ネットワークエンジニアの自分がクラウドインフラとどう向き合ってきたか

エクイニクス・ジャパン株式会社
丸野 達矢

EQUINIX®

自己紹介

名前：丸野　達矢　(MARUNO Tatsuya)
出身：熊本
趣味：バイク、キャンプ、スキー

業務経歴

- 2013年～2017年：
  → L2-L3メインのネットワーク設計構築運用
- 2018年6月ごろ：
  → クラウドインフラの設計構築チームにジョイン

©2019・Equinix.com

自身をとりまく環境

～本題に入る前に…～

オンプレのプライベート仮想基盤(VMwareベース)＋物理基盤で
自社サービスのWeb環境を運営しているお客様にて

- よりよいパフォーマンスの向上、最適化
- コスト削減

などを実現したい要望を受けた。

→ パブリッククラウド(主にAWS)＋物理基盤での
ハイブリット環境を設計構築

※AWSを主とした説明が多くなります。ご了承ください。

©2019 ・ Equinix.com

クラウドインフラの世界に. . .

©2019 ・ Equinix.com

テーマ

ネットワークエンジニアの自分が
クラウドインフラと、どう向き合ってきたか

©2019 ・ Equinix.com

クラウドインフラと向き合ってきたこと①

～ネットワーク図の話～

©2019 ・ Equinix.com

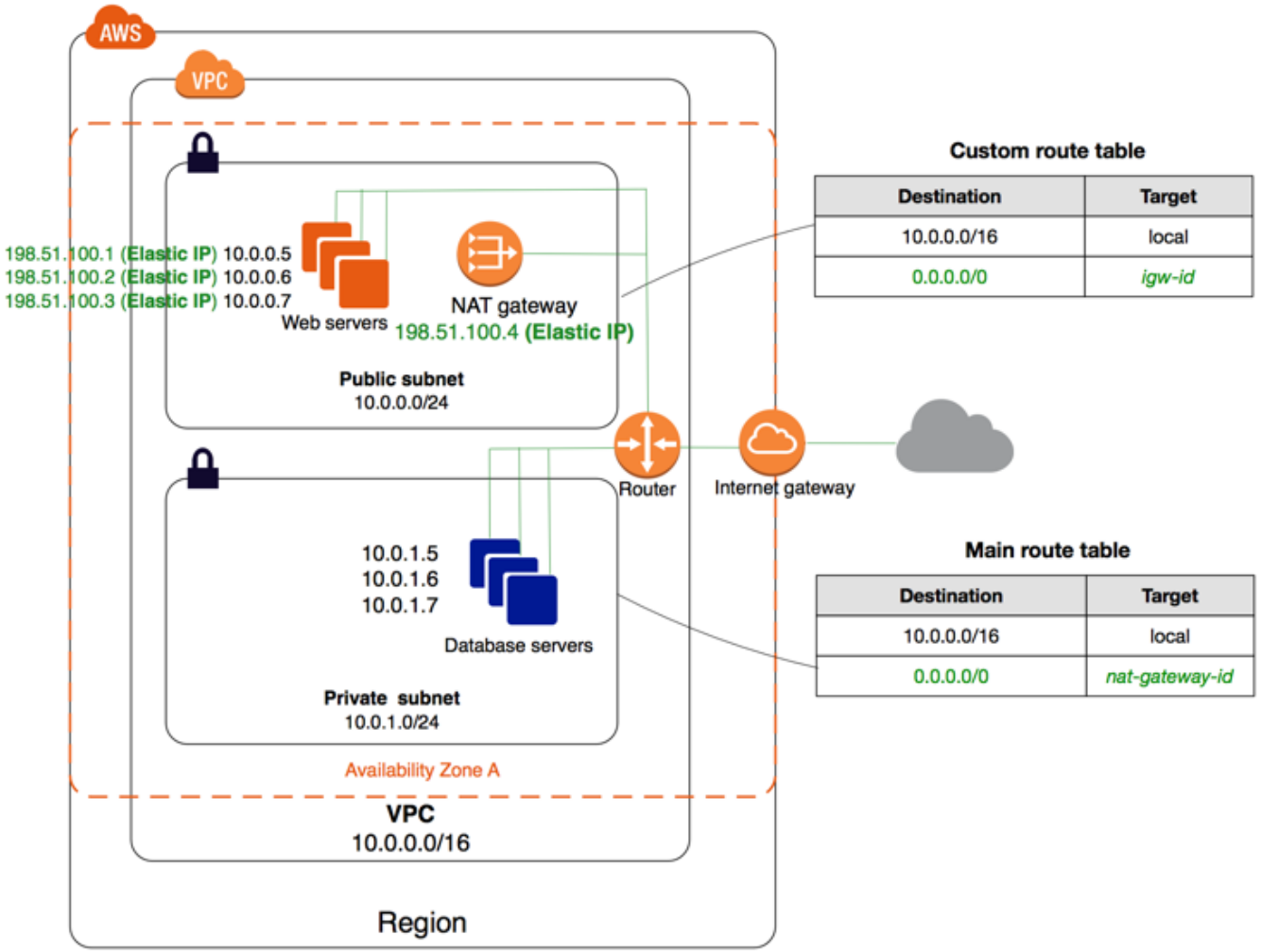
クラウドを触ってまず感じたこと

ネットワーク図が分かりづらい

©2019 ・ Equinix.com

クラウドネットワーク図の例

※各社の公式ドキュメントの図

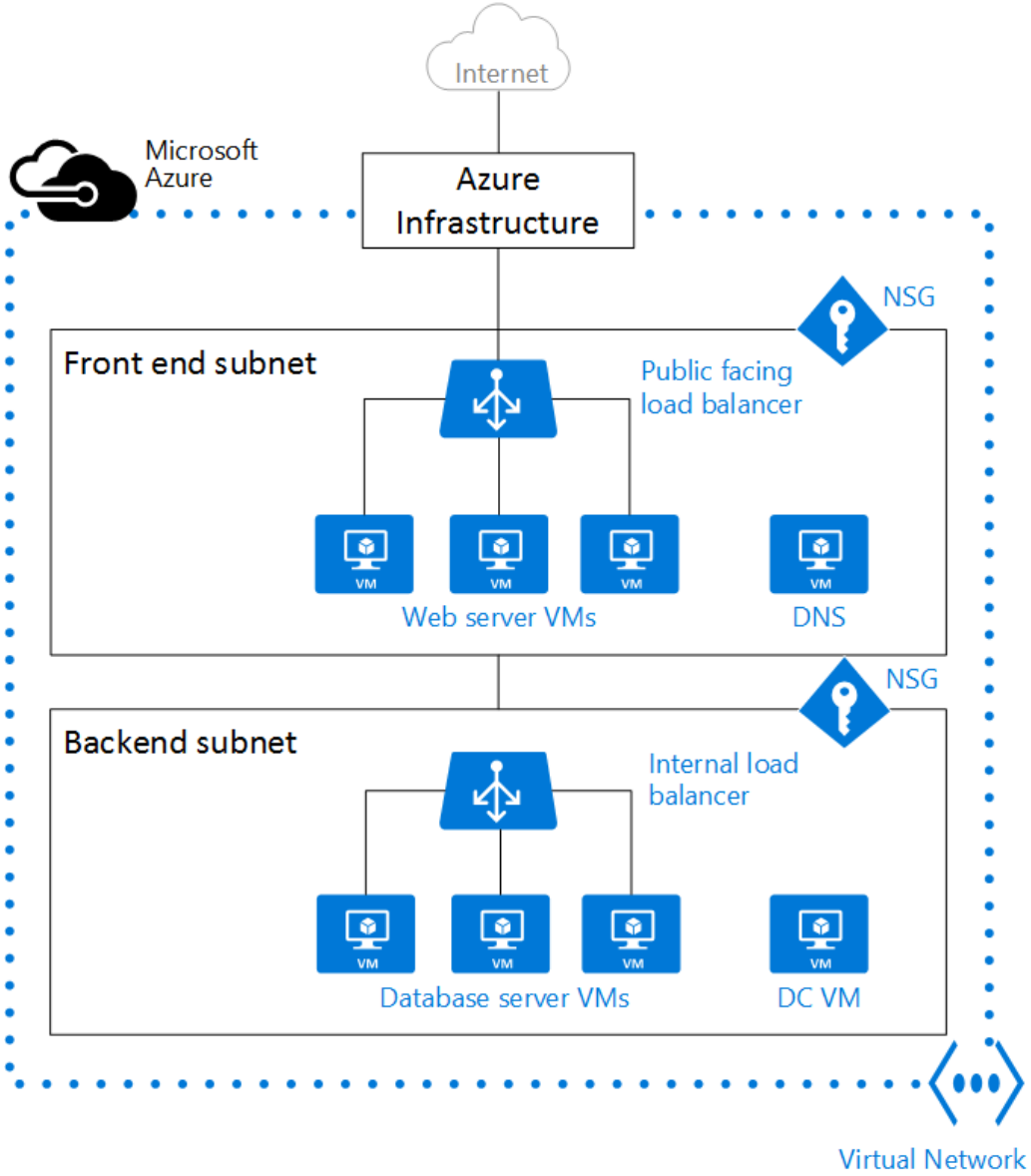


Custom route table

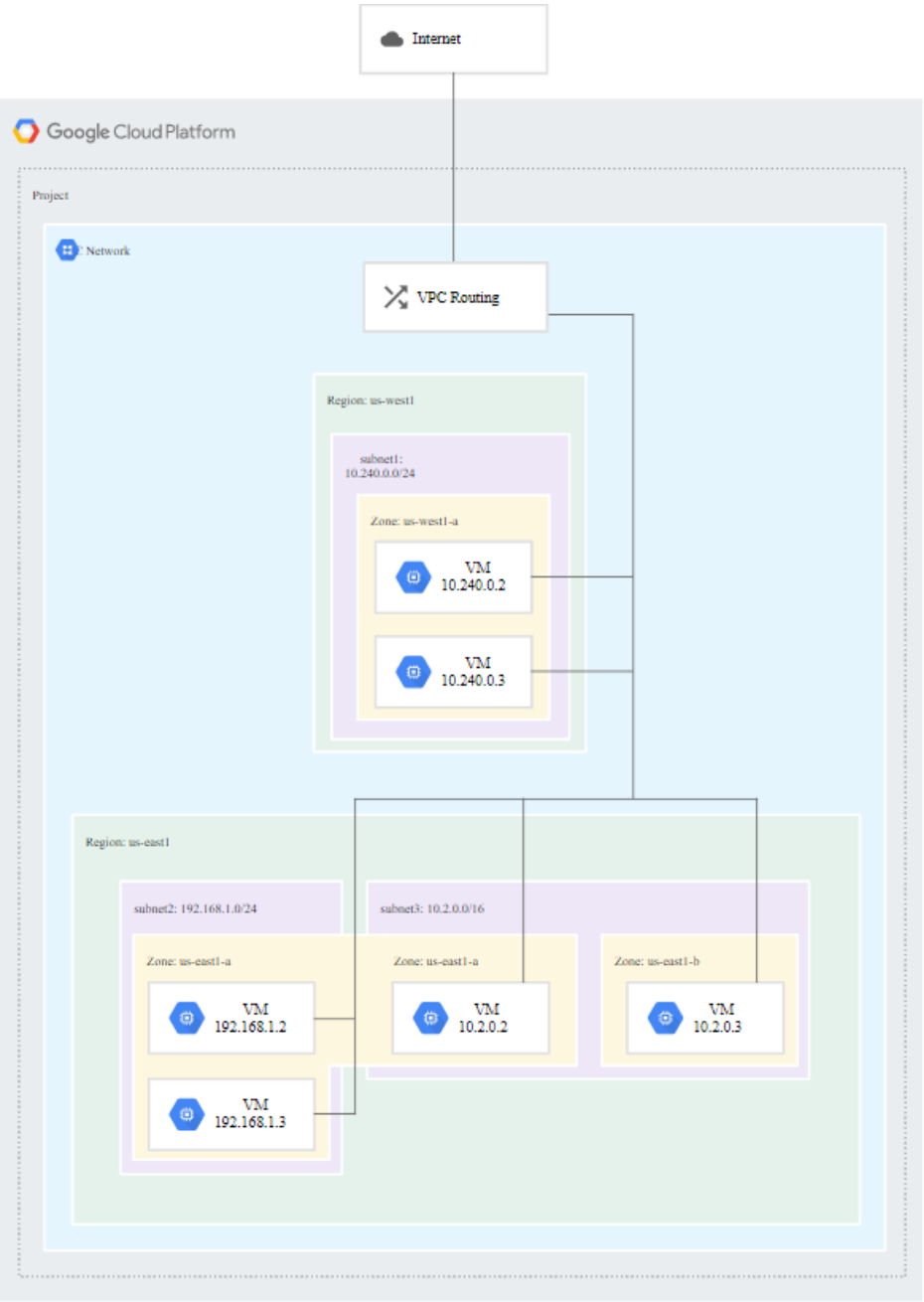
Destination	Target
10.0.0.0/16	local
0.0.0.0/0	*igw-id*

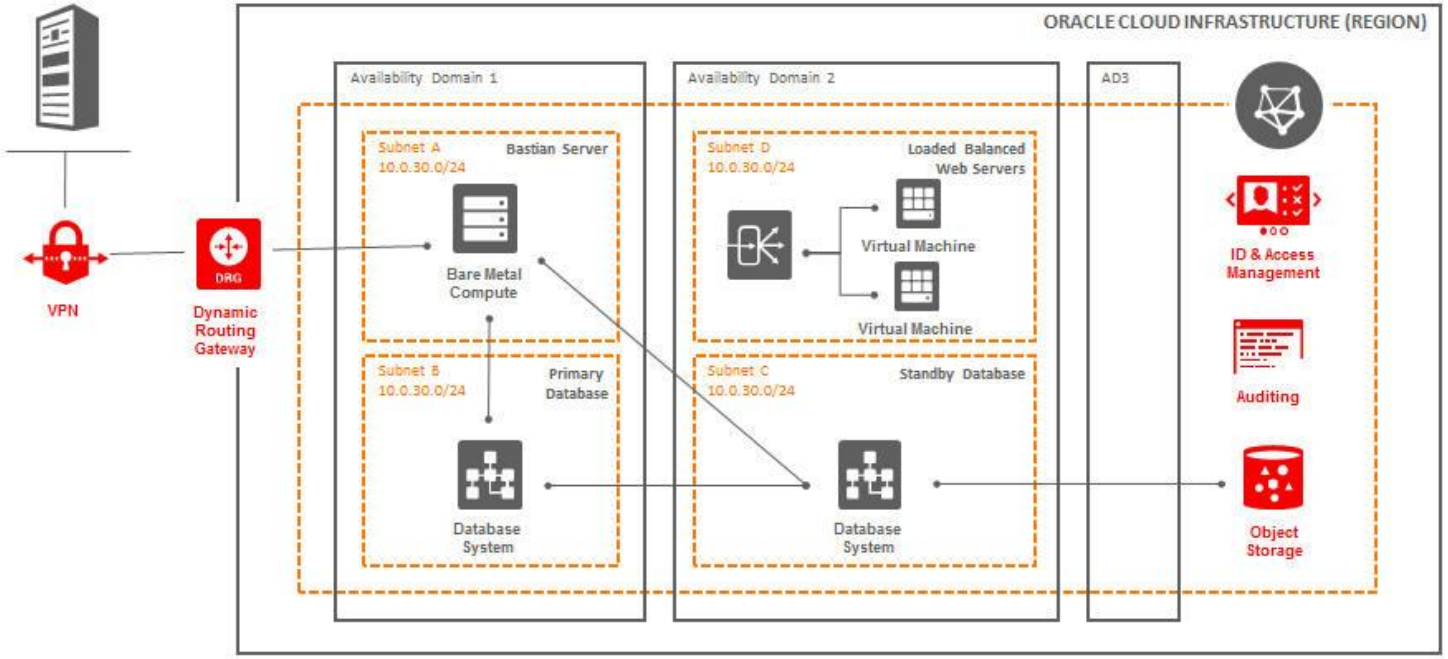
Main route table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	*nat-gateway-id*



