AWS

SUMMIT

AWS のネットワーク設計入門

アマゾン ウェブ 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　岡本　京
2017/5/31

amazon web services

自己紹介

- **岡本　京（おかもと　ひろし）**
  - 所属と職種
    - アマゾン ウェブ サービス ジャパン株式会社
      技術統括本部 ストラテジックソリューション部
      ソリューション アーキテクト
  - 経歴
    - プリセールスエンジニア（ネットワーク）→ AWS
  - 好きなAWSサービス
    - Amazon VPC

amazon web services | Certified

Solutions Architect - Professional

DevOps Engineer - Professional

本セッションの内容

- AWS上でシステムを構築するにあたり、**ネットワーク面ではどのような検討や設計が必要なのか**をお伝えします

- 機能の詳細や操作手順ではなく、**考え方やデザインの説明**にフォーカスさせて頂きます

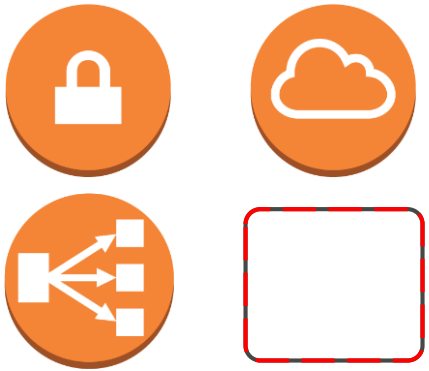
- IPアドレスのサブネッティング、ルーティング、DNSなどの基本的な知識を前提とさせていただきます

目次

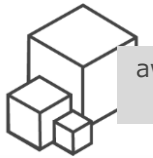
AWS上でのネットワーク設計のポイント

物理設計の
検討、構築が不要

マネージドサービス
による運用負荷の軽減



プログラマブルな
作成、管理、展開



```
aws ec2 create-vpc
    --cidr-block 10.0.0.0/16
```

Amazon Virtual Private Cloud

テンプレート名	説明	表示	デザイナーで表示	作成する
Amazon VPC における単一の Amazon EC2	VPC を作成し、Elastic IP アドレスとセキュリティグループを持つ Amazon EC2 インスタンスを追加します。	表示	デザイナーで表示	Launch Stack
既存の VPN への静的ルーティングを使用する Amazon VPC	既存の VPN エンドポイントへの静的ルーティングを使用して VPN 接続を行うプライベートサブネットを作成します。	表示	デザイナーで表示	Launch Stack
Amazon VPC における Auto Scaling および負荷分散機能を備えたウェブサイト	既存の VPC 内に負荷分散および Auto Scaling 機能を備えたサンプルウェブサイトを作成します。	表示	デザイナーで表示	Launch Stack
DNS およびパブリック IP アドレスを持つ Amazon VPC	DNS サポートおよびパブリック IP アドレスが有効な VPC を作成します。	表示	デザイナーで表示	Launch Stack

AWSのネットワーク関連サービス

Amazon Virtual Private Cloud (VPC)

AWS上にプライベートネットワークを構築

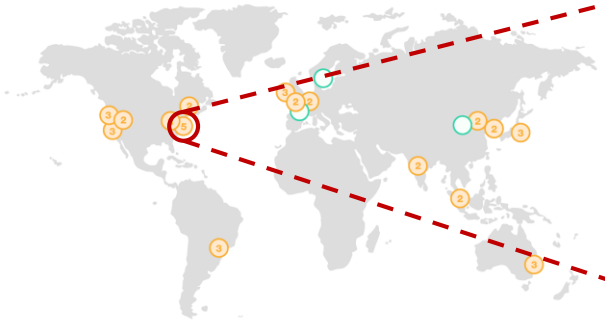
AWS Direct Connect (DX)

AWSと自社拠点/DCの専用線接続

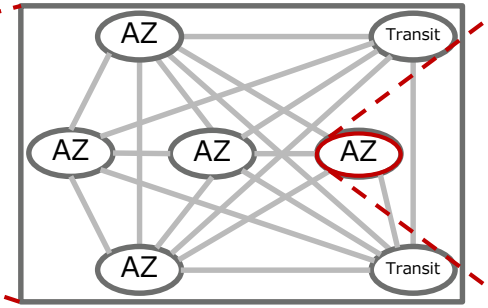
Amazon Route 53

パブリック/プライベートに対応したマネージドDNSサービス

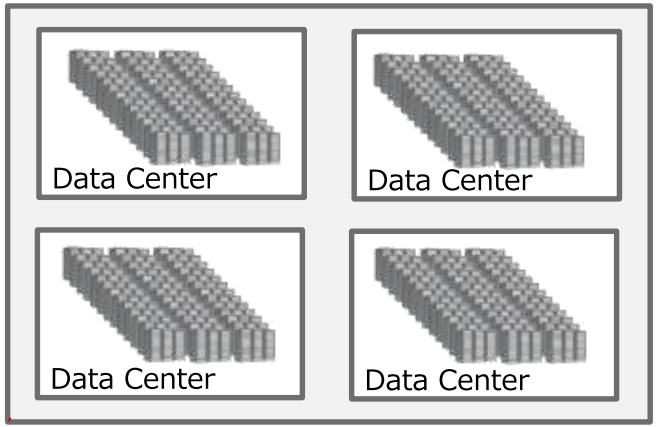
AWSインフラストラクチャとネットワーク関連サービス



リージョンの配置図



リージョンの構成
US East (Northern Virginia) の例



アベイラビリティゾーンの構成

16の**リージョン**
- 42の**アベイラビリティゾーン（AZ）**で構成

VPCは
リージョン内で稼働

53の**Direct Connect ロケーション**
- リージョンとお客様拠点の相互接続ポイント
- 日本は東京、大阪の2箇所

DXは
DXロケーションで物理接続

77の**エッジロケーション**
- CDN（CloudFront）エッジサーバーなどが配置

Route 53は
エッジロケーション内で稼働

※ 2017年5月現在

設計をはじめましょう

AWS上で何をしますか?

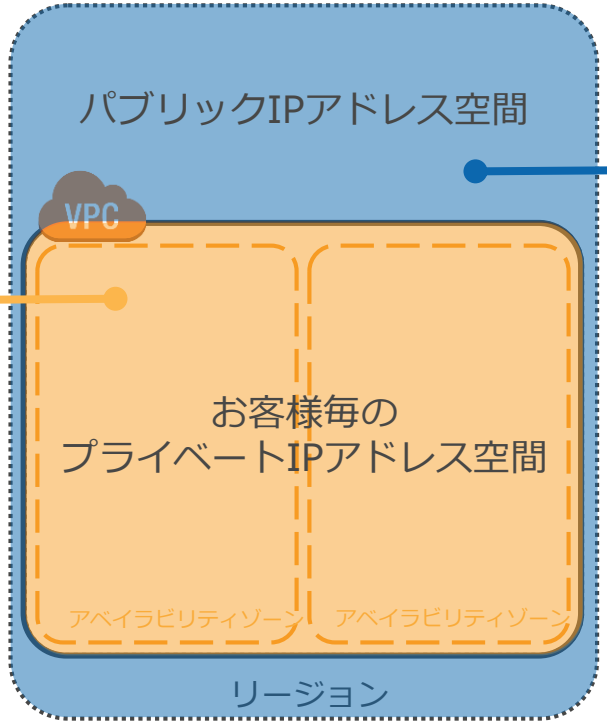
AWS上でのネットワークの検討ポイントは
使いたいサービスによって異なります

AWSサービスのネットワーク観点での分類

プライベートIPアドレス空間上で使用するサービス

- VPCを用いてアドレス空間を構成
- インスタンスの配置をお客様が意識して管理

例）



パブリックIPアドレス空間上で使用するサービス

- 抽象度が高く、お客様は構成を意識せずにサービスを使用
- AWSマネジメントコンソール、各APIエンドポイントもここに存在

例）

※ LambdaはVPC内での起動も選択可能

システム要件とネットワーク関連サービスのマッピング

要件を実現するためのAWSサービスの選定/組み合わせは是非SAにご相談ください！

システムの構成要素		社内環境との通信要件	
プライベートIPアドレス空間で使用するサービスを含む 例） • EC2やRDSを使用した社内システム 本セッションのフォーカス	VPC	大量のデータ連携	DX プライベート接続
		閉域網での接続が必要	
		コスト重視で検討	VPC VPN接続
		特になし	HTTPS/SSH アクセス
プライベートIPアドレス空間で使用するサービスは不要 例） • S3へのデータバックアップ • Lambda, API Gateway, DynamoDBによるサーバーレスWebアプリ	サービスのパブリックDNS名を使用	大量のデータ連携	DX パブリック接続
		閉域網での接続が必要	
		コスト重視で検討	HTTPS/SSH アクセス
		特になし	

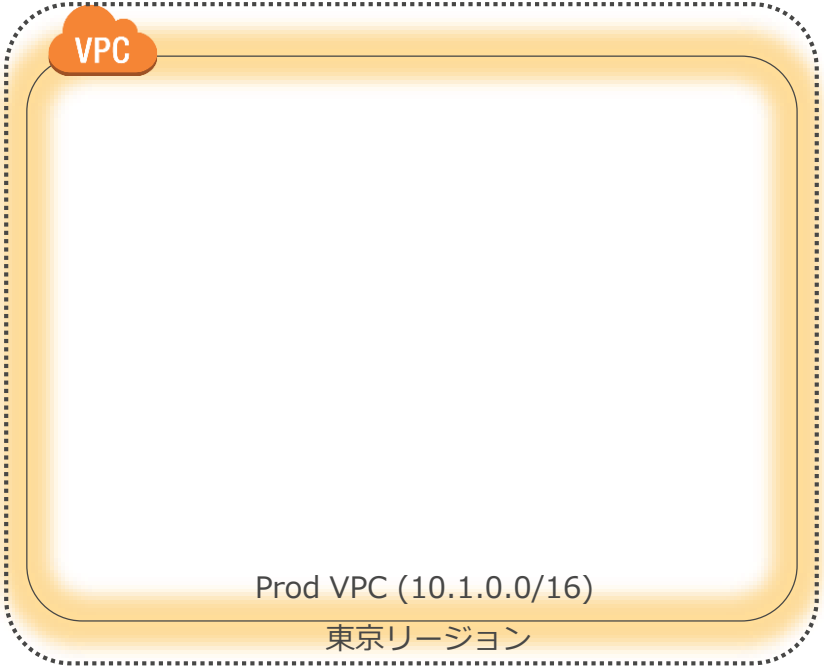
ドメインと名前解決	
独自ドメイン名で公開サービスを展開する	Route 53 パブリックホストゾーン
独自ドメイン名で社内サービスを展開する	Route 53 プライベートホストゾーン
独自ドメイン名を使用せずサービスを展開する	VPC Amazon DNS Server

