

# 【初心者向けWebinar】 RDBのAWSへの移行方法（Oracleを例に）

2015/7/28

アマゾン データ サービス ジャパン株式会社

ソリューションアーキテクト 下佐粉 昭

（2015/09/18更新版）

# 初心者向けWebinarのご紹介

- AWSについてこれから学ぶ方向けのソリューションカットの技術Webinarです
- 過去のWebinar資料
  - AWSクラウドサービス活用資料集ページにて公開  
<http://aws.amazon.com/jp/aws-jp-introduction/>
- イベントの告知
  - 国内のイベント・セミナースケジュールページにて告知  
<http://aws.amazon.com/jp/about-aws/events/>  
(オンラインセミナー枠)

# 自己紹介

名前：下佐粉 昭（しもさこ あきら）

Twitter - @simosako

所属：

- アマゾンデータサービスジャパン株式会社
- 技術本部 ソリューションアーキテクト

好きなAWSサービス: Redshift, RDS, S3

- 人間が運用等から開放されて楽になる系のサービスが好きです



# 本日の内容

- このWebinarでは既存RDBをアマゾン ウェブ サービス (AWS) 上の**EC2（仮想サーバ）**、もしくは**RDS(Relational Database Service)**に移行するための手法をご説明します
- AWSへ移行するメリットは、「**運用管理が楽になる**」こと
- Oracleを例に移行手法のパターンや注意点をご説明します
  - 他RDBをご利用の方にも、基本的な考え方として役に立つ内容になっています

# アジェンダ

## Agenda

- AWSの基本
- 移行の背景
- データベース on EC2かRDSかの選択
- 移行のための手法
- まとめ

# アジェンダ

## Agenda

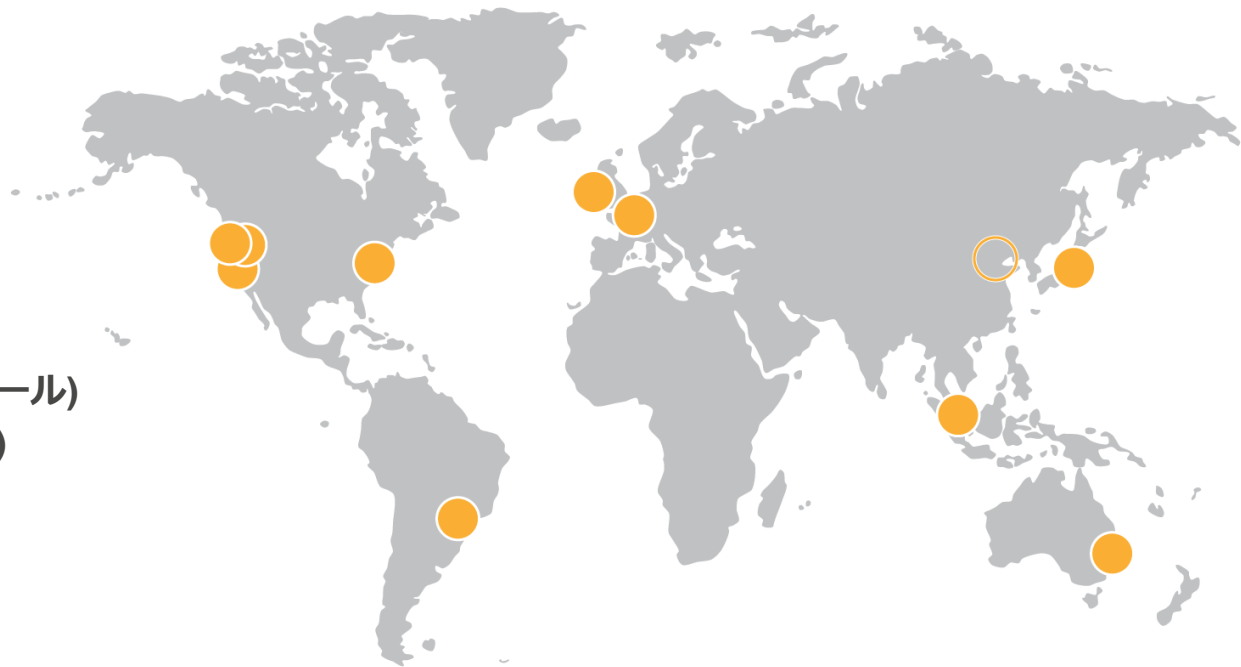
- AWSの基本
- 移行の背景
- データベース on EC2かRDSかの選択
- 移行のための手法
- まとめ

# 世界中に広がるAWSの拠点

## 1つのアカウントで世界中のリージョンが利用可能

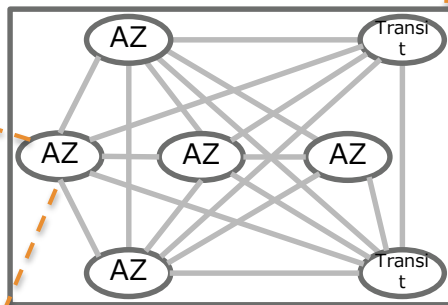
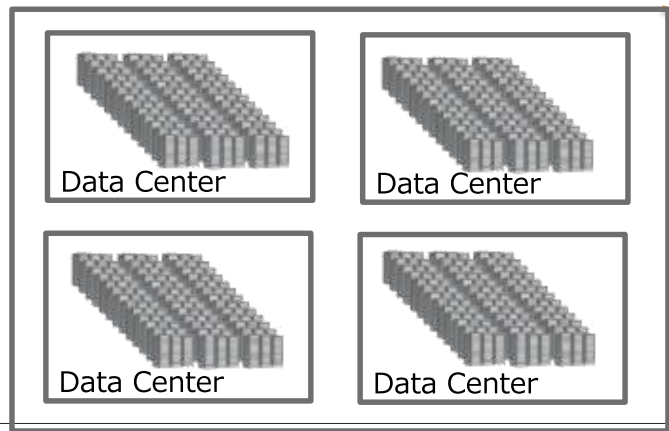
### 11 のリージョン（地域）

1. 米国東部 (北バージニア)
2. 米国西部 (北カリフォルニア)
3. 米国西部2 (オレゴン)
4. 欧州 (アイルランド)
5. 日本 (東京)
6. 南米 (サンパウロ)
7. アジアパシフィック (シンガポール)
8. アジアパシフィック (シドニー)
9. 米国GovCloud \*
10. 中国 (北京) リミテッドプレビュー \*
11. 欧州 (フランクフルト)



# リージョンとアベイラビリティゾーン (AZ)

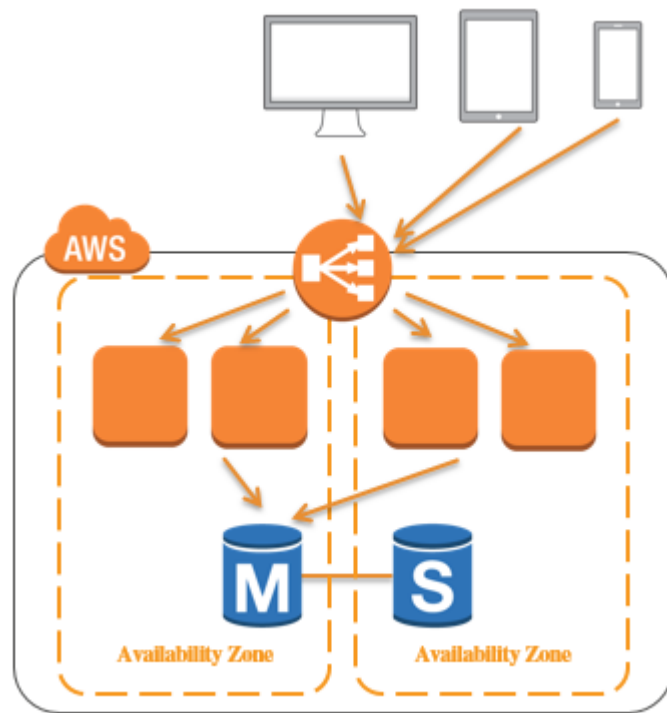
- リージョンは複数のアベイラビリティゾーン (AZ) で構成されています
- AZは、**複数のデータセンター**によって構成され、**高い耐障害性を提供**できる設計になっています





# 複数のAZにまたがった設計で高い可用性を実現

- AWSの基本はマルチAZ構成
  - 複数のAZにまたがった構成
- 利用する基本的なサービス
  - Load Balancer (ELB)
  - WEBサーバ (EC2 + EBS)
  - DBサーバ
    - EC2もしくはRDS