©2019 ・ Equinix.com





©2019 · Equinix.com

実際に起こった問題

- クラウドをしらない人が図を見て理解してもらえない
- 理解するまでのハードルが高い
- 今までのオンプレ時代とのギャップが激しすぎる
- クラウド完結の構成であればまだわかるが、ハイブリット環境になるとどう書けばいいか難しい

→ 対ネットワークエンジニアでも説明が辛い
移行に向けた説明(到達性 / 冗長性の説明とか)、
お客さん相手には尚更困難。。。

©2019 ・ Equinix.com

構成図のルールを新たに設ける事にした

～改良するうえで意識したこと～

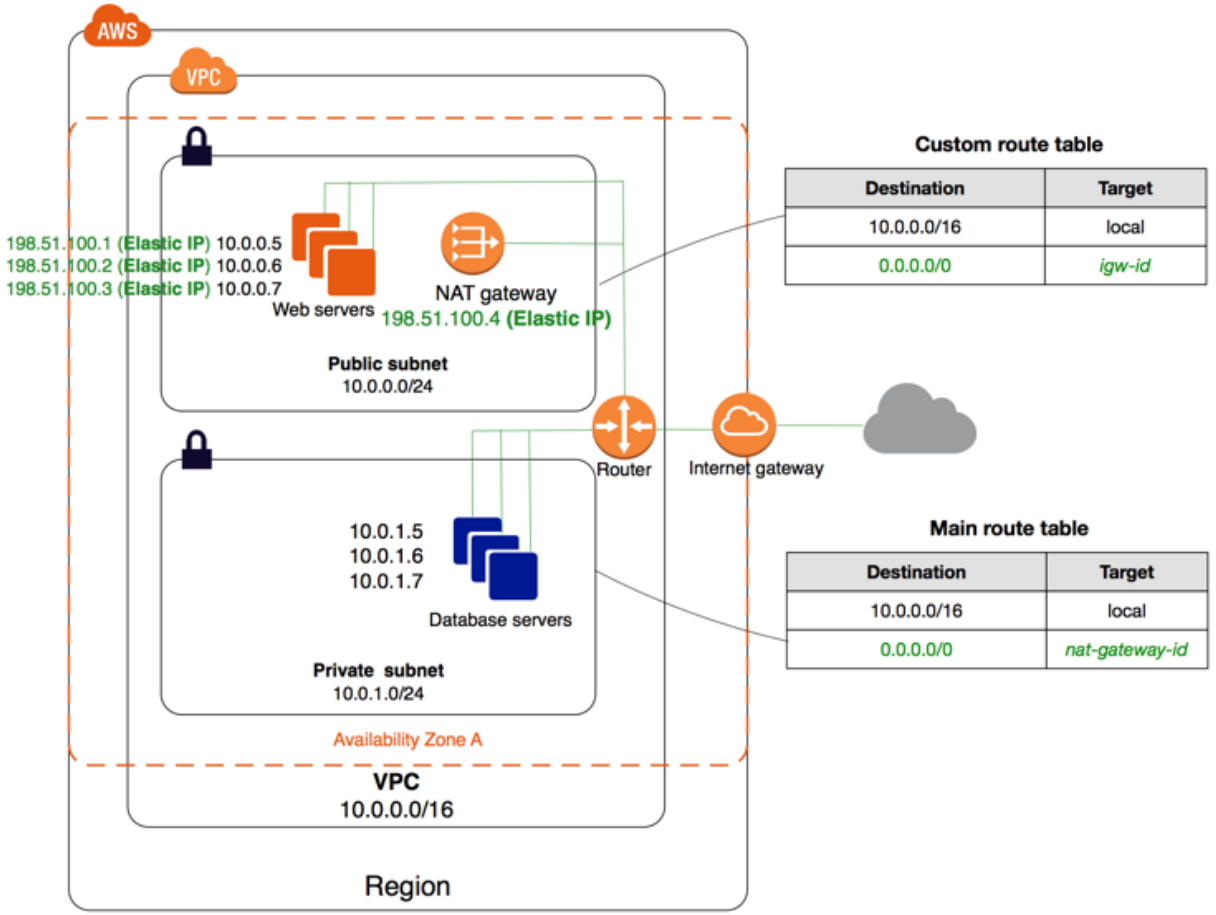
- 運用でもある程度使えるような、正確な図
（ルートテーブルを書かずに、経路を理解する）

