amazon web services

TA-08: Tech Deep Dive by Amazon

ネットワーク Deep Dive

アマゾンデータ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　吉田 英世

amazon web services

■Gold Sponsors

■Global Sponsors

■Silver Sponsors

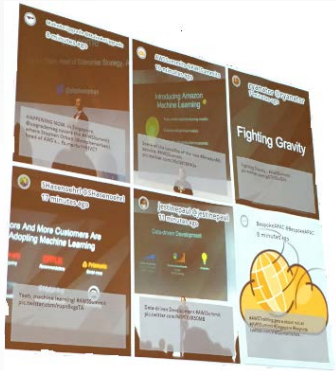
■Bronze Sponsors

■Global Tech Sponsors

■Logo Sponsors

amazon web services

ハッシュタグ #AWSSummit
で、**皆さんのツイート**が展示エリア
の**大画面に表示されます**



公式アカウント@awscloud_jp
を**フォローすると**、ロゴ入り
コースターを**プレゼント**

【コースター配布場所】
メイン展示会場、メイン会場1F受付、デベロッパーカンファレンス会場

amazon web services

自己紹介

吉田 英世

ソリューションアーキテクト
ストラテジックソリューション部（エンタープライズ）

ネットワークSME (Amazon VPC, AWS Direct Connect)

amazon web services

アジェンダ

- VPC ベストプラクティス
- Direct Connect ベストプラクティス
- ネットワークのコード化

amazon web services

AWSのサービス

Technical & Business Support

- Support
- Professional Services
- Partner Ecosystem
- Training & Certification
- Solutions Architects
- Account Management
- Security & Pricing Reports

Enterprise Applications

- Virtual Desktop
- Sharing & Collaboration
- Business Email

Platform Services

Analytics

- Hadoop
- Real-time Streaming Data
- Data Warehouse
- Data Pipelines

App Services

- Queuing & Notifications
- Transcoding
- Workflow
- Email
- App Streaming
- Search

Developer Tools & Operations

- Deployment
- Resource Templates
- DevOps
- Containers
- Application Lifecycle Management
- Event-driven Computing

Mobile Services

- Identity
- Sync
- Mobile Analytics
- Push Notifications

Administration & Security

- Identity Management
- Access Control
- Resource & Usage Auditing
- Key Management & Storage
- Monitoring & Logs

Core Services

- Compute (VMs, Auto-scaling & Load Balancing)
- Storage (Object, Block and Archival)
- CDN
- Databases (Relational, NoSQL, Caching)
- Networking (VPC, DX, DNS)

Infrastructure

- Regions
- Availability Zones
- Points of Presence

http://aws.amazon.com/jp/products/

amazon web services

VPC ベストプラクティス

amazon web services

AWS Marketplaceを活用する

236のネットワーク製品が利用可能！（6/1現在）

ファイアウォール	SOPHOS, paloalto networks., Check Point We Secure the Internet.
ロードバランサ	f5, A10, CITRIX
WAF (Web Application Firewall)	IMPERVA, Barracuda
ルータ	CISCO, BROCADE
ネットワーク高速化	riverbed, SKEED

ネットワークの製品
https://aws.amazon.com/marketplace/b/2649366011/ref=gtw_navlft_node_2649366011?page=1&category=2649366011

amazon web services

MarketPlace製品の利用

1. Choose AMI | 2. Choose Instance Type | 3. Configure Instance | 4. Add Storage | 5. Tag Instance | 6. Configure Security Group | 7. Review

Step 1: Choose an Amazon Machine Image (AMI)

Cancel and Exit

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can sel
an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

- Quick Start
- My AMIs
- AWS Marketplace
- Community AMIs

Categories

All Categories

- Software Infrastructure (294)
- Developer Tools (22)
- Business Software (61)

Operating System

Clear Filter

All Windows

- Windows 2008 R2 (7)
- Windows 2012 (1)

network

1 to 25 of 321 Products

SOPHOS

Sophos UTM 9

Select

(8) | 9.310 Previous versions | Sold by Sophos

Starting from $0.00/hr or from $789/yr (up to 18% savings) for software + AWS usage fees

Free Trial

Linux/Unix, Other 9.3 | 64-bit Amazon Machine Image (AMI) | Updated: 4/20/15

A complete security platform for Amazon Web Services (AWS) Secure your AWS EC2 instances and VPCs with Sophos Unified Threat Management (UTM) for AWS. Its your complete ...

More info

OPENVPN

OpenVPN Access Server (HVM)

Select

(22) | 2.0.17 Previous versions | Sold by OpenVPN Technologies, Inc.

Bring Your Own License + AWS usage fees

Free tier eligible

Linux/Unix, Ubuntu 14.04 | 64-bit Amazon Machine Image (AMI) | Updated: 4/27/15

OpenVPN Access Server is a full featured SSL VPN software solution that integrates OpenVPN server capabilities, enterprise management capabilities, simplified OpenVPN ...

More info

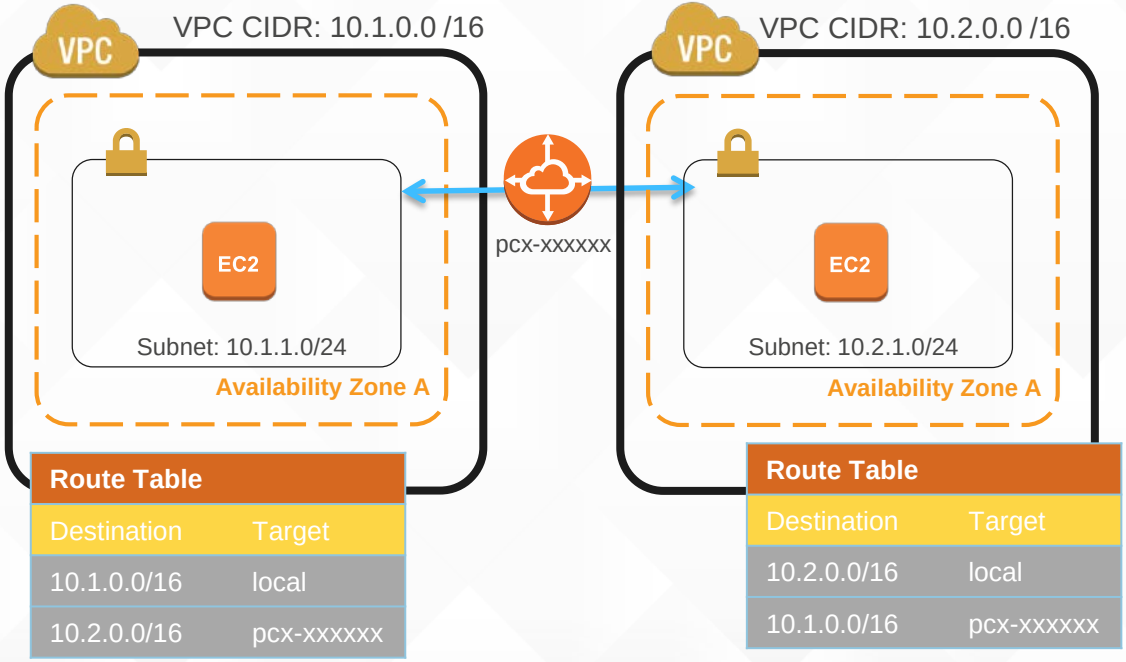
Cisco Cloud Services Router: CSR1000V - Security Technology Package

Select

(2) | 03.14.01.S Previous versions | Sold by Cisco Systems, Inc.

