AWS 資料
プライベートネットワーク設計

改訂履歴

年月	版	作成者	詳細
2021年11月29日	1.0	青山	新規作成

内容

1. VPC の作成

1.1. 操作方法

① AWS マネジメントコンソールより VPC を選択する

② VPC の作成をクリックする

③ 下記項目を入力または選択し、VPC を作成する

名前タグ	任意(例：test-vpc1)
Ipv4 CIDR ブロック	10.0.0.0/16
Ipv6 CIDR ブロック	Ipv6 CIDR ブロックなし(デフォルト)
テナンシー	デフォルト

1.2. VPCとは

VPC の主な概念は次のとおりです。

- Virtual Private Cloud (VPC) — AWS アカウント専用の仮想ネットワーク。
- サブネット — VPC の IP アドレスの範囲。
- ルートテーブル — ネットワークトラフィックの経路を判断する際に使用される、ルートと呼ばれる一連のルール。
- インターネットゲートウェイ — VPC 内のリソースとインターネット間の通信を可能にするために VPC にアタッチするゲートウェイ。
- VPC エンドポイント - PrivateLink を使用してサポートされている AWS サービスや VPC エンドポイントサービスに VPC をプライベートに接続できます。インターネットゲートウェイ、NAT デバイス、VPN 接続、および AWS Direct Connect 接続は必要ありません。VPC のインスタンスは、サービスのリソースと通信するためにパブリック IP アドレスを必要としません。
- CIDR ブロック —クラスレスドメイン間ルーティング。インターネットプロトコルアドレスの割り当てとルート集計方法。

1.3. 使用するCIDRブロックを決定する

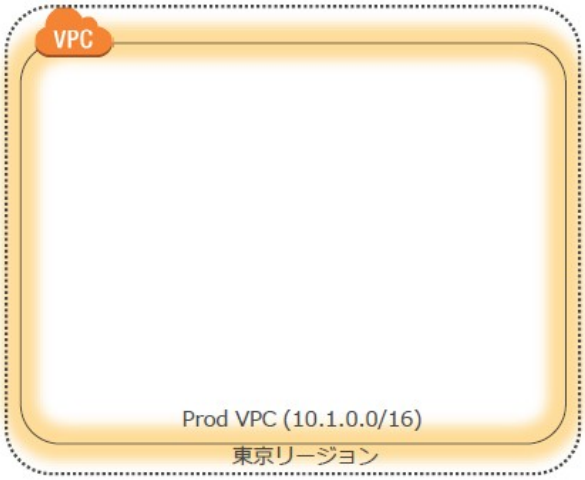
- 大きさは/28 から/16
- レンジは RFC1918 を推奨

1.4. 作成後は変更不可のため大きめに

- /16 が推奨

1.5. オンプレミスや他 VPC のレンジと重複させない

- 相互接続する可能性を見越して



2. サブネットの作成

2.1. 操作方法

① AWS マネジメントコンソールより VPC を選択する

② サブネットを選択してサブネットの作成をクリックする

③ 下記項目を入力または選択し、サブネット 1 を作成する

VPCID	作成した VPC を選択する(例：test-vpc1)
関連付けられた CIDER	表示される

サブネット名	任意(例：test-subnet-1)
アベイラビリティーゾーン	アジアパシフィック (東京) / ap-northeast-1a
IPv4 CIDR ブロック	10.0.11.0/24

④ 新しいサブネット追加をクリックし、下記項目を入力または選択し、サブネット 2 を作成する

VPCID	作成した VPC を選択する(例：test-vpc1)
関連付けられた CIDER	表示される

サブネット名	任意(例：test-subnet-2)
アベイラビリティーゾーン	アジアパシフィック (東京) / ap-northeast-1a
IPv4 CIDR ブロック	10.0.21.0/24

今回は冗長化やロードバランサは考慮しておりません

サブネット 1 は Web サーバ用、サブネット 2 は DB サーバ用に利用する予定です

2.2. VPC の CIDR ブロックの範囲から IP アドレスレンジを切り出す

- 必要な IP アドレス数を見積もる
- /24 が標準的

2.3. サブネット分割はルーティングポリシーに応じて行う

- インターネットアクセスの有無
- 拠点アクセスの有無など

2.4. サブネットは AZ の中に作成される

- 高可用性のために 2 つ以上の AZ の使用を推奨

2.5. サブネットのサイズの検討

サブネットマスク	/16 の VPC 内に 作成可能なサブネット数	サブネットあたりの IP アドレス総数 2^(32-mask) -2	ホストに割り当て可能な IP アドレス数 総数- 3
/18	4	16382	16379
/20	16	4094	4091
/22	64	1022	1019
/24	**256**	**254**	**251**
/26	1024	62	59
/28	16384	14	11

