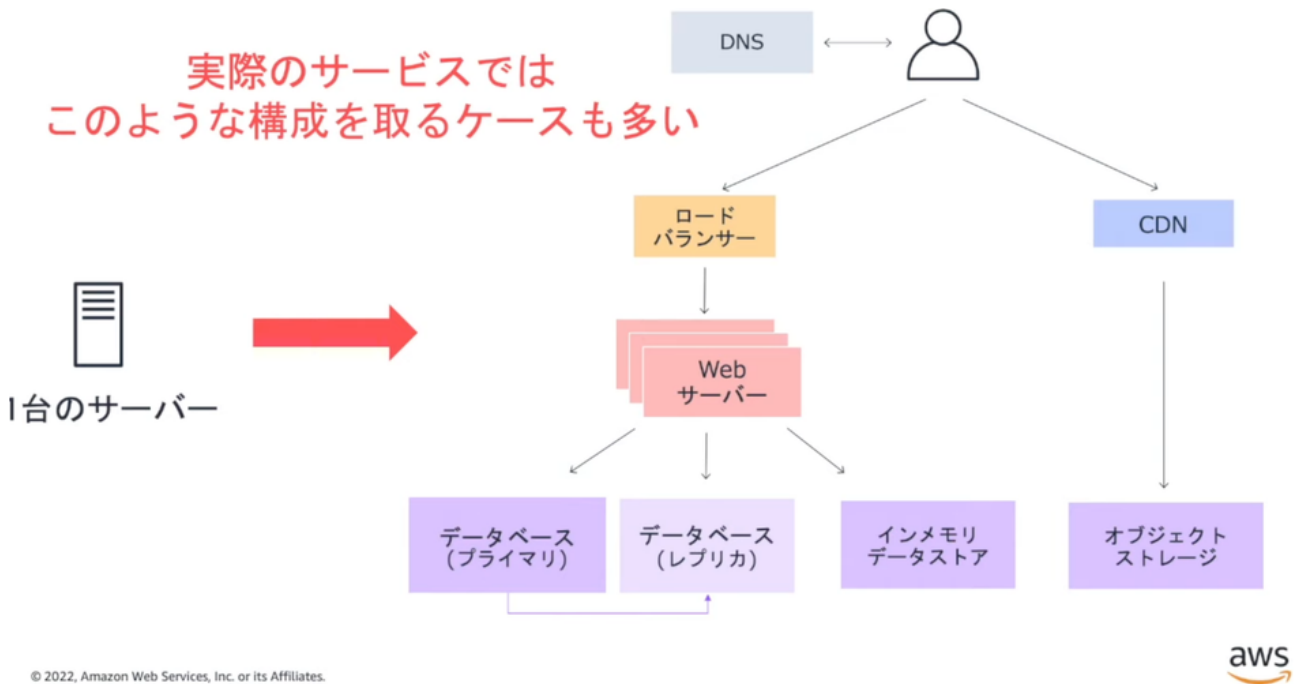


AWS Jump Start事前学習

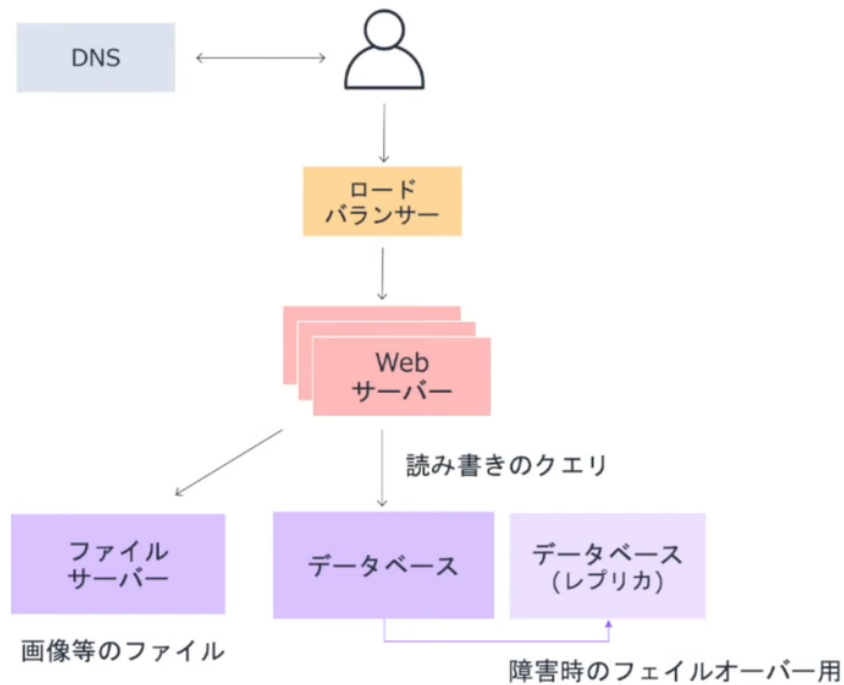
事前学習 ① はじめてのアーキテクティング



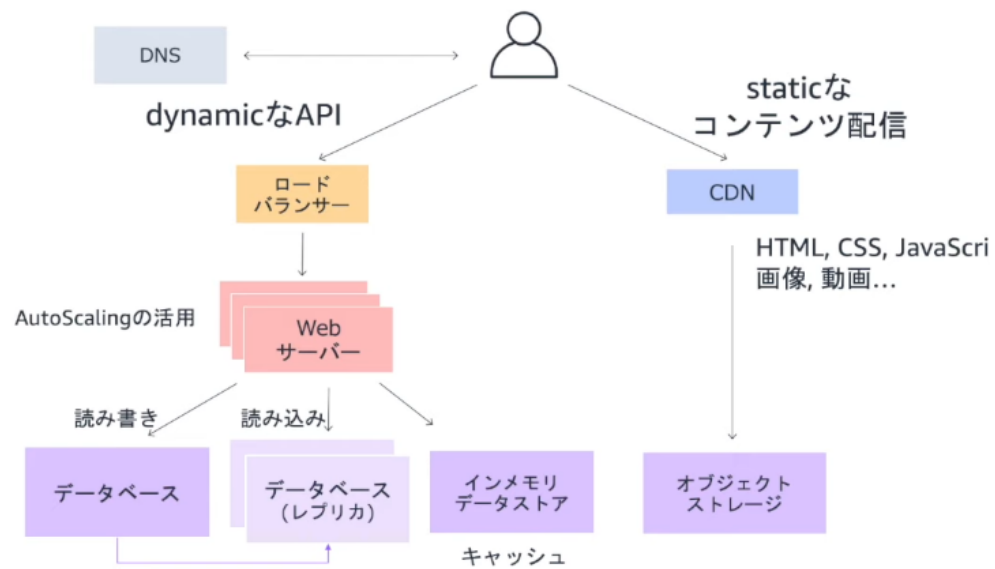
- アーキテクチャ設計
 - 非機能要件、制約要件も踏まえて、システム全体とインフラの設計を行うこと
- 設計で意識すること
 - 信頼性(アクセスできない時間がある、復旧が遅いなど)
 - どのように対策できるか
 - ロードバランサで分散させる。ロードバランサはヘルスチェックをしているので、生きているサーバにしかリクエストしない。ロードバランサ時代はAWSは冗長構成がとられる
 - データセンター分散
 - DBのレプリケーションする。フェイルオーバー
 - バックアップ戦略 AWS Backup
 - スケーラビリティ、パフォーマンス(遅くなってしまう)
 - スケールアウト、スケールアップ
 - DBではリードレプリカを作って、読み込みは素早くやる
 - インメモリデータストア
 - よくある製品Redis、Memcached。失われてもいい情報
 - staticなコンテンツは、オブジェクトストレージと、手前にCDNをかませて配信
 - 素早い運用(ログの確認が大変、レイテンシの確認がしにくい)
 - どのくらいSLAを達成しているか、アラートを上げる仕組み
 - CIでビルド、テスト。CDでデプロイ自動化
 - Infra as code
 - コスト最適化(コスト削減を要求されたが、何したらいいのか分からない)
 - AutoScaling

