\Orchestrating a brighter world NEC

Tungsten Fabricを用いたサービスチェイニングを試してみた

–クラウドサービスで提供できる機能とは?–

2019年 7月 25日

中原一彦
日本電気株式会社

池内　望
日本電気通信システム株式会社

Orchestrating a brighter world

未来に向かい、人が生きる、豊かに生きるために欠かせないもの。
それは「安全」「安心」「効率」「公平」という価値が実現された社会です。

NECは、ネットワーク技術とコンピューティング技術をあわせ持つ
類のないインテグレーターとしてリーダーシップを発揮し、
卓越した技術とさまざまな知見やアイデアを融合することで、
世界の国々や地域の人々と協奏しながら、
明るく希望に満ちた暮らしと社会を実現し、未来につなげていきます。

目次

1. はじめに
2. サービスチェイニングについて
3. なぜTungsten Fabricなのか？
4. Tungsten Fabricでサービスチェイニングを検証してみた
5. ディスカッションしたいこと

はじめに

今回の発表について

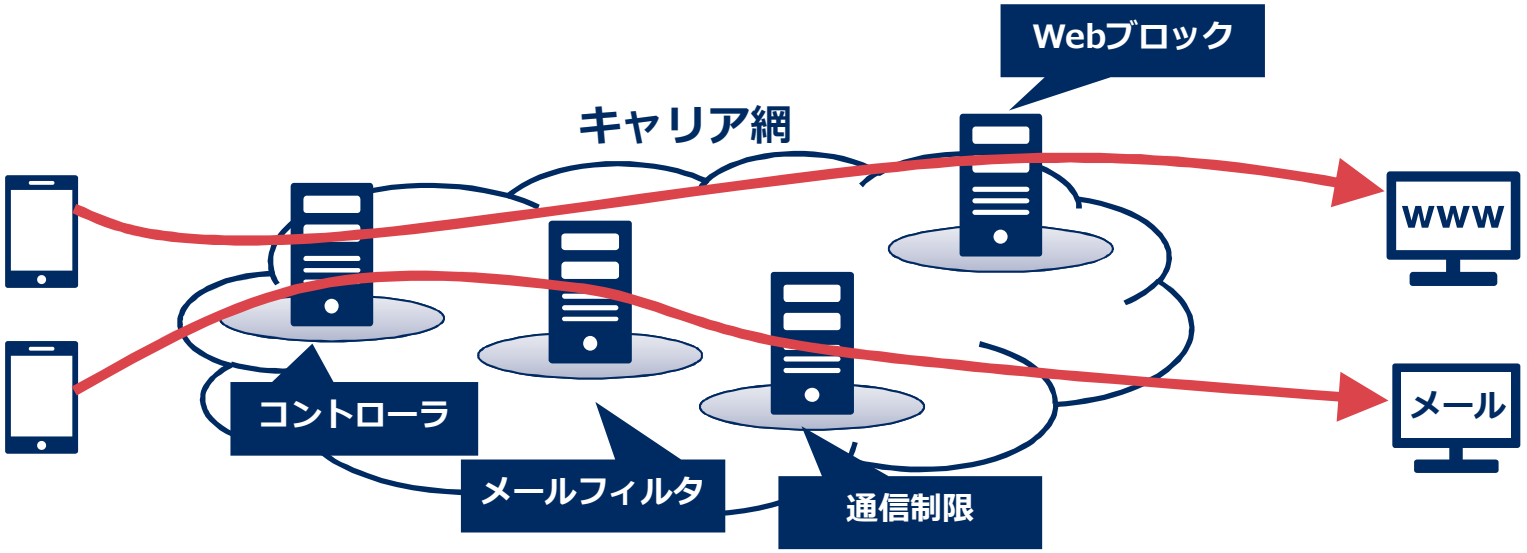
- サービスチェイニングをクラウドの世界で有効的に使うには、何を気を付ければ良いんだろう？どんなサービスが向いているんだろう？