サブネットに割り当てられた IP アドレスのうち下記は割り当て不可：：：総数- 3

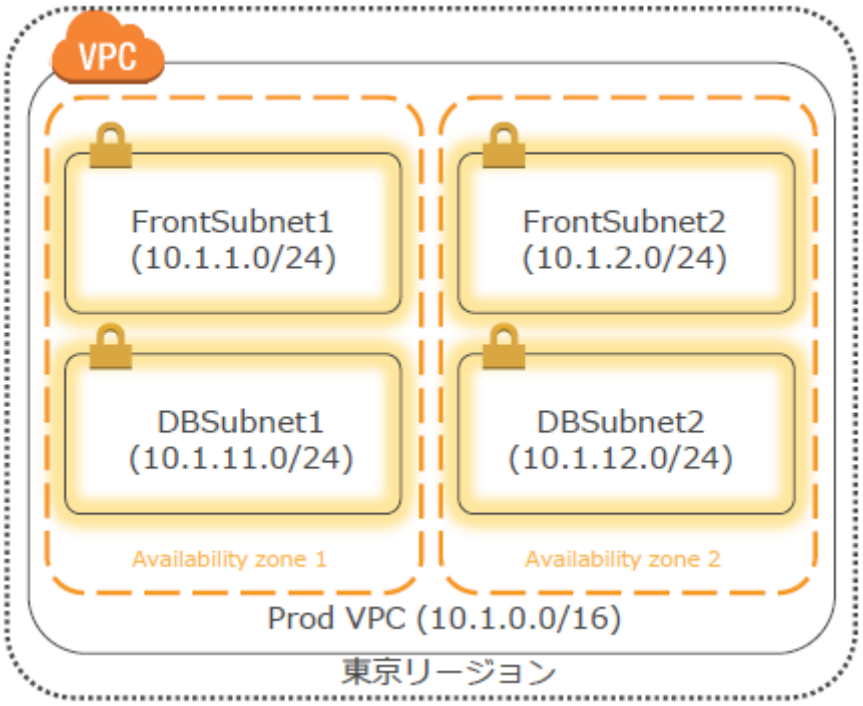
- .1：VPC ルータ（VPC 内のインスタンスにルーティング機能を提供）
- .2：Amazon DNS サーバーのため予約
- .3：将来用途のための予約

仮想プライベートクラウド (VPC) は、AWS アカウント専用の仮想ネットワークです。VPC は、AWS クラウドの他の仮想ネットワークから論理的に切り離されています。VPC 内には、Amazon EC2 インスタンスなどの AWS リソースを起動できます。VPC の IP アドレス範囲を指定して、サブネットを追加し、セキュリティグループを関連付けて、ルートテーブルを設定できます。

サブネットは、VPC の IP アドレスの範囲です。AWS リソースは、指定したサブネット内に起動できます。インターネットに接続する必要があるリソースにはパブリックサブネットを、インターネットに接続しないリソースにはプライベートサブネットを使用してください。

各サブネットの AWS リソースを保護するには、セキュリティグループやネットワークアクセスコントロールリスト (ACL) など、複数のセキュリティレイヤーを使用できます。

オプションで、VPC に IPv6 CIDR ブロックを関連付け、VPC のインスタンスに IPv6 アドレスを割り当てることができます。



2.6. 補足説明：リージョンとアベイラビリティーゾーン

2.6.1. リージョン

AWS にはリージョンという概念が存在します。これは、データセンターが集積されている世界中の物理的ロケーションのことです。また、論理的データセンターの各グループは、アベイラビリティーゾーンと呼ばれます。各 AWS リージョンは、1 つの地理的エリアにある、複数の、それぞれが隔離され物理的にも分離された AZ によって構成されています。