- セキュリティ
- たたき台となる構成

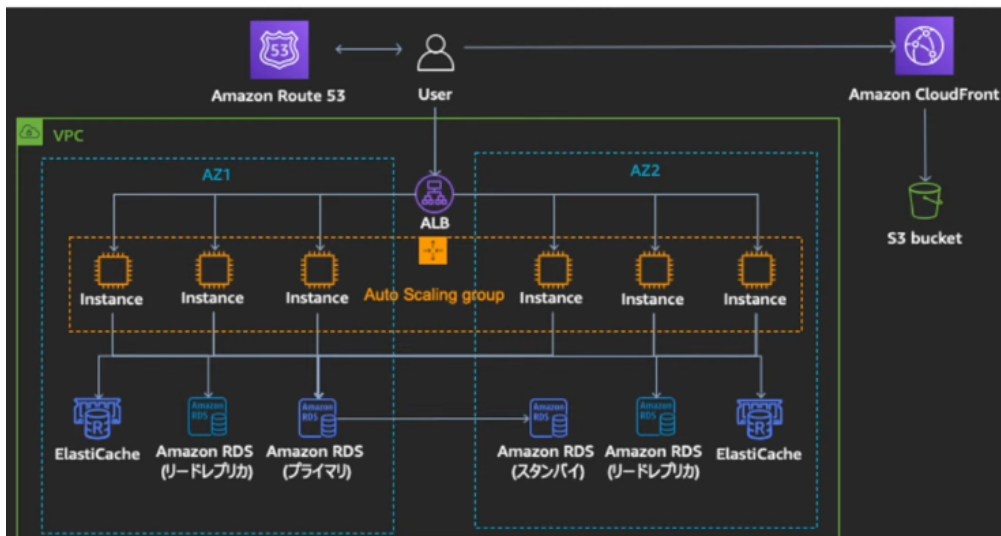
高い信頼性が求められる際の構成例



高い信頼性やスケーラビリティが求められる際の構成例



◦ AWSでの一例



アーキテクチャ設計において大切なこと

- すべての要件、制約を満たすのが難しい場合は、**ビジネスを成功させる**上で優先させるべき点に立ち返り、トレードオフの中で判断する
- 要件を満たす中でできるだけ**シンプル**な構成を検討する

リージョン内の構成

- 複数のDCから構成されるAZ
- リージョン内に地理的に離れた複数のAZを用意



AWSサービスの範囲

- グローバルサービス
 - IAM, CloudFront, Route 53
- リージョンサービス
 - Amazon VPC, Amazon DynamoDB, AWS Lambda
- AZサービス
 - EC2, Amazon RDS

- グローバル、リージョンはAWS側がAZ考慮してくれている。AZサービスは自前で設定する必要あり

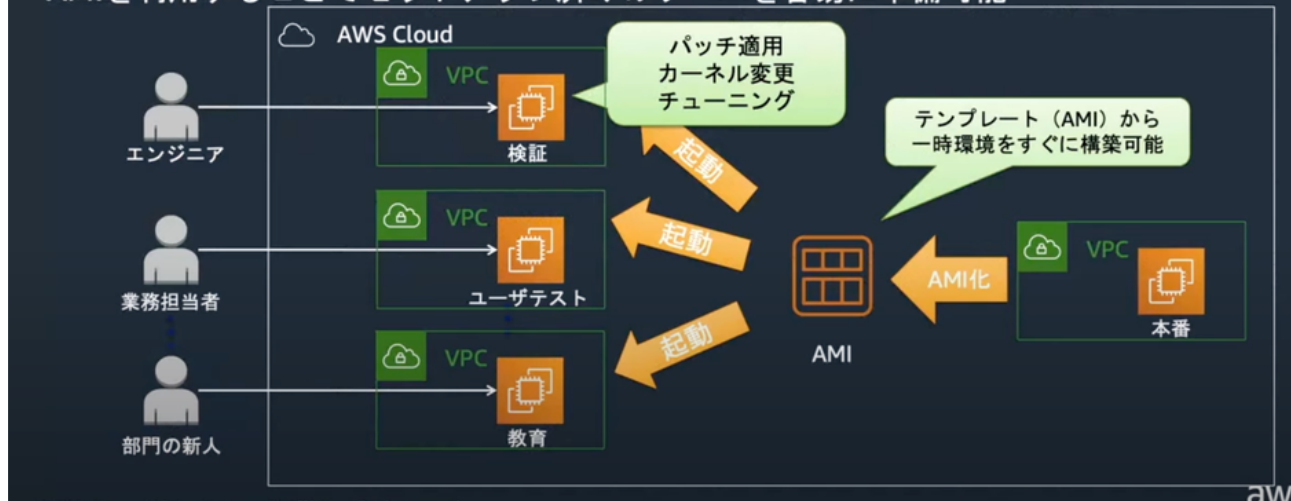
コンピューティングサービス

- EC2 インスタンスの種類によって割引可能。予約しておいたり、余ったものを使ったり
 - EC2の機能①AMI amazon machine image(むずかしい)
 - ソースコードまでは入ってない。amazon-linuxなどからひな形を選ぶ感じ
 - IAMとまちがわない！