- 私たちは、サービスチェイニングで期待される効果の仮説を立て、それがTungsten Fabricを使って実現できるかを検証してみました。

サービスチェイニングについて

サービスチェイニングの代表的なユースケース

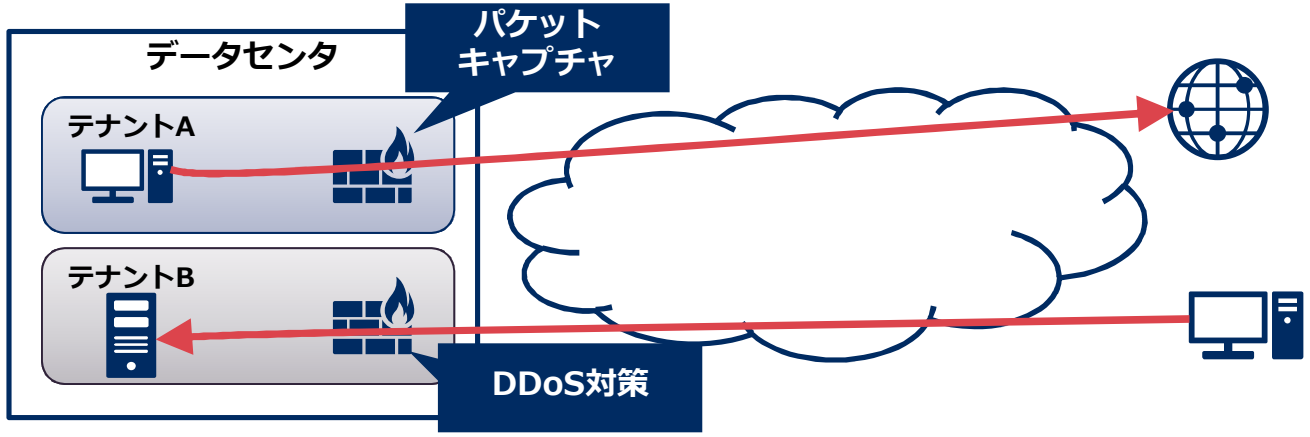
- サービスチェイニングと言えば、キャリア系ではメールフィルタやアプリ通信制限などのサービスを提供する際に用いられる



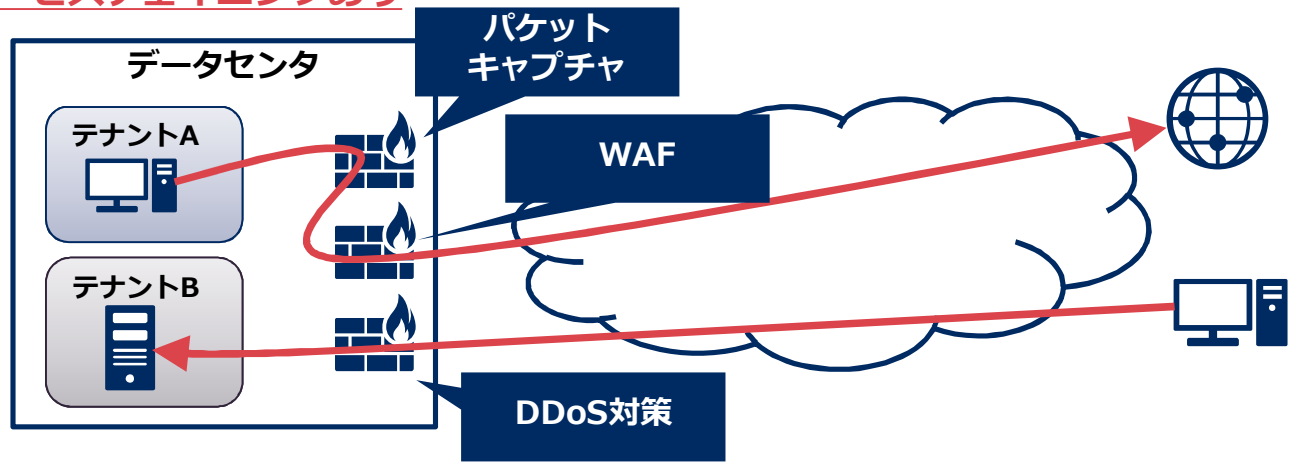
サービスチェイニングをクラウド事業者で応用できないか？

- クラウド事業者の世界でもテナント毎にセキュリティ機器を置かずに、
  サービスチェイニングによって共通的なサービスを提供出来ないだろうか？

サービスチェイニングなし



サービスチェイニングあり



今回検討したモチベーション

クラウドの世界でサービスチェイニングを導入する事によって
得られる効果の仮説

1. **ユーザの運用が楽になる**
   サーバのデフォルトGWなどのNW設定の変更をせず、
   必要なサービスを申し込むだけで、サービス利用を可能にしたい。

