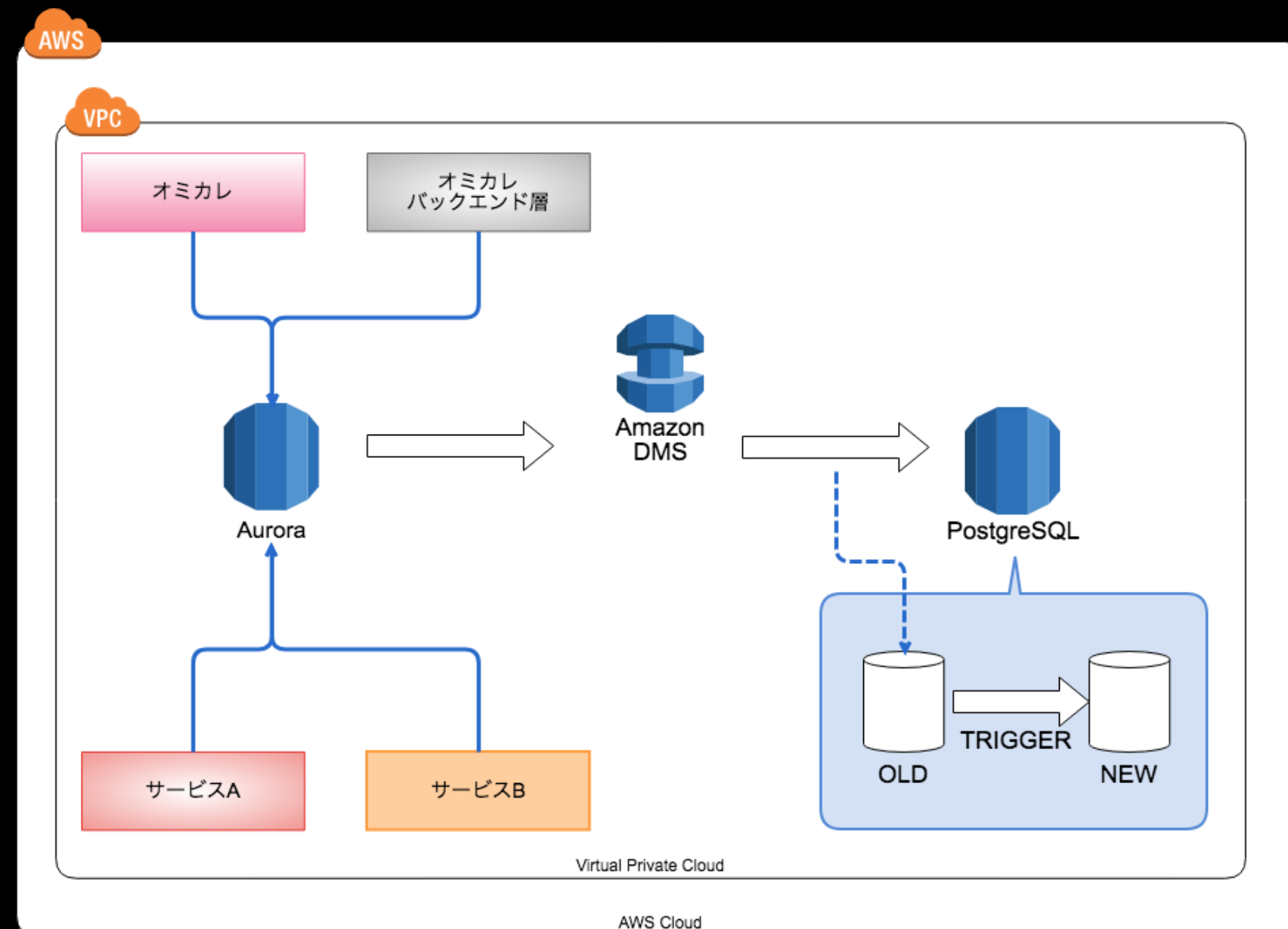


# MySQL(Aurora)からPostgreSQLへの DB移行事例のご紹介

2018-09-23 OSC広島

# 概要

- ➔ **MySQL(AWS Aurora) から PostgreSQLへの異種間DB移行手法**
- ➔ **DBリファクタリングの実施方法**

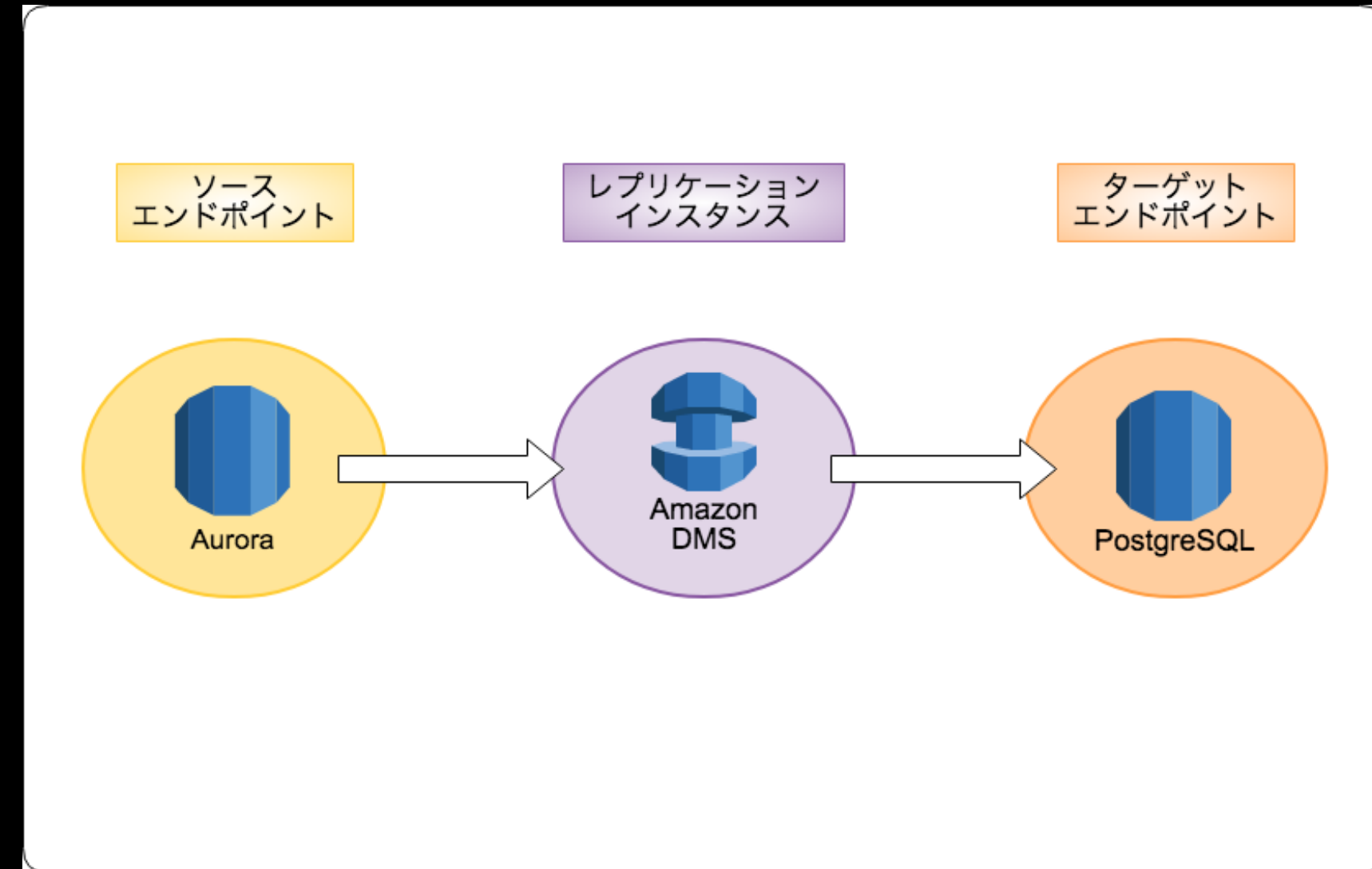


# AWS DMSについて

## Database Migration Service

データの移行を行う事が出来るサービス

※ コピー元、コピー先のどちらかが  
**AWS**のエンドポイントであればレプリケーションをする事が出来る。



# AWS DMSのキーワード

## → ソースエンドポイント

→ データのコピー元。EC2 on DB, RDS, オンプレDB, S3 など。

