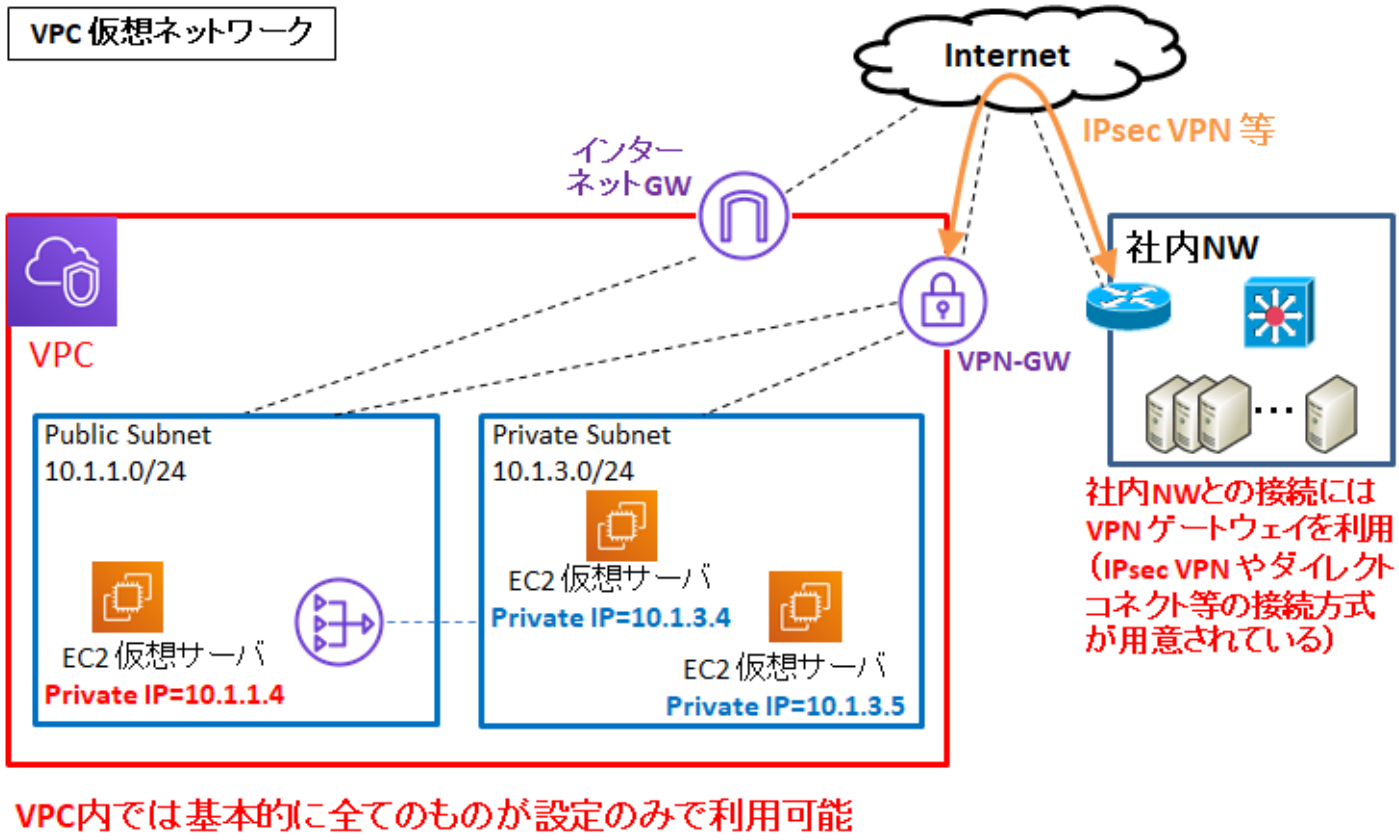
VPC とは

VPC（読み方：ぶいぴーしー）とは Virtual Private Cloud のことで、AWS 内に EC2 等のインスタンスを配置する**仮想ネットワーク空間**です。本記事ではこの VPC について解説していきます。