各 AZ には個別の電力源、冷却システム、そして物理的セキュリティが備わっており、これらは冗長的でレイテンシーが非常に低いネットワークを介し接続されています。高度な可用性の実現にフォーカスしている AWS のお客様は、複数の AZ で実行するようにアプリケーションの設計をすることで、より強力な障害耐性を実現できます。AWS のインフラストラクチャにおけるリージョンは、セキュリティ、コンプライアンス、データ保護からの要求を最も高いレベルで満たします。

AWS では、他のクラウドプロバイダーより広範囲にグローバル展開しています。このグローバル展開を支え、世界中のお客様に確実なサービスを提供するため、AWS では新たなリージョンを迅速に開設していきます。AWS では、北米、南米、欧州、中国、アジアパシフィック、南アフリカ、中東などのリージョンを含む、複数の地理的なリージョンを整備しています。

2.6.2. アベイラビリティーゾーン

アベイラビリティゾーン (AZ) とは、1 つの AWS リージョン内でそれぞれ切り離され、冗長的な電力源、ネットワーク、そして接続機能を備えている 1 つ以上のデータセンターのことです。AZ によって、単一のデータセンターでは実現できない高い可用性、耐障害性、および拡張性を備えた本番用のアプリケーションとデータベースの運用が実現されています。AWS リージョン内のすべての AZ は、AZ 間に高スループットかつ低レイテンシーのネットワーキングを提供する、完全に冗長性を持つ専用メトロファイバー上に構築された、高帯域幅、低レイテンシーのネットワーキングで相互接続されています。AZ 間のすべてのトラフィックは暗号化されます。AZ 間の同期レプリケーションを実行するのに十分なネットワークパフォーマンスを備えています。AZ により、高可用性実現を目的にしたアプリケーションの分割が簡単になります。アプリケーションが AZ 間で分割されている場合、企業は停電、落雷、竜巻、地震などの問題からより安全に隔離され保護されます。各 AZ はそれぞれ他の AZ から物理的に意味のある距離、つまり数キロメートル離れていますが、互いにすべて 100 km (60 マイル) 以内に配置されています。

2.6.3. AWS Local Zones

AWS Local Zones では、コンピューティング、ストレージ、データベース、およびその他の選択された AWS のサービスを、エンドユーザーから近い場所に配置します。AWS Local Zones を使用すると、メディア & エンターテインメントのコンテンツ制作、リアルタイムゲーミング、貯水池のシミュレーション、電子自動設計、そして機械学習など、エンドユーザーに対するレイテンシーが 10 ミリ秒未満であることが要求される高性能なアプリケーションを簡単に実行できます。

各 AWS Local Zone でのロケーションは AWS リージョンを拡張したものであり、Amazon Elastic Compute Cloud、Amazon Virtual Private Cloud、Amazon Elastic Block Store、Amazon File Storage および Amazon Elastic Load Balancing などの AWS のサービスを使用して、地理的にエンドユーザーと近い場所で、レイテンシーの影響を受けやすいアプリケーションを実行できます。AWS Local Zones では、ローカルと AWS リージョンでそれぞれ実行中のワークロード間で高帯域幅かつ安全な接続が利用できます。同じ API とツールセットを介してすべてのリージョン内サービスにシームレスに接続します。

3. VPC コンポーネントの配置とルーティング設定

3.1. 操作方法

インターネットゲートウェイは VPC 内部からインターネット接続するためのコンポーネントであり、VPC にアタッチすることで可能となります。

① AWS マネジメントコンソールより VPC を選択する

② 左ペインよりインターネットゲートウェイ を選択し、インターネットゲートウェイ の作成をクリックする

③ 下記項目を入力または選択し、インターネットゲートウェイ を作成する

名前タグ	任意(例：test-internetgateway)

作成してすぐは、どこにもアタッチされていないので、

作成したインターネットゲートウェイを選択し、

「アクション」→「VPC にアタッチ」を選択し

インターネットゲートウェイの作成より作成した VPC にアタッチする

3.1.1. ルートテーブルの編集

ローカルの通信は VPC 以外はインターネットゲートウェイにサブネットのルートテーブルを編集する。

① 作成したサブネットをクリックし、下部にあるルートテーブルを表示させ、ID を右クリックして新しいタブで開く

② 画面下部のルートを選択しルートの編集をクリック

サブネット 1

送信先	ターゲット
10.0.0.0/16	local
0.0.0.0/0	作成した igw を選択する

※0.0.0.0/0 についてこの IP アドレスは全てと言う意味があり、「デフォルトルート」と呼ばれる。流れとしては、デフォルトでは、ターゲットがインターネットゲートウェイになるが、VPC 内部の IP アドレスのみターゲットがローカルになる。