プライベートネットワーク設計のステップ

1. VPCの作成
2. サブネットの作成
3. VPCコンポーネントの配置とルーティング設定
4. インスタンスの配置
5. 名前解決の検討

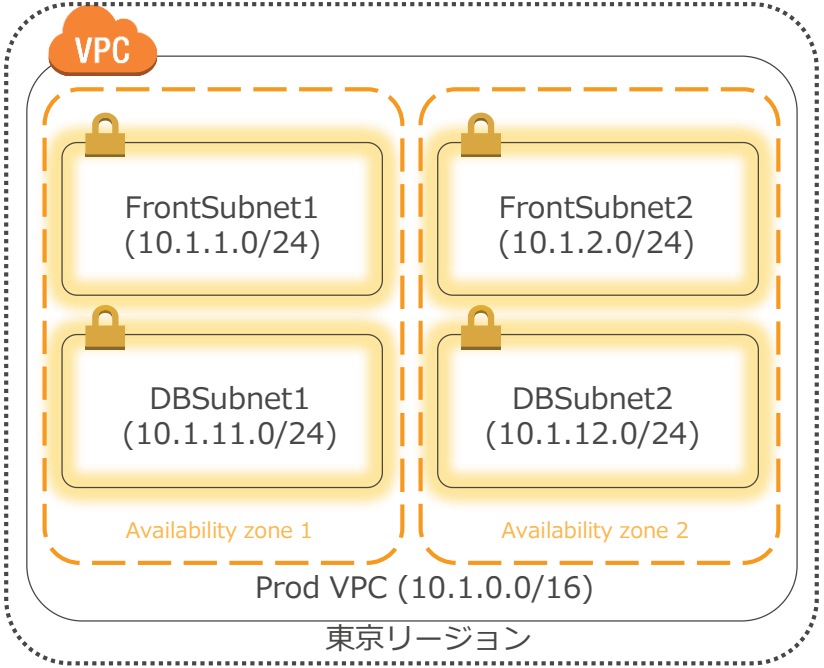
ステップ1. VPCの作成

- 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
- 作成後は変更不可のため大きめに
  - /16 が推奨
- 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



ステップ2. サブネットの作成

- 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
- 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
- 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



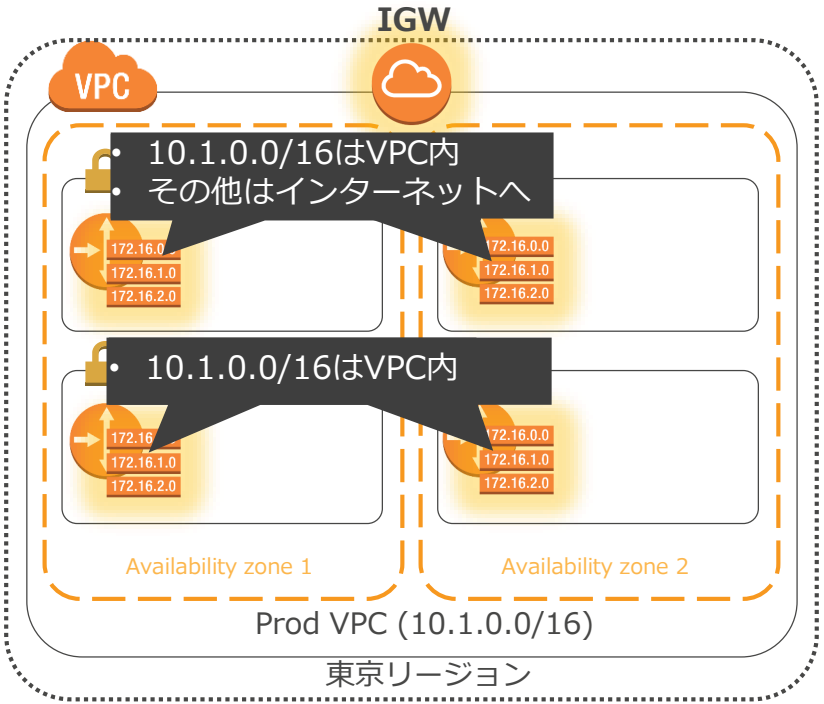
サブネットのサイズの検討

	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
推奨	**/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
  - .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

ステップ3. VPCコンポーネントの配置とルーティング設定

- 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
- 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



VPCコンポーネントの種類（抜粋）

VPC単位で配置するコンポーネント

VGWの接続先

カスタマーゲートウェイ（CGW）
VPN接続

AWS Direct Connect（DX）
専用線接続

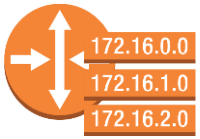
仮想プライベートゲートウェイ（VGW）
拠点との接続

インターネットゲートウェイ（IGW）
インターネット接続

VPCピア接続
他VPCとの接続

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

サブネット単位で配置するコンポーネント



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）

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

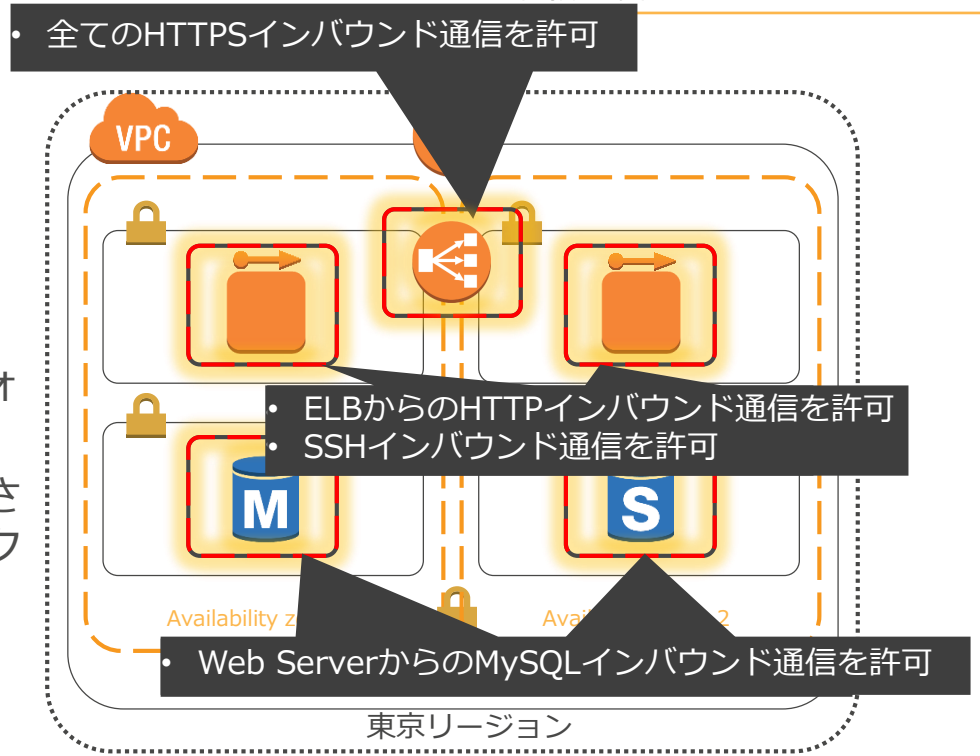
インスタンス単位で配置するコンポーネント

Elastic IP（EIP）
固定パブリックIPアドレス

抽象化されたVPCコンポーネントを活用することで管理工数を削減、自動化を促進

ステップ4. インスタンスの配置

- 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
- 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）



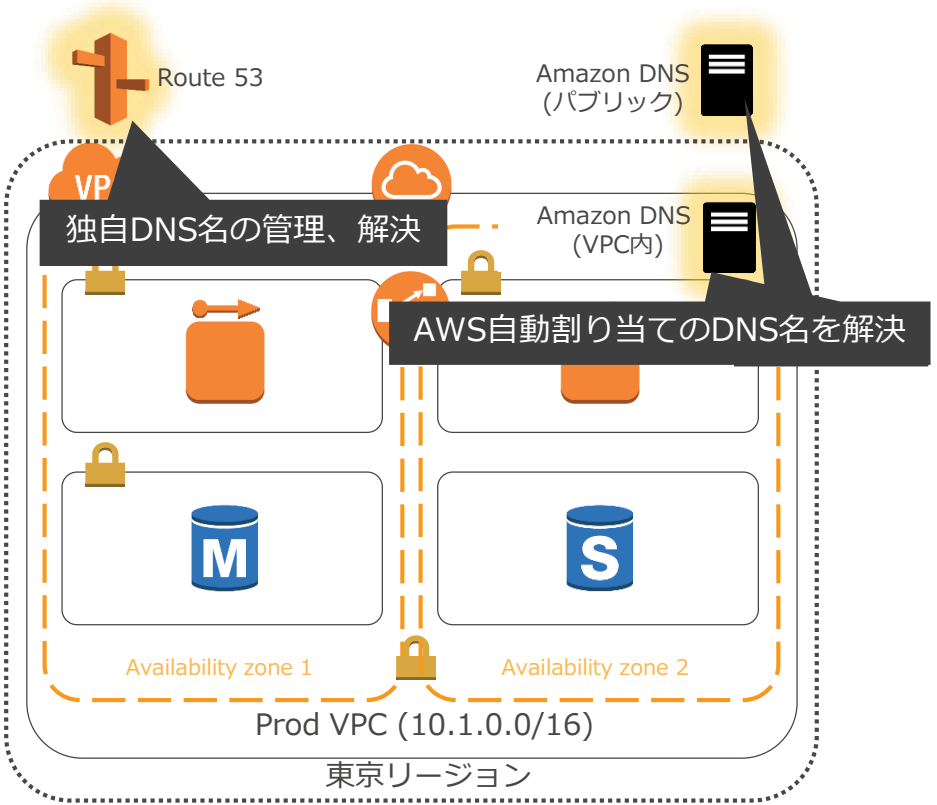
VPCのセキュリティコントロール

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

ステップ5. 名前解決の検討

- 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
- 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