- クラウドの仕様をそこまで理解してなくても、
ある程度繋がりがわかるような図
（でも仕様は正しく表現されてる）

※ 大前提としては、なるだけシンプルなネットワーク構成を作る

©2019 ・ Equinix.com

この図が、、



Custom route table

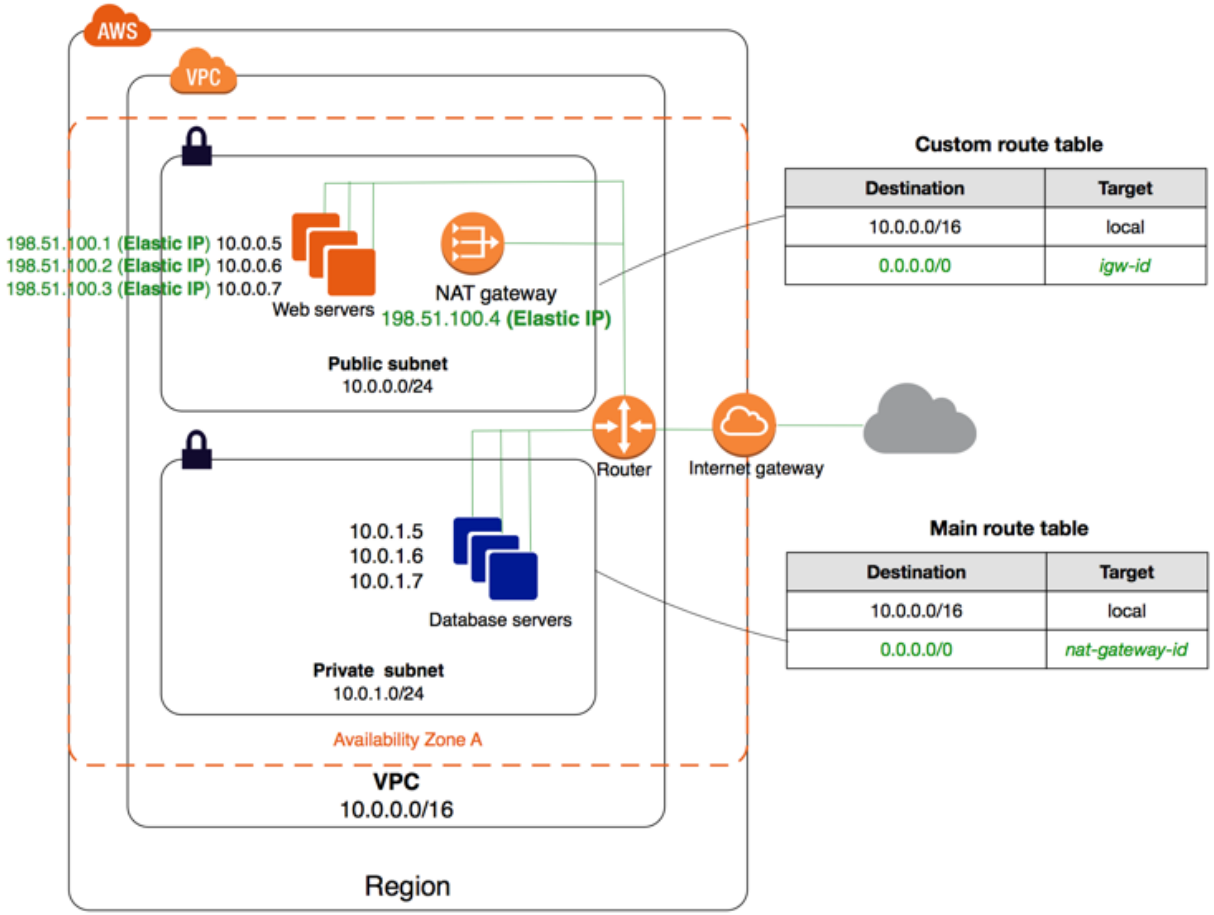
Destination	Target
10.0.0.0/16	local
0.0.0.0/0	*igw-id*

Main route table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	*nat-gateway-id*

©2019 ・ Equinix.com

こんな感じ

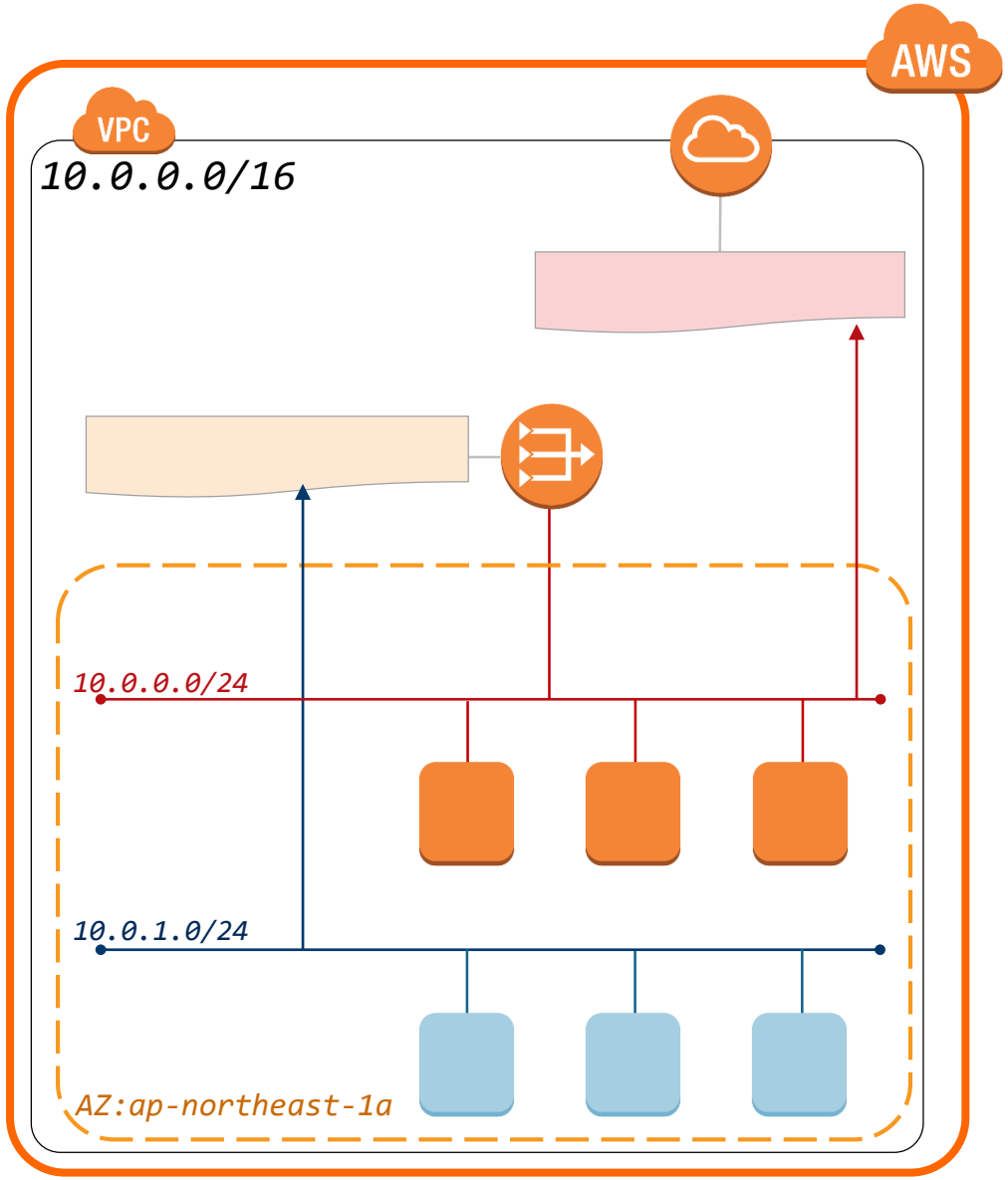


Custom route table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	*igw-id*

Main route table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	*nat-gateway-id*