## → ターゲットエンドポイント

→ データのコピー先。EC2 on DB, RDS, オンプレDB, S3, DynamoDB など。

## → レプリケーションインスタンス

→ レプリケーションを行うインスタンス

→ 内部にストレージを持っていてソースエンドポイントのデータをここに一時保存したりもする。

# AWS DMSのデータ移行の流れ

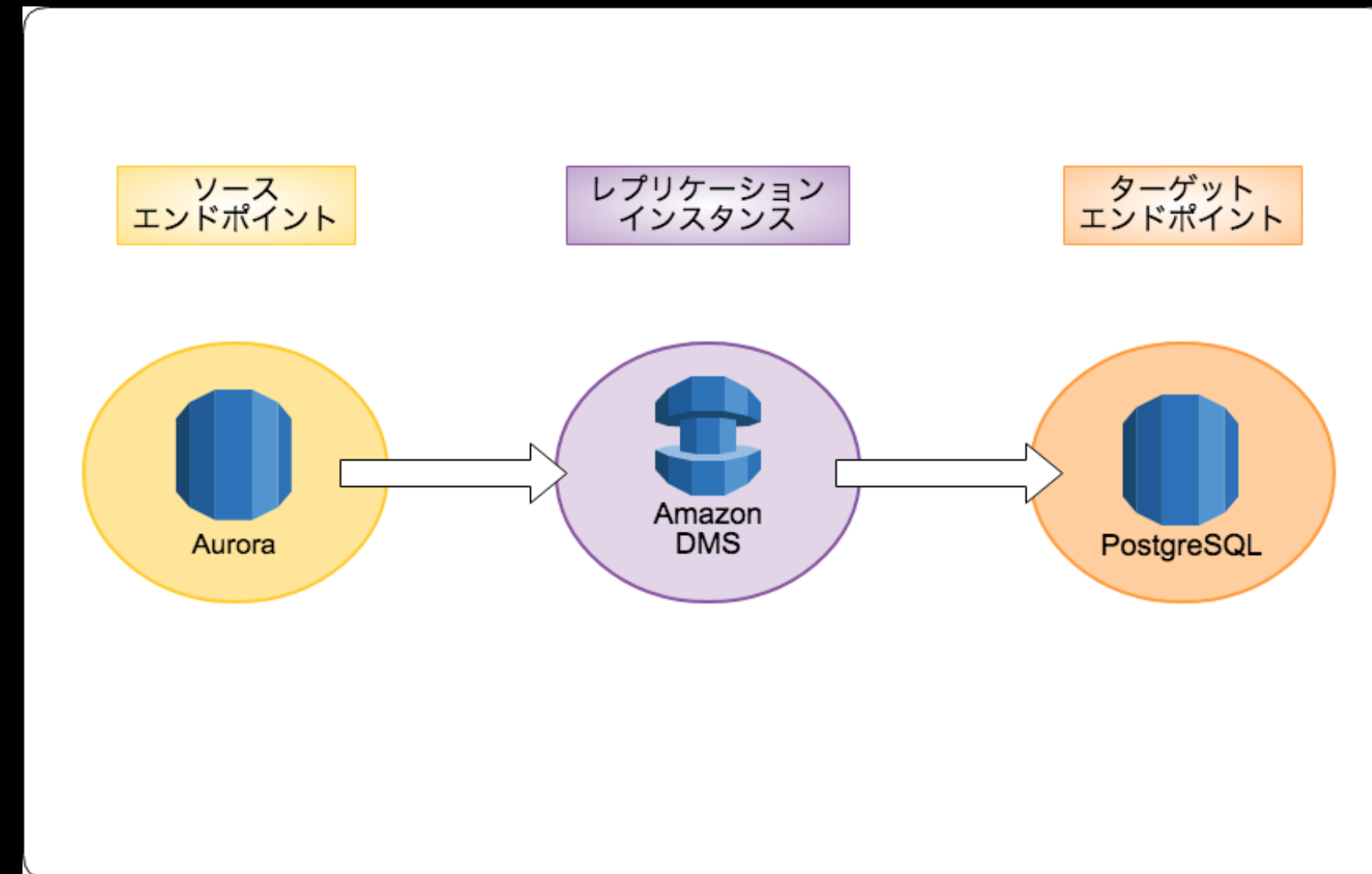
## 1. フルロード

ソースDBからレプリケーションインスタンスにデータをフルロードして、ターゲットDBに書き込む。

## 2. CDC(変更データキャプチャ)

フルロード後はトランザクションログを参照して継続的なレプリケーションを行う事が出来る。

(フルロードのみ行うなども出来る。)