サブネット 2

送信先	ターゲット
10.0.0.0/16	local

※サブネットは VPC 内部のみを対象としています。今回は NAT 接続は省略しています

DB をネット経由でセキュリティパッチなどを適用する場合、NAT 経由でインタネットの接続します

3.2. VPC コンポーネントを配置する

- インターネットに疎通が必要な場合は IGW、社内に接続が必要な場合は VGW など

3.3. サブネット毎のルートテーブルを編集する

- デフォルトで VPC 内宛ての経路は作成済み
- IGW などに向けた経路を作成
- プライベートサブネットとパブリックサブネットの大別

3.4. VPC 単位で配置するコンポーネント(最低限必要なもの)

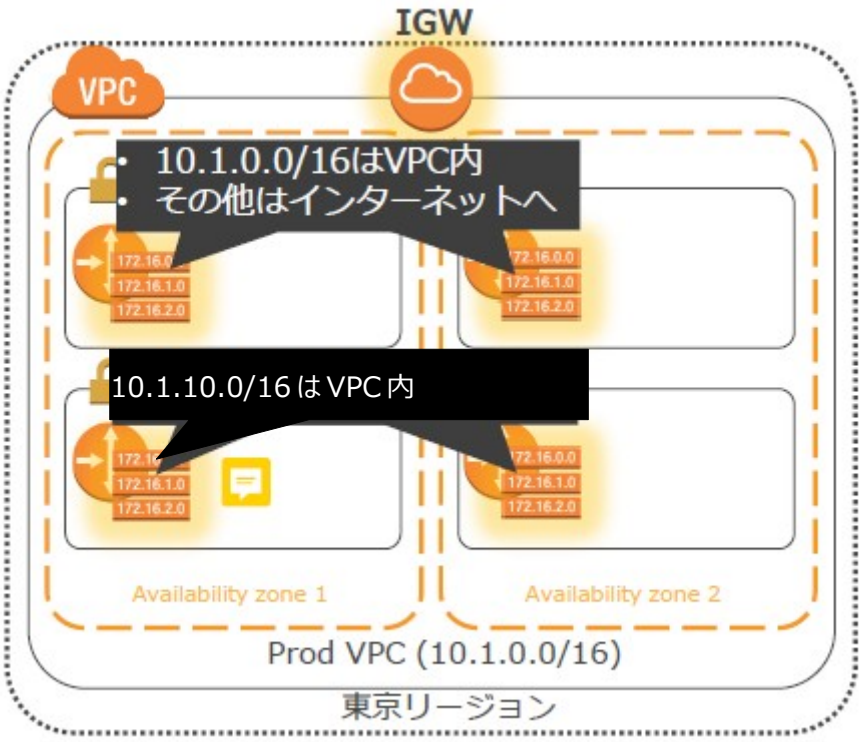
<table>
<tr><td rowspan="2">仮想プライベートゲートウェイ(VGW)
拠点との接続</td><td>カスタマーゲートウェイ(CGW)
VPN 接続</td></tr>
<tr><td>AWS Direct Connect(DX)
専用線接続</td></tr>
<tr><td>インターネットゲートウェイ(IGW)
インターネット接続</td><td></td></tr>
<tr><td>VPC ピア接続
他 VPC との接続</td><td></td></tr>
<tr><td>VPC エンドポイント(VPCE)
VPC 外の AWS サービスとの接続 S3 に対応</td><td></td></tr>
</table>

3.5. サブネット単位で配置するコンポーネント

<table>
<tr><td>VPC ルータ
ルートテーブルに基づいたルーティング（自動的に配置）</td><td></td></tr>
<tr><td>NAT ゲートウェイ
プライベートサブネットに NAT 機能を提供</td><td></td></tr>
</table>