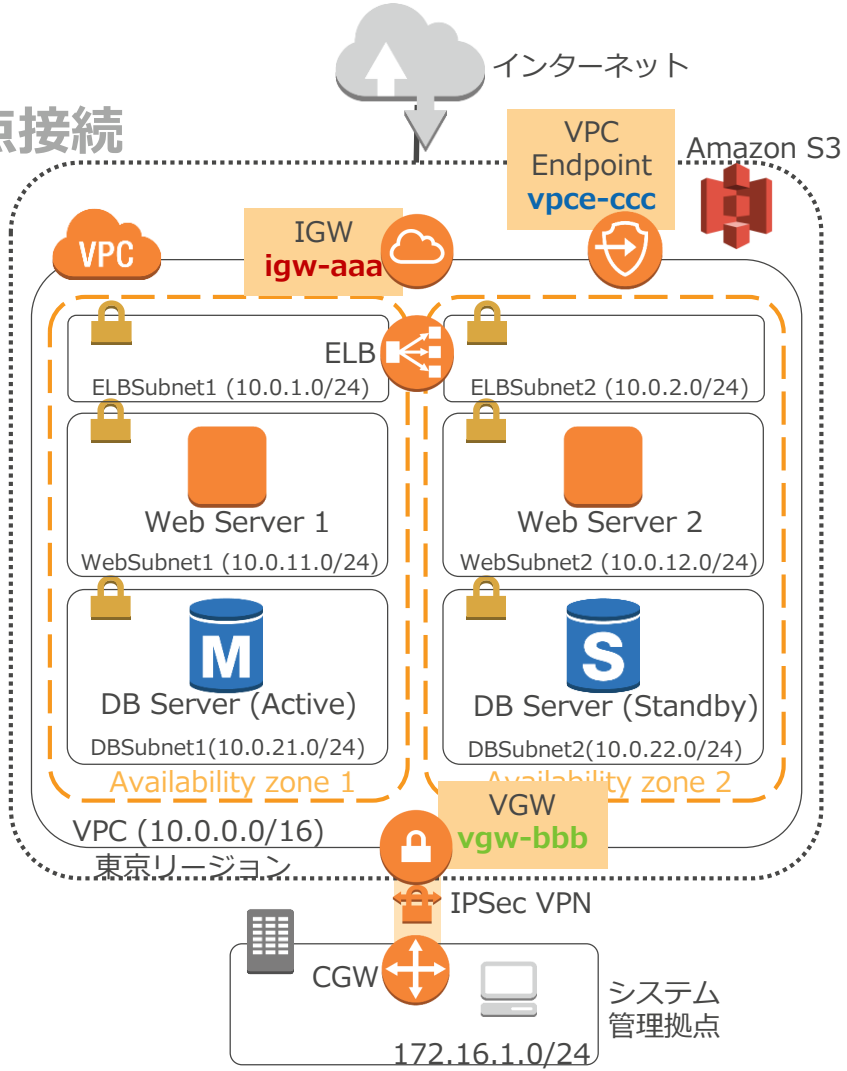


ユースケース別ネットワーク設計例

1. 公開サービス基盤 - 管理拠点とVPN接続
2. 社内システム基盤 - オンプレミスとハイブリッド運用

例1. 公開サービス基盤

Webサービス基盤、管理用にVPNで拠点接続

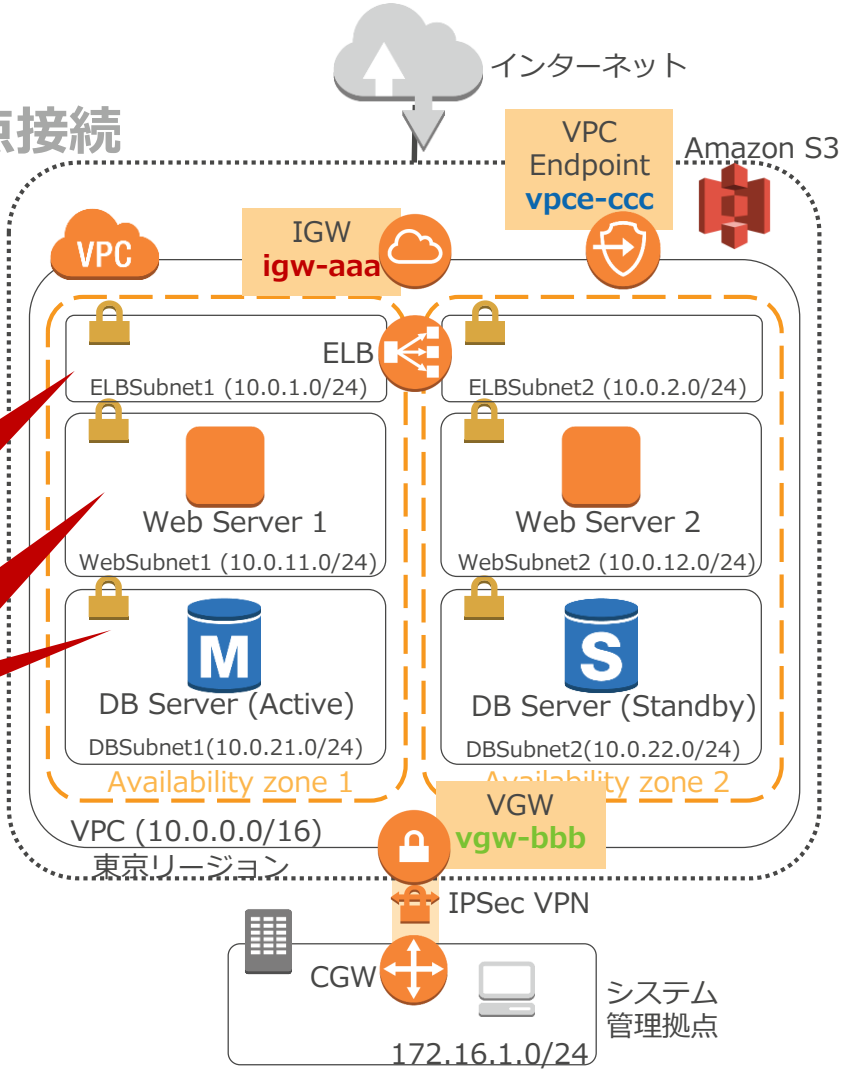


例1. 公開サービス基盤

Webサービス基盤、管理用にVPNで拠点接続

ポイント

- パブリックサブネットは必要最低限に
  - パブリックサブネット
  - プライベートサブネット

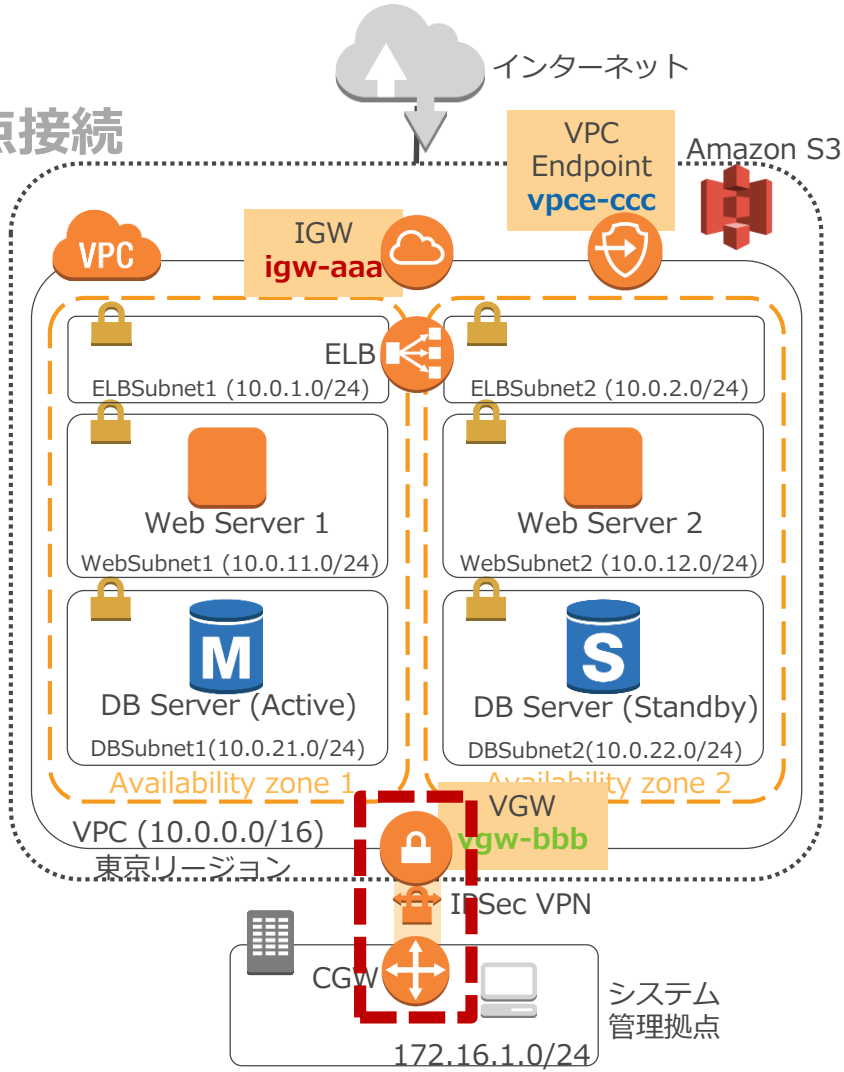


例1. 公開サービス基盤

Webサービス基盤、管理用にVPNで拠点接続

ポイント

- パブリックサブネットは必要最低限に
- 管理拠点とVPN接続

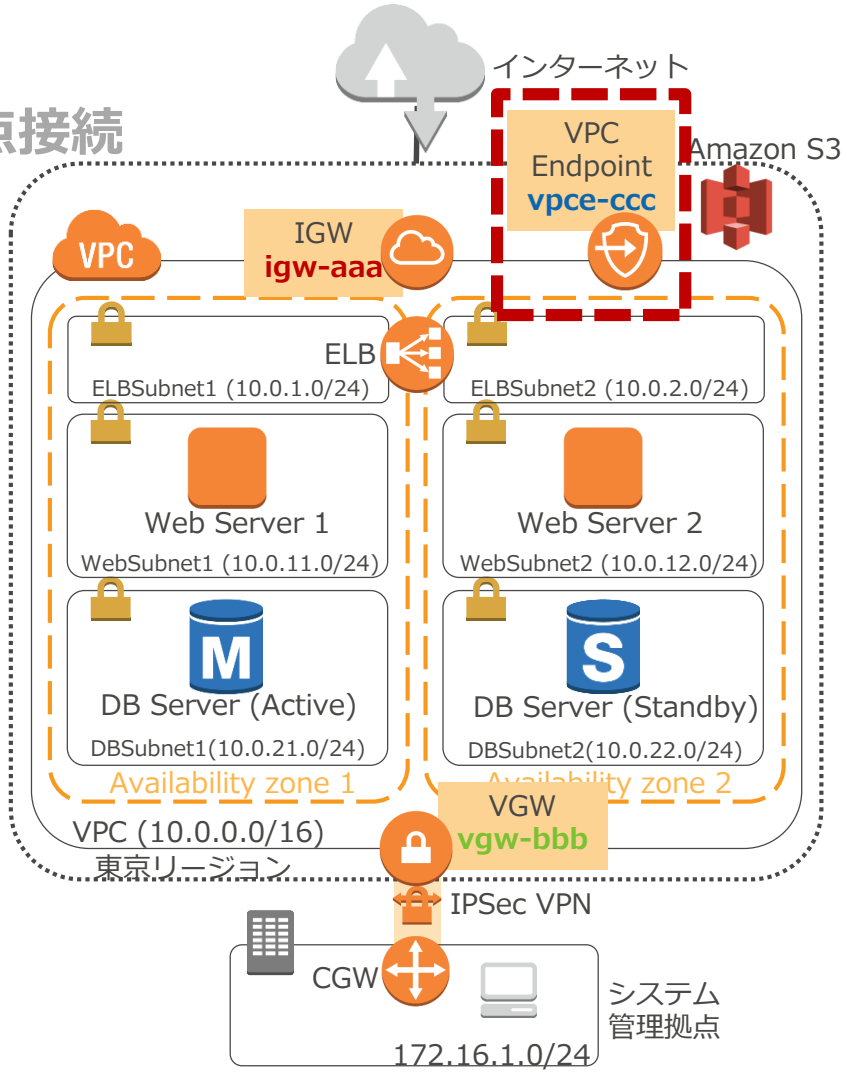


例1. 公開サービス基盤

Webサービス基盤、管理用にVPNで拠点接続

ポイント

- パブリックサブネットは必要最低限に
- 管理拠点とVPN接続
- Webサイトのアセットを S3に保存しているのでVPCエンドポイントを活用