# AWSの豊富なサービス

## お客様のアプリケーション

ライブラリ & SDKs  
Java, PHP,  
.NET,  
Python, Ruby



管理インター  
フェイス  
Management  
Console, CLI



認証とログ  
IAM, Cloud Trail,  
Cloud HSM,  
Config



ディレクトリ  
Directory  
Service



モニタリング  
Cloud Watch,  
Trusted Advisor



コード管理  
CodeDeploy,  
CodeCommit,  
CodePipeline



デプロイと自動化  
Elastic Beanstalk,  
Cloud Formation,  
OpsWorks



### アプリケーション

WorkSpaces, WorkDoc, WorkMail



### モバイルサービス

Mobile Analytics, Cognito, SNS,  
Device Farm



### データベース

RDS, DynamoDB,  
Redshift, ElastiCache



### アプリケーションサービス

AppStream, Cloud Search, SWF,  
SQS, SES, Elastic Transcoder,  
API Gateway



### 分析

Elastic MapReduce,  
Kinesis, Data Pipeline,  
Machine Learning



### コンピュート処理

EC2, Auto Scaling, Elastic,  
Load Balancing, Lambda  
EC2 Container Service



### ストレージ

EBS, S3, Glacier, Storage Gateway



### コンテンツ配信

CloudFront



### ネットワーク

VPC, Route 53, Direct Connect

AWS

Region

AWS

AZ

### グローバルインフラ

リージョン、アベイラビリティゾーン、エッジロケーション



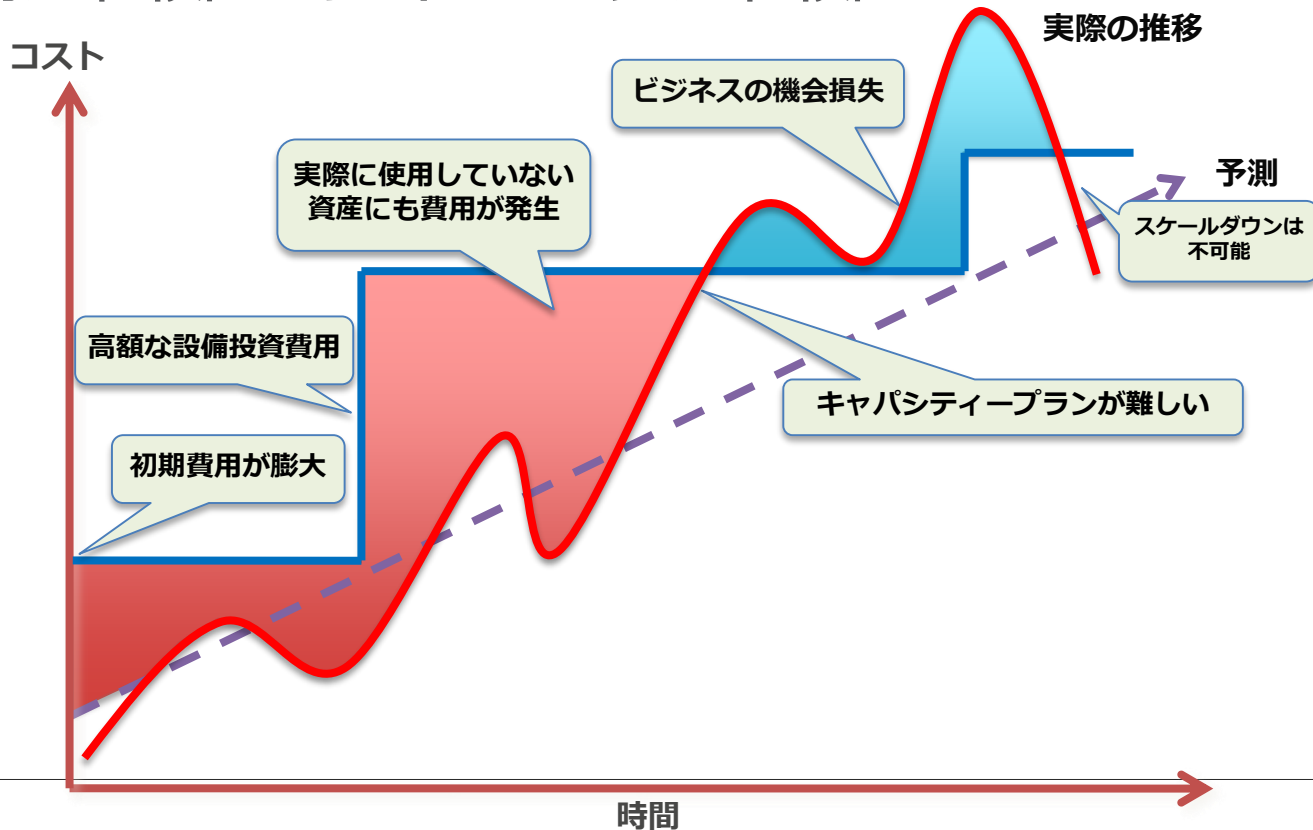
# アジェンダ

## Agenda

- AWSの基本
- 移行の背景
- データベース on EC2かRDSかの選択
- 移行手法
- まとめ

# インフラ全体の課題

- 予測が困難 = サイジングが困難



# 課題：RDBは設計・導入だけでなく運用の負荷が高い

- 運用前（設計・導入）
  - サイジング
  - 導入作業
  - 可用性設計
  - バックアップ設計
- 運用開始後
  - バックアップの自動実行
  - リストア
  - モニタリング
  - サイズ調整（ディスク追加等）
  - SQLチューニング
  - 統計情報の更新
  - フラグメンテーションの解消

# 課題：RDBは設計・導入だけでなく運用の負荷が高い

- 運用前（設計・導入）
  - サイジング
  - 導入作業
  - 可用性設計
  - バックアップ設計
- 運用開始後
  - バックアップの自動実行
  - リストア
  - モニタリング
  - サイズ調整（ディスク追加等）
  - SQLチューニング
  - 統計情報の更新
  - フラグメンテーションの解消



クラウド化で楽になる部分



変わらず残る部分

# クラウドへ移行することで解決できる部分とそうでない部分

- 運用前（設計・導入）

- サーバサイジング
- ストレージサイジング
- 導入作業
- 可用性設計
- バックアップ設計



## サイジングと導入が大幅に軽減

- 数クリックでサーバ起動
- 後からCPUやメモリ、台数を調整可能
- IOPSを保証できるディスク環境

- 運用開始後

- バックアップ&リストア
- パッチのテストと適用
- モニタリング
- サイズ調整（ディスク追加等）
- SQLチューニング
- 統計情報の更新
- フラグメンテーションの解消



# Amazon EC2(Elastic Compute Cloud)



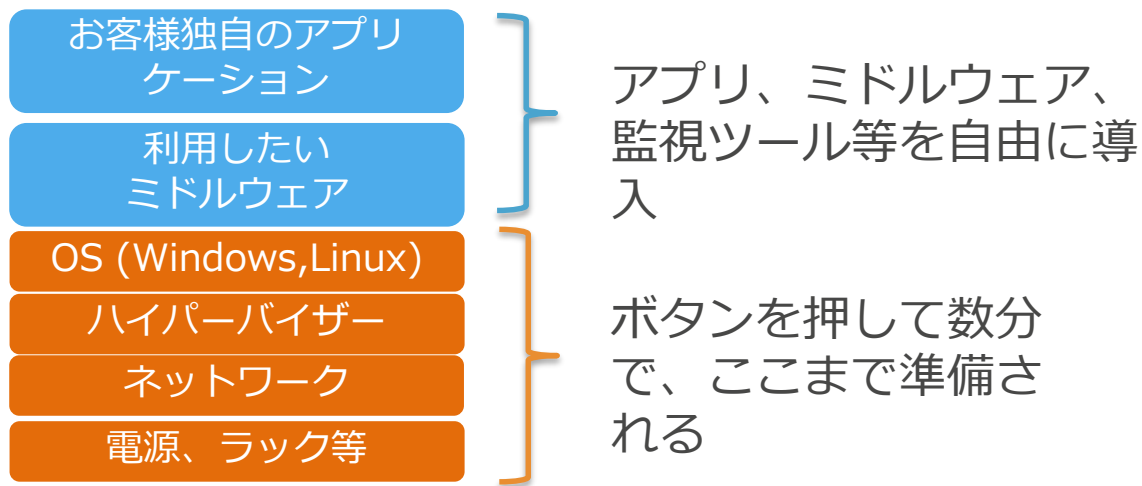
Windows



redhat



- 数分で起動、**1時間ごとの従量課金**で利用可能な仮想サーバ
- 多数のOSをサポート、ライセンス費用込みで従量課金
- 自由にソフトウェアのインストールが可能
- スケールアップ/ダウン、アウト/インが容易に可能

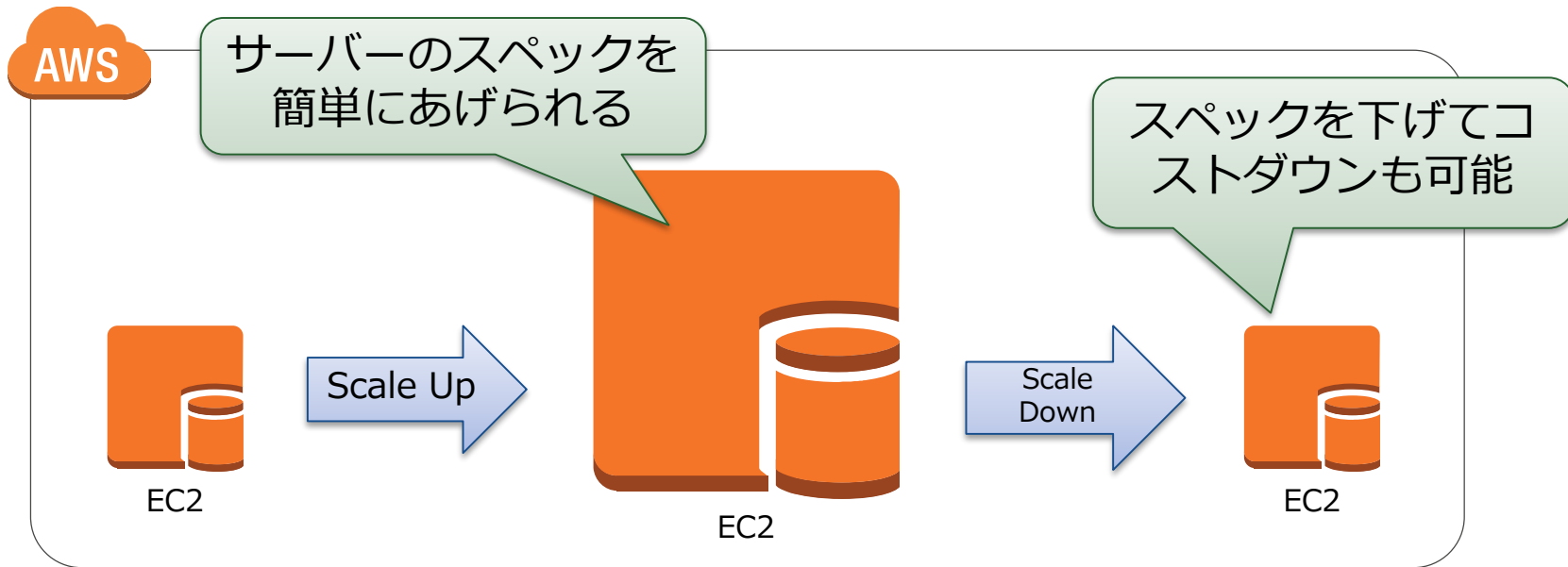






# 柔軟なキャパシティ変更で環境の変化に対応

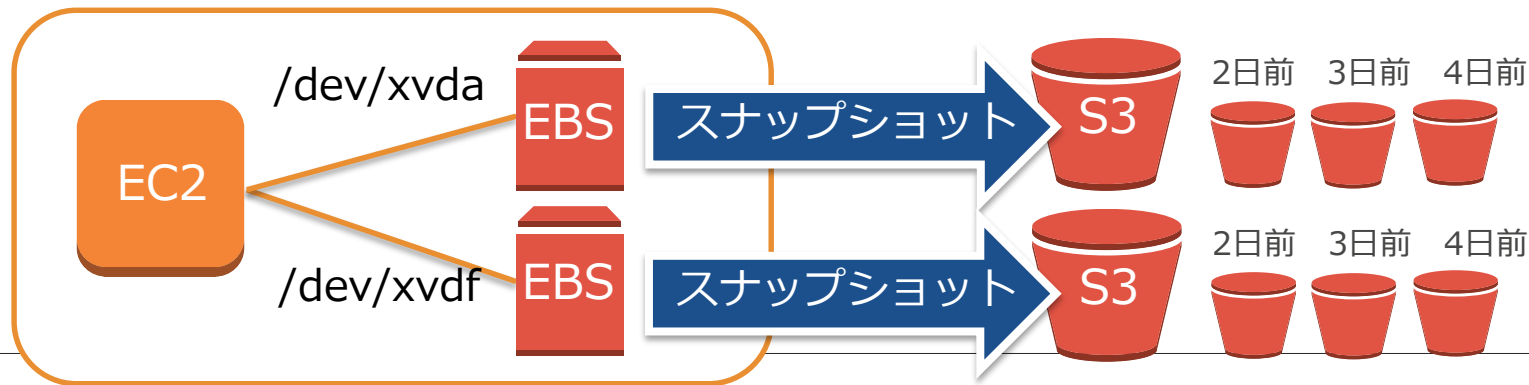
- 仮想CPU数追加、メモリ追加等、スペック変更は容易
- 台数の増減も任意のタイミングで可能
  - 例) AM9時から15時までサーバ台数を増加し、それ以外は減らす