©2019 ・ Equinix.com

やったことは3つ

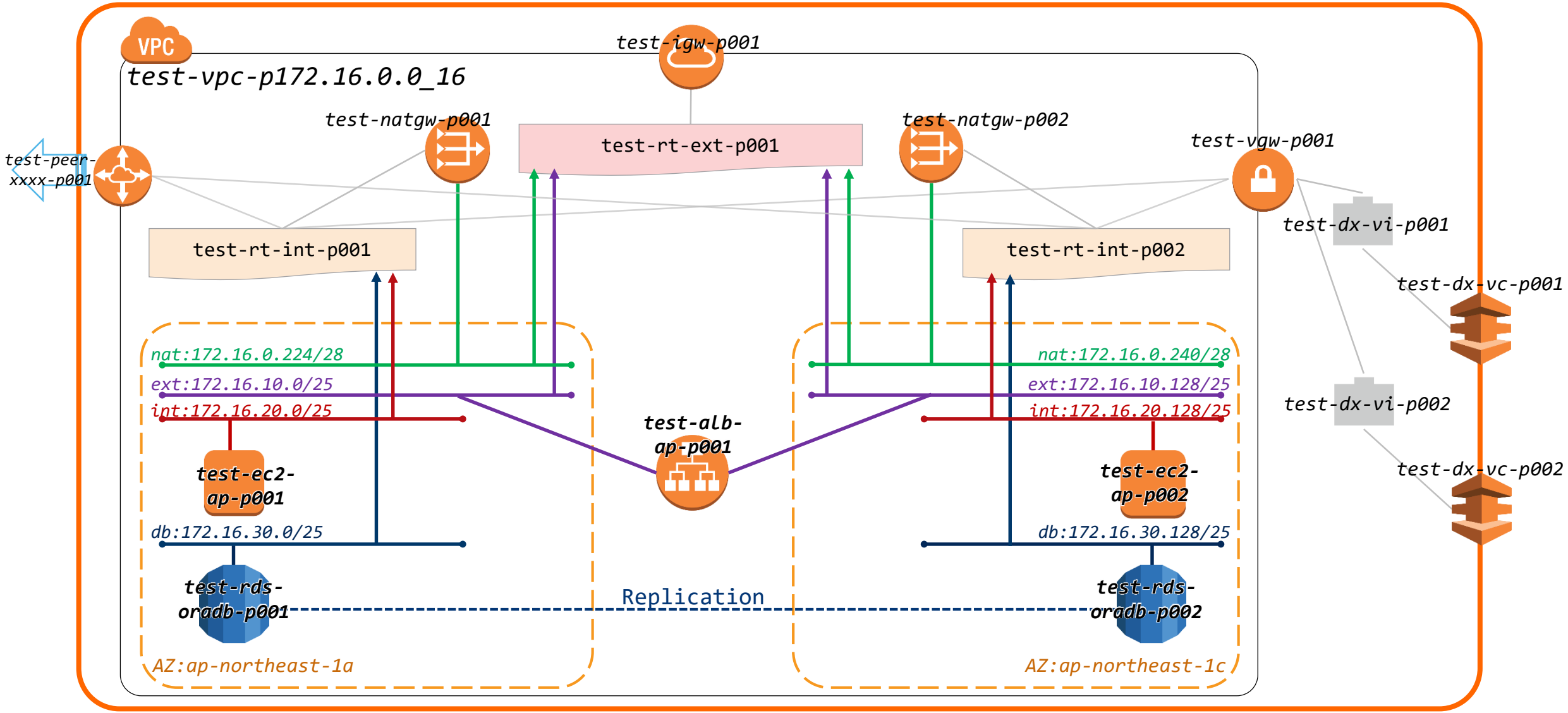
- **route-table：**
  ひと目でわかりやすいようなアイコンを追加し、到達性に合わせて線をつなげるルールに変更。また、経路が把握できればいいのでルーターアイコンは削除。

- **Subnet：**
  オンプレで用いられる論理構成の書き方になるだけ近づけるべく横線に変更。

- **Subnetから伸びる矢印：**
  instanceのoutbound通信は、そのinstanceのsubnetが紐づくroute-tableに流れる（AWSの仕様）。その意味を表現するために矢印を追加。逆に下りはどのroute tableからでも到達できるので要素として書かない、片方向矢印にしてるのもそのため。