例1. 公開サービス基盤

Webサービス基盤、管理用にVPNで拠点接続

ルートテーブル

172.16.0.0
172.16.1.0
172.16.2.0

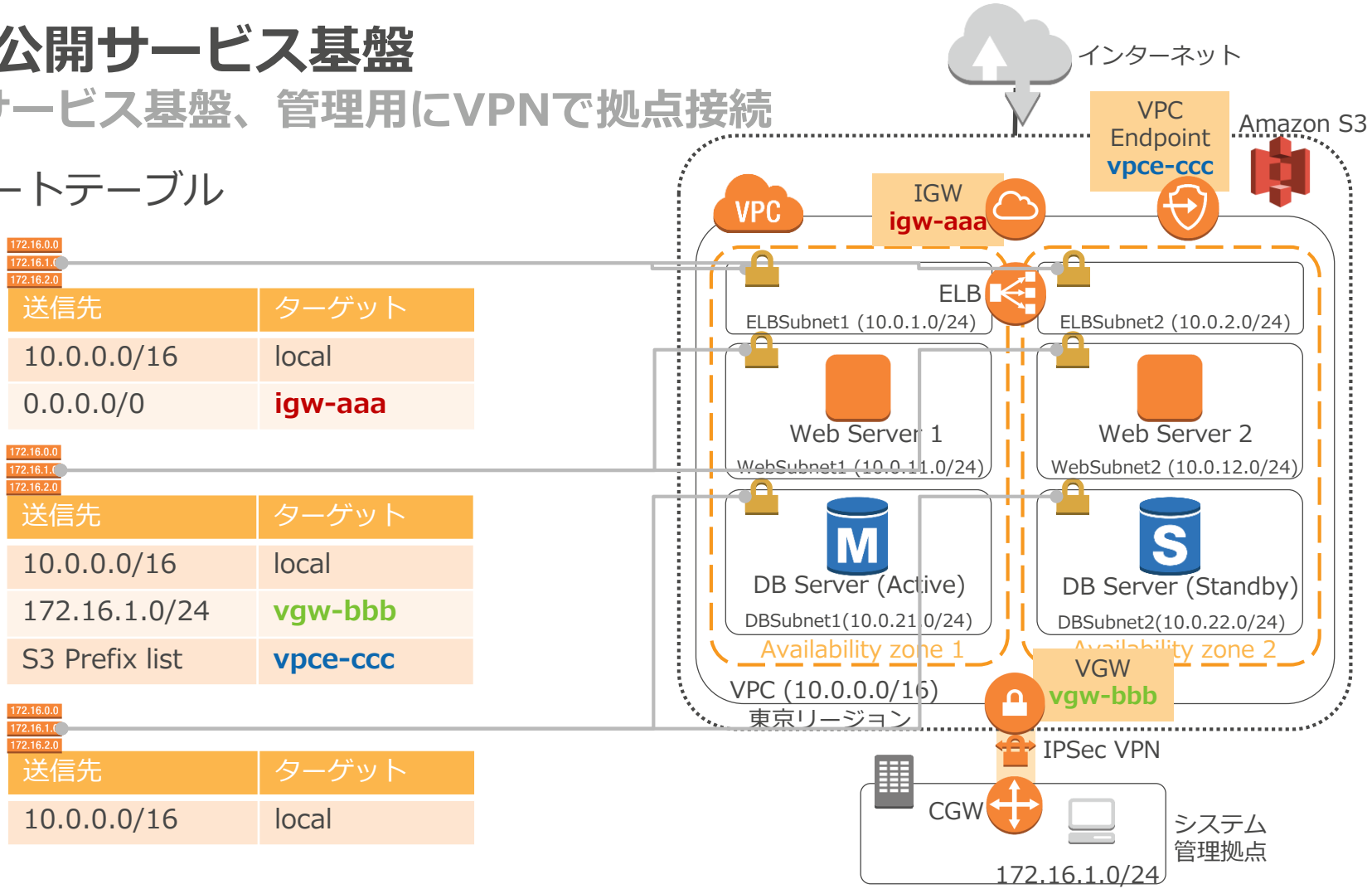
送信先	ターゲット
10.0.0.0/16	local
0.0.0.0/0	igw-aaa

172.16.0.0
172.16.1.0
172.16.2.0

送信先	ターゲット
10.0.0.0/16	local
172.16.1.0/24	vgw-bbb
S3 Prefix list	vpce-ccc

172.16.0.0
172.16.1.0
172.16.2.0

送信先	ターゲット
10.0.0.0/16	local



例1. 公開サービス基盤

Webサービス基盤、管理[illegible]点接続

通信フロー

- ユーザーアクセス
- 管理者アクセス
- EC2からS3へ

ルートテーブル

172.16.0.0
172.16.1.0
172.16.2.0

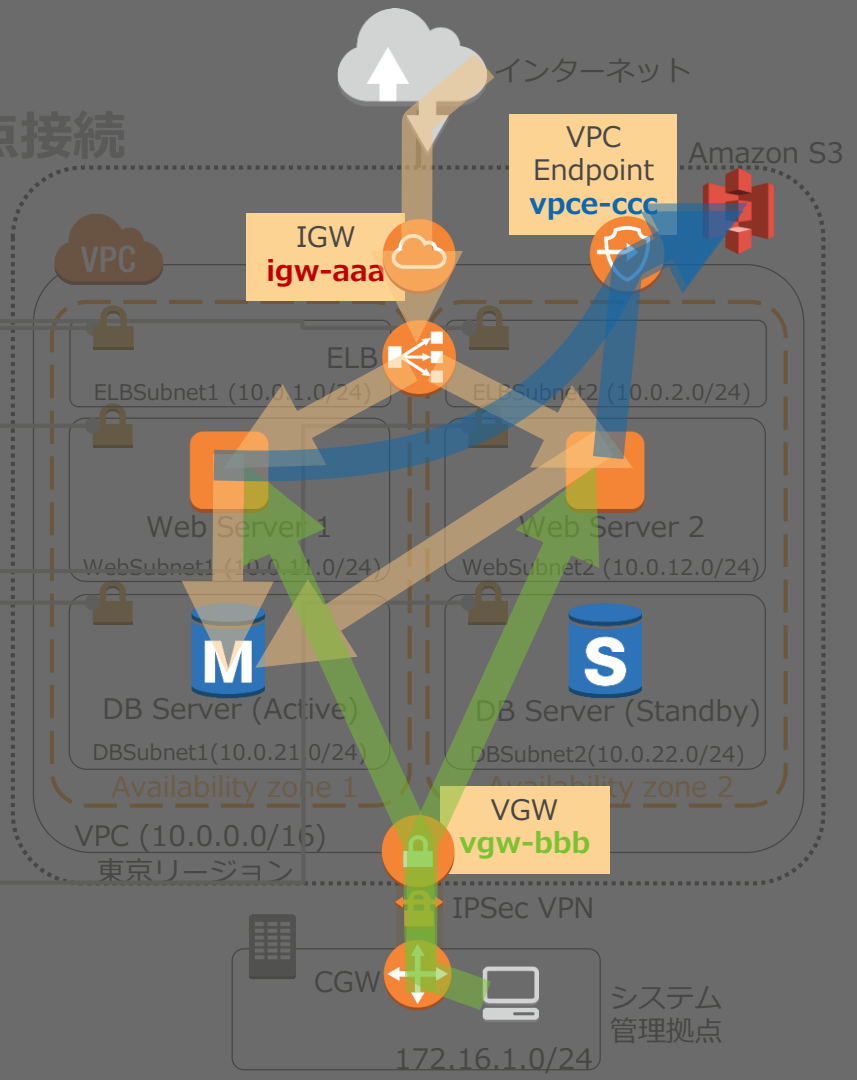
送信先	ターゲット
10.0.0.0/16	local
0.0.0.0/0	igw-aaa

