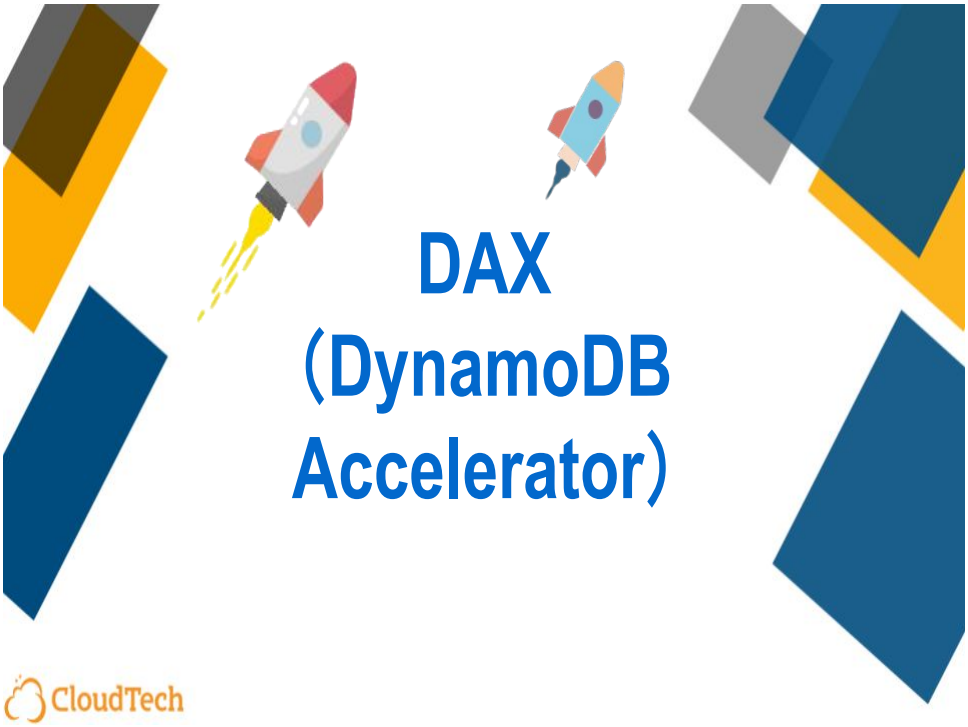
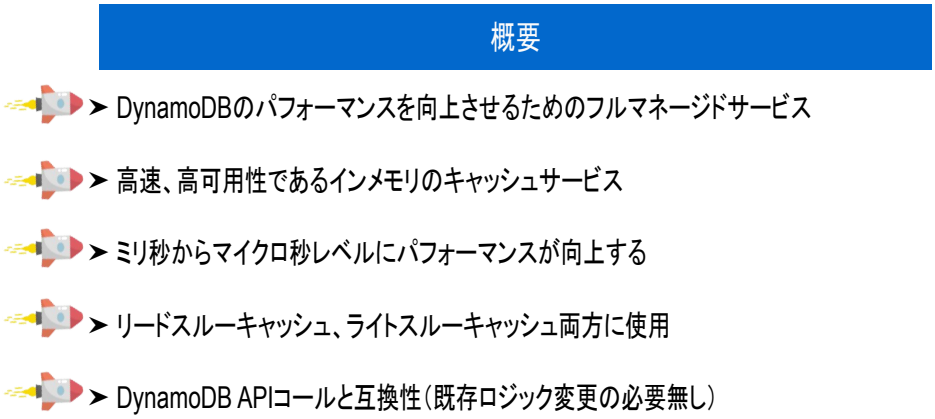


DAX (DynamoDB Accelerator)