3.6. インスタンス単位で配置するコンポーネント

<table>
<tr><td>Elastic IP(EIP)
固定パブリック IP アドレス</td><td></td></tr>
</table>


IGW
VPC
10.1.0.0/16はVPC内
その他はインターネットへ
172.16.0.0
172.16.1.0
172.16.2.0
10.1.10.0/16 はVPC内
Availability zone 1
Availability zone 2
Prod VPC (10.1.0.0/16)
東京リージョン

4. インスタンスの配置

4.1. 操作方法

インターネットゲートウェイは VPC 内部からインターネット接続するためのコンポーネントであり、VPC にアタッチすることで可能となります。

① AWS マネジメントコンソールより EC2 を選択する

② インスタンスを起動をクリックし作成するインスタンスの設定を行う

③ ステップ 2 で「Amazon Linux2 AMI」を選択する。

④ ステップ 3 の設定

インスタンス数	1
購入オプション	チェック無
ネットワーク	作成した VPC
サブネット	作成したサブネット 1
自動割り当てパブリック IP	有効

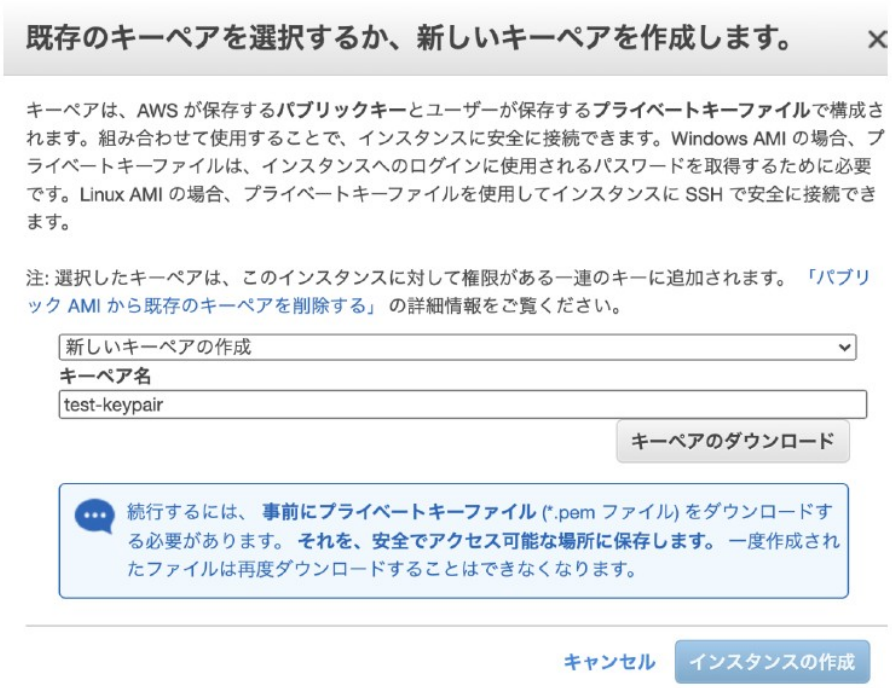
⑤ セキュリティグループは今回はデフォルトのままにする。

⑥ タグとして Name に testec2-1 と名前を付ける。

⑦ 新しいキーペアを作成し、ダウンロードする。

全て終了したらインスタンスを作成ボタンをクリックして作成を完了させる。

※作成に時間がかかる場合があるので、起動中になるのを待つ。



4.2. セキュリティグループの作成

① AWS マネジメントコンソールより EC2 を選択する

② セキュリティグループを選択してセキュリティグループの作成をクリックする

名前	test-sg-1
説明	test-sg-1
VPC	作成した VPC

③ インバウンドの編集