AMI (Amazon Machine Image)

EC2インスタンスを起動する際に必要な情報が格納されたマシンイメージ
EBSのスナップショットから独自のカスタマイズAMIを保存可能
AMIを利用することでセットアップ済みのサーバを容易に準備可能



- EC2の機能②セキュリティグループ
 - EC2だけでなく他のサービスにも設定できる

Security Groupとは

インスタンスへのトラフィックを制限する**ファイアウォール**機能

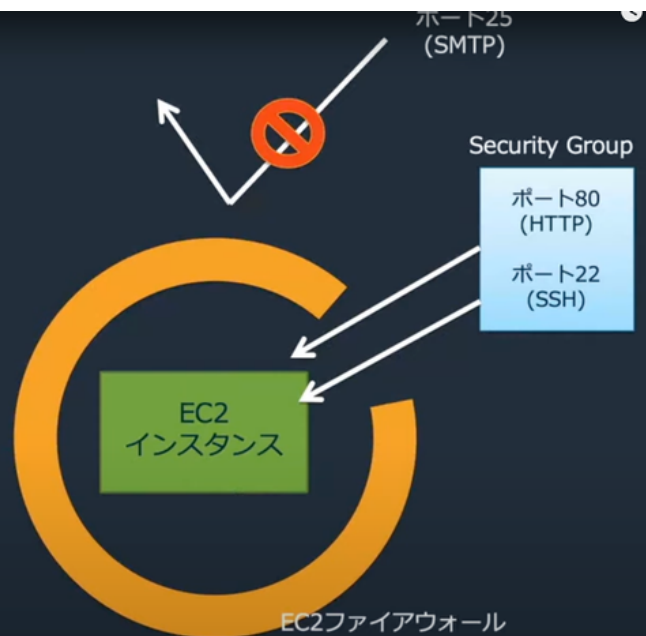
デフォルトでは全トラフィックが閉じた状態

必要な受信アクセスに対してアクセスルールを定義する

- プロトコル (TCP/UDP)
- 宛先ポート