amazon web services

2 VPC Peeringを利用する

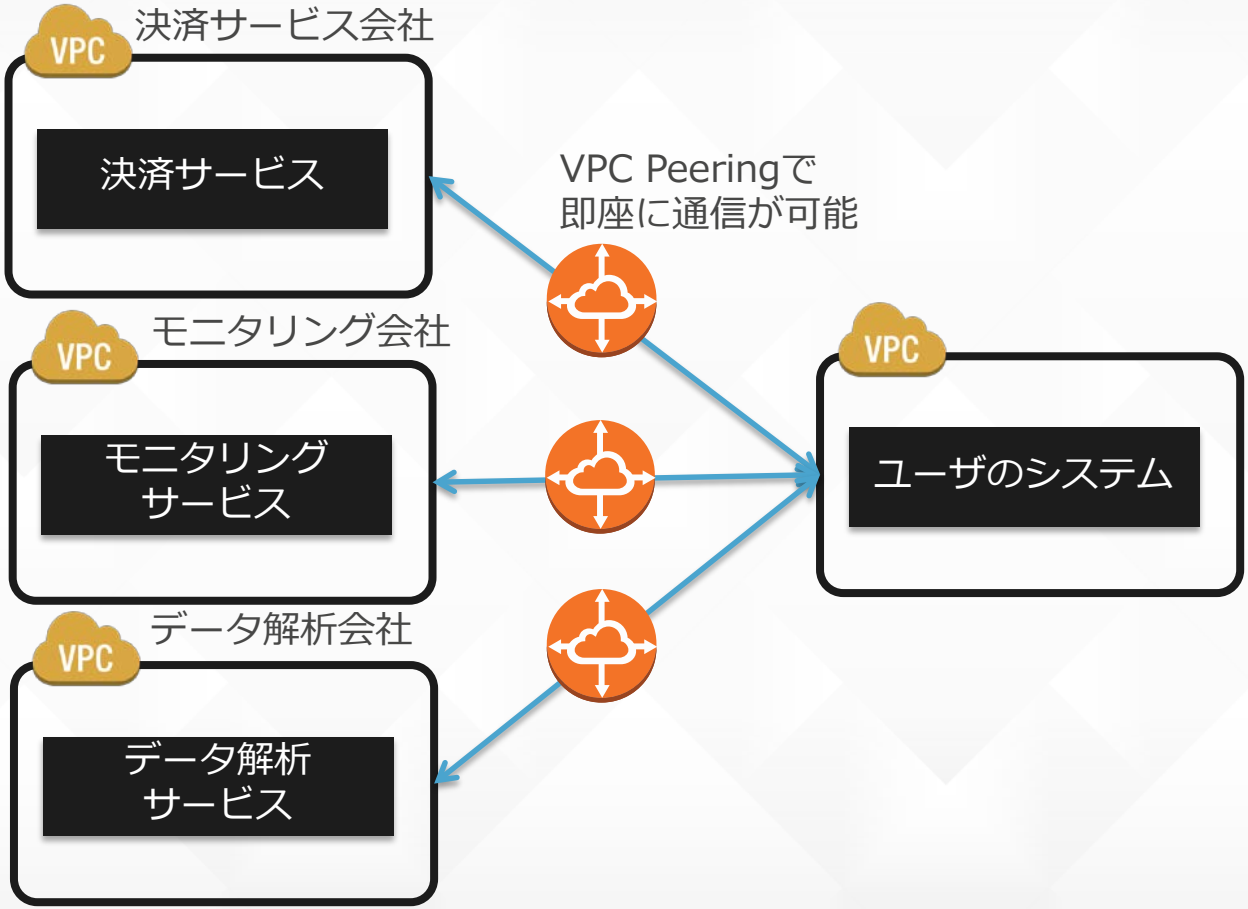


Route Table	
Destination	Target
10.1.0.0/16	local
10.2.0.0/16	pcx-xxxxxx

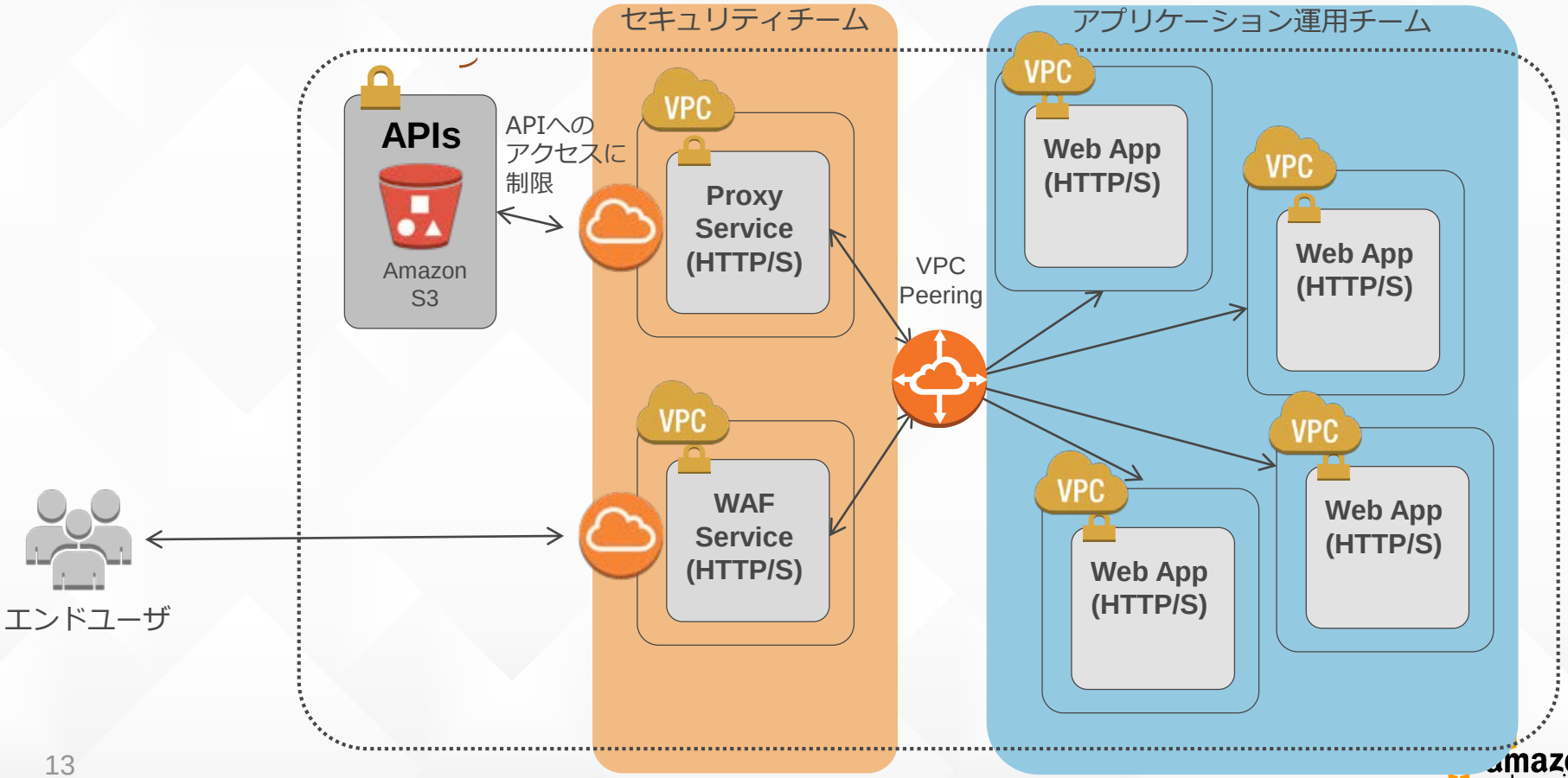
Route Table	
Destination	Target
10.2.0.0/16	local
10.1.0.0/16	pcx-xxxxxx