一般に仮想ネットワークといっても色々な意味合いを持っていますが、ここでいう仮想ネットワークとは「仮想基盤ホスト上がメインとなって作られる IP ネットワーク」です。

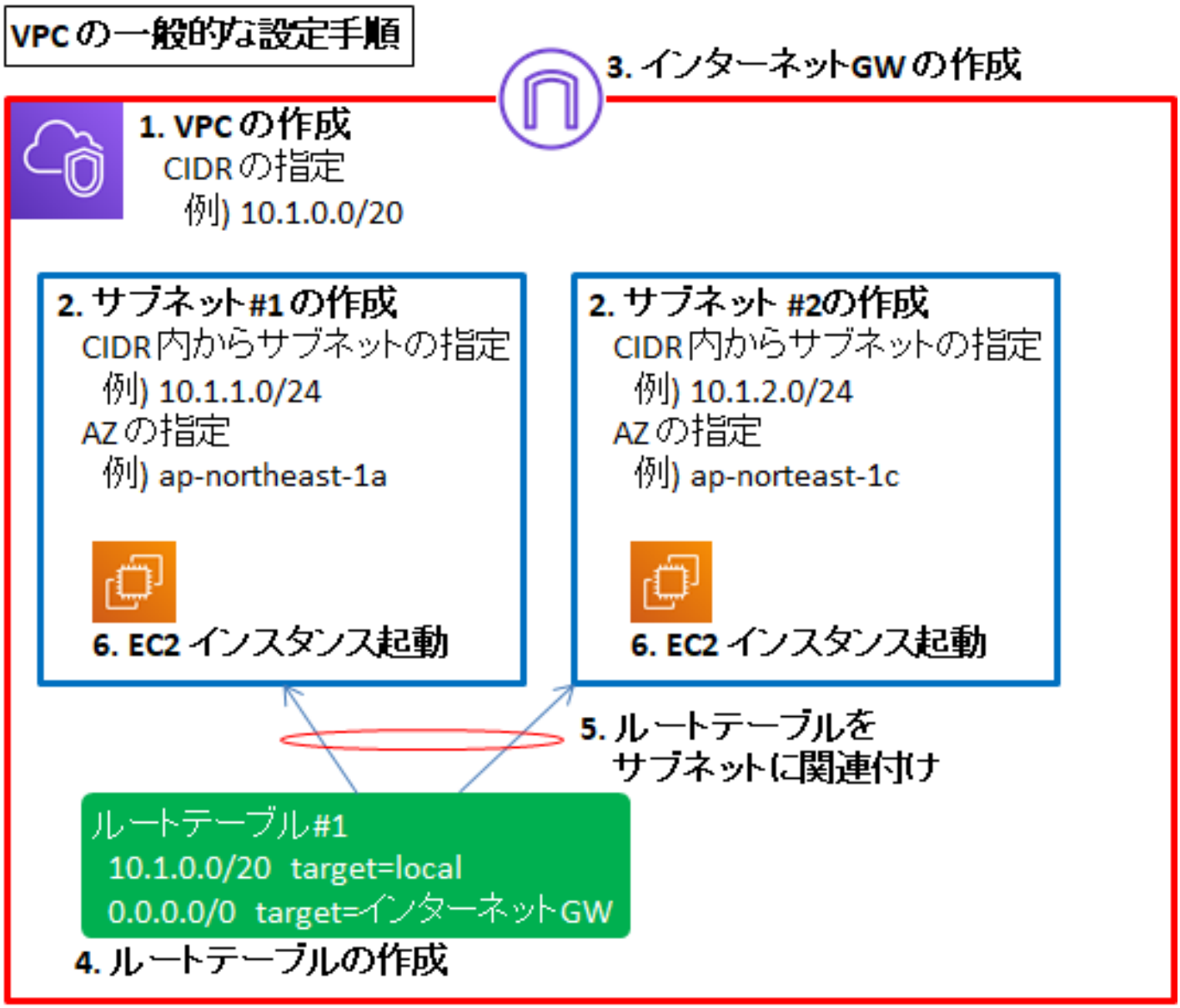
仮想であるがゆえに、**物理的な配線作業やハードウェア追加をすることなく、設定だけで**簡単にインスタンスを NW 接続したり、サブネットを増やしたり、仮想ロードバランサを増やしたりすることができます。