# Amazon EBS (Elastic Block Store)

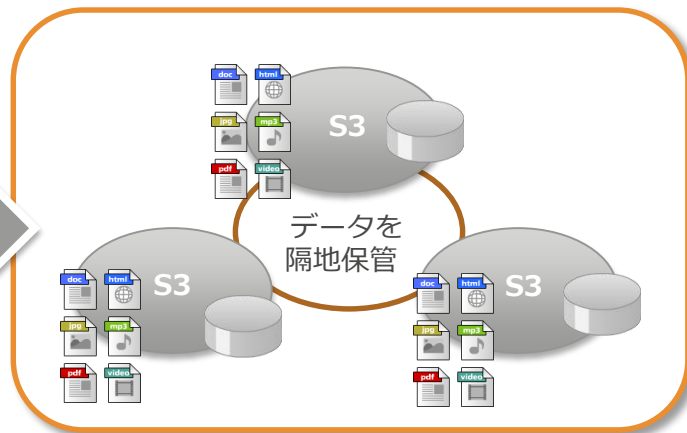
- EC2にマウント可能なブロックストレージ
- 1つのEBSは最大16TB, **最大20,000IOPSまで性能を確保可能**
- 内部的に冗長化されているため、RAID1での冗長化は不要
- サイズ変更可能
- スナップショット機能でS3に差分バックアップ





# Amazon S3 (Simple Storage)

- データ保存・バックアップ用途に向くオブジェクトストレージ
- 自動的に**三箇所以上のDCに隔地保管**
- 設計上のデータ耐久性は、99.9999999999%
- **容量無制限、サイジング不要**
- 従量課金 1GByteあたり月間約4.0円
- WEBの静的コンテンツ配信機能



# クラウドへ移行することで解決できる部分とそうでない部分

- 運用前（設計・導入）

- サーバサイジング
- ストレージサイジング
- 導入作業
- 可用性設計
- バックアップ設計

- 運用開始後

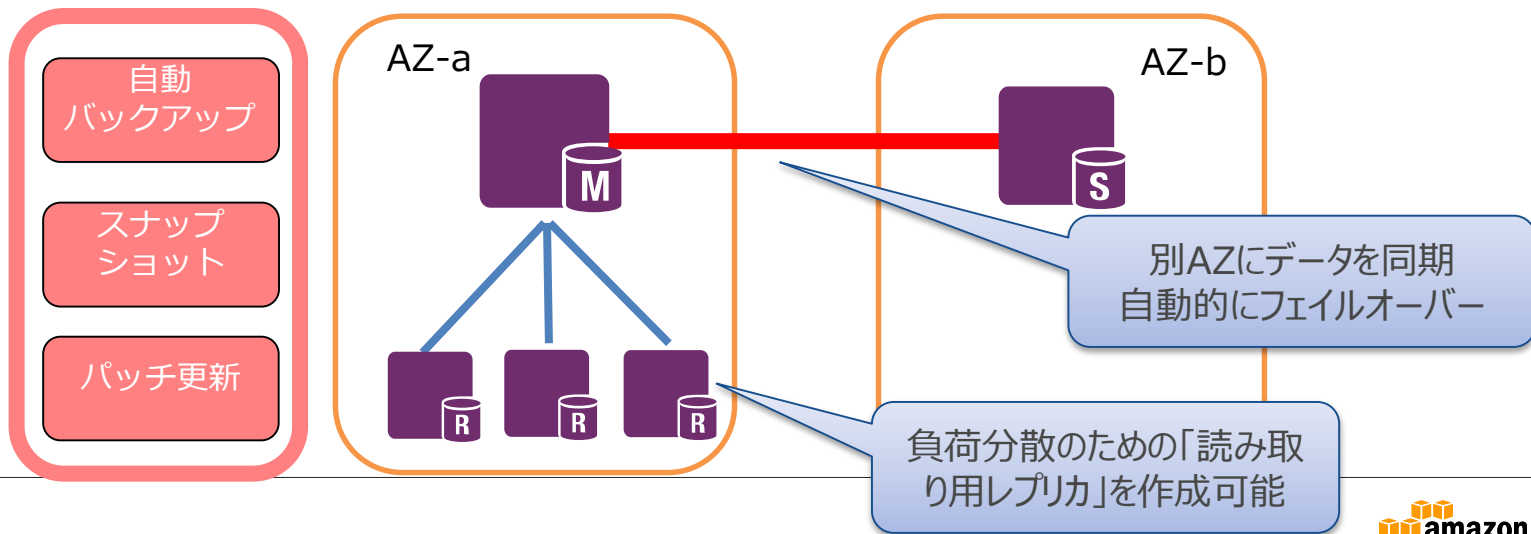
- バックアップ&リストア
- パッチのテストと適用
- モニタリング
- サイズ調整（ディスク追加等）
- SQLチューニング
- 統計情報の更新
- フラグメンテーションの解消

可用性、バックアップ&リストア、  
モニタリングといった、基本要件が  
**設計済みのRDS**を利用することで解消  
**サイズ調整が極めて容易に**



# Amazon RDS (Relational Database Service)

- フルマネージドのRDBMSサービス
  - MySQL、Oracle、SQLServer、PostgreSQL、Auroraから選択可能
- バックアップやフェイルオーバーに対応したDBを数クリックで利用可能
- メンテナンスコストを大幅に削減（パッチ当てやバックアップの自動化）



# RDSでデータベースを作成するのは簡単

- 数クリックでDBが起動
  - DBエンジン
  - インスタンスクラス
  - ディスクの種類とサイズ等を選ぶだけ
- 構成は後から変更可能
- 必須の運用管理機能が実装済み
  - バックアップ（スナップショット）
    - 自動スナップショット
    - 手動スナップショット
  - 同期レプリケーションによる可用性向上
  - 監視（CloudWatch）
- OSへのログイン、常駐アプリの追加等はいできない

### DB 詳細の指定

#### インスタンスの仕様

|               |                                 |
|---------------|---------------------------------|
| DB エンジン       | postgres                        |
| ライセンスモデル      | postgresql-license              |
| DB エンジンのバージョン | 9.4.1                           |
| DB インスタンスのクラス | db.t2.micro – 1 vCPU, 1 GiB RAM |
| マルチ AZ 配置     | はい                              |
| ストレージタイプ      | 汎用 (SSD)                        |
| ストレージ割り当て     | 5 GB                            |

高スループットの作業負荷に対する 100 GB 以下の汎用 (SSD) でのプロビジョニングによって、初期の汎用 (SSD) IO クレジットバランスを使い切った時点で、レイテンシーが大きくなる場合があります。詳細は [ここをクリック](#) をご覧ください。

この DB インスタンスの計画されたワークロードで必要となるコンピューティング能力、ネットワーク、メモリ容量が割り当てられる DB インスタンスのクラスを選択します。 [詳細はこちら](#)。

詳細: db.t2.micro

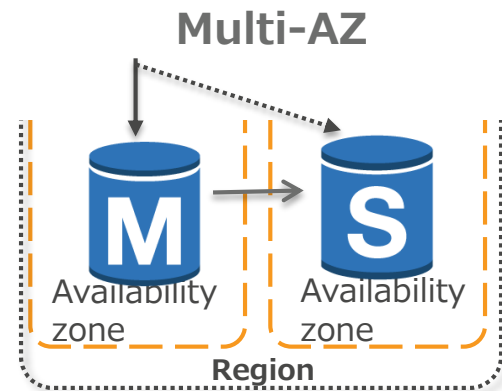
|               |                   |
|---------------|-------------------|
| タイプ           | マイクロインスタンス - 現行世代 |
| vCPU          | 1 vCPU            |
| メモリ           | 1 GiB             |
| EBS 最適化       | いいえ               |
| ネットワークパフォーマンス | 低                 |
| 無料利用枠の対象      | はい                |

# RDSのインスタンスクラス

| DB Instance Class | vCPU | メモリ(GiB) | EBS<br>最適化 | ネットワーク |
|-------------------|------|----------|------------|--------|
| db.t2.micro※      | 1    | 1        | (n/a)      | 低 – 中  |
| db.t2.small※      | 2    | 2        | (n/a)      | 低 – 中  |
| db.t2.medium※     | 4    | 4        | (n/a)      | 低 – 中  |
| db.m3.medium      | 1    | 3.75     | (n/a)      | 中      |
| db.m3.large       | 2    | 7.5      | (n/a)      | 中      |
| db.m3.xlarge      | 4    | 15       | 500Mbps    | 高      |
| db.m3.2xlarge     | 8    | 30       | 1000Mbps   | 高      |
| db.r3.large※      | 2    | 15       | (n/a)      | 中      |
| db.r3.xlarge※     | 4    | 30.5     | 500Mbps    | 中      |
| db.r3.2xlarge※    | 8    | 61       | 1000Mbps   | 中      |
| db.r3.4xlarge※    | 16   | 122      | 2000Mbps   | 高      |
| db.r3.8xlarge※    | 32   | 244      | (n/a)      | 10Gbit |

# RDSはマルチAZデプロイメントに対応

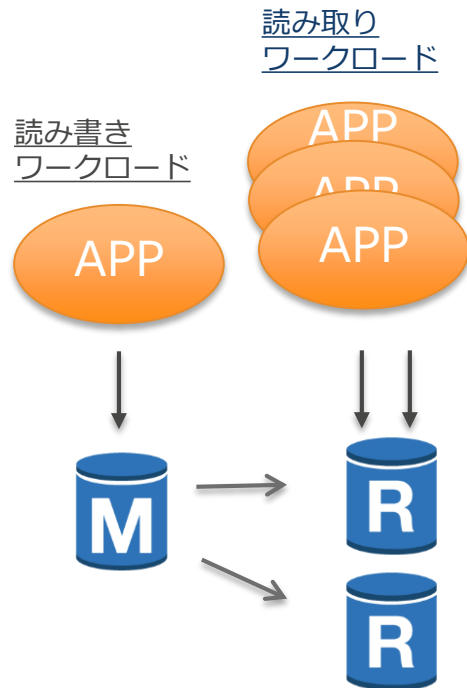
- ワンクリックで**耐障害性**を向上可能なソリューション
  - 高い技術力を持つDBAが行っていた設計をそのままサービス化
  - AWS内部の仕組みで同期レプリケーションを実現（※Data Guardではない）
- 同期レプリケーション＋自動フェイルオーバー
  - アプリ側での対処は必要なし(エンドポイントは変わらない)
  - スタンバイ状態のDBはアクセス不可
- フェイルオーバーの実施タイミング
  - インスタンスやハードウェア障害
  - パッチ適用などのメンテナンス時間
  - 手動リブート時に強制フェイルオーバー指定





# リードレプリカ(RR)機能

- 読み取り専用のレプリカDB
  - 現時点でMySQLとPostgreSQLに対応
  - 5台まで増設可能（※上限緩和申請可能）
  - RRのディスクタイプやインスタンスタイプをソースとは別のタイプに変更可能
- 想定ユースケース
  - 読み取りのスケーリング、BI等の解析処理の分散
  - マルチAZによる耐障害性の代替ではない



リードレプリカ

# クラウドへ移行することで解決できる部分とそうでない部分

- 運用前（設計・導入）

- サーバサイジング
- ストレージサイジング
- 導入作業
- 可用性設計
- バックアップ設計

- 運用開始後

- バックアップ&リストア
- パッチのテストと適用
- モニタリング
- サイズ調整（ディスク追加等）
- SQLチューニング
- 統計情報の更新
- フラグメンテーションの解消

} 残念ながら全ての運用が無くなるわけではありません…

# アジェンダ

## Agenda

- AWSの基本
- 移行の背景
- データベース on EC2かRDSかの選択
- 移行のための手法
- まとめ

# データベース on EC2か？RDSか？

RDSを使うことには導入＆運用面でメリットが多い

- RDBの導入が不要、パッチ適用も容易
- バックアップ＆リストアがビルトイン
- マルチAZへの同期レプリカ設定が容易
- リードオンリーの読み取り専用レプリカのセットアップが容易
- 1時間単位で、OracleやSQL Server等の商用データベースが利用できる