amazon web services

企業間コラボレーション例



セキュリティ機能の共有例



3 VPCのルーティングとセキュリティ

ルーティング

Route Table

IGW

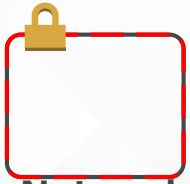
VGW

Customer Gateway

VPC Peering

セキュリティ

Security Group

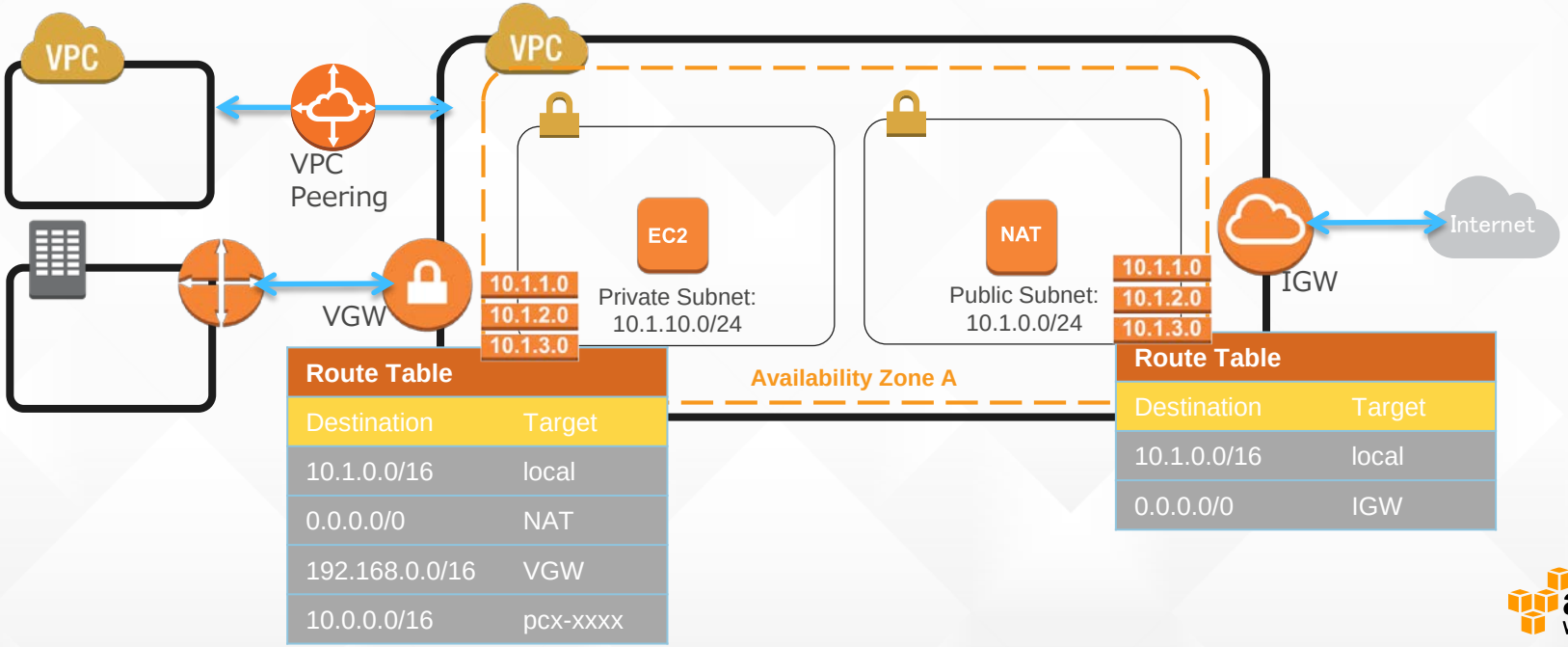


Network ACL

amazon web services

VPCのルーティング

- VPC内のルーティングはすべてルートテーブルに基づく
- IPレベルでの接続性を確保が目的
- どのオブジェクトにトラフィックを転送すればよいかを設定



主なルーティングのエントリ

Route Table	
Destination	Target
10.1.0.0/16	local
0.0.0.0/0	NAT
192.168.0.0/16	VGW
10.100.0.0/16	pcx-xxxx

プライベートサブネットの
ルートテーブル

VPC内のCIDR内のトラフィックは
すべて直接ルーティング

インターネットへのトラフィック（デフォルトルート）は
・プライベートサブネットはNATインスタンス、
・パブリックサブネットはIGWへルーティング

オンプレミスのトラフィックはVGW
へルーティング

他VPCへのトラフィックはPeering Connectionへ
ルーティング

amazon web services

ネットワークACL vs セキュリティグループ

ネットワークACL	セキュリティグループ
サブネットレベルで効果	サーバレベルで効果
Allow/DenyをIN・OUTで指定可能 (**ブラックリスト型**)	AllowのみをIN・OUTで指定可能 (**ホワイトリスト型**)
ステートレスなので、戻りのトラフィックも明示的に許可設定する	ステートフルなので、戻りのトラフィックを考慮しなくてよい
番号の順序通りに適用	全てのルールを適用
サブネット内のすべてのインスタンスがACLの管理下に入る	インスタンス管理者がセキュリティグループを適用すればその管理下になる

ネットワーク構築時に不要な通信を禁止

普段の運用時に必要な通信を許可

amazon web services

ルーティングとセキュリティ

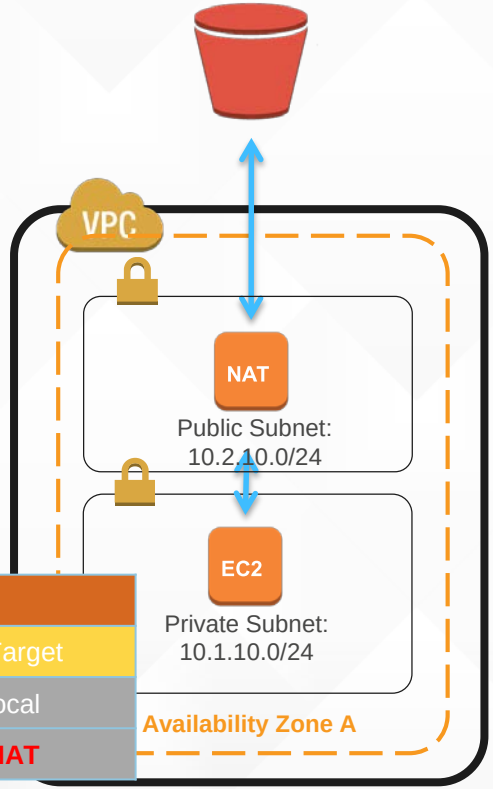
1. ルーティングで疎通性を確保しておく

2. 全体ポリシーで不必要な通信を
ネットワークACLで禁止(構築時)

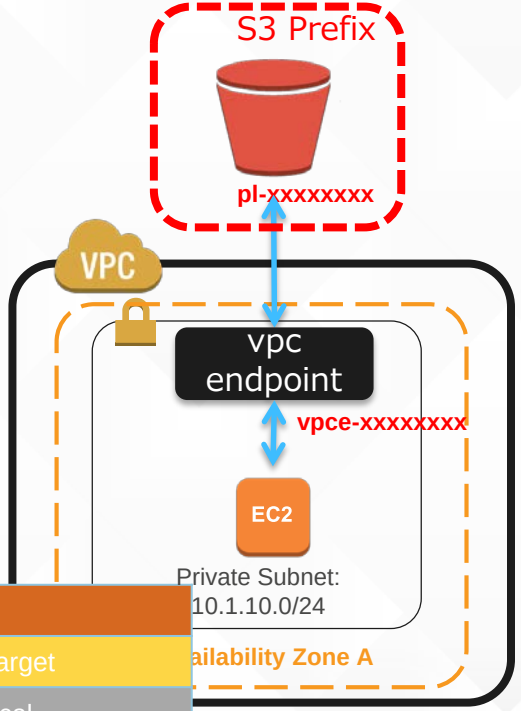
3. 個別に必要な通信を
セキュリティグループで許可(運用時)

amazon web services

4 VPCエンドポイントを利用する



Route Table	
Destination	Target
10.1.0.0/16	local
0.0.0.0/0	NAT



Route Table	
Destination	Target
10.1.0.0/16	local
pl-xxxxxxxx	vpce-xxxxxxxx

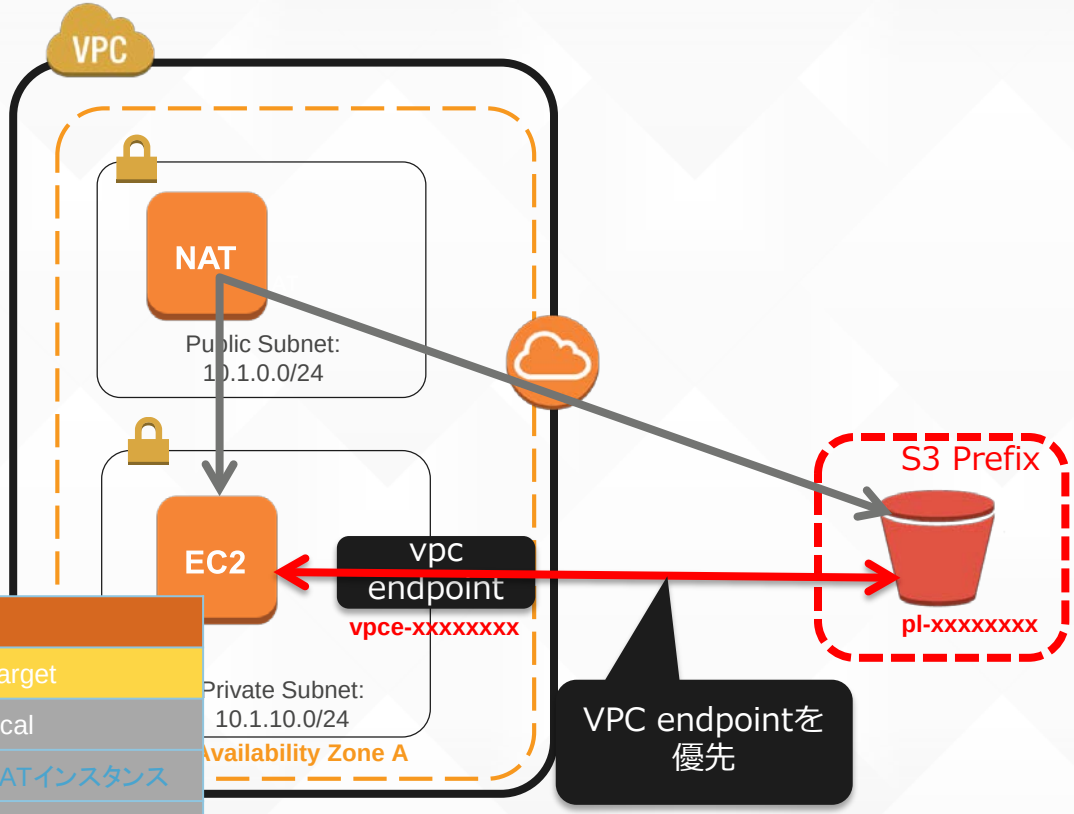
amazon web services

エンドポイントポリシー

```
{
  "Statement": [
    {
      "Sid": "Access-to-specific-bucket-only",
      "Principal": "*",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::my_secure_bucket",
                   "arn:aws:s3:::my_secure_bucket/*"]
    }
  ]
}
```


amazon web services

移行について



Route Table	
Destination	Target
10.1.0.0/16	local
0.0.0.0/0	NATインスタンス
pl-xxxxxxxx	vpce-xxxxxxxx