# AWS DMSのコピーテーブル指定の粒度

## → 文字列一致での指定

**users => users** テーブルのみを対象とする。

## → 正規表現での指定

**user\_% => user\_active, user\_access\_log** と正規表現に合致するテーブルのみを対象にする。

# MySQLからPostgreSQLへの移行プロジェクト

1. ソースエンドポイントにMySQLのデータベースを登録する。
2. ターゲットエンドポイントにPostgreSQLを登録する。
3. レプリケーションインスタンスを作成する。
4. 移行したいテーブルを適当なグループに分割してDMSのタスクを作成してレプリケーションする。

例. **user\_replication** => user と user\_～ など.

5. ターゲットのPostgreSQLのテーブルにトリガーを指定してDBリファクタリングを実施する。(一番やりたいこと)

# トリガー設定構文について

```
CREATE TRIGGER 'トリガー名'  
    trigger_time trigger_event  
    ON 'テーブル名' FOR EACH ROW '実行関数'
```

	MySQL	PostgreSQL
<b>trigger_time</b>	BEFORE, AFTER	BEFORE, AFTER
<b>trigger_event</b>	INSERT,UPDATE,DELETE から一つ	INSERT,UPDATE,DELETE から1個以上

※ もっとあるけど、省略



# PostgreSQLの強み(手続き型言語)

名称	言語	配布方法
PL/pgSQL	SQL	基本配布
PL/Perl	Perl	基本配布
PL/Python	Python	基本配布
PL/Java	Java	拡張
PL/v8	JavaScript	拡張