まずはRDSで検討し、適当ではない場合にEC2

# オンプレミス vs. データベース on EC2 vs. RDS

|                    |                    |                    |
|--------------------|--------------------|--------------------|
| App optimization   | App optimization   | App optimization   |
| Scaling            | Scaling            | Scaling            |
| High availability  | High availability  | High availability  |
| Database backups   | Database backups   | Database backups   |
| DB s/w patches     | DB s/w patches     | DB s/w patches     |
| DB s/w installs    | DB s/w installs    | DB s/w installs    |
| OS patches         | OS patches         | OS patches         |
| OS installation    | OS installation    | OS installation    |
| Server maintenance | Server maintenance | Server maintenance |
| Rack & stack       | Rack & stack       | Rack & stack       |
| Power, HVAC, net   | Power, HVAC, net   | Power, HVAC, net   |
| オンプレミス             | データベース on EC2      | RDS                |

お客様がご担当する作業

AWSが提供するマネージド機能

# データベースをEC2に構築する理由①

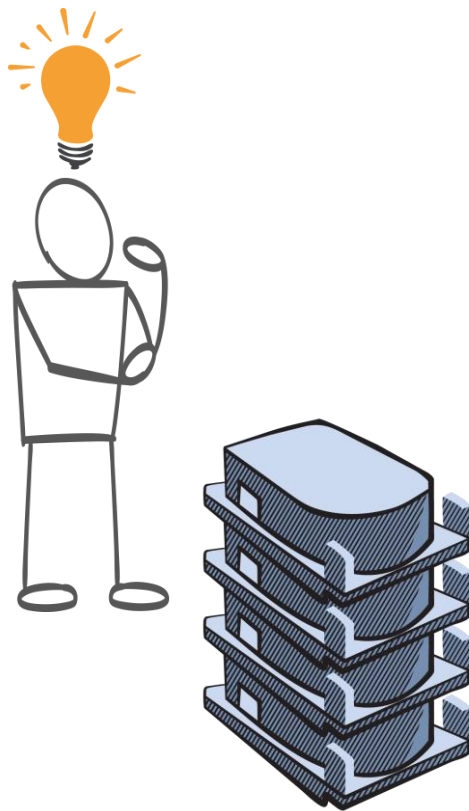
## RDSで用意されたリソースとニーズが合わないケース

- RDSが対応していないRDBやバージョンを選択したい場合
- RDSの最大CPU数やメモリサイズでは不足する場合
- RDSの最大ストレージ量より大きいディスクが必要な場合
  - ストレージ領域の最大サイズは6TB (SQL Serverのみ最大4TB)
  - **[Update] RDS for Oracle でもR3インスタンスが選択可能になりました**
- メンテナンス時間を完全にユーザがコントロールしたい場合

# データベースをEC2に構築する理由②

## チューニングの幅が広い

- OS側のパラメータ調整、常駐プログラム
  - データベースと同じOS上で常駐プログラムを実行可能
  - シェルログインでの作業が必要な場合
  - OS（カーネル）に何か特殊な設定が必要な場合
- RDSが変更に対応していないDBパラメータの変更
- ストレージ領域構成の自由度が高い
  - 多くのボリュームをOSにマウントし高速化
  - REDOログやUNDO表領域のみ別のボリュームに分離
  - 一部の表領域に専用のボリュームを割り当て



# EBSの3つのボリュームタイプ

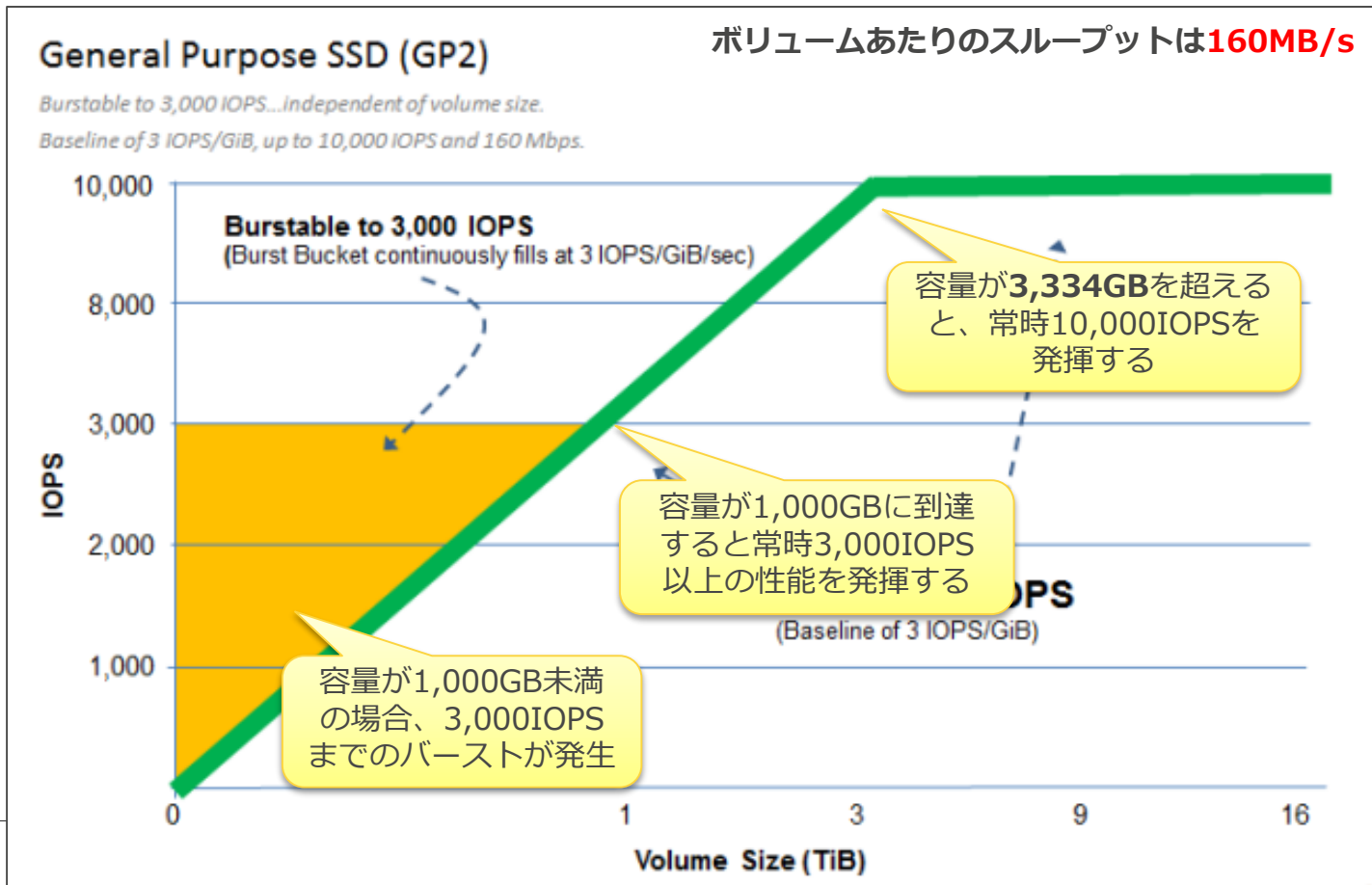
| Volume Type     | General Purpose (GP2) (SSD)  | Provisioned IOPS(SSD)   | Magnetic   |
|-----------------|--|---|--|
| ユースケース          | <ul style="list-style-type: none"> <li>起動ボリューム</li> <li>仮想デスクトップ環境用のディスク</li> <li>開発およびテスト環境用のディスク</li> <li>中小規模のデータベース</li> </ul>   | <ul style="list-style-type: none"> <li>10,000IOPSよりも高いI/Oパフォーマンスを要求するミッションクリティカルなアプリケーション</li> <li>大規模なデータベース<br/>各種NoSQL<br/>各種RDB</li> </ul>              | <ul style="list-style-type: none"> <li>アクセス頻度の低いデータ</li> <li>コストを最重要視する場合</li> </ul>             |
| ボリュームサイズ        | 1GiBから <b>16TiB</b> まで   | 4GiBから <b>16TiB</b> まで  | 1GiBから1TiBまで   |
| IOPS            | <ul style="list-style-type: none"> <li>1GiBあたり3IOPSのベースパフォーマンス。<b>最大値は10,000IOPS</b></li> <li>1,000GiB以下のボリュームでは、3,000IOPSまでバーストが可能</li> <li><b>1,000GB以上のボリュームではバーストの概念はなく、常にベースパフォーマンスを発揮する</b></li> </ul> | <ul style="list-style-type: none"> <li>プロビジョニングしたIOPS値のパフォーマンスを発揮する</li> <li><b>最大IOPSは20,000IOPS</b></li> <li>容量とIOPS値の比率に制限があり、容量の30倍のIOPS値が上限</li> </ul> | 平均100IOPS。数百IOPSまでバースト可能。  |
| スループット          | <ul style="list-style-type: none"> <li><b>170GB未満：128MB/s</b></li> <li><b>170GB以上：128MB/s~160MB/s</b></li> <li><b>214GB以上：160MB/s</b></li> </ul>   | <ul style="list-style-type: none"> <li><b>320MB/s</b></li> </ul>  | <ul style="list-style-type: none"> <li>40~90MB/s</li> </ul>                                      |
| 料金<br>(東京リージョン) | <ul style="list-style-type: none"> <li>1GBあたり1ヶ月0.12ドル</li> </ul>  | <ul style="list-style-type: none"> <li>1GBあたり1ヶ月0.142ドル</li> <li>プロビジョニングしたIOPS値に応じて1IOPSあたり1ヶ月0.114ドル</li> </ul>   | <ul style="list-style-type: none"> <li>1GBあたり1ヶ月0.08ドル</li> <li>100万I/Oリクエストあたり0.08ドル</li> </ul> |

RDBでは大きめのGP2を複数束ねるのが基本



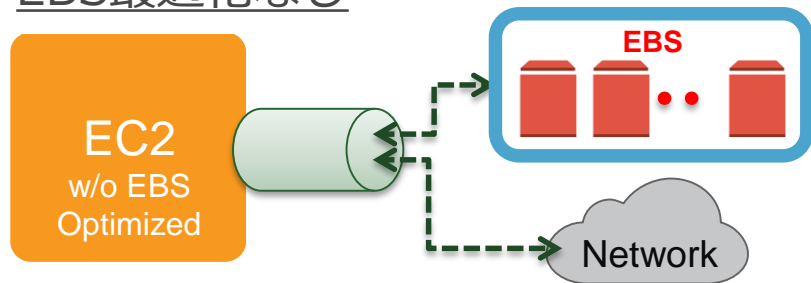
# General Purpose (GP2) (SSD)