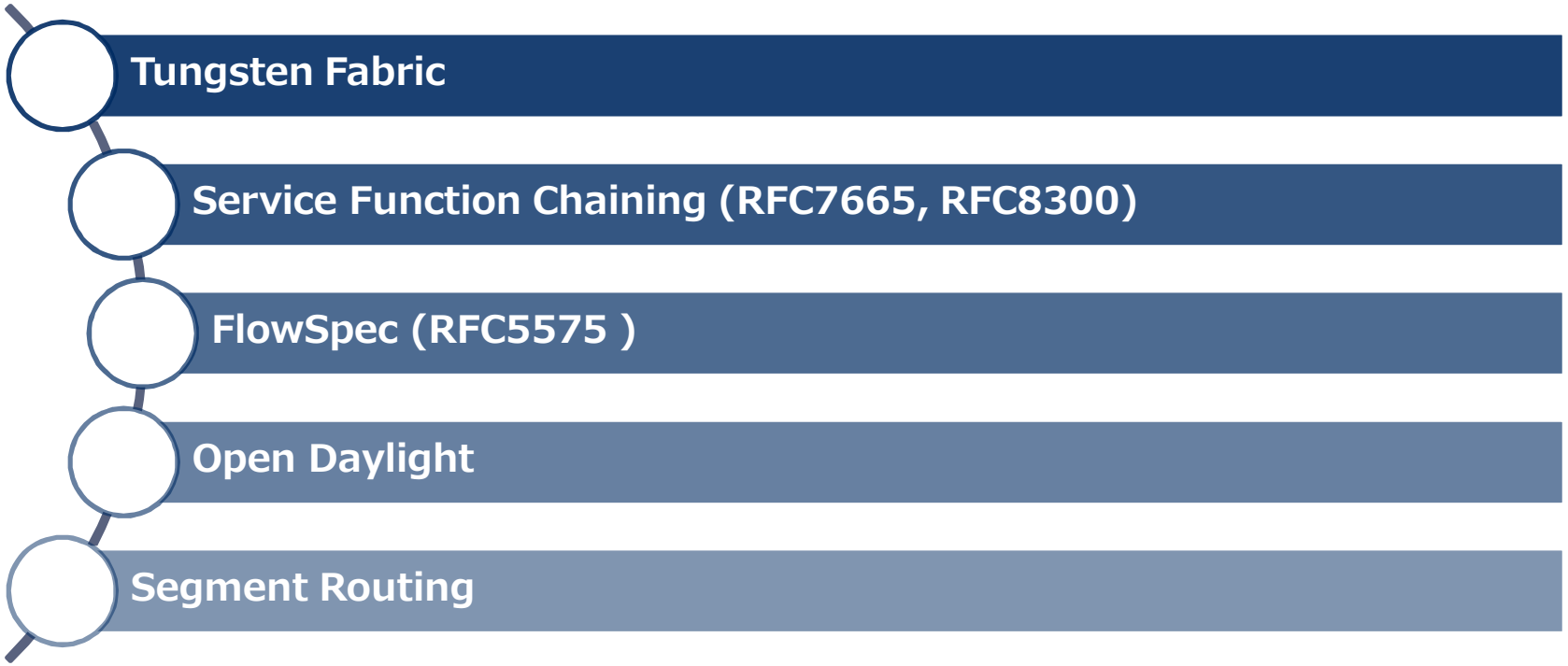
2. **クラウド事業者側の運用者の負荷を軽減できる**
   運用者がユーザからの申告で実施するパケットキャプチャなどの
   トラブルシューティングをユーザが実施する仕組みを構築し
   運用者の作業負荷を軽減したい。

3. **攻撃を受けたときの緊急回避を素早く行える**
   事業者側でWAFやDDoS対策などの攻撃対策を
   柔軟に導入したい。

なぜTungsten Fabricなのか？

実現方式の検討

- 一口で「サービスチェイニング」と言っても、実現する手段は、この世の中に色々とある。



実現方式の検討

今回は、サービスチェイニングを実現する手段の中で、実例が多くOpenStackとの親和性が高い**Tungsten Fabric**を用いた検証を行った。

No	方法	選定理由
1	**Tungsten Fabric**	● **旧 OpenContrail も含めて事例が多く、知識のあるOpenStackとの親和性も高いため採用。**
2	Service Function Chaining (RFC7665, RFC8300)	● AWS 等で使用できる事は把握したが、実現解などを見いだせなかった為、今回は保留とした
3	FlowSpec (RFC5575)	● FlowSpecだけではサービスチェイニングが実現できる訳ではない。また、オープンな実現解を見いだせなかった為、今回は保留とした
4	Open Daylight	● OpenFlowを前提としており、制限事項を見切れなかった為、今回は保留とした
5	Segment Routing	● 検討はじめた段階では研究、実験段階という認識であったため、今回は保留とした ● Interopなどで登場しておりトレンドなので、将来、検討したい