VPCエンドポイントの注意事項

- リージョンをまたいでAWSクラウドへ通信することはできない

- VPN、AWS Direct Connect、VPC Peering、ClassicLinkの接続先からVPCエンドポイントを利用することはできない

amazon web services

Direct Connect ベストプラクティス

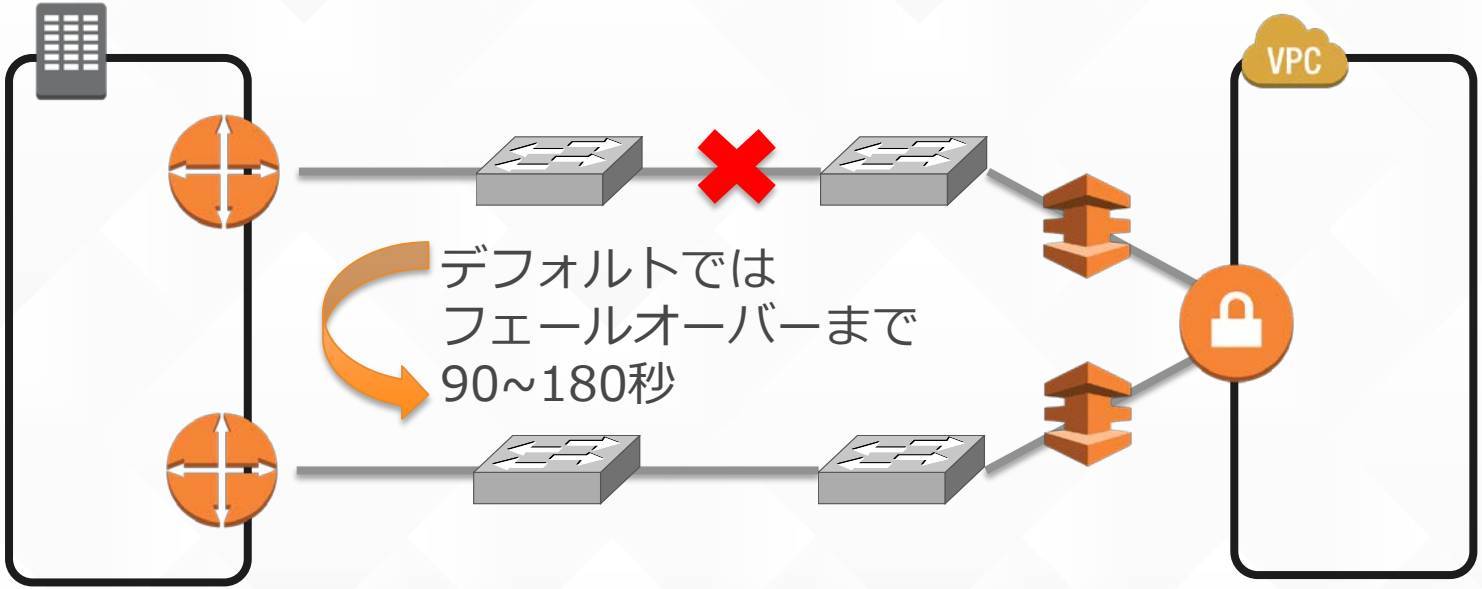
amazon web services

AWSのBGPの動作

- ルートにBGP属性値は付与しない
- お客様ルータのBGP属性値を評価
- ロードシェアリング可能（マルチパスが有効）
- プライベート接続ではVPCのプレフィックス(CIDR)を広告
- パブリック接続ではリージョン内のAWSクラウドのプレフィックスを広告
- BFDは有効

amazon web services

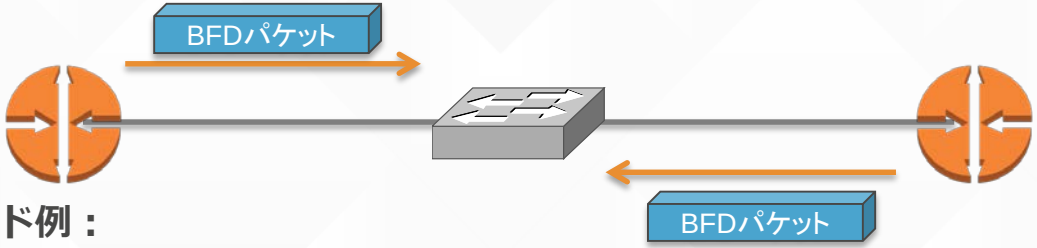
1 回線のフェールオーバーを高速化する



amazon web services

BFD(Bidirectional Forwarding Detection)

- 高速な障害検出を行なう機能(RFC5880)
- ミリsecレベルのBFDパケットの送受信
- ルーティングプロトコル（今回はBGP）へ障害通知
- 音声や映像などを扱っている場合におすすめ
- BFD対応ルータを利用



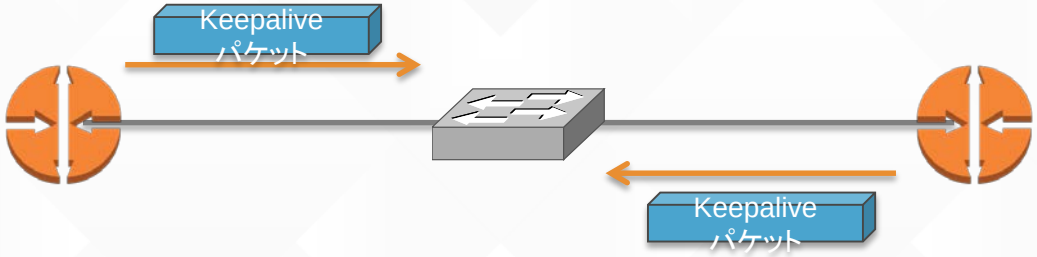
Ciscoのコマンド例：

```
interface GigabitEthernet x/x
  bfd interval <送信ms> min_rx <受信ms> multipllier <受信回数>
router bgp <AS番号>
  neighbor <ネイバーIP> fall-over bfd
```


amazon web services

KeepaliveとHoldtime

- BGPの機能の一部（すべてのBGPルータで利用)
- Keepaliveパケットを指定した間隔で送受信
- KeepaliveがHoldtime時間ないに受信できないと障害と判定
- Holdtimeのデフォルト値は90~180秒が多い
- 値を比較、低い値でネゴシエーションされる



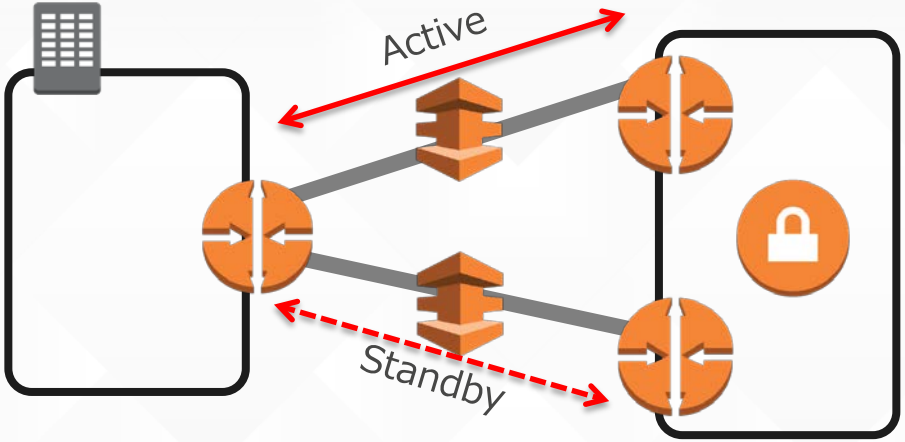
Ciscoのコマンド例：

```
neighbor <ピアIP> timers <Keepalive間隔(秒)> <Hold time(秒)>
```