ベースパフォーマンス



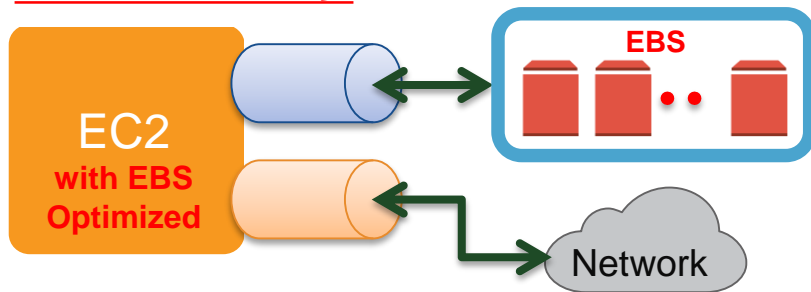
# EBS最適化インスタンスによるEBS帯域の確保

## EBS最適化なし



- EBS最適化を有効にすることで独立したEBS帯域を確保
- 大きいインスタンスタイプほど使える帯域が広い

## EBS最適化あり



| インスタンスタイプ  | EBS帯域                  |
|------------|------------------------|
| c3.xlarge  | 500 Mbps (62.5 MB/sec) |
| c3.4xlarge | 2,000 Mbps(250 MB/sec) |
| c4.2xlarge | 1,000 Mbps(125 MB/sec) |
| c4.8xlarge | 4,000 Mbps(500 MB/sec) |

# 【ご参考】 Oracle License for AWS



クラウド・コンピューティング環境における Oracle ソフトウェアのライセンス

本資料は、以下のベンダーが提供するクラウド・コンピューティング環境に適用されます:

**Amazon Web Services**

– Amazon Elastic Compute Cloud (EC2), Amazon Simple Storage Service (S3)

- プロセッサライセンスでEditionによって係数が異なる。
- Standard EditionおよびStandard Edition One (係数0.25)
  - 1 台の EC2 (コア数 4) ⇒ 1 Processor ライセンス
  - 1 台の EC2 (コア数 8) ⇒ 2 Processor ライセンス
- Enterprise Edition (係数0.5)
  - 1 台の EC2 (コア数 4) ⇒ 2 Processor ライセンス
  - 1 台の EC2 (コア数 8) ⇒ 4 Processor ライセンス

参考URL <http://www.oracle.com/jp/store/cloud-lic-170290-ja.pdf>

# アジェンダ

## Agenda

- AWSの基本
- 移行の背景
- データベース on EC2かRDSかの選択
- 移行のための手法
- まとめ

# OracleのAWSへの移行方法

- この章の情報は、AWSホワイトペーパー「Strategies for Migrating Oracle Database to AWS」から引用しています
- 利用するスクリプト等は以下のホワイトペーパーを参照してください
- Strategies for Migrating Oracle Database to AWS (PDF)
  - <http://d0.awsstatic.com/whitepapers/strategies-for-migrating-oracle-database-to-aws.pdf>

# DB移行の手法 - その前に確認すべきこと

- 移行データサイズ
- 許容可能なダウンタイム
- AWSとのネットワーク速度

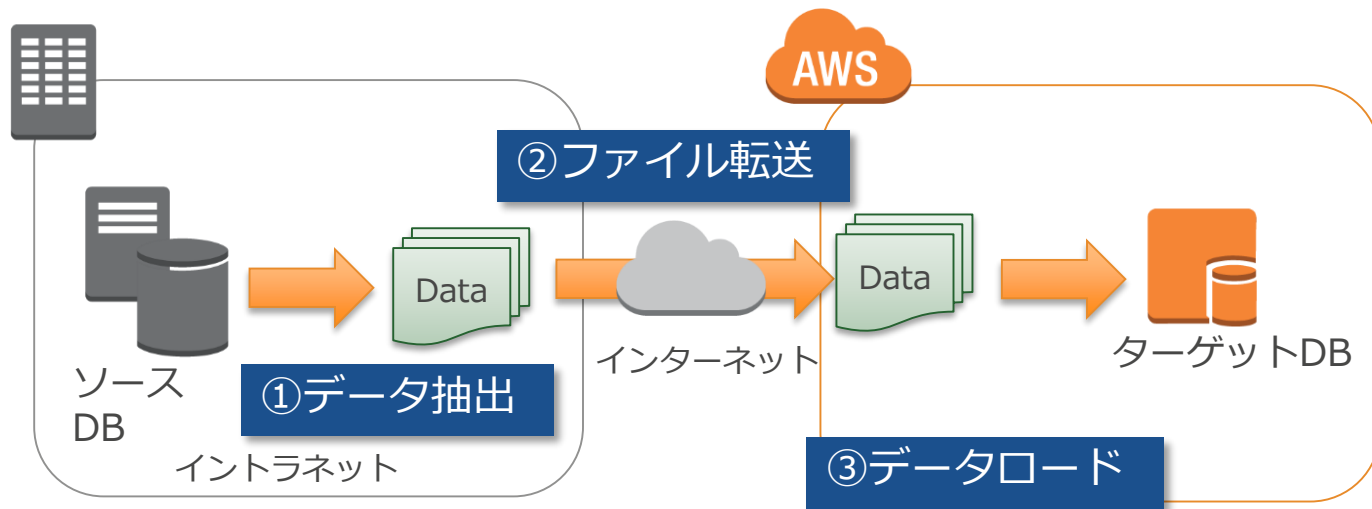
サイズと時間。サイズが小さく、時間が長い方が、移行方法の選択肢が多くなる

- 通信経路暗号化の必要性
  - SCP、VPN、専用線
  - ZIPファイルの暗号化…

移行元-AWS間通信中の暗号化方法

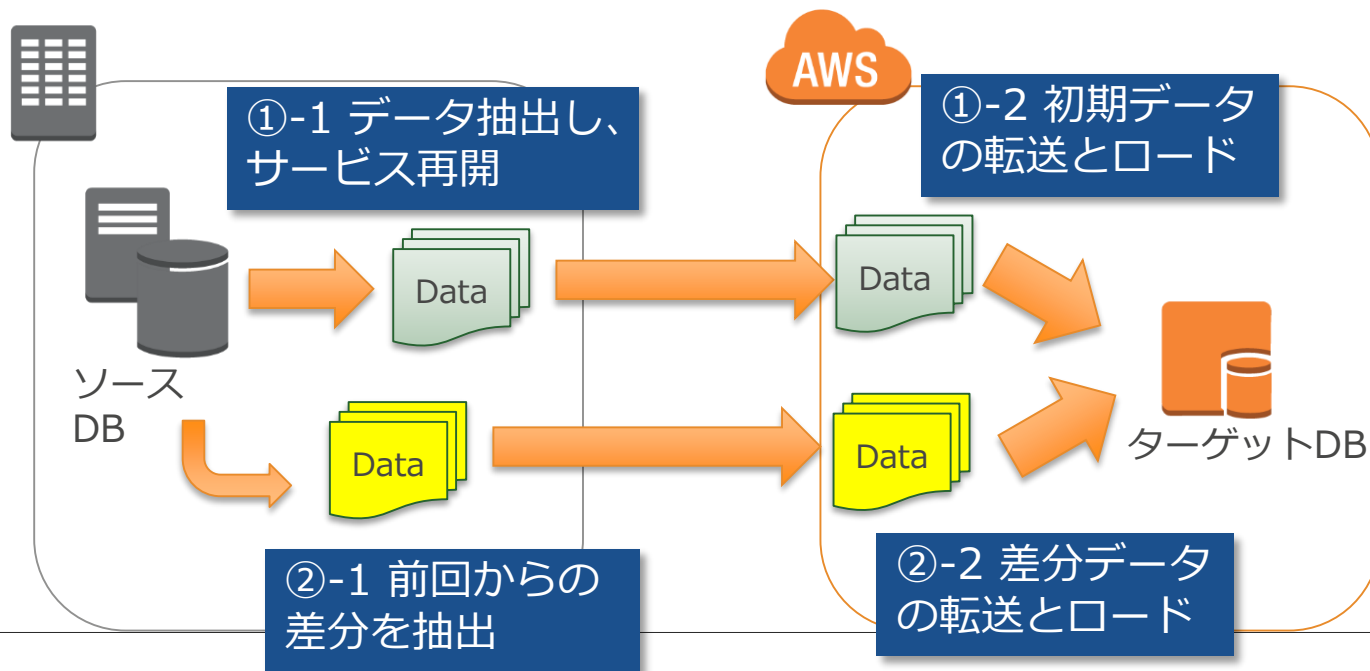
# 移行の考え方①ワンステップ移行

- 抽出～ファイル転送～ロードまでを一度に実施するシンプルな方法
- 完了までDBは停止。1～3日間程度DBが使えない時間が許容される前提
- 小規模DBに向く



# 移行の考え方② 2ステップ移行

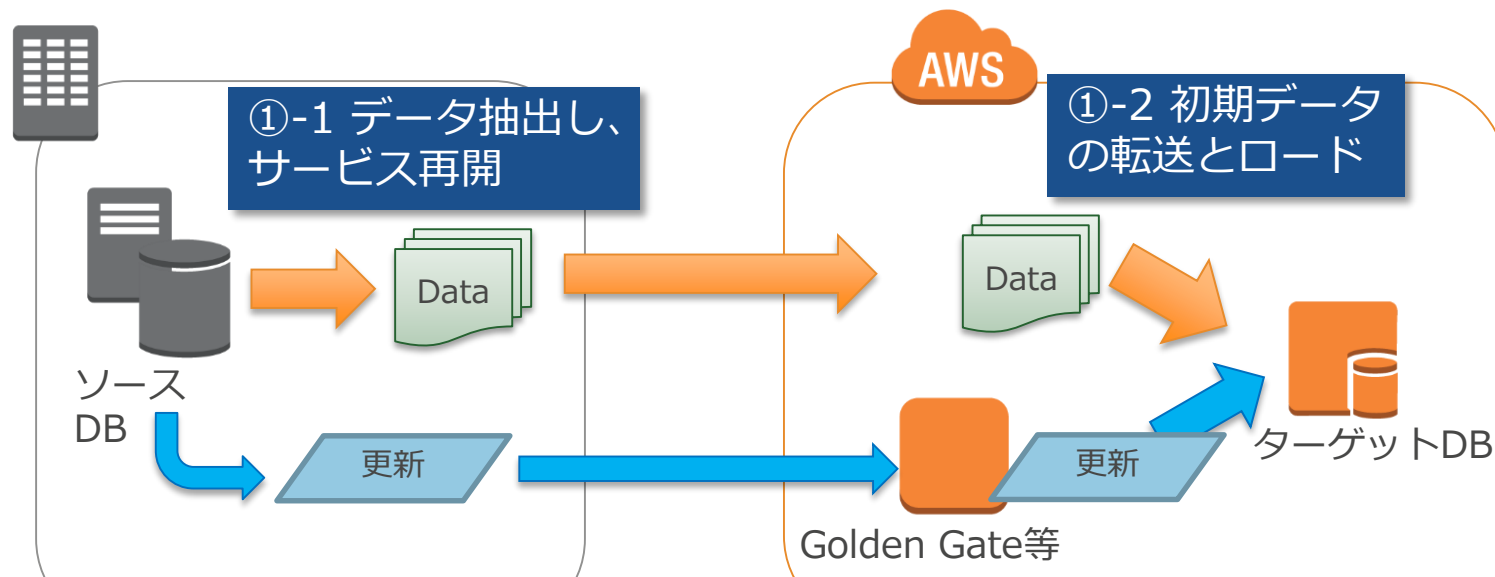
- 初期データのロードと、最新データのロードの二段階に分けた方法
- サービス停止時間を短くしたい中~大規模向け





# 移行の考え方③ゼロ・ダウンタイム移行

- 大規模、もしくはダウンタイムがほとんど取れないシステム向け
- ソースDBから**更新データを継続的に**ターゲットDBに転送し、更新内容を反映することでソースとターゲットを常に同じ内容に保つ
  - Continuous Data Replication もしくは Continuous Data Protection (CDP)
  - Oracle GoldenGate、もしくはAttunity Replicateといったソフトウェアの導入が必要