VPC 内のインスタンスにはプライベート IP やパブリック IP（グローバル IP）を付与することができ、パブリック IP があれば（適切なアクセス許可設定があれば）インターネットからのアクセスが可能です。

また、VPC 内に VPN ゲートウェイを作成し、IPsec VPN やダイレクトコネクト等を使って社内 NW と接続すれば、プライベート IP のままアクセスすることも可能です。なお、ダイレクトコネクトに関しては AWS 指定のデータセンター内で AWS ユーザーが『回線敷設、AWS機器へのEthernet接続』といった物理的な作業が必要になります。

VPC を構成して EC2 を起動するまでの一般的な設定手順例を以下に示します。



1. VPCの作成: CIDRを指定
2. サブネットの作成: CIDRから切り出し、AZを指定
   ★AZの異なる複数のサブネットを作成し冗長化することで信頼性向上
3. インターネットGWを作成
4. ルートテーブルを作成
5. ルートテーブルをサブネットに関連付け
6. EC2インスタンスを起動

AZ（アベイラビリティーゾーン）とは

VPC の設計をする上でまず大きなポイントとなるのが AZ (Availability Zone) です。AWS では 1つのデータセンターを 1つの AZ に紐付けています。つまり、複数の AZ に同等のサーバを分散配置することで、「例えデータセンター単位での大障害が発生したとしても冗長性が担保される」のです。

AZ は VPC 内に配置する「サブネット」を作成する際に選択する必要があります。つまり AZ が異なる場合はサブネットも異なりますので、同等のサーバを AZ に跨って分散配置する場合は必ず別サブネットになります。

ルートテーブル

一般的なルーティングの考え方だと各NW機器にルートテーブルを持つのですが、AWS のルートテーブルは特殊で、作成したルートテーブルをサブネットに紐付けます。

ネットワークエンジニアの方はちょっと戸惑うと思います。ここではイメージをしやすいように私なりの解釈を付けてみたいと思います。

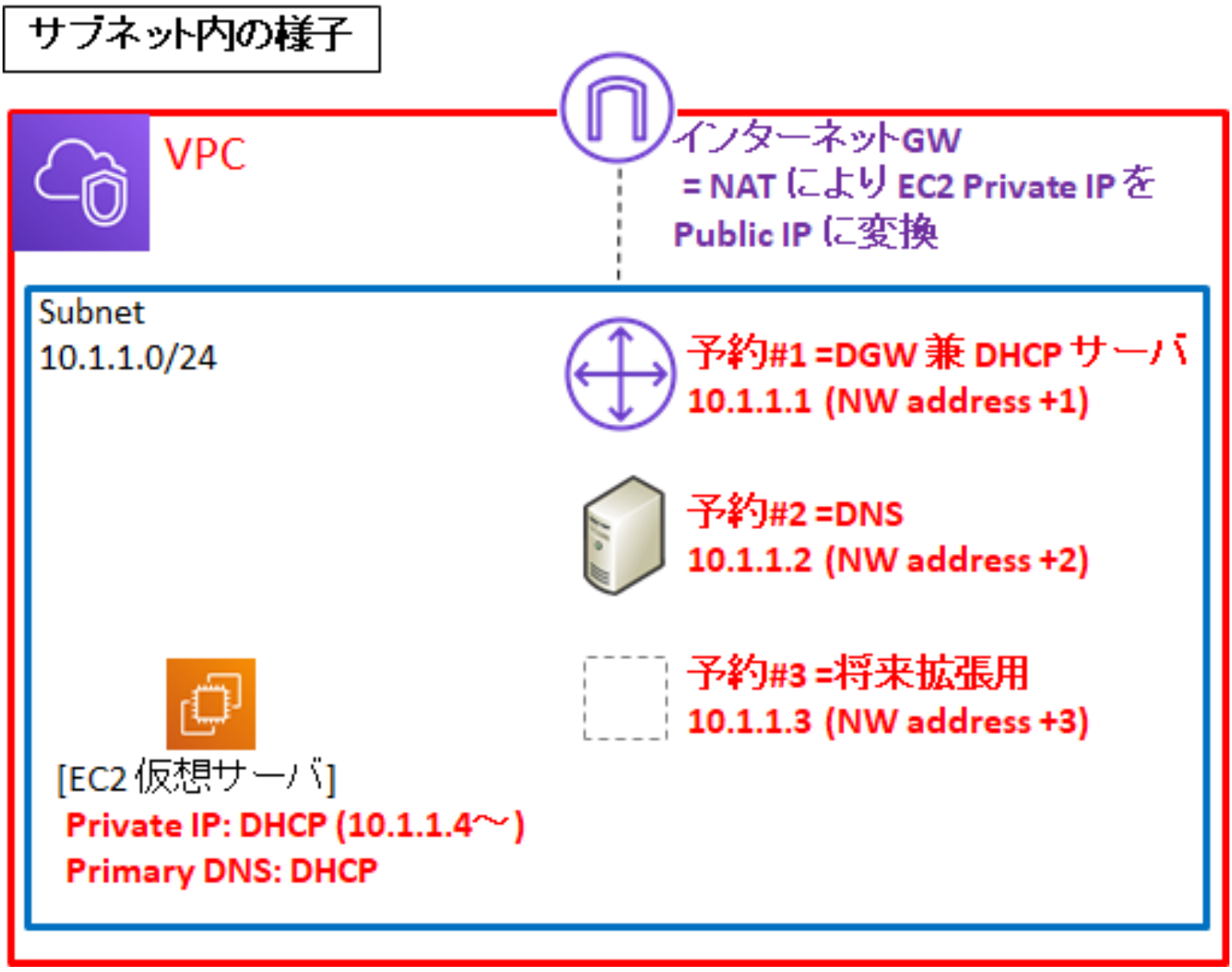
まず前提として、AWS のサブネットは利用可能 IP アドレスレンジのうち最初の 3つは以下の用途で予約されます。