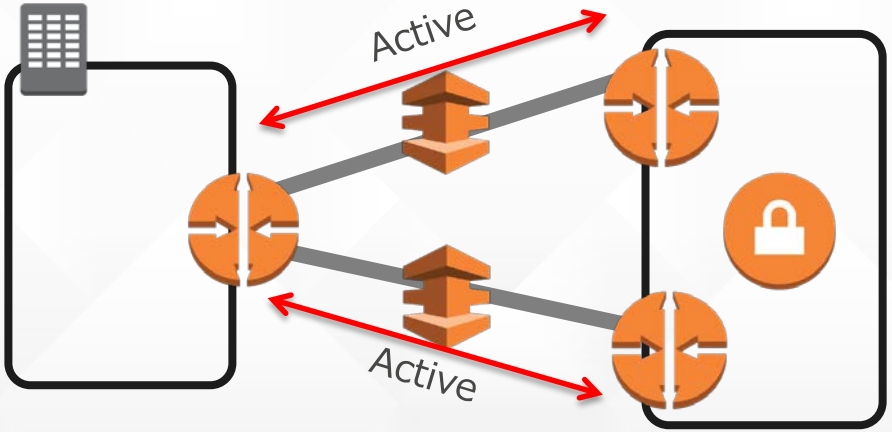

amazon web services

2 Direct Connectのトラフィック設定

Active/Standby

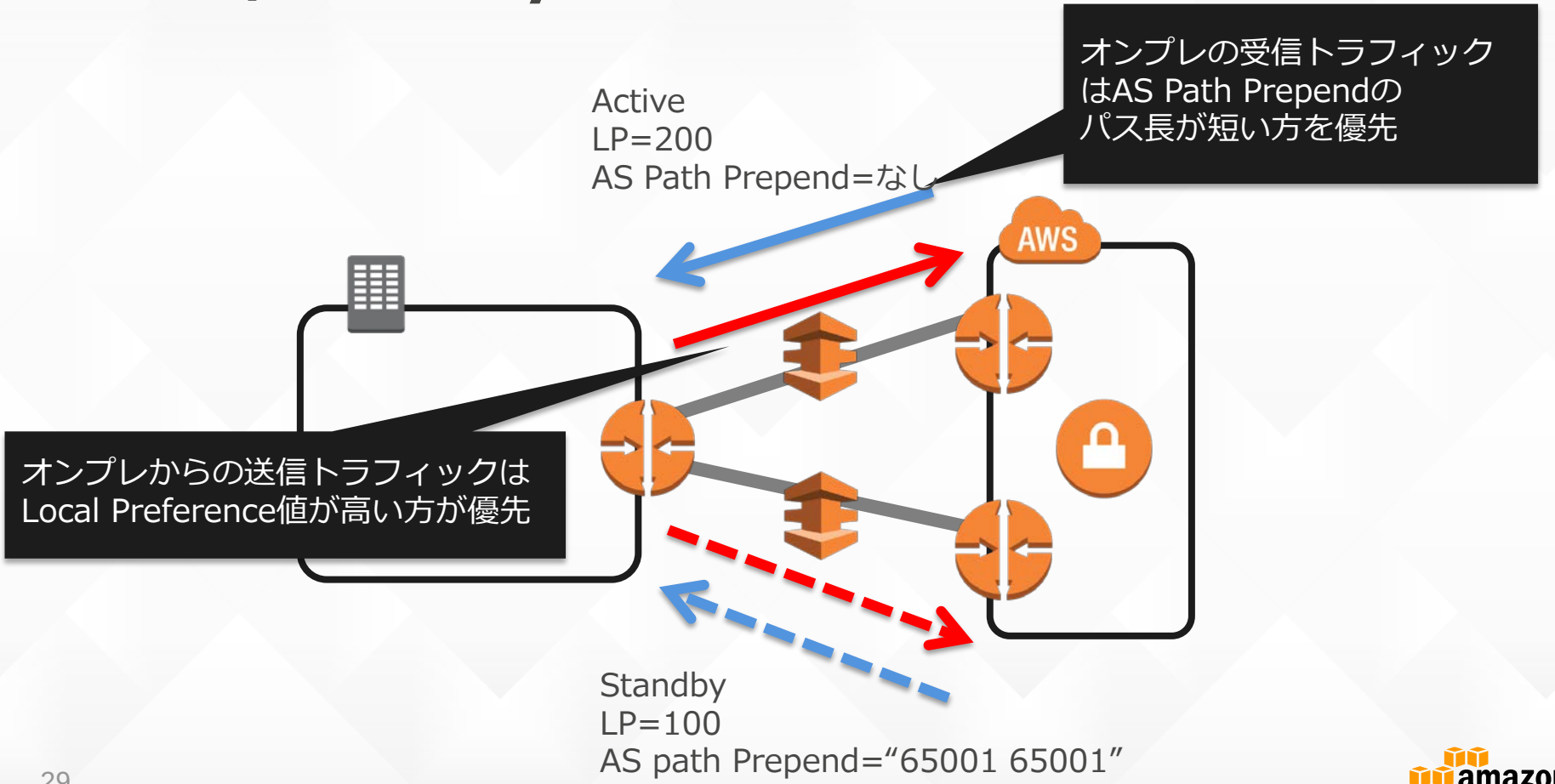


Active/Active



amazon web services

Active/Standbyの構成

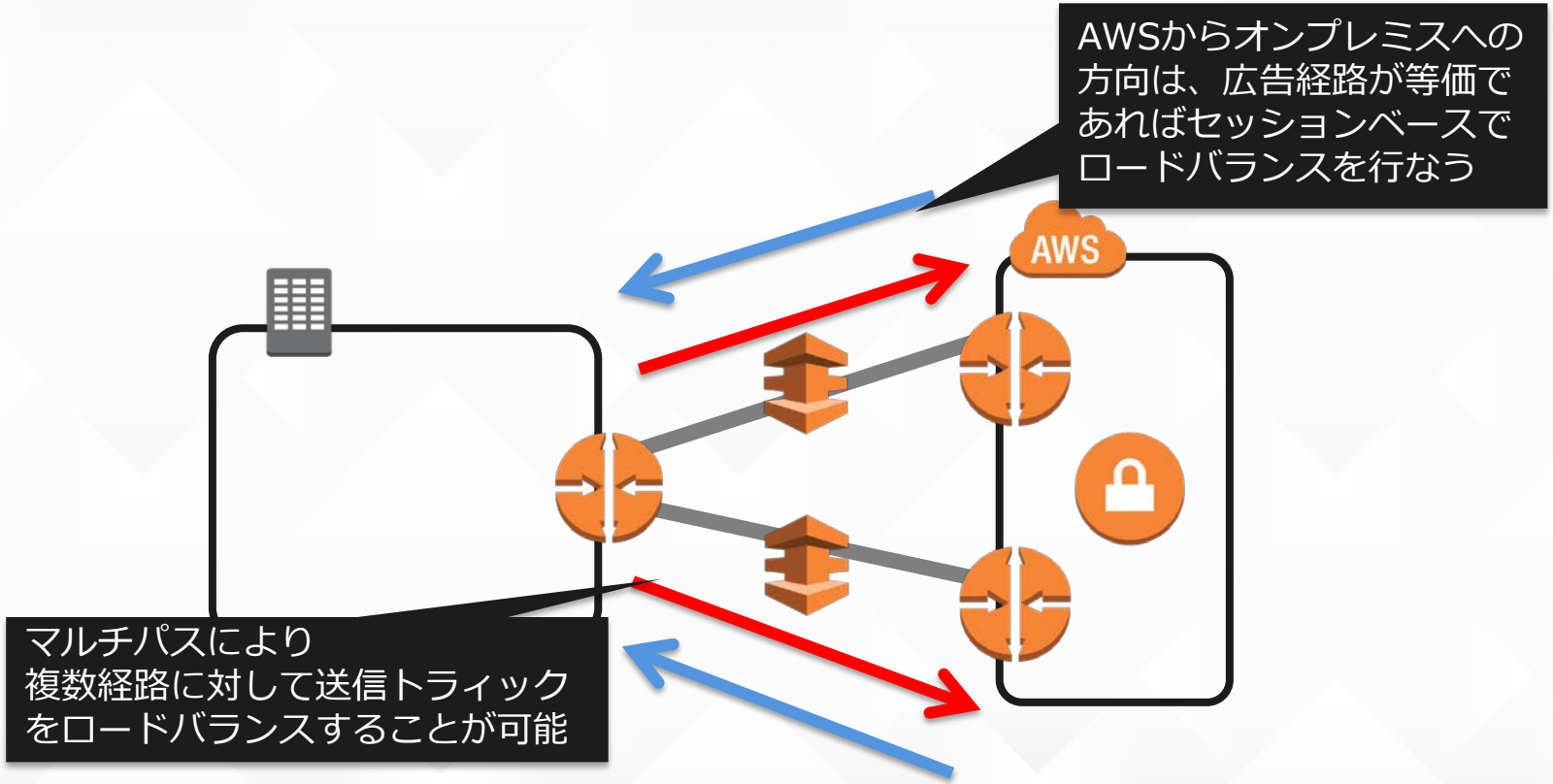


Active/Standby 利用時の注意点

- どちらの回線がActiveなのかをきちんと管理
  - VPNやベストエフォート回線がActiveにならないように
- 上りと下りのトラフィックを意識
  - BGP属性値の設定によっては上りと下りが違う非対称ルートになることもある