※ その他

## AWS サービス

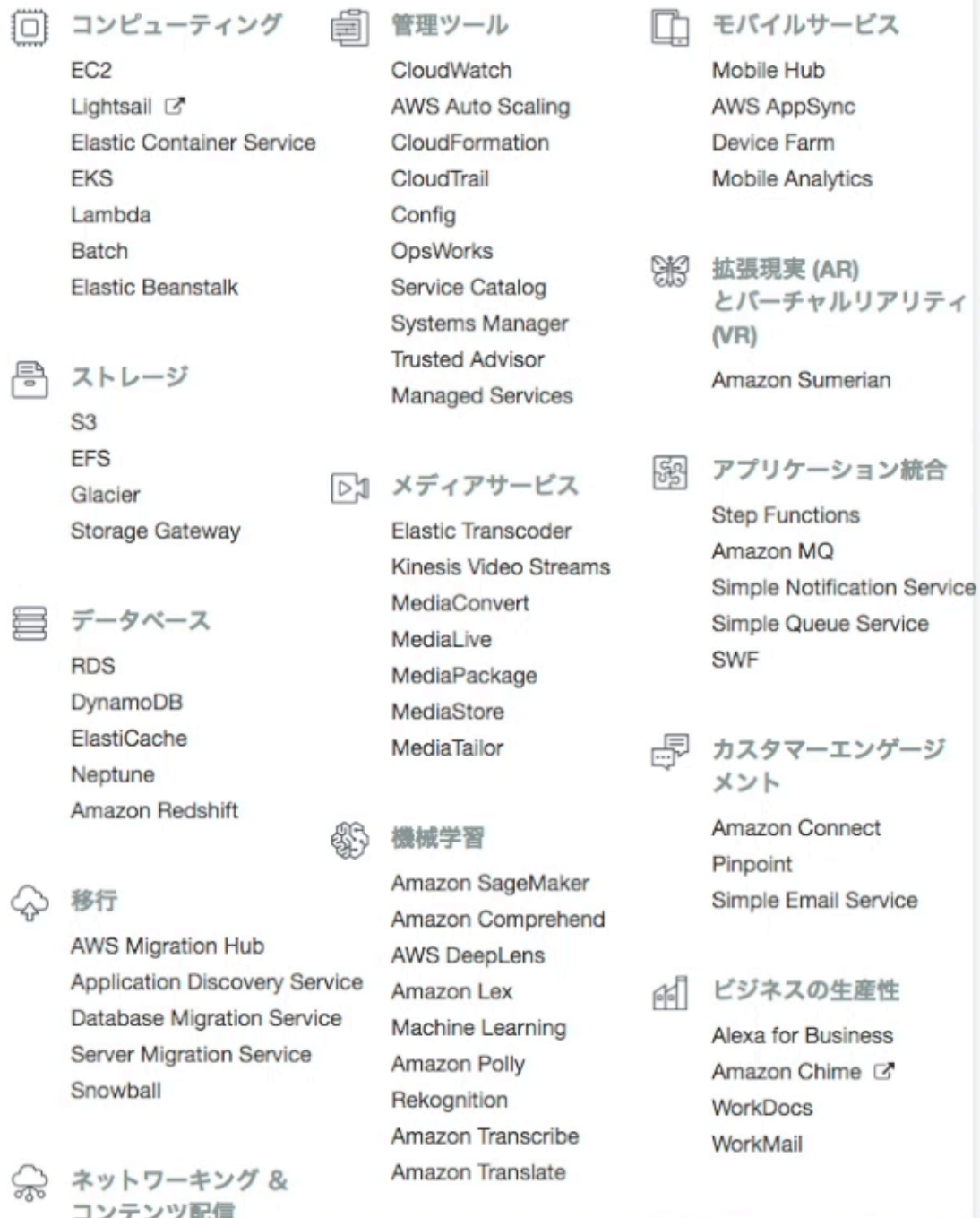
サービスを名前、あるいは機能で検索 (例: EC2、S3、VM、ストレージ)。



## 最近アクセスしたサービス



## すべてのサービス



## 役に立つヒント



## コストの管理

AWS Budgets を使用して、AWS のコスト、使用量、および予約をモニタリングします。 [今すぐ開始](#)



## Organizations の作成

複数の AWS アカウントのポリシーベースの管理に AWS Organizations を使用します。 [今すぐ開始](#)

## AWS を試す

## Amazon SageMaker を使った機械学習

機械学習モデルを最短の時間で構築、トレーニング、デプロイできます。 [詳細はこちら](#)

## Amazon Relational Database Service (RDS)

RDS を使用して、お客様のデータベースを管理し、スケールできます。RDS は Aurora、MySQL、PostgreSQL、MariaDB、Oracle、SQL Server をサポートします。 [詳細はこちら](#)

## AWS Fargate でコンテナを実行

Amazon ECS と統合された AWS Fargate によって、コンテナの実行およびスケールができます。すばやくスケールし、あらゆるサイズのアプリケーションを実行できます。お支払いいただくのは、必要なコンピューティングリソースに対する料金のみです。 [詳細はこちら](#)

## AWS Marketplace

AWS で実行する人気のソフトウェア製品を検索、購入、デプロイできます。 [詳細はこちら](#)

pythomic — root@ikki-vg-pythomic: ~ — ssh • vagrant ssh — 93×25

```
root@~# psql -h target-sample.ci8fuyhipk33.ap-northeast-1.rds.amazonaws.com -U tester -d dms_sample
```