©2019 ・ Equinix.com

実用だとこんな感じ（マルチAZ）



©2019 ・ Equinix.com

新ルール導入の結果

- 通信経路が明確になり、クラウド内のネットワークの動き（到達性,冗長性）が伝わりやすくなった。

- 図を通して仕様の理解が深まることで、公式ドキュメントの表現でも問題なく読めるようになった。クラウドネットワークを理解するハードルを下げることができた。

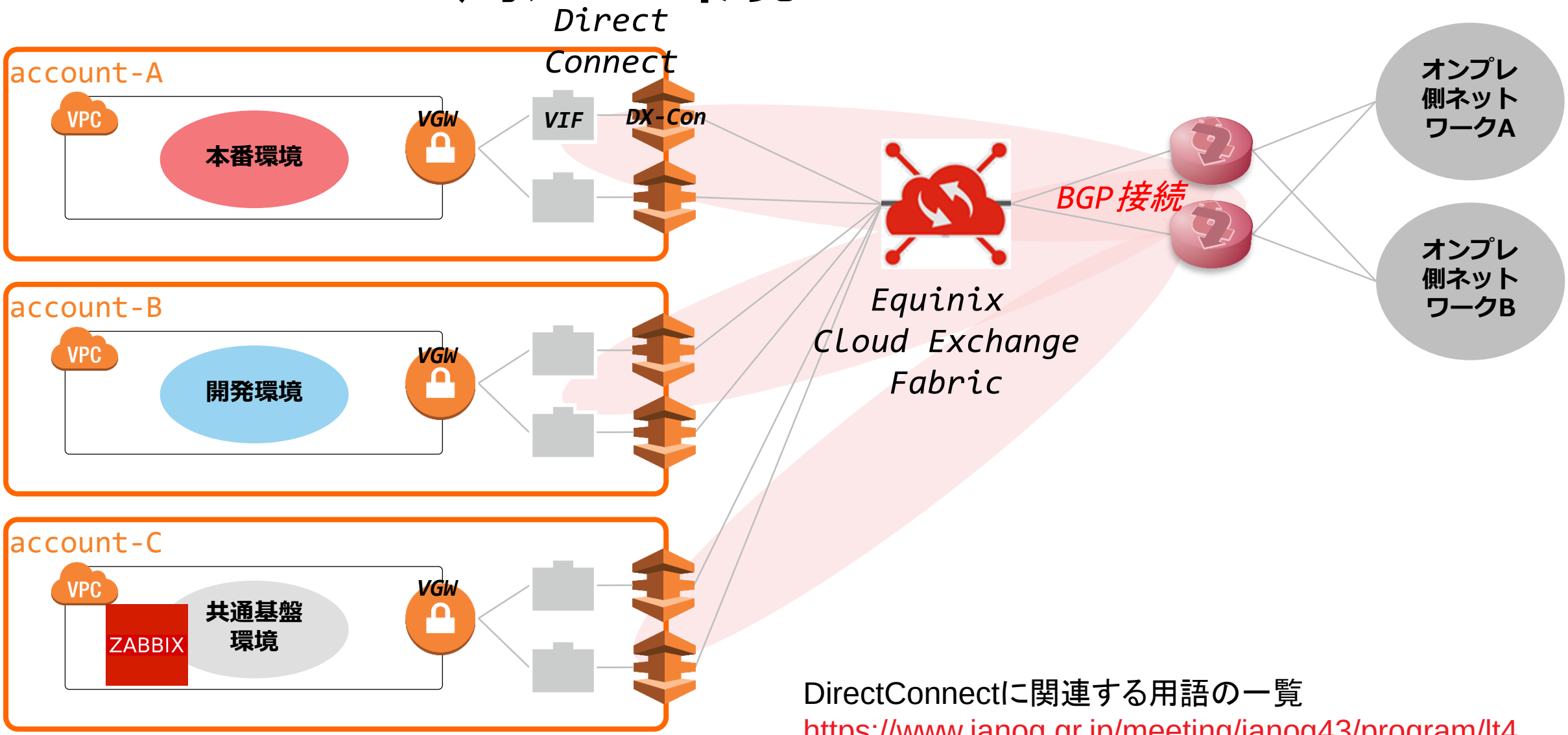
©2019 ・ Equinix.com

クラウドインフラと向き合ってきたこと②

～DirectConnect周辺の監視の話～

©2019 ・ Equinix.com

DirectConnect周辺の環境

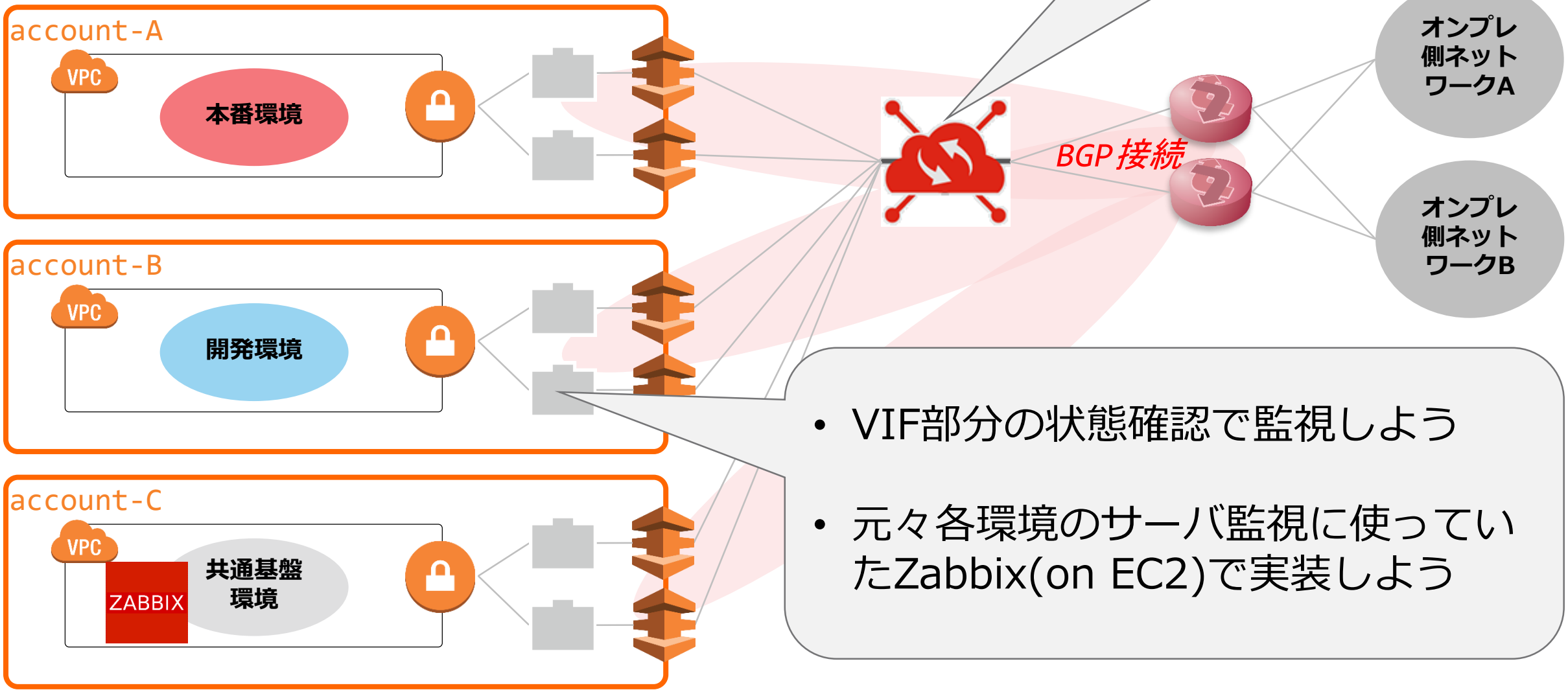


DirectConnectに関連する用語の一覧
https://www.janog.gr.jp/meeting/janog43/program/lt4

©2019 ・ Equinix.com

DirectConnectの監視がしたい

※サービス内機能で障害時
メール送信する仕組みあり



- VIF部分の状態確認で監視しよう
- 元々各環境のサーバ監視に使っていたZabbix(on EC2)で実装しよう

©2019 ・ Equinix.com

VIF状態確認方法

～確認するAWS API～

aws directconnect describe-virtual-interfaces

→ 「virtualInterfaceState」の項目

→→ Zabbix-itemで取得、“available“の文字列を含まない場合はアラート通知

「virtualInterfaceState」の状態	意味
confirming	VIF作成が承認されるまで保留中
verifying	パブリック仮想インターフェイスにのみ適用 各パブリック仮想インターフェイスは、仮想インターフェイスを作成する前に検証が必要
pending	作成承認後、VIFがトラフィックを転送できるまで準備中
available	トラフィックを転送できる
down	BGPがダウンしている（トラフィックを転送できない）
deleting	VIF削除中
deleted	VIF削除済み
rejected	VIF作成が承認されなかった
unknown	VIFの状態不明

https://docs.aws.amazon.com/cli/latest/reference/directconnect/describe-virtual-interfaces.html