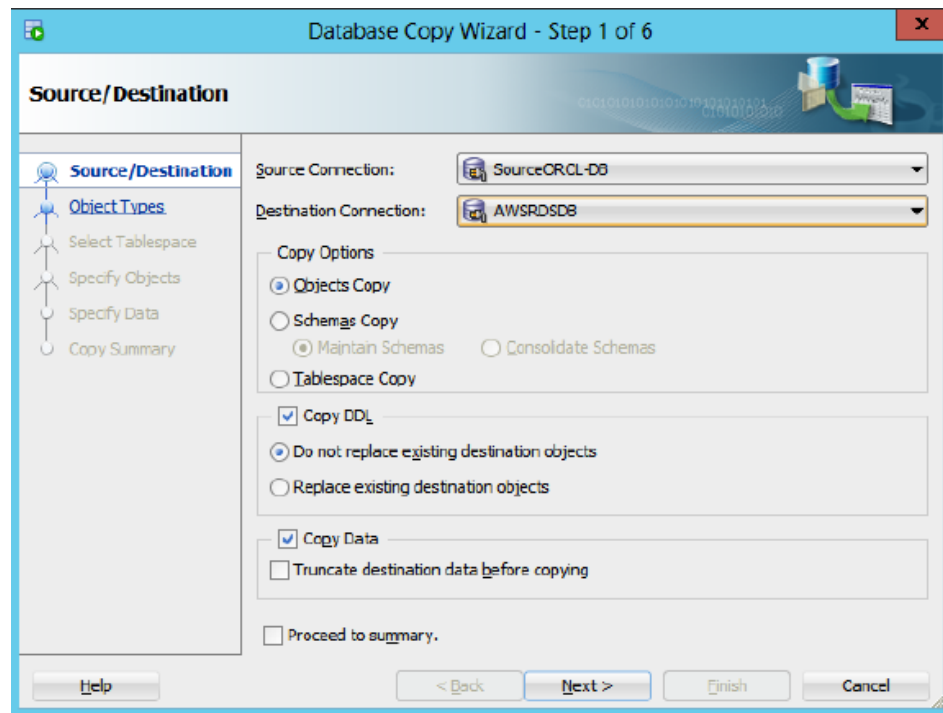
② 表が更新されるたびに、**継続的に**更新データが伝搬、反映される

# DB移行手法（Oracle）

| DB移行手法                       | 移行サイズ（目安） | 対象移行先                          | 推奨サイズ・状況                                  |
|------------------------------|-----------|--------------------------------|---|
| SQL Developerの Database Copy | ~200MB    | Oracle on EC2<br>もしくはRDS       | 小規模DBの移行<br>テスト・評価環境の構築                   |
| マテリアライズド・ビュー+DB LINK         | ~500MB    | Oracle on EC2<br>もしくはRDS       | 小規模かつ、特定の表のみ移行                            |
| SQL*Loader ★                 | ~10GB     | Oracle on EC2<br>もしくはRDS       | 小~中規模。いくつかの表のみ移行                          |
| Export/Import<br>※非推奨        | ~10GB     | Oracle on EC2<br>もしくはRDS       | 小~中規模。DB全体を移行<br>10gR2以降はサポート外            |
| Data Pump<br>（データ・ポンプ）★      | ~5TB      | Oracle on EC2<br>もしくはRDS       | 中~大規模（10GB~ 5 TB）の DB移行                   |
| 外部表                          | ~1TB      | Oracle on EC2<br>もしくはRDS       | 外部表をすでに使っている環境のみ推奨                        |
| RMAN ★                       | 全てのサイズに対応 | Oracle on EC2のみ<br>（RDSでは使えない） | RMANバックアップがすでに AWS(S3)上にある。もしくは 5TB以上のサイズ |
| Golden Gate ★                | 全てのサイズに対応 | Oracle on EC2<br>もしくはRDS       | ゼロ・ダウンタイムの移行が必要な場合                        |

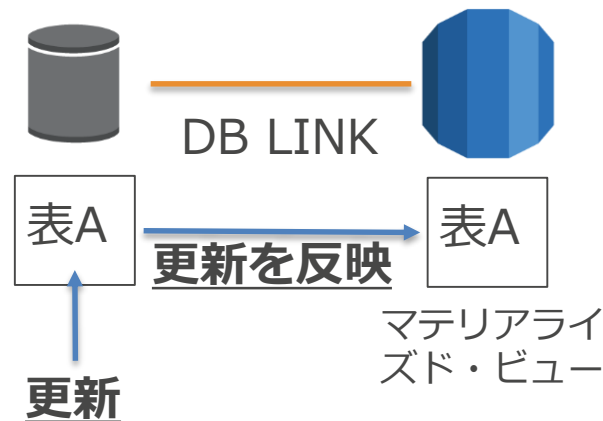
# Oracle SQL DeveloperのDatabase Copy

- Oracle SQL Developer の“Database Copy Wizard”を利用したワンステップ方式の移行
- 移行先までGUIで指定可能
- サイズが小さく（数百MB以下）、ネットワーク速度が十分であればお手軽な方法
- テスト・評価用



# マテリアライズド・ビュー + DB Link

- マテリアライズド・ビューはソース表が更新されると、そのソース表を元に構成されたターゲット表も更新される仕組み（計算済表機能の一種）
- DB間を繋ぐDB LINKを組み合わせることで、ソース側DBの表が更新されると、ターゲットDB上のマテリアライズド・ビューの更新が可能
- DB LINKで繋ぐため、DB間の通信をSQL\*NETで実施している点に注意（暗号化する場合はAdvanced Security Optionが必要）
- ごく一部の表だけリアルタイムにターゲットDBに反映したい場合に有効**



# マテリアライズド・ビュー + DB Linkでの移行手順 (抜粋)

## 1. DB LINKを作成@ターゲット

```
CREATE DATABASE LINK SourceDB_Lnk CONNECT TO SourceUser IDENTIFIED BY PASS USING  
' (desceiption=(address=(protocol=tcp) (host=your.domain) (port=1521))  
(connect_data=(sid=ORACLCRM)))'
```

## 2. マテリアライズド・ビュー・ログを作成@ソース

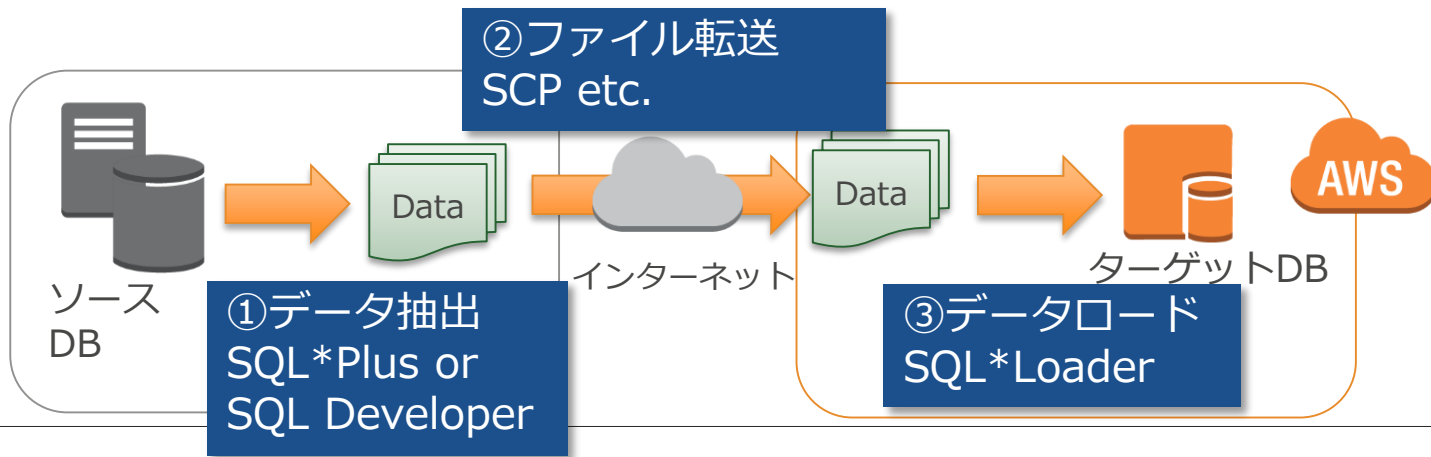
```
CREATE MATERIALIZED VIEW LOG ON customers (※対象の表それぞれに対して作成)
```

## 3. マテリアライズド・ビューを作成@ターゲット

```
CREATE MATERIALIZED VIEW customer BUILD IMMEDIATE REFRESH FAST AS  
SELECT * FROM customer@ SourceDB_Lnk (※対象の表それぞれに対して作成)
```

# SQL\*Loader

- SQL\*LoaderはDDLやデータ（CSV等）をDBに投入するコマンド
- ソースの表データはCSV形式等、メタデータをDDLとして抜き出し、そのファイルを転送。ターゲットDBでSQL\*Loaderで投入
- 速度効率は後述の方法に劣るが、テキストファイルなので、データの修正や、DDLの変更など細かい調整が効くのがメリット
  - 小規模(10GB未満)、限定的な表に対して、ワンステップでの移行

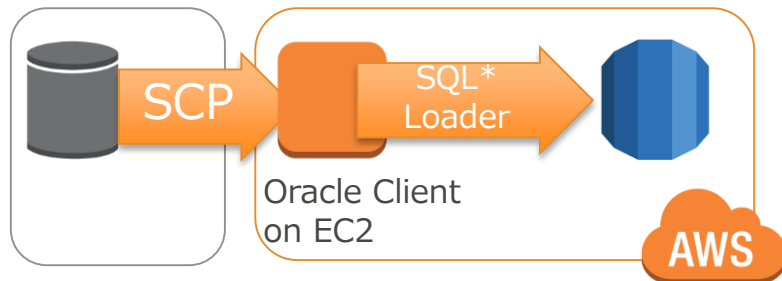
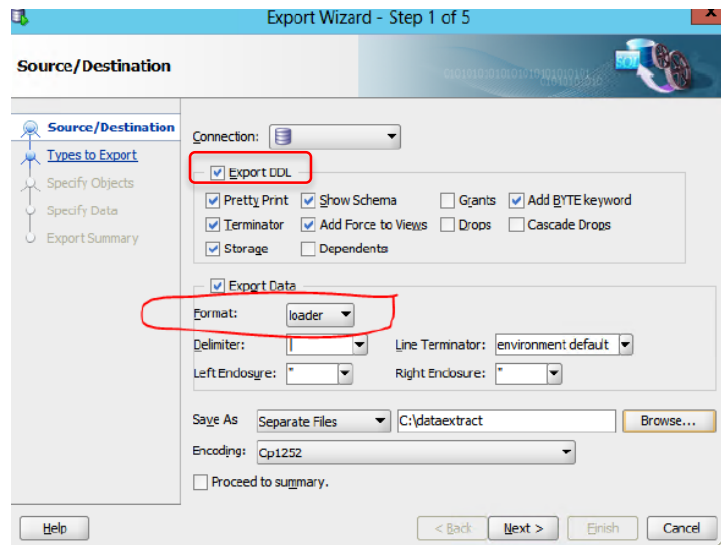