amazon web services

Active/Activeの構成



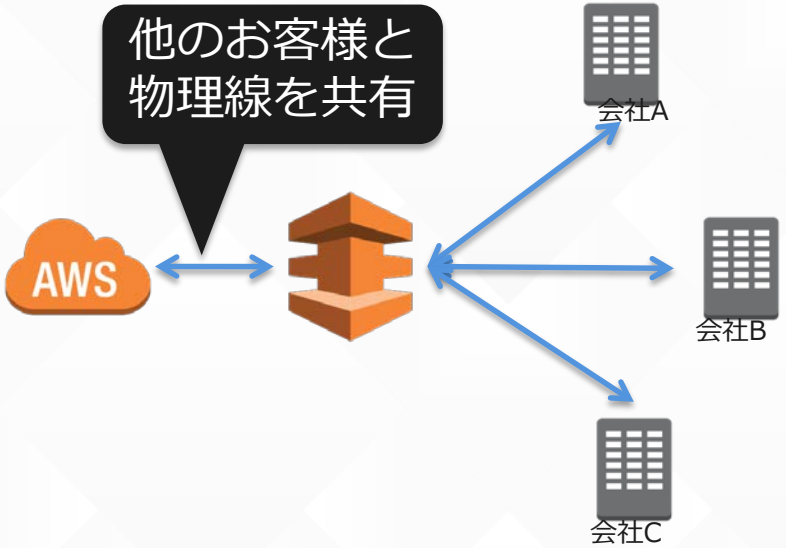
Active/Active 利用時の注意点

- 回線の切断時、正常な回線にトラフィックが移動するため帯域あふれに注意
  - マルチパスのペアをもう一つ準備する
  - Active/Standby構成とする
- トラフィックの偏りに注意
  - きれいにロードバランスされるかはアルゴリズム次第
  - パケットベースの場合は２つの回線の遅延差が懸念

amazon web services

3 占有型と共有型

占有型



共有型

amazon web services

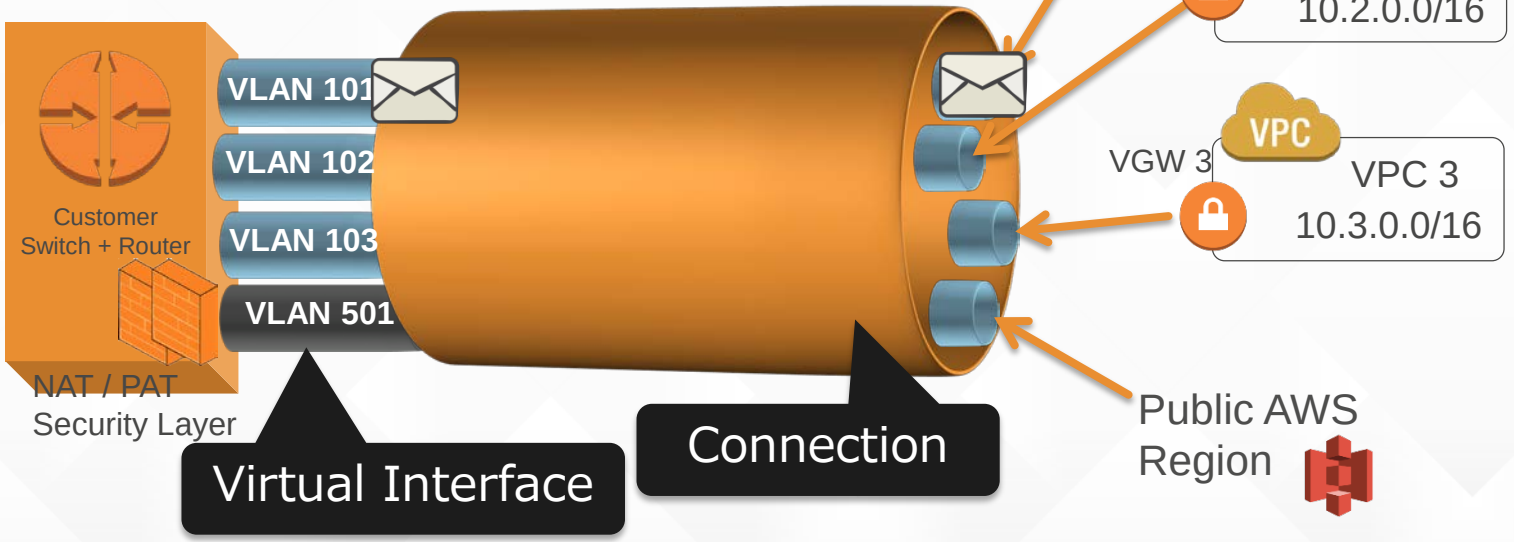
Direct Connectの構成

Customer Interface 0/1.101	
VLAN Tag	101
BGP ASN	65001
BGP Announce	Customer Internal
Interface IP	169.254.251.2/30

Private Virtual Interface 1	
VLAN Tag	101
BGP ASN	10124
BGP Announce	10.1.0.0/16
Interface IP	169.254.251.1/30

Customer Internal Network

Route Table	
Destination	Target
10.1.0.0/16	PVI 1
10.2.0.0/16	PVI 2
10.3.0.0/16	PVI 3
AWS Cloud	PVI 5



amazon web services

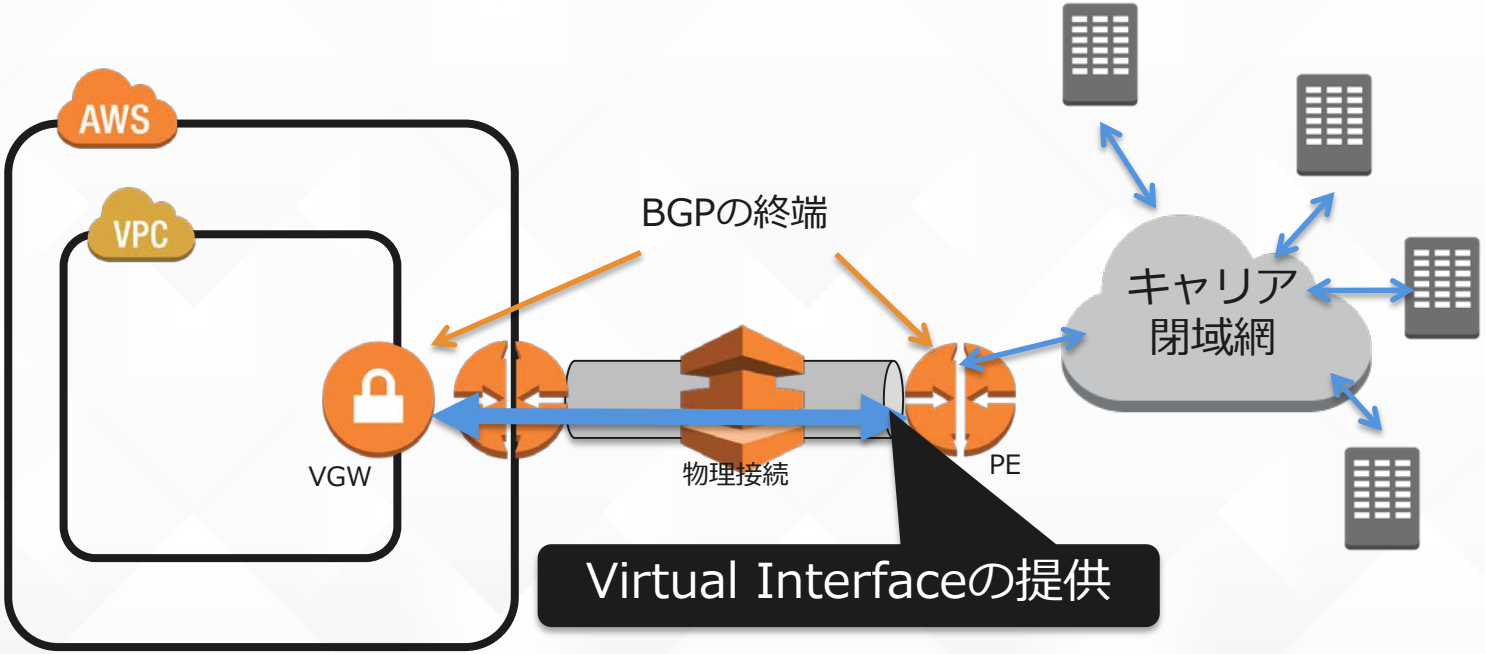
占有型 VS 共有型

	占有型	共有型
ユーザの管理単位	Connections （物理接続単位）	Virtual Interface （論理接続単位）
1契約で接続できるVPC数	複数(VLANの数だけ)	1つ
トポロジ	ポイントツーポイント	ポイントツーマルチポイント IP-VPNなどの閉域網
お客様ルータの管理	自社 （マネージド契約可）	キャリア側
利用できる帯域	1Gbps, 10Gbps	~1Gbps ベストエフォートも有
リードタイム	遅い(数ヶ月)	早い
1契約あたりの単価	高	安

amazon web services

キャリア閉域網の共用型の例

キャリアの閉域網を利用する場合、エンドユーザには１契約あたり
１つのVirtual Interfaceの提供となり、１つのVPCに接続。

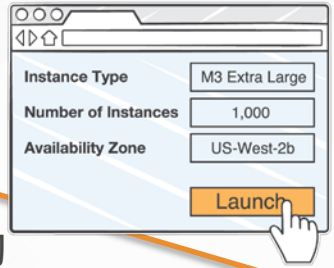


ネットワークのコード化
(Infrastrucute as Code)