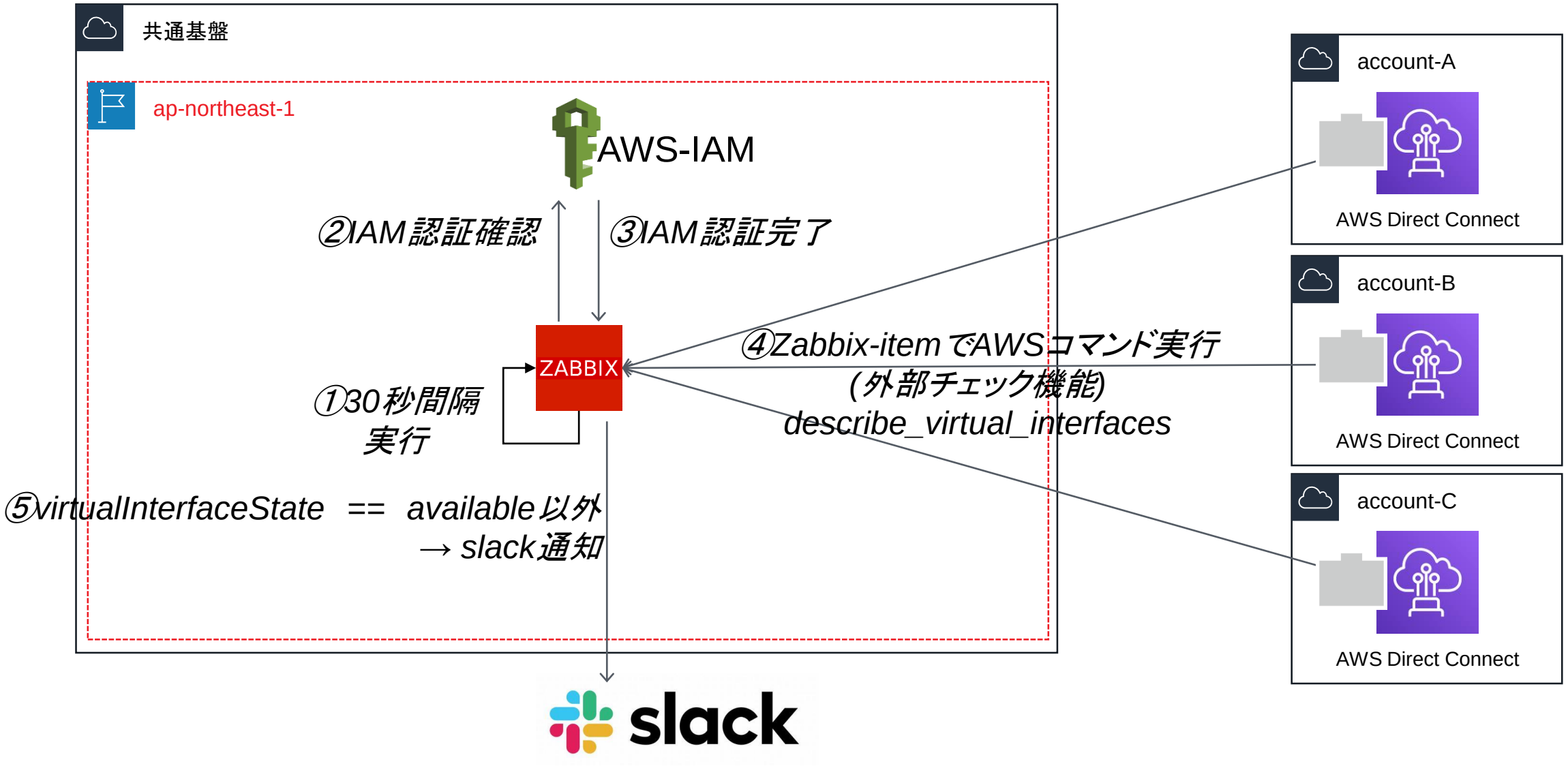
©2019 ・ Equinix.com

問題発生

想定外の不要なアラート頻発
&
zabbixサーバリソースが高負荷状態

©2019・Equinix.com

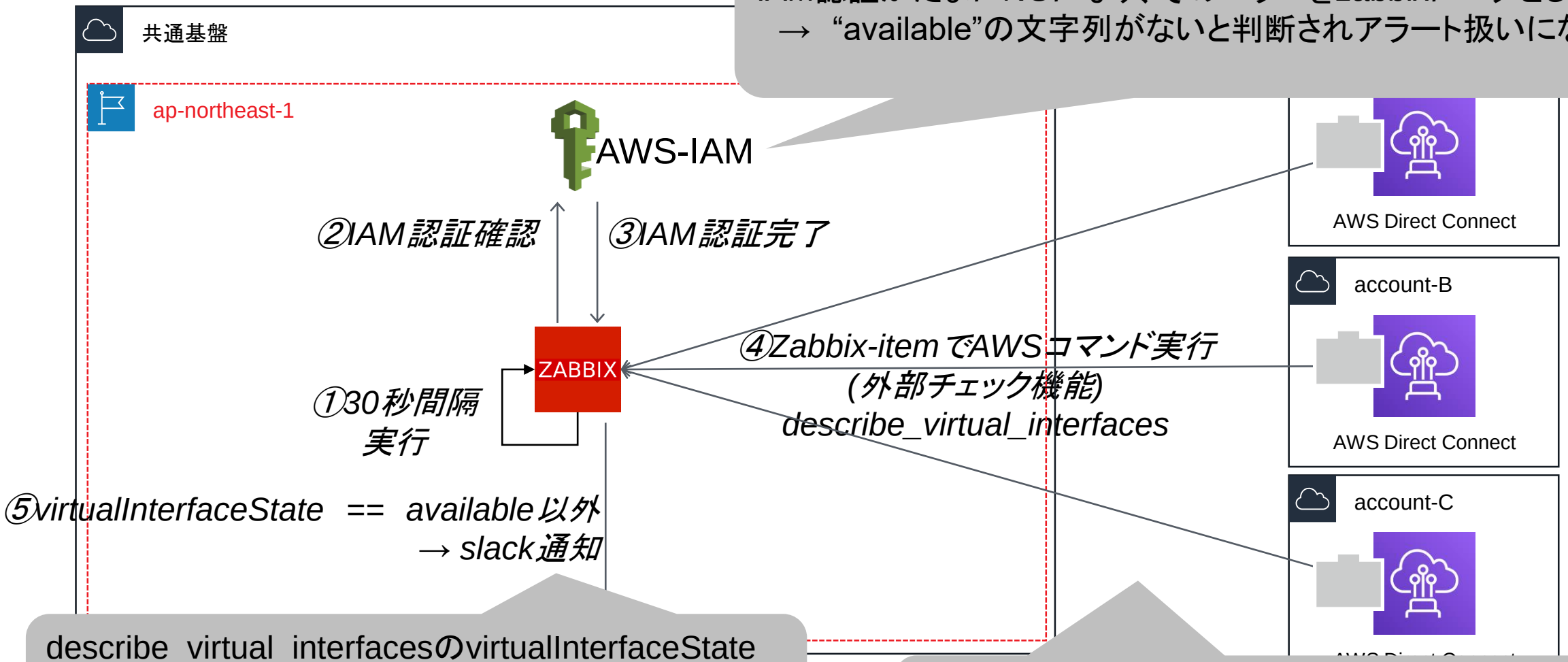
構成



©2019 ・ Equinix.com

なにが起きてたか

IAM認証がたまにNGになり、そのエラーをzabbixにログとして返す
→ “available”の文字列がないと判断されアラート扱いになる



describe_virtual_interfacesのvirtualInterfaceState
→ BGP-peerは切れてないのにVIFの
状態がunknownに遷移