# SQL\*Loaderを使った移行：ポイント

- SQL DeveloperのExport Wizardで「Export DDL有り」、フォーマットを「loader」で実行（もしくはSQL\*Plus）
- ファイルを**圧縮して**SCP転送
- SQL\*Plusでデータ投入。INDEXメンテナンスをスキップしつつダイレクトパスロードをすることで速度アップ可能。ただし後で全インデックスのREBUILDが必要

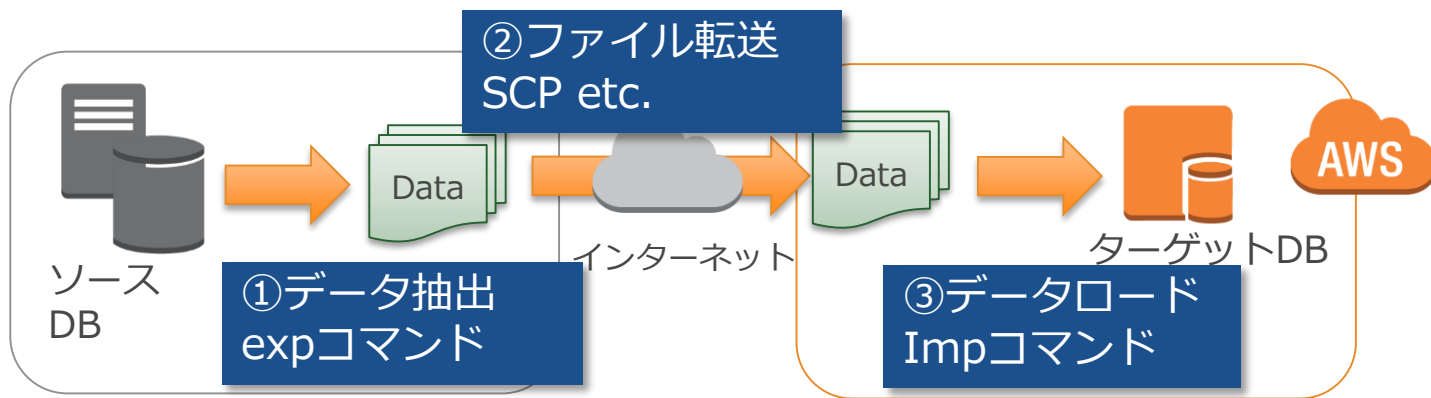
```
> sqlldr userid= ... data=load.dat direct=y  
skip_index_maintenance=true errors=0  
:  
> ALTER INDEX ind1 REBUILD
```

- ターゲットDBがRDSの場合、「Oracleクライアント導入済みのEC2インスタンス」を一時的に作成して、そこからSQL\*loaderでデータ投入する。移行が終わったらインスタンスを削除する



# Export/Importコマンド (※非推奨)

- SQL\*Loaderパターンとほぼ同じ。こちらは個別の表ではなくDB一括での移行
- Export(expコマンド)はDB内の表を一括でファイルに出力
- Import(impコマンド)は一括でDBに取り込む
- Oracle 10gR2以降はサポート対象外





# Export/Importコマンドでの移行

- 抽出:

```
> exp userID/password@$service FILE=exp_file.dmp LOG=exp_file.log
```

- 転送 : emp\_file.dmpを圧縮してSCP転送

- ロード:

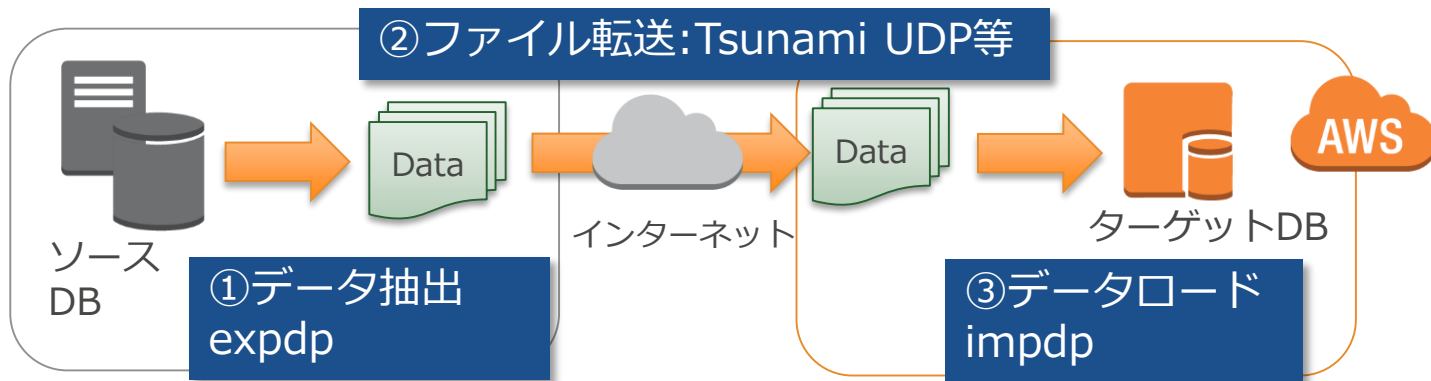
```
> imp userID/password@$service FROMUSER=cust_schema  
TOUSER=cust_schema FILE=exp_file.dmp LOG=imp_file.log
```

## 考慮点

- データが小さく、移行先がOracle on EC2の場合は、ファイル転送せずに直接ソースDB側からimpコマンドを実行することも可能。  
その場合、サーバ間の通信はSQL\*Netになる点に注意
- 移行先がRDSの場合にはOracle Clientを導入済のEC2が必要になる

# Data Pump

- 中規模以上のデータ移行にはData Pumpが第1候補
  - Oracle 10gR1以降で利用可能
  - ターゲットがRDSの場合、RDS側がOracle 11.2.0.3以降である必要がある
- ソースDBからexpdpでデータをファイルとして抽出し、impdpで投入。ワンステップのDB全体移行に向く
- 2ステップで移行する場合は、Data Pumpのexpdpに差分管理の仕組みがないので、前回の差分をQueryで検索できる必要がある



# EC2 on OracleへData Pumpで移行する①

- 抽出 : expdpコマンド
  - 圧縮=ON、暗号化を指定、パラレルで実行

```
expdp demoreinv/demo full=y  
  dumpfile=data_pump_exp:reinvexp1$U.dmp,  
    data_pump_exp:reinvexp2$U.dmp,  
    data_pump_exp:reinvexp2$U.dmp  
  filesize=20G parallel=8  
  logfile_data_pump_exp1:reinvexpdp.log compression=all  
  ENCRYPTION=all ENCRYPTION_PASSWORD=... job_name=reInvExp
```

- full=yでDB全体
- 出力先ディスクを3つ用意し、出力速度を向上させる
- 8パラレル実行
- 圧縮ON
- 暗号化ON

# EC2 on OracleへData Pumpで移行する②

## Tsunami UDPで通信速度の向上

ファイルサイズが大きくなると転送方法の検討が必要

- アップロードはパラレルで実行することで速度向上を検討
- 通信距離回線によっては、通常のTCPベースの通信(FTP等)は、巨大ファイルの転送において速度が出づらくなってくる。それを解消するため、高速ファイル転送に最適化されたプロトコルやツールが開発されている
- Tsunami UDPもその一つ (OSS)
  - <http://tsunami-udp.sourceforge.net/>

### Tsunami UDP Protocol

SOURCEFORGE.NET®

Tsunami UDP Protocol: A fast user-space file transfer protocol that uses TCP control and UDP data for transfer over very high speed long distance networks ( $\geq 1$  Gbps and even 10 GE), designed to provide more throughput than possible with TCP over the same networks. The project is based on original Indiana University 2002 Tsunami source code, but has been significantly improved and extended. As such, large portions of the program today are courtesy by [Aalto University Metsähovi Radio Observatory](#).

Includes FTP-like client and server command line applications for normal file transfers. It has additionally been extended for high rate real-time data streaming in eVLBI radio astronomy and geodesy (VSIB, PCEVN DAQ). Licensed under the original IU open source license.

# EC2 on OracleへData Pumpで移行する③

- コード : impdpコマンド
  - 暗号のパスワードを指定して、パラレルで実行

```
impdp demoreinv/demo full=y DIRECTORY=DPUMP_DIR  
dumpfile=reinvexp1%U.dmp,reinvexp2%U.dmp,reinvexp3%U.dmp  
parallel=8 logfile=Dpimp.og ENCRYPTION_PASSWORD=... job_name=DPImp
```

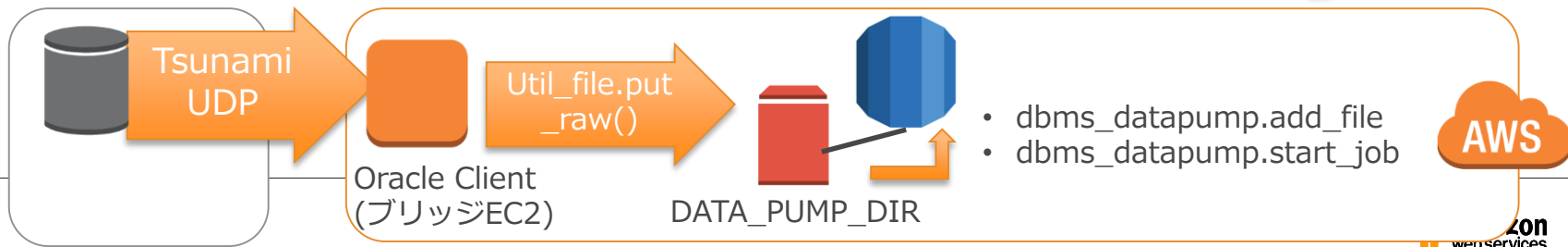
# RDSをターゲットにData Pumpで移行する

- ターゲットDBがRDSの場合は注意が必要
  - RDSにはSSHログインできないしSCPもできない。しかしData Pumpでロードするデータ（ファイル）は、Oracleと同じOS上に存在しないといけない
  - RDS Oracle 11.2.0.3 以降では「DATA\_PUMP\_DIR」からデータを読み込める。ただし結果として一時的にディスク領域はデータサイズの2倍必要になる
  - 移行作業中はMulti-AZは非推奨（巨大DBをRDSに移行する場合）

## 流れ

- AWS上に「ブリッジEC2」を立て、そこにOracle Clientを導入
- ソースDBでexportした\*.dmpファイルをブリッジEC2に転送
- ブリッジEC2からutl\_file.put\_raw()を使ってRDSの「DATA\_PUMP\_DIR」にファイルをコピー
- RDS Oracle側でdbms\_datapump.\*プロシージャで「DATA\_PUMP\_DIR」からデータを読み取る

※サンプルスクリプトがホワイトペーパーに掲載されています



# 外部表 (External Table)

- 基本的にはData Pumpと同じ
  - Data Pumpの方がより大きいサイズに対応できるため、推奨されるのは既存で外部表を使っているケースのみ
  - 移行先がRDSの場合、Data Pumpと同様に、データコピーをDATA\_PUMP\_DIRにする工夫が必要

## RDSへの移行の流れ

- 1) ソースDBで取得したデータファイルをRDSのDATA\_PUMP\_DIRにコピーする
- 2) DATA\_PUMP\_DIR上のファイルに対し、外部表を作成
- 3) 外部表をSELECTし、それを通常表にINSERTする