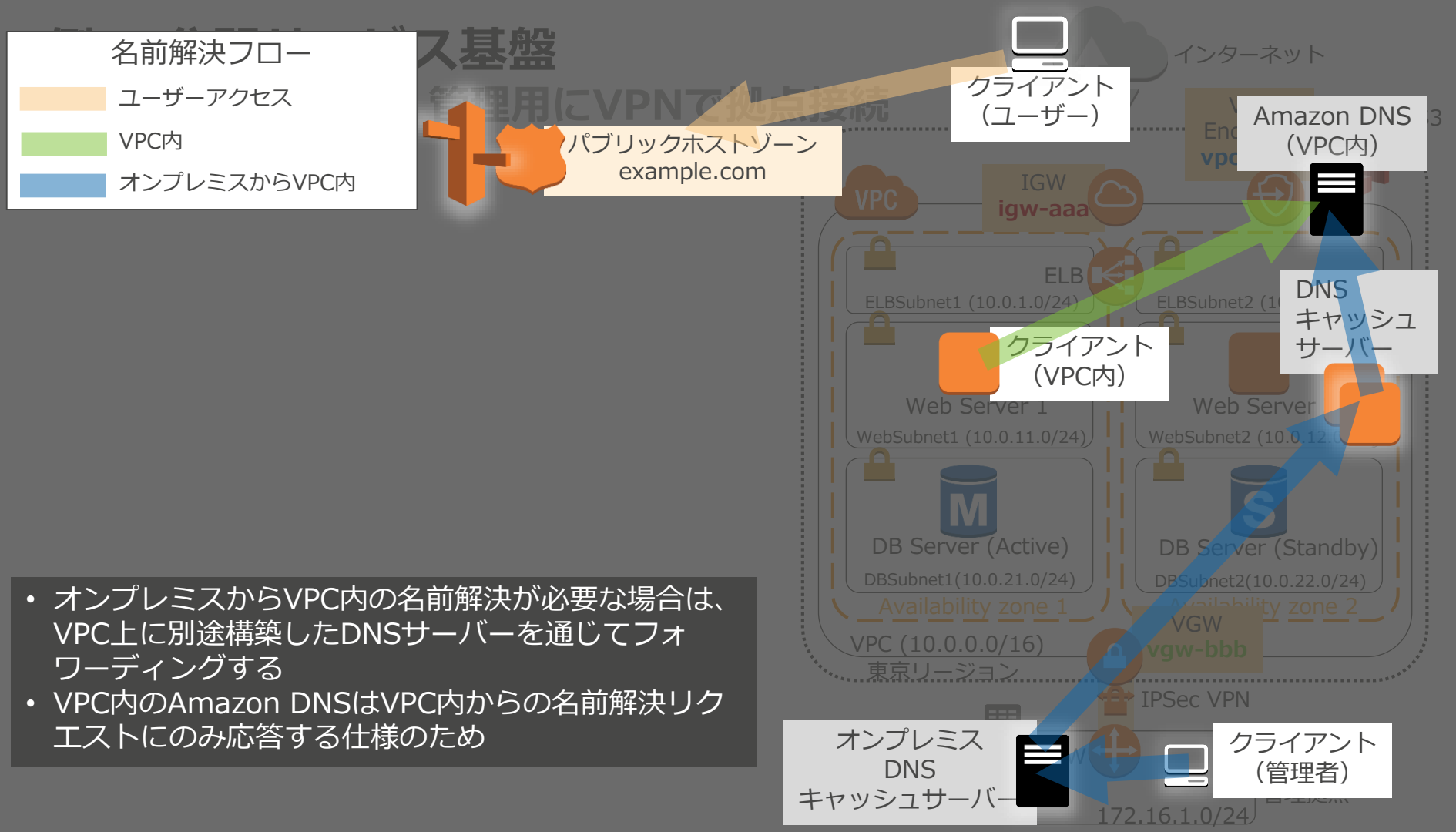
172.16.0.0
172.16.1.0
172.16.2.0

送信先	ターゲット
10.0.0.0/16	local
172.16.1.0/24	vgw-bbb
S3 Prefix list	vpce-ccc

172.16.0.0
172.16.1.0
172.16.2.0

送信先	ターゲット
10.0.0.0/16	local

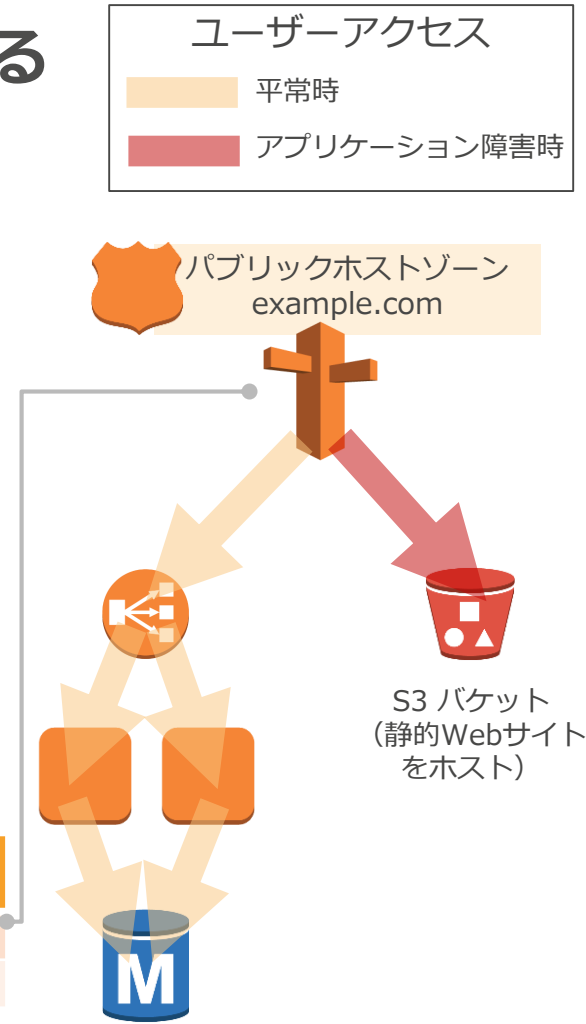




- オンプレミスからVPC内の名前解決が必要な場合は、VPC上に別途構築したDNSサーバーを通じてフォワーディングする
- VPC内のAmazon DNSはVPC内からの名前解決リクエストにのみ応答する仕様のため

Route 53とAWSサービスの連携を活用する

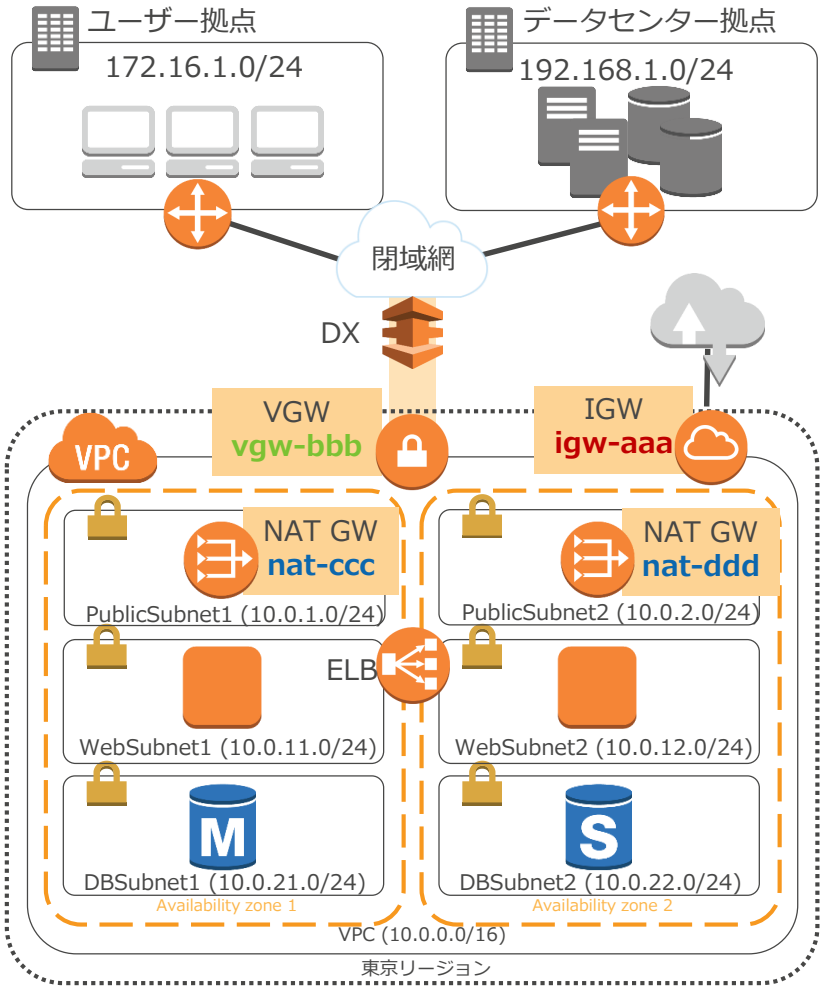
- ALIASレコード
  - AWSのサービスエンドポイントのIPアドレスを直接返答する仮想リソースレコード
  - CNAMEと比較してクエリ回数を削減できレスポンスが高速化
- ELBと連携したDNSフェイルオーバー
  - Route 53のヘルスチェック機能とELBが連携
  - アプリケーションの障害時にSorryページに切り替える場合などに活用可能
  - S3の静的Webサイトホスティング機能との組み合わせも有効



DNS名	タイプ	値	ルーティングポリシー	フェイルオーバーレコードタイプ
www.example.com	A	Alias **<ELBのDNS名>**	Failover	Primary
www.example.com	A	Alias **<S3静的WebサイトのDNS名>**	Failover	Secondary

