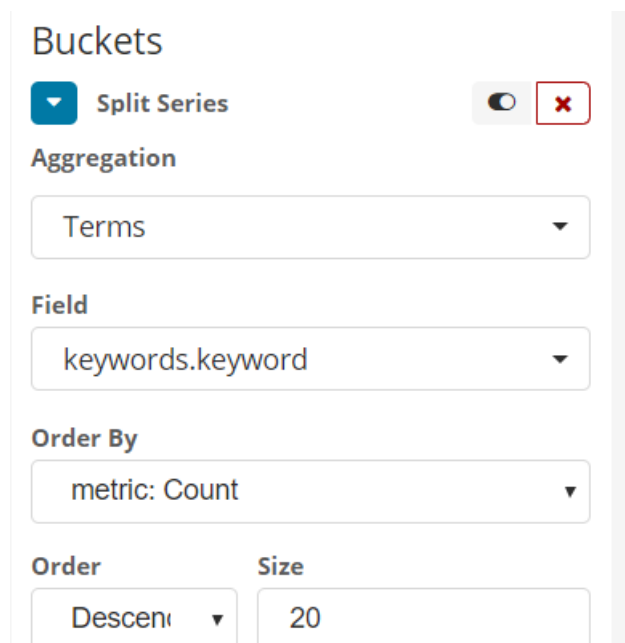
6-48. また[support\*]を選びます。



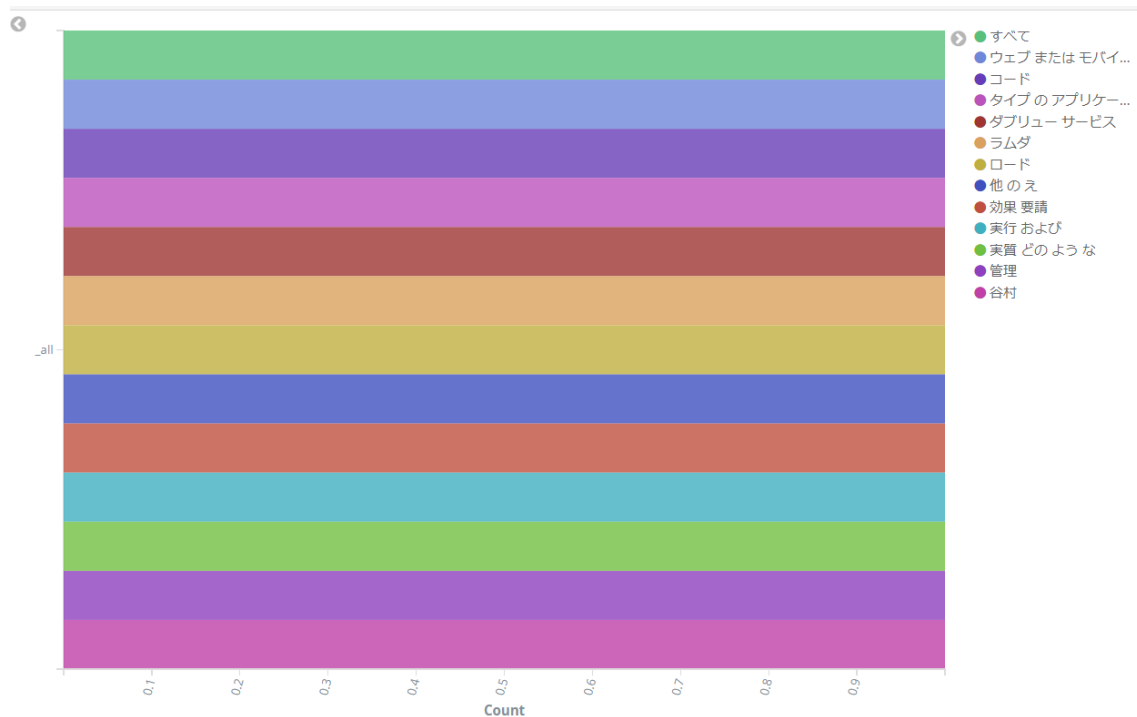
6-49. 今度は[Split Series]を選びます。



6-50. 以下の画面のように設定し、Apply changes を押します。



6-51. 以下のように、取り込んだ単語が件数毎にグラフ化されます。



まだ 1 件しか取り込んでいないため、全て 1 件ですが、複数回繰り返すうちにグラフ生成されていきます。

## 7. リソースの削除

本日生成した Elasticsearch Service, Cloud9, S3 バケットを削除してください。

おつかれさまでした！