CloudTech

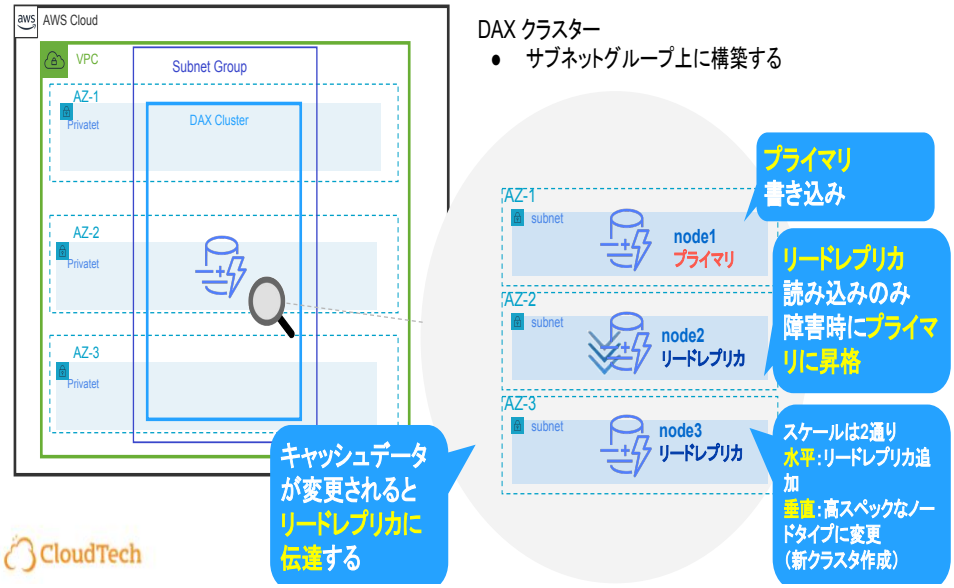
DAX（Amazon DynamoDB Accelerator 概要）



- DAXの概要ですが、
  - これはDynamoDBのパフォーマンスを向上させるためのフルマネージドサービスです。
  - DynamoDB用の、インメモリのキャッシュサービスで、高速であり、高可用性という特徴があります。
    - RDSのように「サブネットグループ」を作成し、複数のAZに所属するような配置ができます。後で見ていきましょう。
  - ミリ秒からマイクロ秒へとパフォーマンスを向上させることができます。最大 10倍の性能改善を実現し、1秒間に数百万のリクエストを処理する場合でも有用です。
    - オークション、ゲーム、特別セールやプロモーションなど、読み込みが多く、バースト性の高いワークロードに最適です。
  - リードスルーキャッシュ、ライトスルーキャッシュ両方に使用できます。

        - つまり、読み込みも書き込みも、両方のパフォーマンス向上が見込めるということです。
    - DAXは既存のDynamoDB APIコールと互換性があるため、アプリケーションロジックを変更する必要はありません。
        - 例えば、既にEC2とDynamoDBで稼働しているのであれば、DAXを新規追加しても、アプリのロジック自体への変更は不要ということです。

DAXクラスターの構成要素



- DAXの構成要素を見ていきましょう
- AWSクラウドのVPC上に
    - DAXクラスターというものを構築して使用します。
        - DAXクラスターは、1つ以上のサブネットを束ねたサブネットグループを指定して構築する必要があります。

- DAXクラスターの中身を少し細かく見てみましょう。
- DAXクラスターの実態は、ユーザー側で管理不要な EC2 で構成されています。
    - このEC2のことをノードと呼び、ノードをひとまとめにして管理しやすくしたものがクラスターです。
    - 構築時、DAXのノードタイプを指定します。これは（t3.small、t3.midiumなど）EC2のインスタンスタイプと似ているスペックのタイプですね。クラスター内の各ノードは同じ必要がある。
        - クラスター内のノードの 1 つがプライマリノード

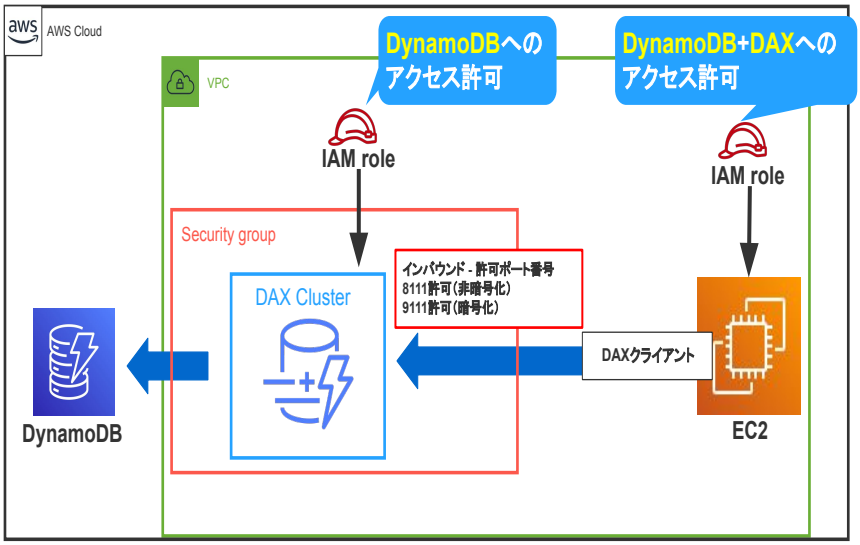
        - として指定されます。
        - もちろん、これの1台だけでも機能しますが、他ノードを追加した場合は全てリードレプリカとして機能することとなります。
            - プライマリノードはDynamoDB への書き込みオペレーションを処理します。
                - プライマリノードにキャッシュされたデータが変更されると、DAX はその変更をすべてのリードレプリカノードに伝達します。
            - リードレプリカは DynamoDB には書き込みまず、読み取りのみです。
            - （ここらへん、RDSのリードレプリカと同様です。）
                - リードレプリカの目的としては、高可用性があります。
                    - プライマリノード障害時に、DAX は自動的にリードレプリカにフェイルオーバーし、新しいプライマリとして起動しますので、
                        - リードレプリカを異なるアベイラビリティーゾーンにデプロイすると、AZ障害に対処できます。