SSH　カスタム：0.0.0.0/0　　今回はすべて

④ アウトバウンドの編集

すべてのトラフィック　カスタム：0.0.0.0/0　　今回はすべて

4.3. インスタンスに割り当てられているセキュリティグループを変更する

① インスタンスを選択し、「アクション」→「セキュリティ」→「セキュリティグループの変更」をクリックする。

② デフォルトのセキュリティグループは削除し、作成したセキュリティグループを追加する。

4.4. EC2 にログインする

ターミナルで EC2 にログイン

キーファイルが必要

ユーザは **ec2-user**

Tera Term を利用

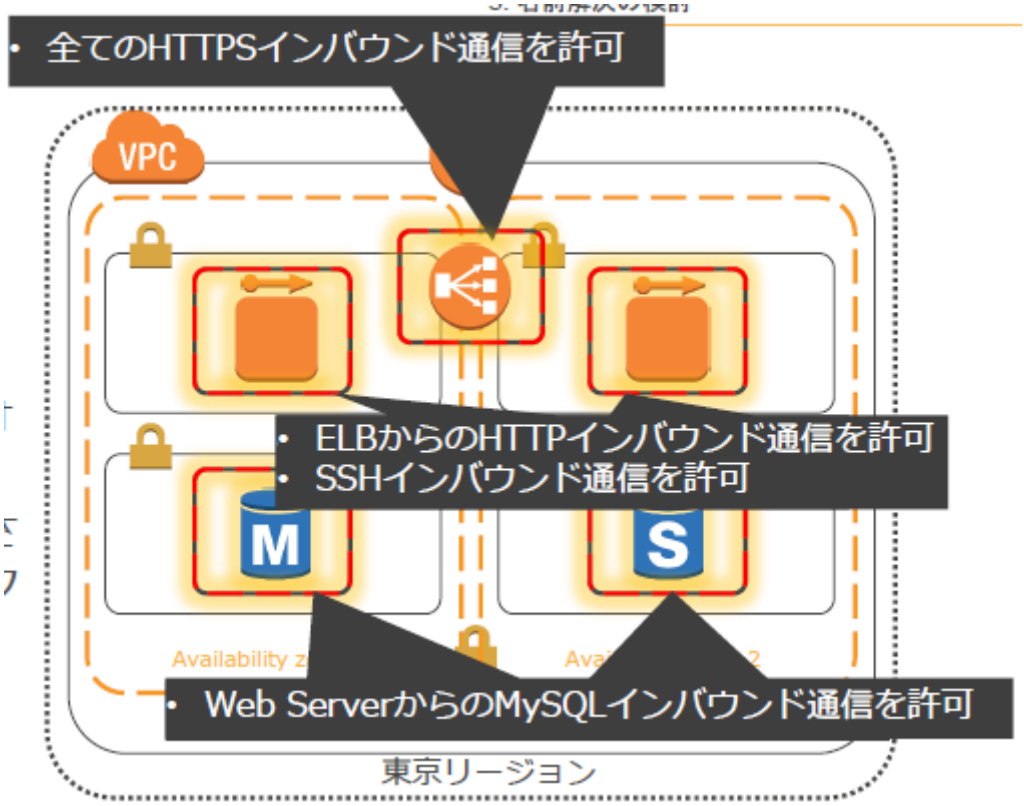
ダウンロード先 https://ja.osdn.net/projects/ttssh2/

4.5. サブネット、インスタンスのセキュリティポリシーを決定する

- セキュリティグループとネットワーク ACL の作成

4.6. インスタンスを配置する

- プライベート IP アドレスはデフォルトで自動割り当て
- インターネットに直接アクセスさせるインスタンスにはパブリック IP アドレスを付与（動的又は EIP）



4.7. VPC のセキュリティコントロール

セキュリティグループ	ネットワークACL
インスタンスに適用	**サブネットに適用**
ホワイトリスト型 Allowのみを指定可能 インバウンド/アウトバウンドに対応	ブラックリスト型 Allow/Denyを指定可能 インバウンド/アウトバウンドに対応
ステートフル 戻りのトラフィックは自動的に許可	**ステートレス** 戻りのトラフィックも明示的に許可設定する
全てのルールを適用	番号の順序通りに適用

例えば下記のような形で相補的に使用

- セキュリティグループ: インスタンスレベルで必要な通信を許可、通常運用でメンテナンス
- ネットワーク ACL ： サブネットレベルでの不要な通信を拒否、メンテナンスは構築時など最小限に