例2. 社内システム基盤

オンプレミスから移行しハイブリッドで運用

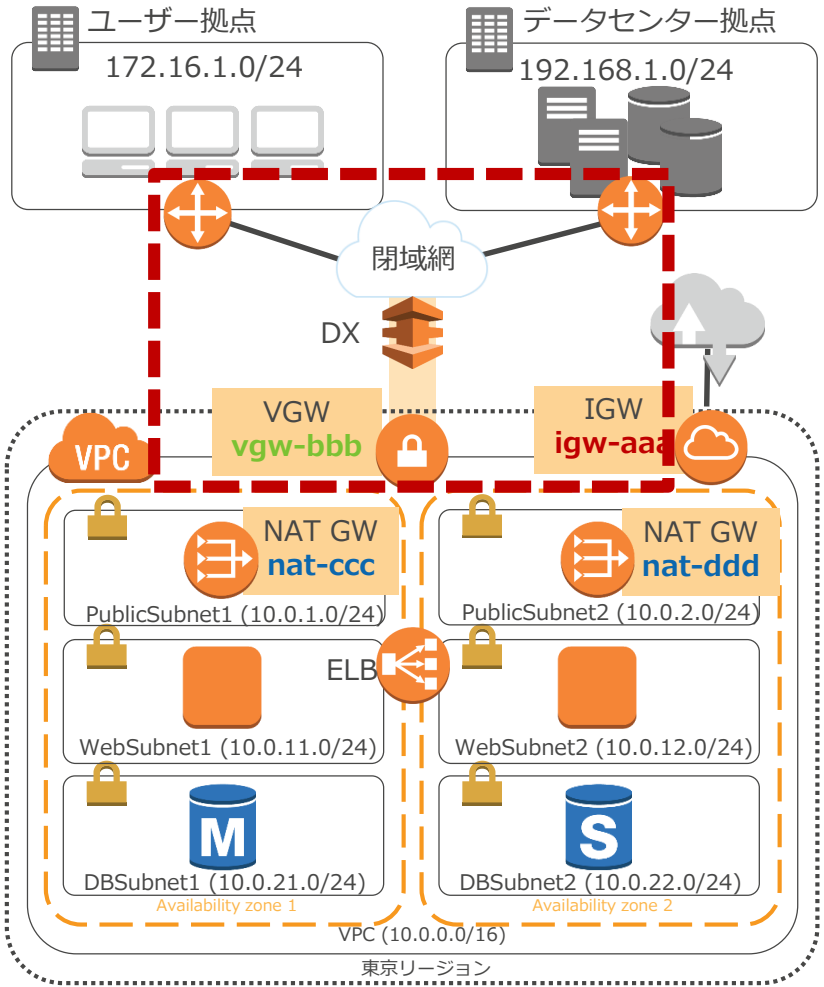


例2. 社内システム基盤

オンプレミスから移行しハイブリッドで運用

ポイント

- DXパートナー様のサービスにより閉域網とAWSリージョンを専用線接続

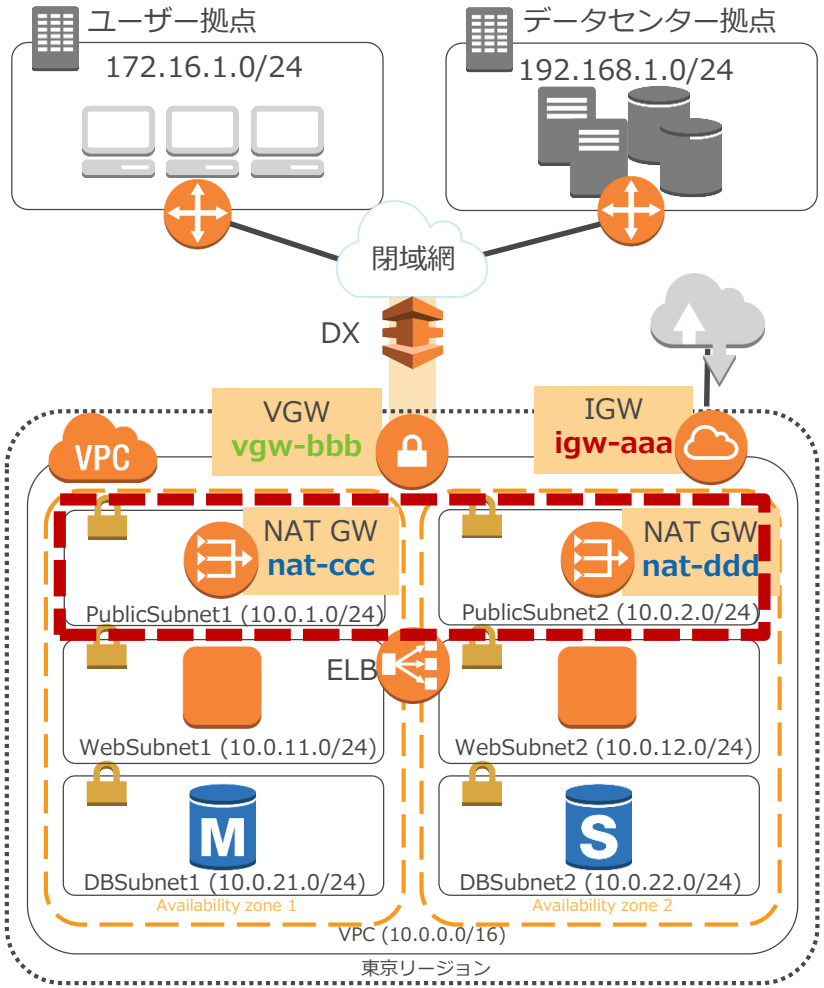


例2. 社内システム基盤

オンプレミスから移行しハイブリッドで運用

ポイント

- DXパートナー様のサービスにより閉域網とAWSリージョンを専用線接続
- プライベートサブネットのサーバーがインターネットに接続するためにNATゲートウェイを利用



例2. 社内システム基盤

オンプレミスから移行しハイブリッドで運用

ルートテーブル

172.16.0.0
172.16.1.0
172.16.2.0

送信先	ターゲット
10.0.0.0/16	local
0.0.0.0/0	igw-aaa

172.16.0.0
172.16.1.0
172.16.2.0

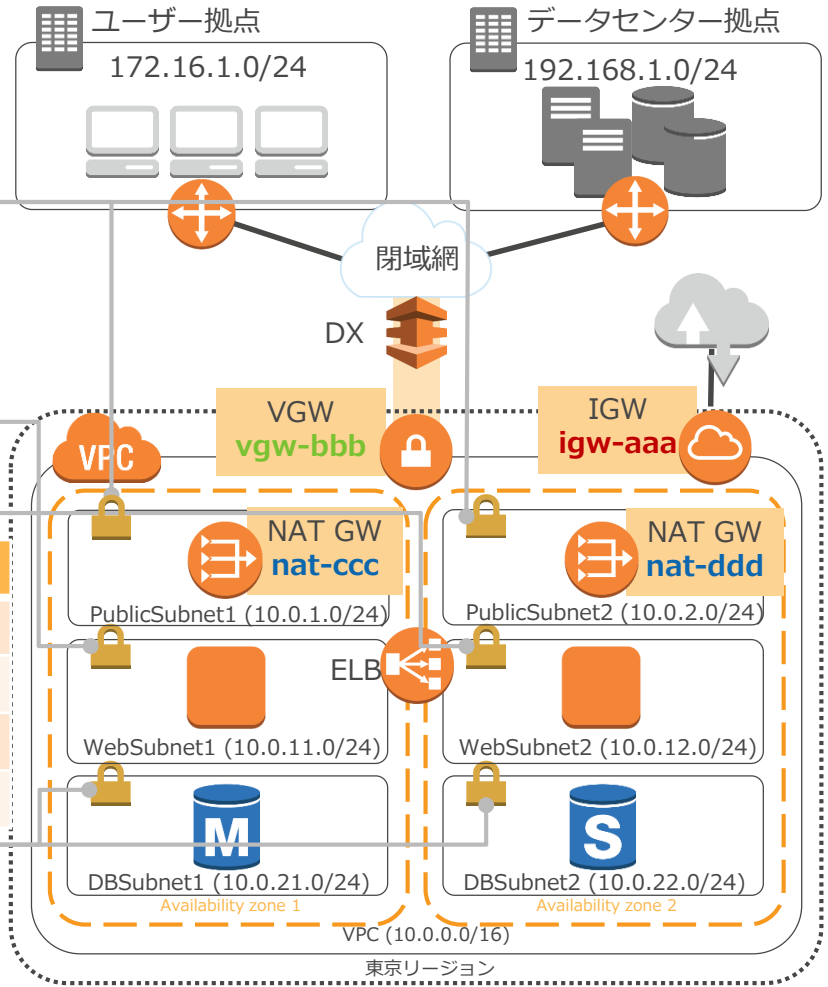
送信先	ターゲット
10.0.0.0/16	local
172.16.1.0/24	vgw-bbb
192.168.1.0/24	vgw-bbb
0.0.0.0/0	nat-ccc

172.16.0.0
172.16.1.0
172.16.2.0

送信先	ターゲット
10.0.0.0/16	local
172.16.1.0/24	vgw-bbb
192.168.1.0/24	vgw-bbb
0.0.0.0/0	nat-ddd

172.16.0.0
172.16.1.0
172.16.2.0

送信先	ターゲット
10.0.0.0/16	local



例2. 社内システ
オンプレミスから移行

ルートテーブル

通信フロー
- ユーザーアクセス
- システム間通信
- EC2からインターネットへ

172.16.0.0
172.16.1.0
172.16.2.0

送信先	ターゲット
10.0.0.0/16	local
0.0.0.0/0	igw-aaa

172.16.0.0
172.16.1.0
172.16.2.0

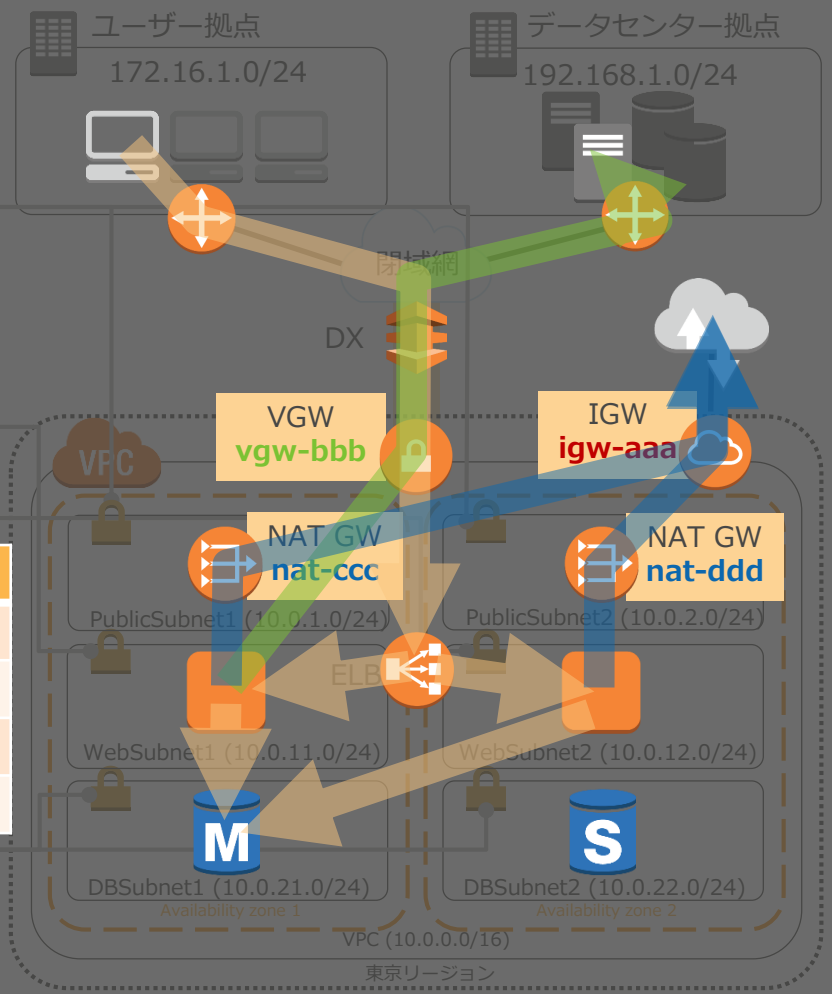
送信先	ターゲット
10.0.0.0/16	local
172.16.1.0/24	vgw-bbb
192.168.1.0/24	vgw-bbb
0.0.0.0/0	nat-ccc