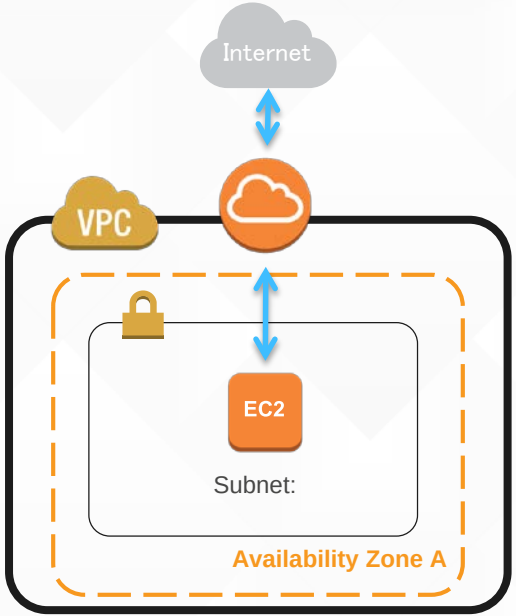
amazon web services

コード化について

手順書



手動
オペレーション



コード

コードの実行

インフラの構成をコードで管理

amazon web services

コード化するツール

CloudFormation

JSONテンプレートを元にAWSリソースの構築を自動化。
レイヤーを分けて定義も可（ex. ネットワークレイヤー）
複製やアップデート、バージョン管理が容易

AWS CLI
AWS SDK
Powershell

AWS CLI/PowershellはAWSの操作をコマンドラインで行なうツール

AWS SDKはいろいろなプログラミング言語に調整されたAPIを利用してAWSの操作を行なう

amazon web services

CloudFormation

```
"Resources" : {
        "VPC" : {
                "Type" : "AWS::EC2::VPC",
                        "Properties" : {
                        "CidrBlock" : “10.0.0.0/16”,
                                "Tags" : [ { "Key" : “Name", "Value" : “VPCName“ } ]
                        }
        },
        "PublicSubnet" : {
                "Type" : "AWS::EC2::Subnet",
                "Properties" : {
                        "VpcId" : { "Ref" : "VPC" },
                "CidrBlock" : “10.0.1.0/24”,
                        "Tags" : [ { "Key" : "Network", "Value" : "Public" } ]
                }
        }
}
```

VPC作成

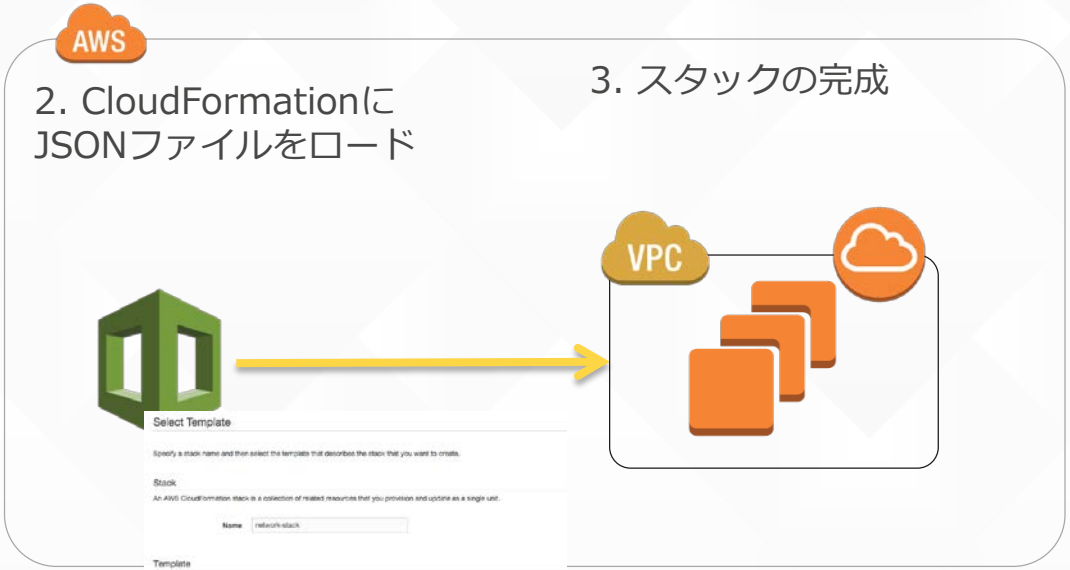
サブネット作成

サンプルコード&テンプレート：
http://aws.amazon.com/jp/cloudformation/aws-cloudformation-templates/

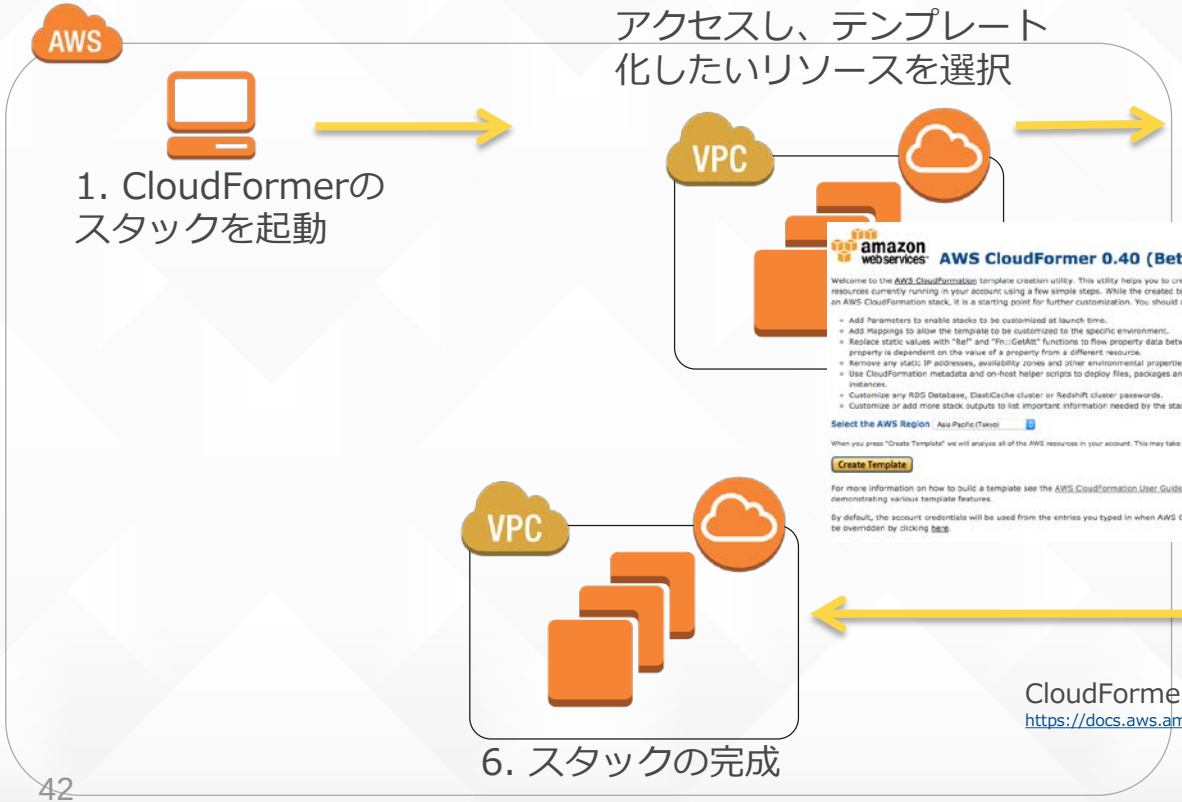
amazon web services

CloudFormationのワークフロー

1. JSONを編集



CloudFormerを利用したテンプレート作成



CloudFormer:
https://docs.aws.amazon.com/ja_jp/AWSCloudFormation/latest/UserGuide/cfn-using-cloudformer.html