まずはセキュリティグループのインバウンド方向でデザイン

4.8.　インスタンスの種類

参照 URL

https://aws.amazon.com/jp/ec2/instance-types/

汎用	汎用インスタンスは、バランスの取れたコンピューティング、メモリ、ネットワークのリソースを提供し、多様なワークロードに使用できます。汎用インスタンスは、ウェブサーバーやコードリポジトリなど、インスタンスのリソースを同じ割合で使用するアプリケーションに最適です。
	ウェブサーバー、コンテナ化されたマイクロサービス、キャッシングフリート、分散データストア、開発環境といったスケールアウトワークロード
コンピューティング最適化	コンピューティング最適化インスタンスは、高パフォーマンスプロセッサの恩恵を受けるコンピューティングバウンドなアプリケーションに最適です。このファミリーに属するインスタンスは、バッチ処理ワークロード、メディアトランスコード、高性能ウェブサーバー、ハイパフォーマンスウェブサーバー、ハイパフォーマンスコンピューティング (HPC)、科学モデリング、専用ゲームサーバーおよび広告サーバーエンジン、機械学習推論などのコンピューティング集約型アプリケーションに最適です。
	バッチ処理、分散分析、ハイパフォーマンスコンピューティング（HPC）、広告配信、拡張性の高いマルチプレイヤーゲーム、ビデオエンコーディングなど、計算量の多いワークロード。
メモリ最適化	メモリ最適化インスタンスは、メモリ内の大きいデータセットを処理するワークロードに対して高速なパフォーマンスを実現するように設計されています。
	オープンソースデータベース、メモリ内キャッシュ、リアルタイムビッグデータ分析などのメモリ集約型アプリケーション

高速コンピューティング	高速コンピューティングインスタンスでは、ハードウェアアクセラレーター (コプロセッサ) を使用して、浮動小数点計算、グラフィックス処理、データパターン照合などの機能を、CPU で実行中のソフトウェアよりも効率的に実行します。
	機械学習、ハイパフォーマンスコンピューティング (HPC)、計算流体力学、金融工学、耐震解析、音声認識、自律走行車、創薬。
ストレージ最適化	ストレージ最適化インスタンスは、ローカルストレージの大規模データセットに対する高いシーケンシャル読み取りおよび書き込みアクセスを必要とするワークロード用に設計されています。ストレージ最適化インスタンスは、数万 IOPS もの低レイテンシーなランダム I/O オペレーションをアプリケーションに提供するように最適化されています。
	NoSQL データベース (例: Cassandra、MongoDB、Redis)、インメモリデータベース (例: Aerospike)、スケールアウトトランザクションデータベース、データウェアハウジング、Elasticsearch、分析ワークロード。

4.9. インスタンスタイプとは

インスタンスファミリーとは	インスタンスファミリーとは、インスタンスタイプが「m5a.large」の場合、先頭部分の「m」になります。 インスタンスファミリーは「汎用」「コンピューティング最適化」「メモリ最適化」「ストレージ最適化」「高速コンピューティング」の 5 種類があり、それぞれ特徴が異なります。 たとえば基本的なタイプである「汎用」の場合、インスタンスファミリーは「t シリーズ」や「m5」「 m6」「 a1」などです。
インスタンス世代とは	インスタンス世代とは、インスタンスタイプが「m5a.large」の場合「5」の部分になります。インスタンス世代は数字が大きいほど新しい世代となるため、たとえば「m5」と「m4」であれば前者の方が新しい世代になります。 基本的に新しい世代になるほど性能が高く、価格も安価になっていくという傾

	向があります。
追加機能とは	追加機能とは、インスタンスタイプが「m5a.large」の場合「a」の部分になります。追加機能はないタイプもありますが、CPU を Intel 製から AMD 製や AWS Graviton 製に変更したり、ネットワークを強化するなどの変更が行われた場合には追加機能が記載されます。 たとえば CPU を AMD 製に変更した場合、追加機能は「a」、メモリ搭載量を強化した場合追加機能は「e」になります。
インスタンスサイズとは	インスタンスサイズとは、インスタンスタイプが「m5a.large」の場合、最後の「large」の部分になります。インスタンスサイズには複数のサイズが用意されており、「nano」「 micro」「 medium」「 small」「 large」「 xlarge」「2xlarge」のようにサイズが大きくなっていきます。 たとえば「m5a」の場合、「large」以降のインスタンスサイズから選択可能です。

4.10. EC2 を停止する際に気を付けるべきインスタンスタイプ

4.10.1. クレジット機能のある T2 と T3

T2、T3 インスタンスは CPU 稼働率が閾値以下の場合にクレジットを貯める仕組みになっており、負荷が掛かった場合は貯めたクレジットによって性能以上の処理ができる仕組みになっています。