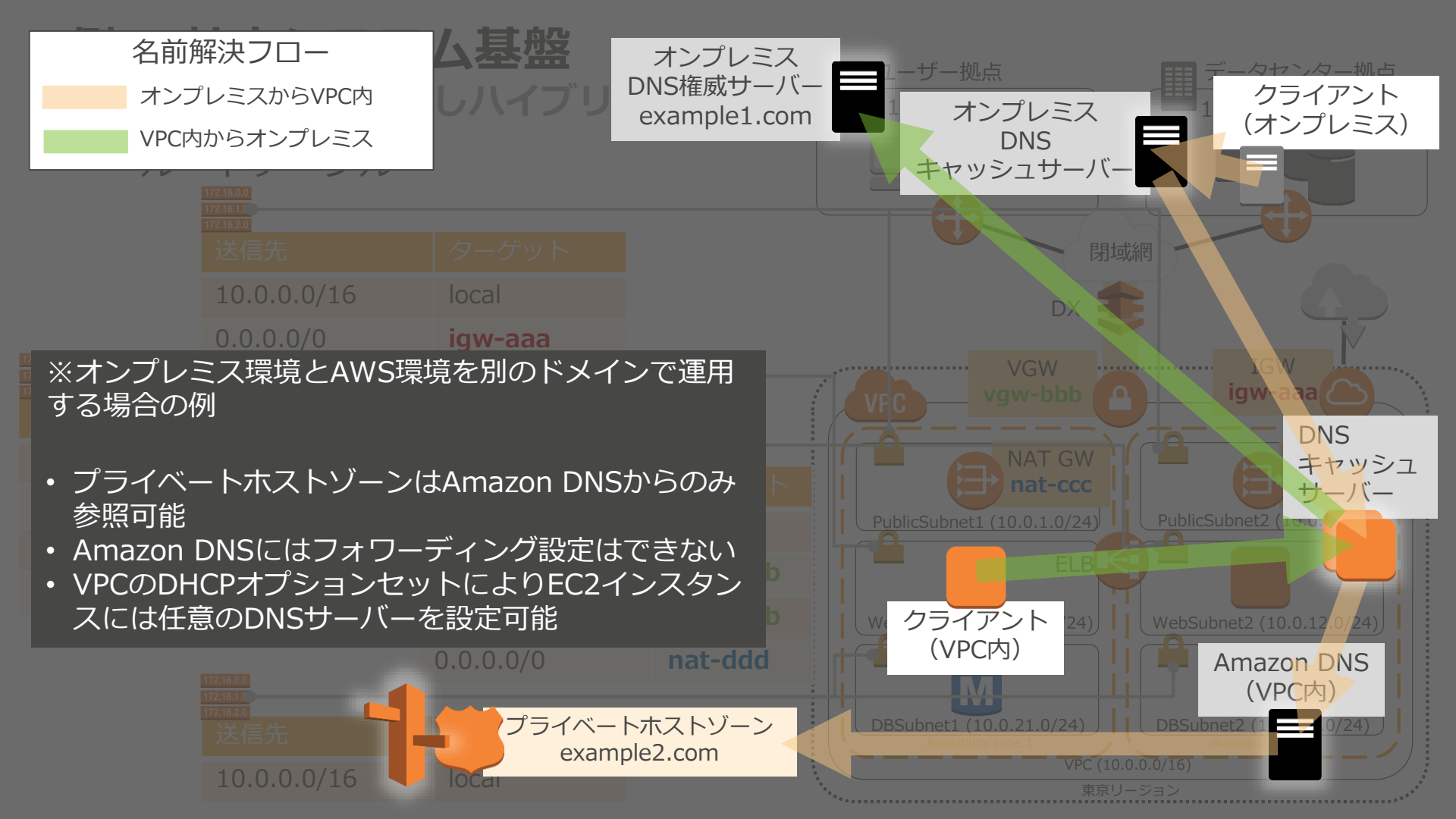
172.16.0.0
172.16.1.0
172.16.2.0

送信先	ターゲット
10.0.0.0/16	local
172.16.1.0/24	vgw-bbb
192.168.1.0/24	vgw-bbb
0.0.0.0/0	nat-ddd

172.16.0.0
172.16.1.0
172.16.2.0

送信先	ターゲット
10.0.0.0/16	local

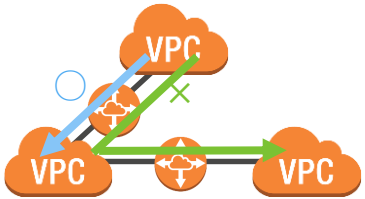


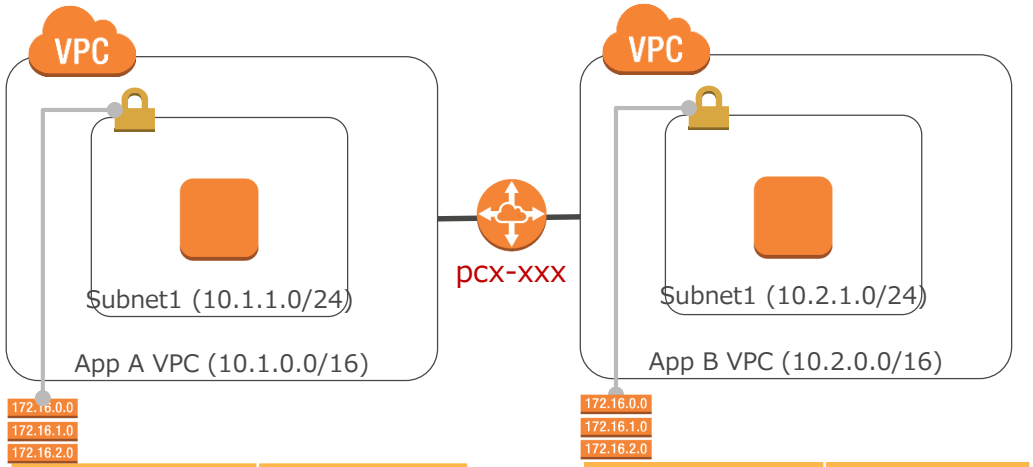

名前解決フロー
オンプレミスからVPC内
VPC内からオンプレミス
オンプレミス
DNS権威サーバー
example1.com
オンプレミス
DNS
キャッシュサーバー
クライアント
（オンプレミス）
閉域網
DX
VGW
vgw-bbb
IGW
igw-aaa
VPC
NAT GW
nat-ccc
DNS
キャッシュ
サーバー
PublicSubnet1 (10.0.1.0/24)
ELB
クライアント
（VPC内）
WebSubnet2 (10.0.12.0/24)
Amazon DNS
（VPC内）
DBSubnet1 (10.0.21.0/24)
VPC (10.0.0.0/16)
東京リージョン
プライベートホストゾーン
example2.com
送信先
ターゲット
10.0.0.0/16
local
0.0.0.0/0
igw-aaa
nat-ddd
※オンプレミス環境とAWS環境を別のドメインで運用する場合の例
• プライベートホストゾーンはAmazon DNSからのみ参照可能
• Amazon DNSにはフォワーディング設定はできない
• VPCのDHCPオプションセットによりEC2インスタンスには任意のDNSサーバーを設定可能

更なる活用に向けて

VPCピア接続 (VPC Peering)

- 2つのVPC間でルーティング
- 異なるAWSアカウントのVPCとも接続可能
- 同一リージョン内のみ
- CIDRの重複は不可
- 直接ピア接続しているVPCにのみルーティング





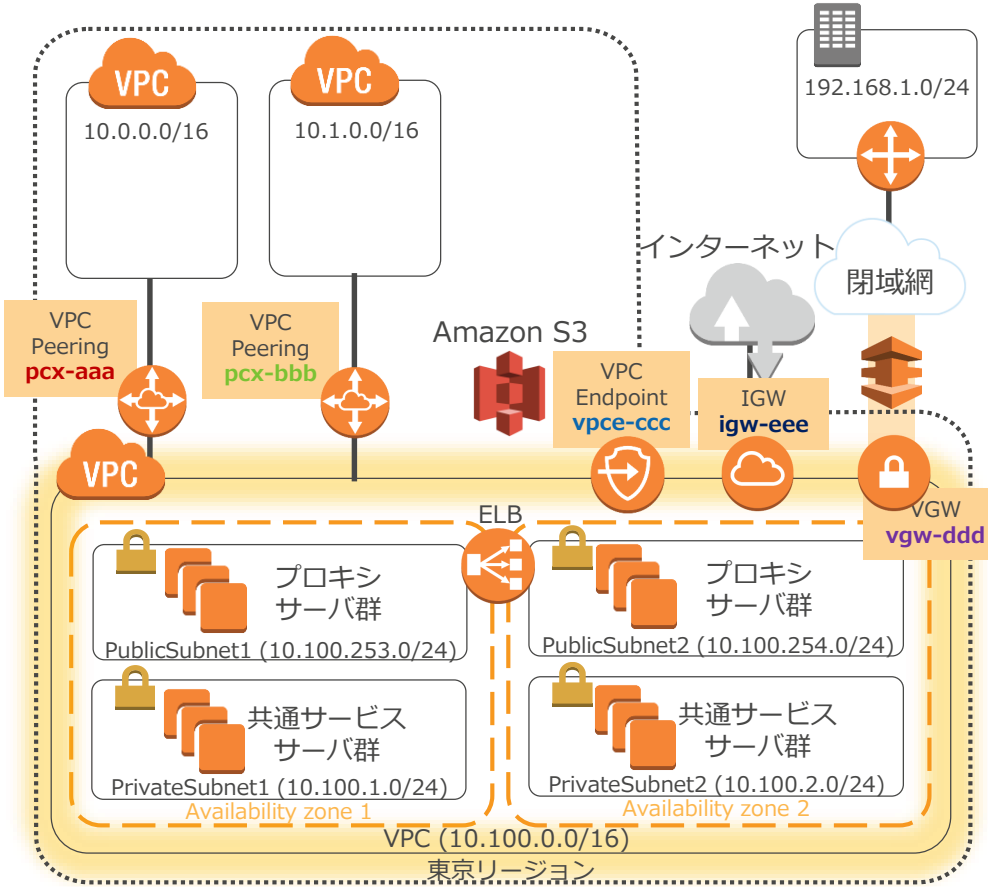
送信先	ターゲット
10.1.0.0/16	local
10.2.0.0/16	pcx-xxx