    - DAX クラスターのスケールについては 2通りの方法がある
        - 水平でスケールする
            - 読み込みをスケーリングできます。DAX クラスター内のすべてのノードに負荷を

        - 均等に分散します。
    - ノードを垂直スケール
        - (新しいノードタイプで新しいクラスターを作成する必要があります。)

https://docs.aws.amazon.com/ja_jp/amazondynamodb/latest/developerguide/DAX.concepts.cluster.html

DAXの動き



CloudTech

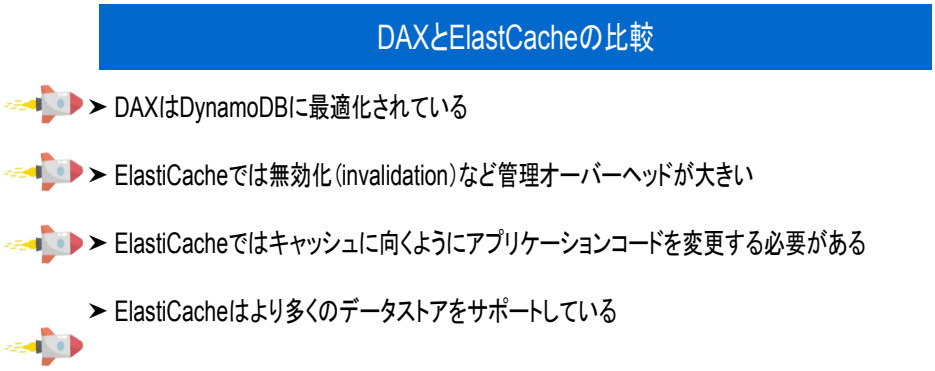
- では、全体的な構成図でDAXを理解していきます。
- まず、DynamoDBがVPCの外側とAWSの間の領域にあるとします。
    - EC2を構築し、内部のアプリケーションがDynamoDBへ通信しているとします。
    - DAXはVPCの環境の中で、両者の間に入って、処理を高速化します。
        - EC2には、DAXのクライアンドソフトウェアをインストールしておきます。
        - EC2は最初に、DAX クラスターのエンドポイントを指定することで DAX にアクセスでき、
            - DAXにキャッシュがないかを問い合わせ、キャッシュがあればそれを返しますし、もしデータがなければ、DynamoDBに取得しにいきます。

- DAX にサービスロールの設定が必要です。DAXが

- DynamoDB テーブルへアクセスすることを許可します。
    - サービスロール、これは AWS同士のリソースのアクセス許可を指定するのに必要な IAMの設定でしたね。
- 同様に、EC2にもサービスロールの設定が必要です。EC2がDynamoDB, もしくはDAXへアクセスすることを許可します。
    - ここらへんは少し細かく定義できるので、参考ドキュメントを参照ください。
- DAXに設定するセキュリティグループについてですが、暗号化されていないクラスターには TCP ポート 8111、暗号化されたクラスターには 9111 のインバウンドトラフィックを許可する必要があります。
    - なお、EC2側のセキュリティグループは要件によって異なってきますので、詳細は割愛します。

https://docs.aws.amazon.com/ja_jp/amazondynamodb/latest/developerguide/DAX.access-control.html

DAXとElastCacheの比較



CloudTech

DAXとElastiCacheの比較

- DAXは、DynamoDBに最適化されています。
  - DynamoDBを使った設計で、よりパフォーマンスを高めたい時はDAXが第一選択肢となるでしょう。
  - 特にアプリケーションのコードが複雑な場合など、手早く威力を発揮してくれます。

- ElastiCacheでは、無効化など管理のオーバーヘッドが大きくなります。
  - 他、キャッシュ戦略など、管理は ElastiCacheの方が色々と考慮しなければならない点は多いと言えるでしょう。

- ElastiCacheでは、キャッシュに向くように、アプリケーションコードを変更する必要があります。
  - 先ほどの件と多少重複しますが、アプリ側の後続作業も多いということですね。

- ElastiCacheはより多くのデータストアをサポートしています。
    - DAXはDynamoDB専用でしたが、ElastiCacheはRDSでサポートされているDBはもちろんサポートされているので、柔軟な対応ができます。