- アクセス元IP / Security Group

ルールをひとまとめにした

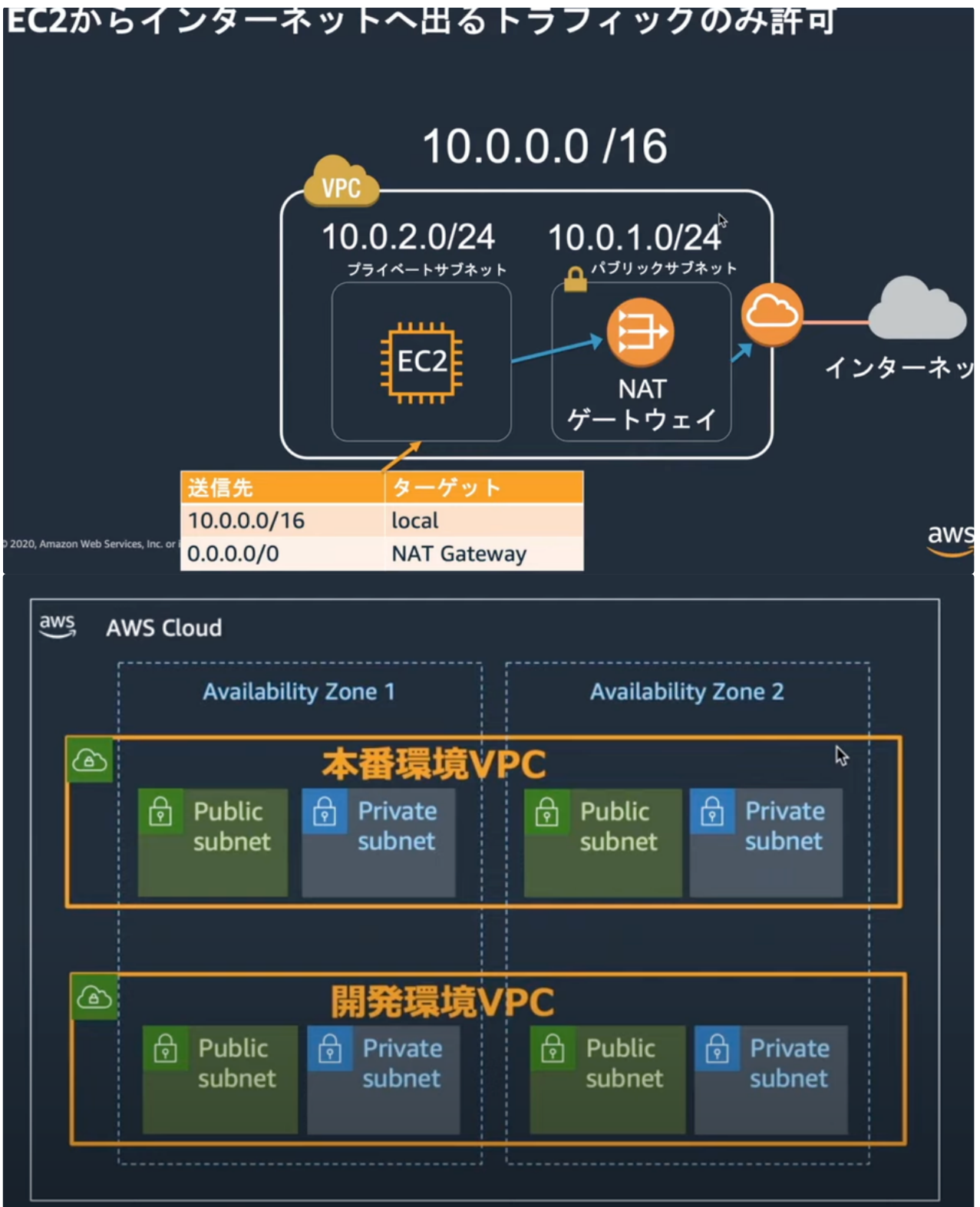
テンプレートを Security Group と呼ぶ



- サーバレス
 - Lambda
 - コードのみを上げればAPIが作れる。常時起動しているわけではないので安い
- オートスケール
 - クラウドウォッチがセット
 - 仮想サーバーのログは見れない
 - OSのなかのメモリは見れない

ネットワークサービス [🔗](#)

- VPC プライベートネットワーク空間
 - むずかしい



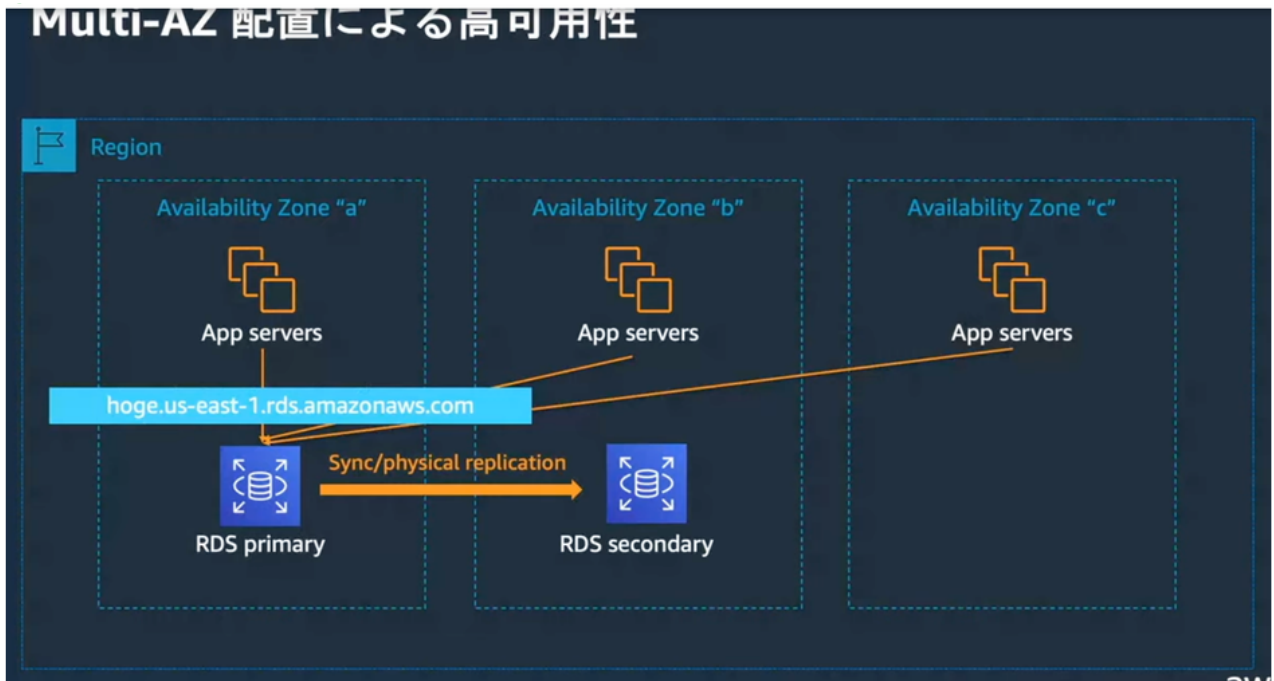
- AZごとにサブネットおく
- Cloud Frond
 - 地理的に近いところから