デフォルトゲートウェイ(DGW)用

DNS 用

将来の拡張用（現時点では用途無し）

例えばサブネットが 10.1.1.0/24 の場合、DGW は 10.1.1.1 であり DNS は 10.1.1.2 であり、EC2 インスタンス等に払い出される IP レンジは 10.1.1.4 以降となります。



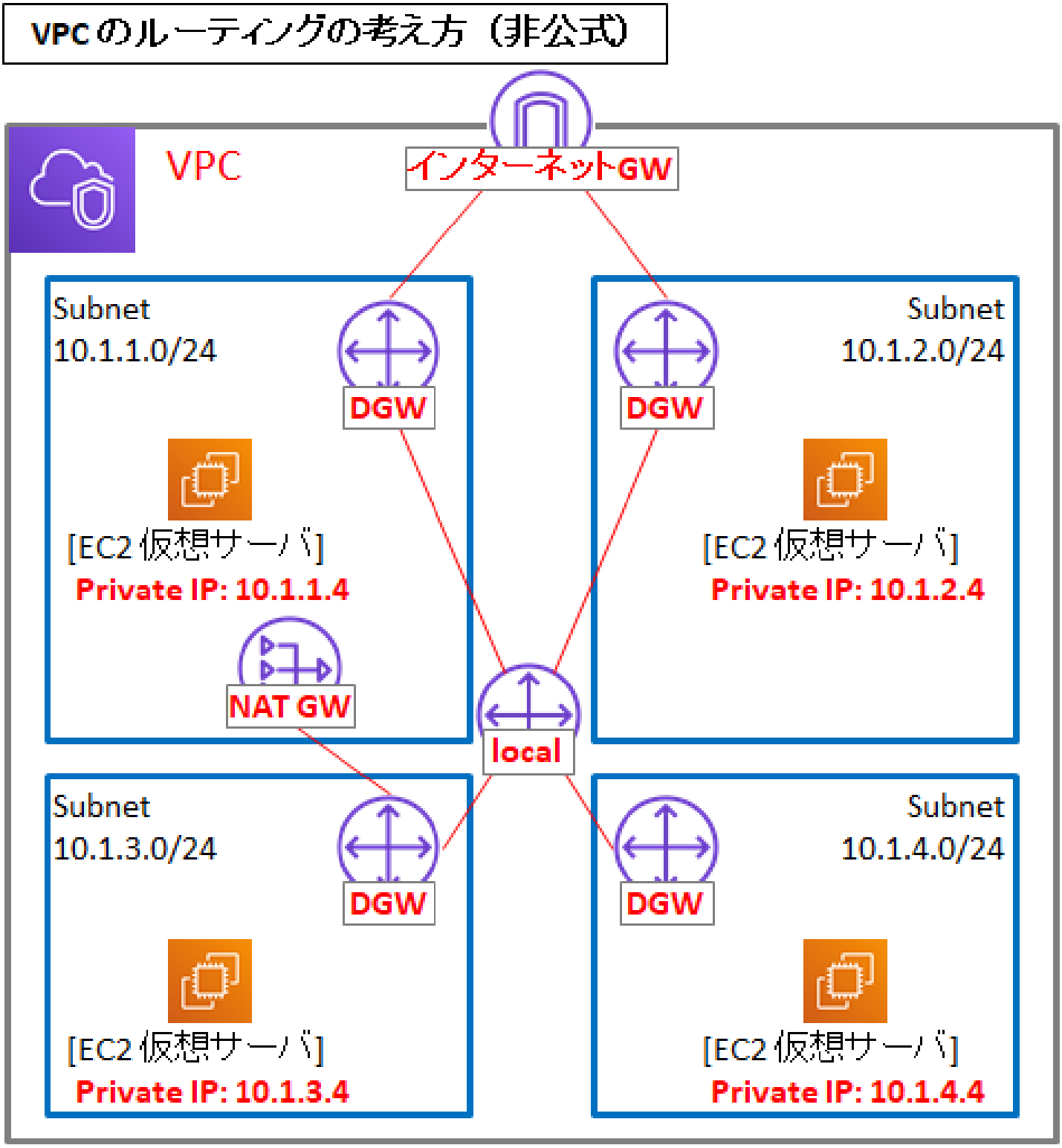
また、EC2 インスタンス等の OS に割り当てられる IP, サブネットマスク, DGW, DNS は AWS が用意した DHCP を使って払い出されます。(/var/log/messages を見ると DGW が DHCP サーバを兼務していることが分かります。)

そして OS が取得する IP は原則プライベート IP です。デフォルト設定だと払い出される IP は起動するたびに変化する可能性がありますが、設定で固定化（DHCP で払い出される IP が必ず同じものになる設定）することも可能です。

パブリック IP は原則 NAT によって提供されます。このあたりについては後述します。

以上が前提知識です。

そして本題の VPC のルートテーブルについてですが、ネットワークエンジニアにとって馴染みがあり、矛盾しない構成図にしようとすると以下のようになります。※ AWS 非公式の個人的な見解です。



- サブネット間の通信はサブネットの中央にある “local” ルータ経由で通信する（local ルータは各 DGW への個別ルートを持っている）
- インターネット GW や NAT GW は各 DGW ルータと直結している
- 各NW機器は p2p 接続および IP unnumbered を使って接続している

そしてルートテーブルをサブネットに関連付けると、DGW にそのルートテーブルを持つ、と考えられます。

サブネット間通信についてはデフォルトでルートテーブルが保有しているエントリ「10.1.0.0/20 Target=local」を使います。10.1.0.0/20 の範囲の通信は local ルータへ送信されます。（各 DGW と local ルータは p2p かつ IP unnumbered で接続）

また、インターネットとの通信をしたいときはデフォルトルート(0.0.0.0/0) のTarget をインターネット GW、もしくは NAT-GW にします。インターネット GW を使う場合は「0.0.0.0/0 Target=igw」、NAT GW を使う場合は「0.0.0.0/0 Target=nat-gw」です。