AWS CLI (VPCとサブネット作成・削除)

```
#!/bin/sh
export AWS_DEFAULT_REGION=“ap-northeast-1"

VPC_ID=`aws ec2 create-vpc --cidr-block 10.0.0.0/16 --output text | awk '{print $6;}'`
SUBNET_ID=`aws ec2 create-subnet --vpc-id $VPC_ID --cidr-block 10.0.1.0/24 --output text | awk '{print $6;}'`

echo "Created $VPC_ID & $SUBNET_ID"

#Clean up
aws ec2 delete-subnet --subnet-id $SUBNET_ID
aws ec2 delete-vpc --vpc-id $VPC_ID
```


amazon web services

AWS CLI (動的な処理)

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

aws ec2 replace-route --route-table-id $ROUTE_TABLE_ID
                      --destination-cidr-block 0.0.0.0/0
                      --instance-id $INSTANCE_ID

aws ec2 attach-network-interface  --network-interface-id $ENI
                                  --instance-id $INSTANCE_ID
                                  --device-index 1

aws ec2 assign-private-ip-addresses --network-interface-id $ENI
                                    --private-ip-addresses 10.0.0.100
```

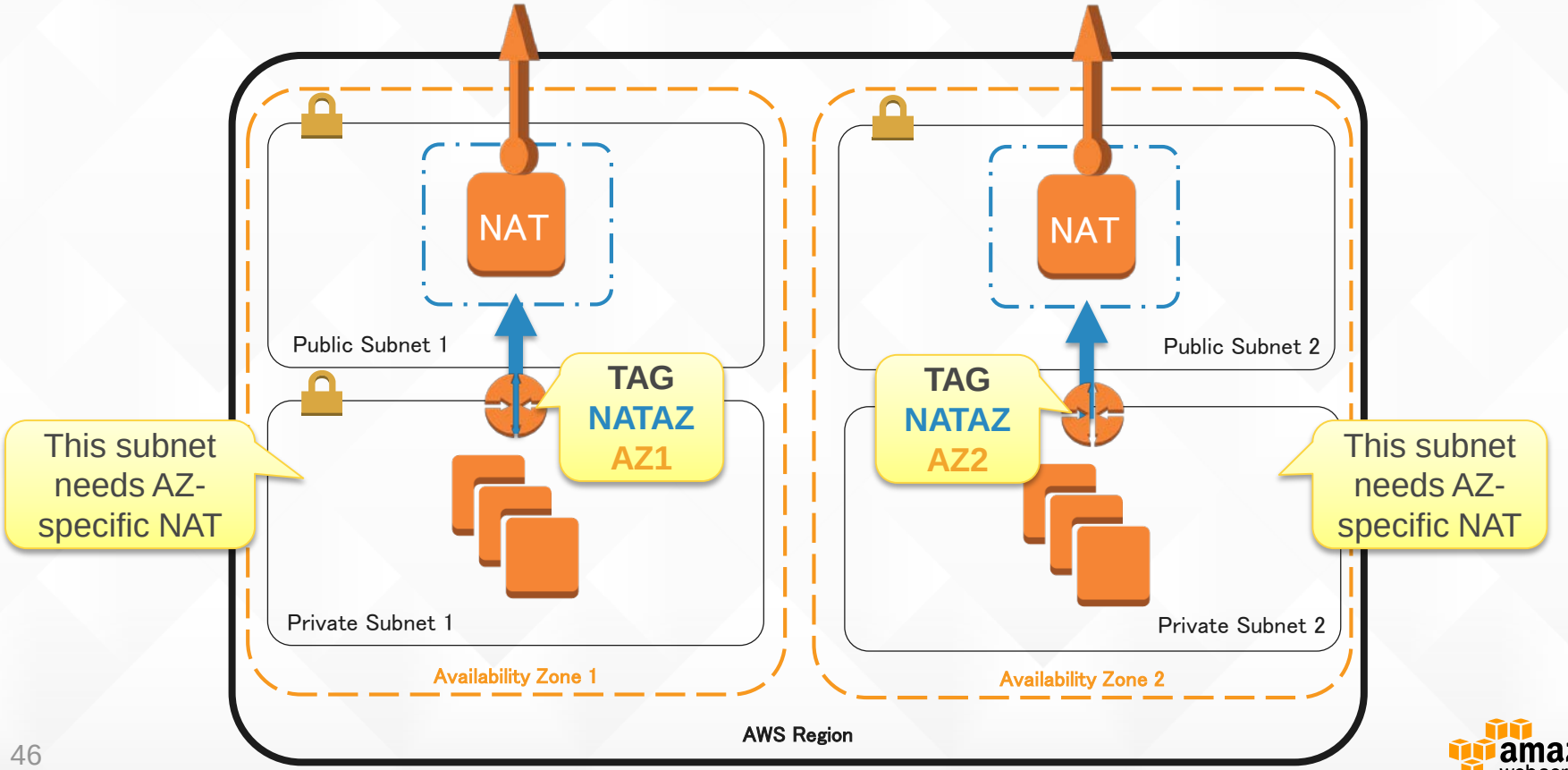

amazon web services

AWS SDK Python (VPCとサブネット作成・削除)

```
#!/usr/bin/python

import boto.vpc
Region="ap-northeast-1"

conn = boto.vpc.VPCConnection(Region)
vpc = conn.create_vpc('10.0.0.0/16')
subnet = conn.create_subnet(vpc.id, '10.0.1.0/24')
print ("created %s & %s" %(vpc.id, subnet.id))

#Clean up
conn.delete_subnet(subnet.id)
conn.delete_vpc(vpc.id)
```


amazon web services

タグによる動的NAT制御



タグによる動的NAT制御

```
#!/bin/bash
INSTANCE_ID=`/usr/bin/curl --silent http://169.254.169.254/latest/meta-data/instance-id`
AZ=`/usr/bin/curl --silent http://169.254.169.254/latest/meta-data/placement/availability-zone`
REGION="${AZ%?}"

MAC=`curl --silent http://169.254.169.254/latest/meta-data/network/interfaces/macs/`
VPC_ID=`curl --silent http://169.254.169.254/latest/meta-data/network/interfaces/macs/$MAC/vpc-id`

ROUTE_TABLES=`aws ec2 describe-route-tables --region $REGION --output text
                    --filters "Name=tag:NATAZ,Values=any,$AZ" | grep ROUTETABLES | awk '{print $2}'`

# Parse through RouteTables that need to be modified
for MY_RT_ID in $ROUTE_TABLES; do
        aws ec2 replace-route --route-table-id $MY_RT_ID --destination-cidr-block 0.0.0.0/0
                                                --instance-id $INSTANCE_ID` --region $REGION
done
```


amazon web services

参考資料

■ドキュメント
http://aws.amazon.com/jp/documentation/vpc/
http://aws.amazon.com/jp/documentation/direct-connect/

■Blackbelt Tech Webinar
http://www.slideshare.net/AmazonWebServicesJapan/aws-black-belt-tech-amazon-vpc-47025202
http://www.slideshare.net/AmazonWebServicesJapan/aws-black-belt-tech-aws-direct-connect

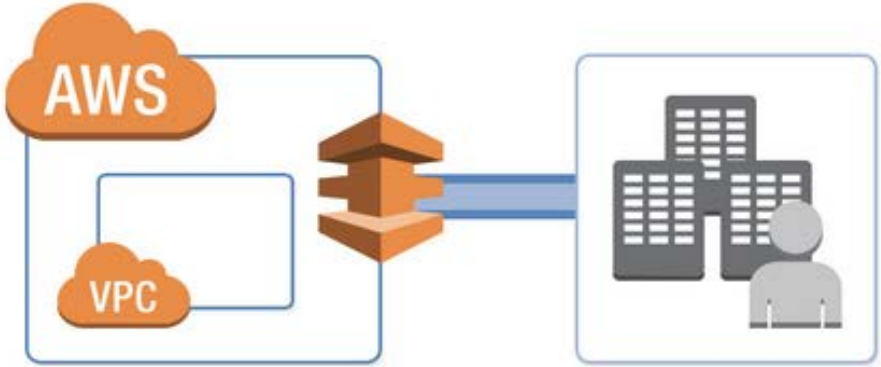
■re:Invent 2014 [SDD422] VPC Deep Dive
http://www.slideshare.net/AmazonWebServices/sdd422-amazon-vpc-deep-dive-aws-reinvent-2014
https://www.youtube.com/watch?v=HexrVfuIY1k

■re:Invent 2015 [ARC401] Black-Belt Networking for the Cloud Ninja
http://www.slideshare.net/AmazonWebServices/arc401-blackbelt-networking-for-the-cloud-ninja-aws-reinvent-2014
https://www.youtube.com/watch?v=-klyaq9R0XU

■Black-Belt Networking for the Cloud Ninja サンプルコード
http://s3.amazonaws.com/reinvent-arc401/index.html

AWS専用線アクセス体験ラボ

sponsored by Intel®



http://aws.amazon.com/jp/dx_labo/

学べる！

試せる！

Thank You

amazon
web services