送信先	ターゲット
10.2.0.0/16	local
10.1.0.0/16	pcx-xxx

共通機能VPCをハブとした大規模AWS環境の構成例

ポイント

- VPCのハブアンドスポーク構成
- ハブVPCに共通機能やVPCコンポーネントを集約
- **ハブVPCのプロキシサーバによりスポークVPCと拠点/インターネットとを通信可能とする**



共通機能VPCをハブとした大規模AWS環境の構成例

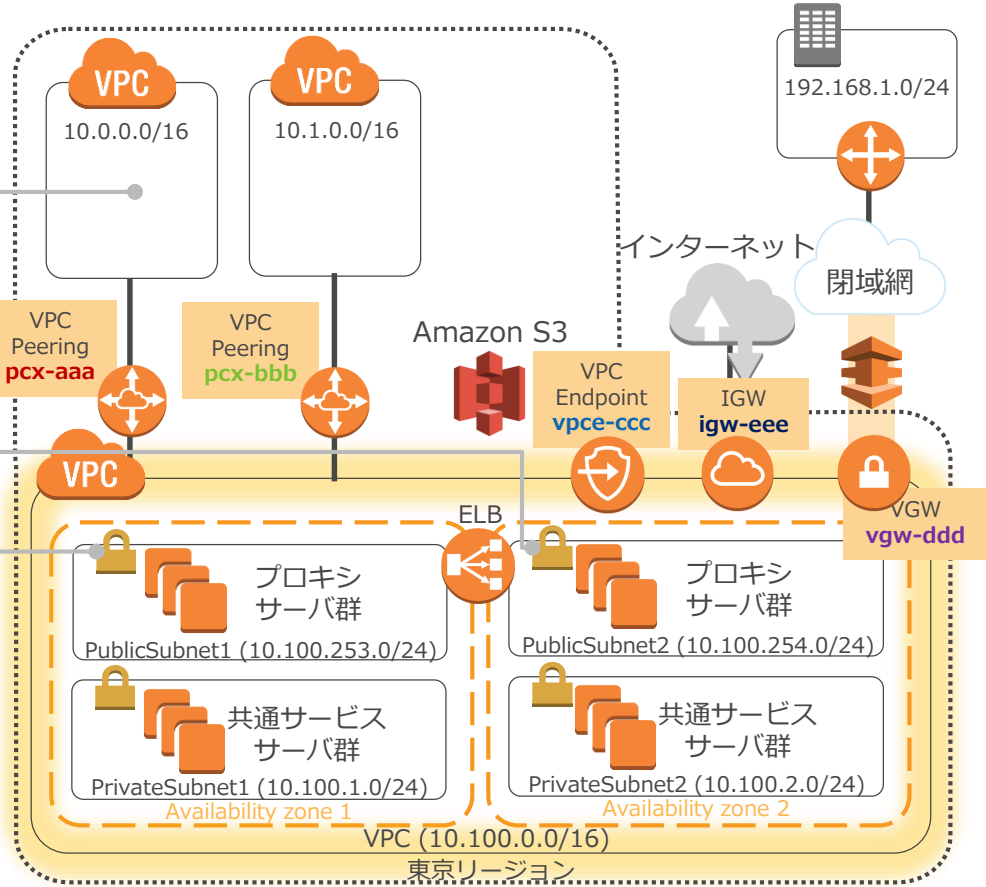
ルートテーブル

172.16.0.0
172.16.1.0
172.16.2.0

送信先	ターゲット
10.0.0.0/16	local
0.0.0.0/0	**pcx-aaa**

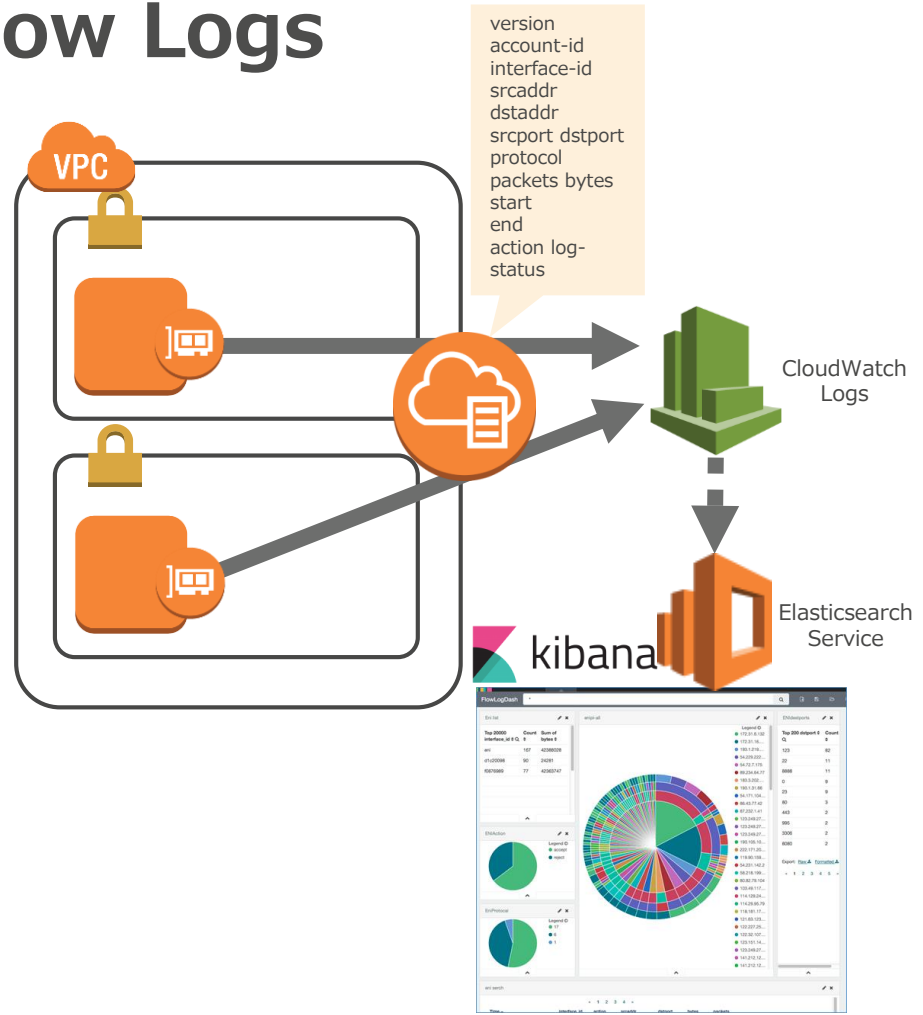
172.16.0.0
172.16.1.0
172.16.2.0

送信先	ターゲット
10.100.0.0/16	local
10.0.0.0/16	**pcx-aaa**
10.1.0.0/16	**pcx-bbb**
S3 prefix list	**vpce-ccc**
192.168.1.0/24	**vgw-ddd**
0.0.0.0/0	**igw-eee**



通信内容の可視化: VPC Flow Logs

- ネットワークトラフィックをキャプチャし、CloudWatch Logsへ Publish
- セキュリティグループとネットワークACLのルールで accepted/rejectされたトラフィックログを取得
- Elasticsearch Service上のKibanaなどでグラフィカルな表示、分析も可能



IPv6にも対応済み

	IPv4	IPv6
アドレス体系	32bit	128bit
VPCでの利用	デフォルトで適用	オプトイン （自動適用ではなく任意）
CIDRブロックサイズ	16～28bitで選択 自分で任意のアドレスを設定可能	56bit固定 Amazon保有のprefixから自動で56bit CIDRがアサインされる（選べない）
サブネット ブロックサイズ	16～28bitで選択	64bit固定
パブリックIP/ プライベートIP	それぞれ存在 （NATを介してパブリックIPをプライマリプライベートIPにMAP）	パブリックのみ （プライベートにするにはEgress-only Internet Gatewayを利用）
インスタンスタイプ	全てのインスタンスタイプ	M3、G2を除く全ての現行世代の インスタンスタイプでサポート
アマゾン提供DNS	プライベートIP、Elastic IPに対する それぞれのDNSホスト名を受信	提供されるDNSホスト名はなし
閉域接続	VPN、DirectConnect	DirectConnectのみ

まとめ

- AWSでは、ネットワークの設計、調達、構築、運用の工数を削減し、**やりたいことに集中できる**

- **VPC, Direct Connect, Route 53**を活用するとシステム要件に沿ったネットワーク環境を構築可能

- **まずは1つのシステムを稼働**してみましょう
  - **すぐに始められます！**
    - VPC作成ウィザードで数クリックで作成
    - 無料利用枠の活用（VPC自体はそもそも無料）
      - https://aws.amazon.com/jp/free/

ご参考資料/情報

- サービス毎の詳細説明資料 – VPC, DX, Route 53
  - https://aws.amazon.com/jp/aws-jp-introduction/#networking
  - 機能、ステップバイステップの実機操作解説など
  - 「クラウド　活用資料集」で検索するとトップに表示されます

- AWS 専用線アクセス体験ラボ sponsored by Intel®
  - https://aws.amazon.com/jp/dx_labo/
  - Direct Connectの接続を無料で体験学習できます！

本セッションのFeedbackをお願いします

受付でお配りしたアンケートに本セッションの満足度やご感想などをご記入ください
アンケートをご提出いただきました方には、もれなく素敵なAWSオリジナルグッズをプレゼントさせていただきます

アンケートは受付、パミール3FのEXPO展示会場内にて回収させて頂きます

AWS

SUMMIT

Thank you!

amazon web services