これらの GW もやはり各サブネットの DGW と p2p かつ IP unnumbered で接続されていると考えられます。

各工程のポイントや注意点

1. VPC作成

EC2 等の「インスタンス」は「サブネット」内で起動します。そして「サブネット」は「VPC」内に配置します。そのため、まずは VPC を作成します。

VPC の作成には CIDR を1つだけ設定する必要があります。2つ以上は割り当てられません。

例えば CIDR を 10.1.0.0/20 に設定した場合、その VPC には /24 のサブネットなら「10.1.X.0/24 (X = 0～15)」の16個のサブネットが作成できますし、/23 のサブネットなら「10.1.2Y.0/23 (Y = 0～7)」の8個のサブネットが作成できます。

このように、VPC の CIDR 設定は IP アドレスの拡張性に影響を与える、ということを意識しなければなりません。

なお、CIDR およびサブネットに適用可能なサブネットマスク長は /16 ～ /28 です。

2. サブネット作成

次に、先ほど作成した VPC の中に、必要な数だけサブネットを作成します。

サブネット作成における重要なポイントは、「パブリック or プライベートの選択」と「AZ の指定」です。

今回はインターネットへ公開する Web サーバーを配置する「パブリックサブネット」と、バックエンドで動作させる DB サーバーを配置する「プライベートサブネット」の 2種類を使います。また、それぞれに対して AZ を 2つずつ用意し、冗長化を図ります。

そのために必要なサブネットは 2 * 2 = 4つです。

そこで今回は 10.1.1.0/24 , 10.1.2.0/24 , 10.1.3.0/24 , 10.1.4.0/24 の4つのサブネットを作成することにします。

設計として 10.1.1.0/24 と 10.1.2.0/24 をパブリックサブネット（Webサーバ配置）、10.1.3.0/24 と 10.1.4.0/24 をプライベートサブネット（DBサーバ配置）とします。また、10.1.1.0/24 と 10.1.3.0/24 を AZ#1、10.1.2.0/24 と 10.1.4.0/24 を AZ#2 としてアベイラビリティゾーンを分けます。

3. インターネット GW 作成および VPC へのアタッチ

インターネット GW を作成し、VPC にアタッチします。ですがまだこの時点ではパブリックサブネットとして設計したサブネットはインターネット接続できません。次に実行する「ルーティングテーブルの作成およびサブネットへの関連付け」が必要になります。

4. ルートテーブル作成

各サブネットが別のサブネットやインターネットと通信するためには、ルートテーブルを作成し、それを各サブネットにアタッチしなければなりません。

まずはパブリックサブネット用のルートテーブルを作成します。

ルートテーブルは VPC の管理画面から作成します。作成されたルートテーブルは VPC 内の複数のサブネットに関連付けが出来ますが、1 つのサブネットには 1つのルートテーブルしか関連付けできません。

ルートテーブルを作成するとデフォルトで「VPC 内のルーティング許可(ターゲットがlocal)」設定が含まれています。このテーブルにインターネット接続を許可する設定として「0.0.0.0/0 のターゲット=インターネットGW」を投入します。

このルートテーブルを、10.1.1.0/24 と 10.1.2.0/24 と関連付けます。この瞬間にこのサブネットは「パブリックサブネット」となり、インターネットとの疎通ができるようになります。

プライベートサブネットは Web サーバと通信ができればよいので、ルートテーブルのデフォルト設定「10.1.0.0/20 のターゲット=local」が設定されたルートテーブルを 10.1.3.0/24 および 10.1.4.0/24 に関連付けします。

EC2 インスタンスの起動

EC2 インスタンスの起動 AMI 等を選び、起動するサブネットを選択します。

Web サーバを起動する場合はパブリックサブネットを選択しますが、パブリックサブネットで起動すると「パブリック IP (グローバル IP)」が払い出し可能です。この場合、自動的にインターネット GW に Static NAT が設定されます。

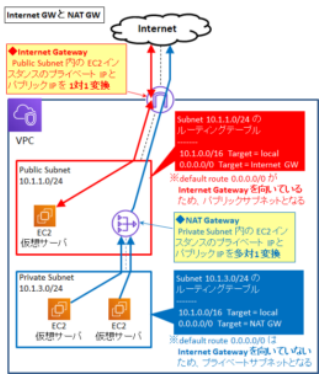
例えば DHCP の払い出し IP が 10.1.1.4 で、パブリック IP の払い出しが 20.30.40.50 だった場合は 20.30.40.50 -> 10.1.1.4 の NAT 変換設定が自動でインターネットGWに組み込まれます。