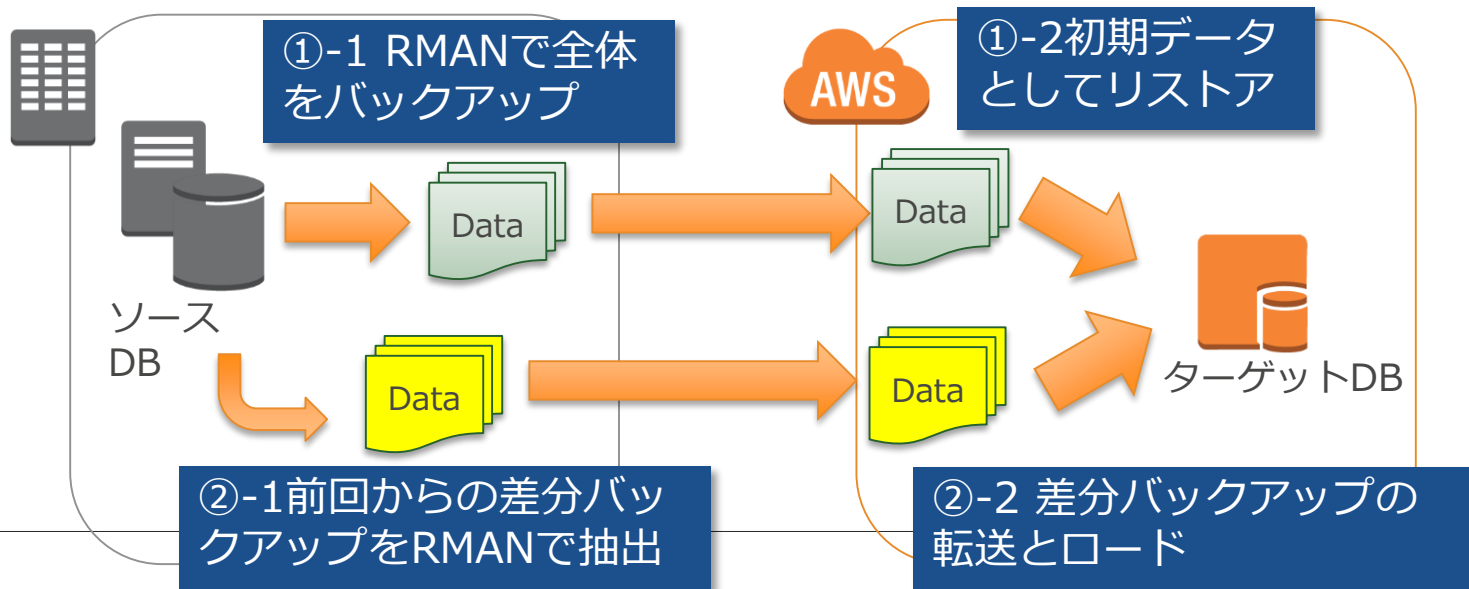
# RMAN

- RMANはOracleのバックアップ&リストア機能
  - RDSに対して、RMANでデータをリストアすることはできません
- RMANを選択するケース
  - 運用ですでにAmazon S3等のRMANバックアップイメージを保存している
  - 巨大DB（数TB以上）のケースではRMANのデータをAWS Import/Exportで取り込む方が速い
    - AWS Importは物理メディア経由でAWS S3/EBSにデータを転送する機能
    - AWS Import/Exportは現在東京リージョンでは使用できません
  - **2ステップ移行を計画しており、差分を適用したい場合**



# RMANを使った2ステップ移行

- RMANを利用すると、前回のバックアップからの「差分バックアップ」が利用できるため、2ステップ移行に適している



# 参考：RMANを使った移行

- フルバックアップの取得例

```
$ rman target=/  
RMAN>
```

```
CONFIGURE CONTROLFILE AUTO BACKUP ON;  
BACKUP DATABASE PLUS ARCHIVELOG
```

- ターゲットDBの作成例

- ターゲットのインスタンスはNOMOUNTモードで起動
- RMANでDUPLICATEを指定して、新ターゲットDBを作成する

```
$ rman AUXILIARY /  
RMAN>
```

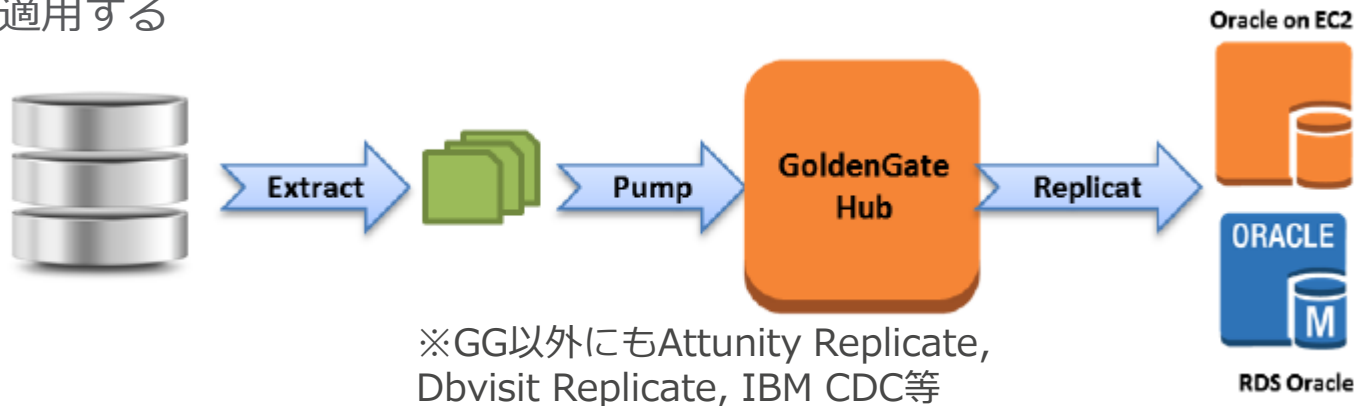
```
DUPLICATE TARGET DATABASE TO DBONEC2  
SPFILE  
NOFILENAMECHECK;
```

# GoldenGateによるゼロ・ダウンタイム移行

ソースDBの変更をキャプチャし、**ターゲットDBに継続的に反映しつづける**仕組み  
ReplicatプロセスはEC2サーバ（Golden Gate Hub）を立ててそこで動かすことを推奨

## 処理の流れ

- GG (GoldenGate)のExtractプロセスが初期データを抽出
- GGのPumpプロセスがHubとなるGGサーバにファイルを転送
- GGのReplicatプロセスが、ターゲットDBに初期データを投入
- その後は、ソースDBに更新が入るたびにそれらキャプチャして、ターゲットDBに継続的に適用する



# アジェンダ

## Agenda

- AWSの基本
- 移行の背景
- データベース on EC2かRDSかの選択
- 移行のための手法
- まとめ

# まとめ①

- AWSへRDBを移行する目的
  - 運用管理を楽に
  - 柔軟で強固なインフラへの移行
- RDSか？ RDB on EC2か
  - 基本はRDS。要件にフィットしない場合にRDB on EC2を検討
- 移行時の考慮点
  - DBサイズ
  - 移行に掛けられる時間（ダウンタイム）
  - ネットワーク速度
- 手法
  - ワンステップ移行、2ステップ移行、ゼロ・ダウンタイム移行

# まとめ②移行の手法（再掲）

| DB移行手法                          | 移行サイズ（目安） | 対象移行先                          | 推奨サイズ・状況  |
|---------------------------------|-----------|--------------------------------|---|
| SQL Developerの<br>Database Copy | ~200MB    | Oracle on EC2<br>もしくはRDS       | 小規模DBの移行<br>テスト・評価環境の構築                         |
| マテリアライズド・<br>ビュー+DB LINK        | ~500MB    | Oracle on EC2<br>もしくはRDS       | 小規模かつ、特定の表のみ移行                                  |
| SQL*Loader ★                    | ~10GB     | Oracle on EC2<br>もしくはRDS       | 小~中規模。いくつかの表のみ移行                                |
| Export/Import<br>※非推奨           | ~10GB     | Oracle on EC2<br>もしくはRDS       | 小~中規模。DB全体を移行<br>10gR2以降はサポート外                  |
| Data Pump<br>(データ・ポンプ) ★        | ~5TB      | Oracle on EC2<br>もしくはRDS       | 中~大規模（10GB~ 5 TB）のDB移行                          |
| 外部表                             | ~1TB      | Oracle on EC2<br>もしくはRDS       | 外部表をすでに使っている環境のみ<br>推奨                          |
| RMAN ★                          | 全てのサイズに対応 | Oracle on EC2のみ<br>(RDSでは使えない) | RMANバックアップがすでに<br>AWS(S3)上にある。もしくは5TB<br>以上のサイズ |
| Golden Gate ★                   | 全てのサイズに対応 | Oracle on EC2<br>もしくはRDS       | ゼロ・ダウンタイムの移行が必要な<br>場合                          |

# Q&A



# 参考文献・リンク

- Oracle RDSにおけるデータ移行（マニュアル）
  - [http://docs.aws.amazon.com/ja\\_jp/AmazonRDS/latest/UserGuide/Oracle.Procedural.Importing.html](http://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/UserGuide/Oracle.Procedural.Importing.html)
- Strategies for Migrating Oracle Database to AWS
  - AWSのホワイトペーパー(PDF)。具体的な作業内容が記載されています
  - <http://d0.awsstatic.com/whitepapers/strategies-for-migrating-oracle-database-to-aws.pdf>



# AWSトレーニングでは様々な学習方法をご提供しています

## メリット

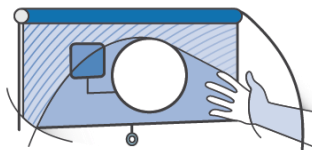
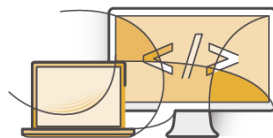
- AWS について**実習**や**実践練習**を通じて学習できる
- **AWS を熟知したエキスパート**から直接 AWS の機能について学び、疑問の答えを得られる
- **自信をもって** IT ソリューションに関する決定を下せるようになる

## 提供方法



e ラーニングや動画

セルフペースラボ



クラスルーム  
トレーニング

詳しくは、<http://aws.amazon.com/training> をご覧ください

# 公式Twitter/Facebook AWSの最新情報をお届けします



@awscloud\_jp



検索



もしくは  
<http://on.fb.me/1vR8yWm>

最新技術情報、イベント情報、お役立ち情報、お得なキャンペーン情報などを  
日々更新しています！

# AWS初心者向けWebinar



- AWSをこれからご使用になる技術者向け、ソリューションカットのセミナー
  - <http://aws.amazon.com/jp/about-aws/events/>

## AWS Black Belt Tech Webinar 2015

AWSのサービスをディープにご紹介

- <http://aws.amazon.com/jp/about-aws/events/>

# AWSの導入、お問い合わせのご相談

- AWSクラウド導入に関するご質問、お見積り、資料請求をご希望のお客様は、以下のリンクよりお気軽にご相談ください

<https://aws.amazon.com/jp/contact-us/aws-sales/>

お問い合わせ

日本担当チームへのお問い合わせ >

関連リンク

フォーラム

## 日本担当チームへのお問い合わせ

AWS クラウド導入に関するご質問、お見積り、資料請求をご希望のお客様は、以下のフォームよりお気軽にご相談ください。平日営業時間内に日本オフィス担当者よりご連絡させていただきます。

※ご請求金額またはアカウントに関する質問は[こちらからお問い合わせください](#)。  
※Amazon.com または Kindle のサポートに問い合わせは[こちらからお問い合わせください](#)。

アスタリスク（\*）は必須情報となります。

姓\*

名\*

ご参加ありがとうございました