describe_virtual_interfacesのコマンドエラー
→ DirectConnect側のAPIで何かしらの障害発生

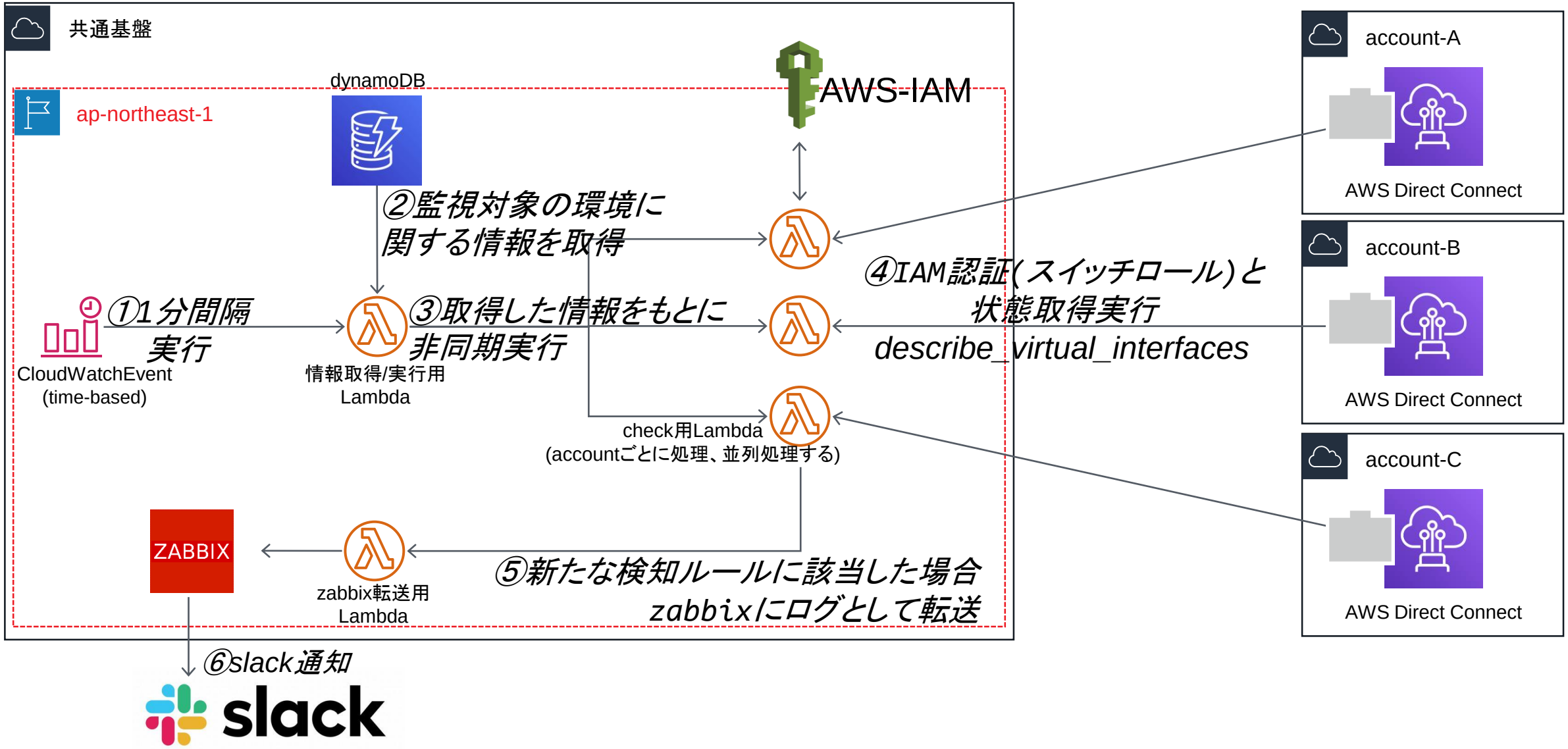
©2019 ・ Equinix.com

対策

実用性向上＆リソース分離を目指し、
再設計&サーバレス化

©2019・Equinix.com

新構成



©2019 ・ Equinix.com

実用性向上を目指した、新たな検知ルール

3パターンで確認

- スイッチロールのエラーが出た場合
- VIFが削除されたなどの理由より、対象が存在しない場合
- virtualInterfaceState, bgpStatusどちらかの状態が“down”と出た場合

※他のLambdaエラーが発生したときは、別の仕組みで検知

”unknown”については、過去の実績で1度もBGP障害が発生していないため、アラートとして扱わない判断をしました

©2019 ・ Equinix.com

結果

- zabbixとのリソース分離ができ、
  VIFについて安定した監視環境を作ることができた

→　目的達成

- 低いコストで監視環境を作ることができた
- サーバーレスとはいえ、実はやってること簡単
  （AWSのAPIを実行して結果を判定してるだけ）

→　今後につながる実績になった

©2019・Equinix.com

課題

- イベントドリブンではなく、約1分間隔
  → 1分未満の断が発生したときに気づけない（解決にむけ実装準備中）

- サービス通信の状況をわかるようにする必要がある
  → アラート検知後エンジニアが確認してる、より簡略化していきたい

  ※LambdaからpingやDNSの問い合わせをすることができない（各環境にEC2を建てるしかない？？）

- メンバーにアプリレイヤの方がいるから実現できた
  → チーム内の技術力、コード管理文化の向上が必要

©2019 ・ Equinix.com

クラウドインフラと向き合って感じた
クラウドネットワークのいいところ悪いところ

©2019 ・ Equinix.com

クラウドネットワークのいいところ

- 思ってたよりネットワークの知識がいる

→ 正直最初は「これからはネットワークの知識なんかなくてもインフラが作れるような時代になるのか」と思ってたが、全然そんなことなかった

L3ルーティング、BGP、VPN、ネットワークセキュリティなどいままで物理でやってきたことと変わらない

クラウドネットワークのいいところ

- ネットワーク管理の簡素化、非属人化

→ 初回のハードルは高かったが、仕様を理解しルール化(ネットワーク図も含めて)すると、とてもシンプルで伝えやすいネットワークになる

→→ 多くの人に理解させやすいので、属人化を防ぐことができる
管理を分散できるので、他のことに注力できる

※自分の場合は、業務効率化チームに加わってansibleやpythonを使った業務効率化/自動化に日々取り組んでます。

©2019 ・ Equinix.com

クラウドネットワークのいいところ

- L2設計の開放、L2ループの面倒さからの開放
  → VLANの設計やループ防止など、考える必要がなくなった

- セキュリティ機能が充実
  → OSに依存することなく、“サーバ単位”でファイアウォール設定が簡単にできる（同じサブネットでもOK）

- グローバルIP利用がすぐできる
  → サービス環境の追加だけでなく、検証やマネジメント環境利用など、インターネットからアクセスできる環境をすぐ作れる

©2019 ・ Equinix.com

クラウドネットワークの悪いところ

- 物理のネットワーク環境と違って、障害試験ができない
  → どうしようもない部分

- クラウド側のネットワーク機能上でのBGPステータス確認、pingやtracerouteなどの到達性/経路確認ができない
  → 過去のJANOGで実施された「インフラエンジニアがクラウドについて語らうBoF」でも、同じ意見を持ってる人は多かった

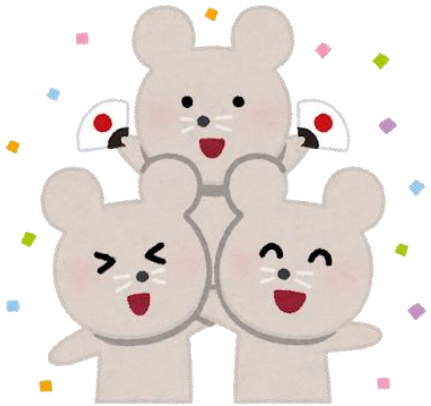
©2019 ・ Equinix.com

まとめ

クラウドインフラと、どう向き合ってきたか

①業務関係者と会話しやすいように、
ネットワーク図の書き方ルールを作った
→ クラウドネットワークの構成や仕様理解のハードルを下げた

② DirectConnect接続の監視、サーバーレスに挑戦
→ サーバレス実現、次につながる実績になった
ただしアプリレイヤのメンバーがいないとスピード感でない

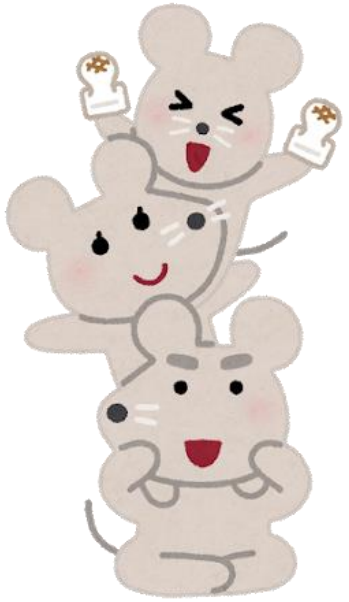


©2019 ・ Equinix.com

クラウドインフラをどう感じてるか

- ネットワーク知識は必ず必要
  → パケットの気持ちになる、仕組みが分かればシンプル
- コード文化必要
  → 自分たちで作る/壊す/増やす/維持管理する

→→ 今までと比べて、半分以上の業務は変わった



- パブリッククラウドをどう選択していくか難しい
  → 適材適所を理解し、最適なインフラを作れるよう目指す

©2019 ・ Equinix.com

会場に聞きたいこと

- こんな部分でクラウドをうまく利用してます。などの
例があれば知りたい

- 使いたいけど使えない or あえて使ってない。などあれば
その理由も含めて教えてほしい

- クラウドを使う際CLIベースの運用は必要、ただなかなか教育や文化作り
をするの難しい。
うまくやっている例などあれば知りたい

©2019 ・ Equinix.com

EQUINIX®