ストレージ [🔗](#)

- S3
 - イン無料、アウト有料、おいている間有料
 - S3 intelligent tieringで、自動的にアーカイブに移動されたりしてコストが良い
- EBS
 - EC2にくっつける容量
 - 1つのEC2のみ
- EFS
 - 複数のEC2で共有できる
- snowball family
 - 物理的にストレージを移行してくれる

DB [🔗](#)

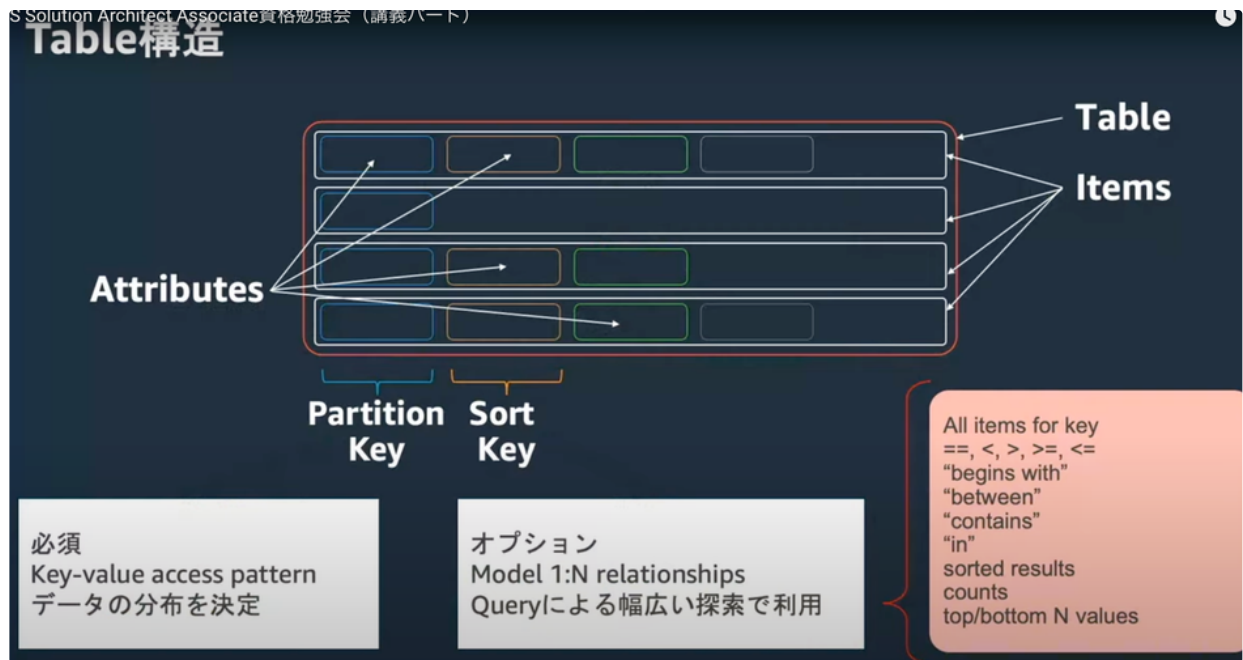
- RDS
 - 複製とリードレプリカは違う
 - リードレプリカもprimaryに昇格できる



- aが死んでも、bでエンドポイントは同じものでアクセスできる

- Aurora
 - リードレプリカ大量に作れる
 - バックアップがS3に溜まっている。クラスタボリュームという
- Dynamo
 - writeの負荷分散が得意。amazonや金融
 - No SQL
 - RDSとは構造が違う

Table構造



- selectのときは、ソートキーを定義しておく

セキュリティ

- IAM
 - AWSアカウントに割り当てる(IAMユーザー、IAMグループ)
 - 誰が何をできる、できない
 - コンソールログイン
 - APIたたく
 - EC2やLambdaに割り当てる(IAMロール)
 - AWSサービス同士の連携につかう