また、プライベートサブネット上のインスタンスはそのままではインターネット通信ができません。例えば Windows Update 等でインターネット通信を行いたい場合は NAT-GW等を使うことになります。

インターネット GW と NAT-GW の使い方や使い分け、違いについては以下を参照して下さい。

【図解/AWS】インターネットGWとNAT-GWの違い～各メリット、パブリックサブネットとは～

インターネットゲートウェイとNAT ゲートウェイの違い インターネットゲートウ...



milestone-of-se.nesuke.com 2021.01.21

ELB の設定イメージ

次に、Web サーバや DB サーバのロードバランスを考えます。ロードバランスを行うには Elastic Load Balancer (ELB) を使います。

ELB には以下の 3種類があります。

Application Load Balancer (ALB：http, https レベルのロードバランス)

Network Load Balancer (NLB：TCP レベルのロードバランス)

Classic Load Balancer (CLB：旧世代ロードバランス)

これから新規作成するインスタンスをロードバランスする場合には 3 の選択肢は推奨されません。これは旧世代のインスタンス向けのものです。

1 の ALB では http, https のみがサポートされますが、その分、httpに特化した機能が充実しています（TLS の SNI 対応やクッキーを識別するスティッキーセッション等）。

2 の NLB では TCP レイヤーをカバーするロードバランサです。ALB と同様、サーバ証明書をセットして TLS の終端をすることも可能ですが、SNI には対応していないためバーチャルドメインが使えないことや、クッキーセッションを意識したバランシングができない等、http(s) には不向きです。

ELB の一般的な構成は、ロードバランス対象インスタンスが存在する複数のサブネットにアタッチし、ターゲットグループとして該当インスタンスを指定します。

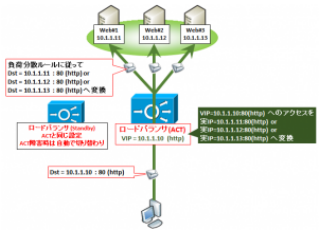
ヘルスチェックによる障害検知も可能ですので障害サーバにバランスされることを回避できます。

ロードバランサを使う上で一番の注意点は「サーバへのアクセス時には送信元 IP が (クライアントIPではなく) ロードバランサの IP に NAT されている」ことです。

これは AWS ELB 固有の性質ではなく、一般的なロードバランサの性質です。

ロードバランサでは TLS 終端（暗号化/復号化）をするためにも、通信内容を往路・復路ともに把握する必要があります。なのでサーバからの戻りの通信が必ずロードバランサ経由となるように、NAT （あるいは Proxy）による送信元 IP 変換が行われるのです。

このあたりの詳細については、一般的なロードバランサの仕組みとして以下を参照して下さい。



【図解】ロードバランサ(L7)の仕組み～セッション維持方式、SSL証明書やリバースプロキシ等の構成例～

ロードバランサとは ロードバランサとは、サーバの負荷分散を行いつつ、冗長化を実...

milestone-of-se.nesuke.com 2020.03.10

この送信元 IP 変換により主に 2つの弊害が発生します。

1つはセキュリティグループ（仮想FW）の設定が意図しないものになる可能性です。セキュリティグループで送信元 IP による制御をしている場合、ELB 経由となることでこれをすり抜けられてしまいます。ロードバランサ自体にもセキュリティグループの設定が可能ですので、こちらにも確実に適用しなければなりません。

もう1つはサーバ側のアクセスログです。通常はクライアントIPが記録されるはずのアクセスログですが、ソースNATにより全てロードバランサのIPが記録されてしまう可能性です。（ネットワークロードバランサの場合はソースNATを使わない方式もできますが条件次第です。）

このようなケースでは一般に、ロードバランサが http ヘッダの "X-Forwarded-For" 属性にクライアント IP を埋め込む方式や、Proxy Protocol でクライアント IP を伝達する方式が使われますので、これらを検討する必要があります。