このような T シリーズを停止させた場合、クレジットがリセットされてしまうため注意が必要です。

4.10.2. インスタンスストアが使用できるインスタンスタイプ

EC2 はインスタンスタイプによって、保存先がインスタンスストアのものと EBS のものとにわかれています。インスタンスストアを使用できるインスタンスタイプを停止させた場合、データが消えてしまうため注意が必要です。

4.11. インスタイプの料金

4.11.1. AWS のリザーブドインスタンスとは？

参照：https://aws.amazon.com/jp/ec2/pricing/reserved-instances/

AWS のリザーブドインスタンスとは、オンデマンドインスタンスに比べて大幅に割安のコストで運用できるサービスです。

コストパフォーマンスが良いのと同時に、一定時間、それもなるべく長期期間使用する事を前提としています。

昼夜関係なく一年を通して休む事が無い、常時動作し続けているアプリケーションや社内システムの商用環境に適用すると、大幅なコストの削減が出来ます。

- 大幅な割引
  AWS のリザーブドインスタンスは、オンデマンドインスタンスに比べて 72%のコストを削減する事が可能です。
  また、支払いオプションも複数種類用意されているので、自分の状況に合わせて柔軟に決定する事が出来ます。
- 1 年もしくは 3 年
  AWS のリザーブドインスタンスは 1 年から 3 年間使用する事を前提として提供されるサービスなので、常時動作させる事を前提としたアプリケーションに最適です。
  支払いは、複数のアカウントを纏めて一括で払う事も出来、支払いの業務を簡素化する事も出来ます。

4.11.2. リザーブドインスタンスのプラン

AWS は、ユーザーの状況になるべく柔軟に対応するために、いくつかあるプランを選べるようにしていますが、リザーブドインスタンスも例外ではありません。

使用する料金や機能、支払い方法などにもいくつか種類があるので、長期的な利用料金を安くしつつも、高機能なインスタンスを使用する事が可能です。

- スタンダードとコンバーチブル
  AWS のリザーブドインスタンスにはスタンダードとコンバーチブルという二つのプランがあり、機能面に差があります
  コンバーチブルの方が、アベイラビリティゾーンや OS の変更が可能になるため、その後のビジネスの状況に合わせて柔軟に変更する事が可能になります。
- 支払い方法は 3 種類
  AWS のリザーブドインスタンスには支払い方法が全部で 3 種類あり、全額前払い、一部前払い、前払いなし、となっています。
  全額前払いを選択した場合、割引率を大きくすることができ、一部前払いと前払いなしを選択した場合は、毎月無理のない金額を支払う、という形になります。

4.11.3. リザーブドインスタンスの注意点

AWS のリザーブドインスタンスは、オンデマンドインスタンスと比べてコストパフォーマンスが大きく優れているので、上手く使えば経費の大幅な削減につながります。

しかし、長期的且つ恒常的にインスタンスを使用する事を目的としているため、気を付けないと、逆に予想外に費用が大きくなることもあり得ます。

- サーバーが休止していても料金がかかる

  AWS は一定期間、継続してサーバーを利用することを前提にしたサービスなので、サーバーが休止していても料金がかかります。

  よって、リザーブドインスタンスは一定の時間しか稼働していない、開発環境やテスト環境に使用する事は不適切で、商用のシステムやアプリケーションを動かすような環境でリザーブドインスタンスを使用しましょう。

  もしもテスト環境や開発環境等にリザーブドインスタンスを使用していた場合、コスト削減どころか、オンデマンドインスタンスを使用するよりも多くの無駄なコストを生んでしまいます。
- AWS の柔軟性を活かしづらくなる

  AWS のリザーブドインスタンスは、同じスペックのインスタンスを継続的に使用する仕様になっており、支払期間の途中でインスタンスのスペックを上げる、若しくは下げる、という行為が取りづらくなっているので、AWS の柔軟性が活かしづらくなっています。

  よって、アプリケーションを運用する場所を変えたい、などの要望があった場合はインスタンスを新規で購入しなくてはいけません。

5. 名前解決の検討

5.1. 自動割り当ての DNS 名を活用する

- AWS では IP アドレスでなく DNS 名を活用してアプリの設計を行うことを推奨
- VPC では暗黙的に DNS が動作
- インスタンスには自動で DNS 名が割り当てられる

5.2. 独自 DNS 名を使用する

- Route 53 により独自 DNS 名を割り当て、管理することが可能