Tungsten Fabricって？

tungstenfabric とは？

- オーバーレイ技術を使用したオープンソースのSDNコントローラ
  - 昔は「Open Contrail」と呼ばれていましたが、2018年3月にLinux Foundationへ移行が完了し「Tungsten Fabric」に。
- OpenStackとの組み合わせで、Compute/Networkの両リソースを連携可能
- オーバーレイにEVPNを採用しており、各種NW機器や他のコントロールドメインとの連携が容易
- **サービスチェイニングにも柔軟に対応可能**

今回、期待するところ

【公式サイト】https://tungsten.io/

Tungsten Fabricで
サービスチェイニングを検証してみた

検証環境

- 仮想化基盤として openstack を使用
- サービスチェイニングの実現手段として tungstenfabric を使用
- バージョン
  - OpenStack Queens
  - Tungsten Fabric 5.1.0

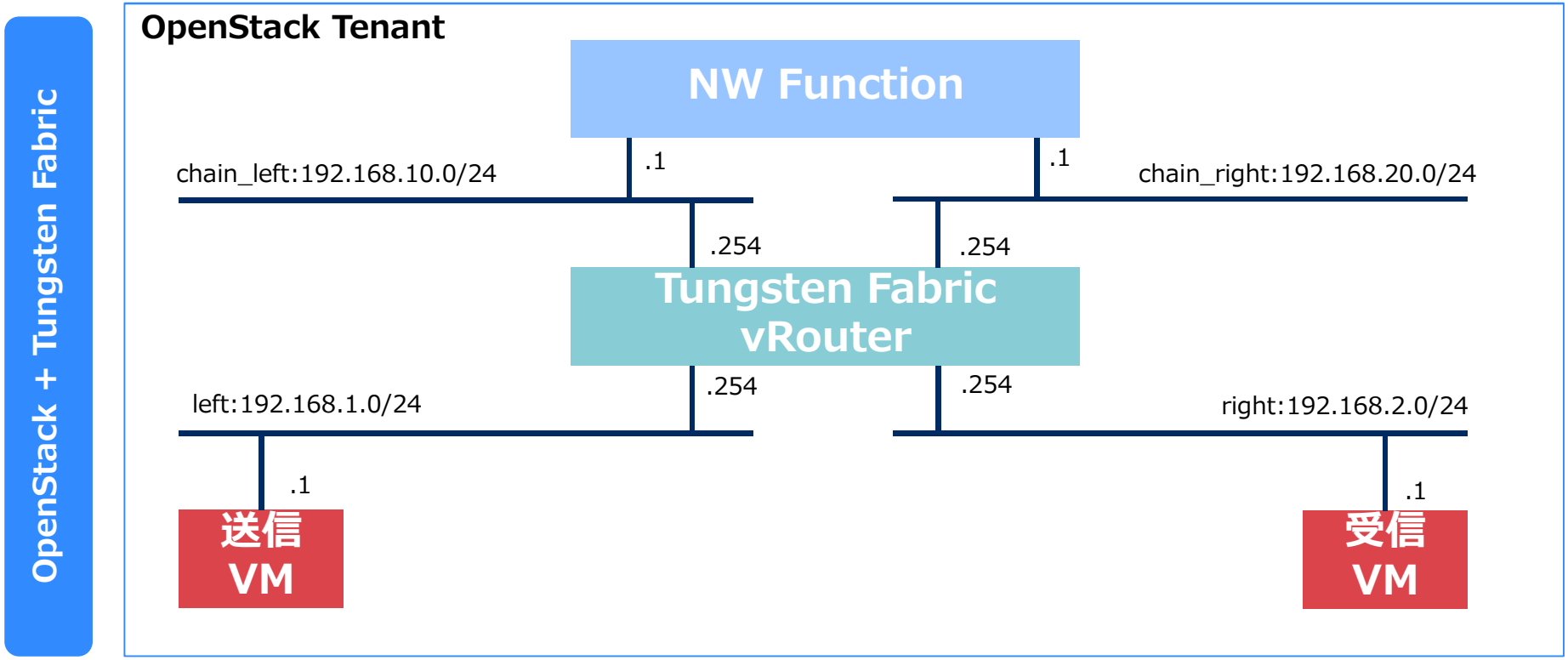
検証した機能とねらい

それぞれ仮説が正しいかを検証してみた

#	仮説	検証方法
1	ユーザの運用が楽になる	**ユーザ側のNW設定なしでサービスチェイニングが出来るかを確認**
2	クラウド事業者側の運用者の負荷を軽減できる	**パケットキャプチャ機能を経路途中に挟み込めるかを確認**
3	攻撃を受けたときの緊急回避を素早く行える	**WAF機能、DDoS対策機能を経路途中に挟み込めるかを確認**

検証構成

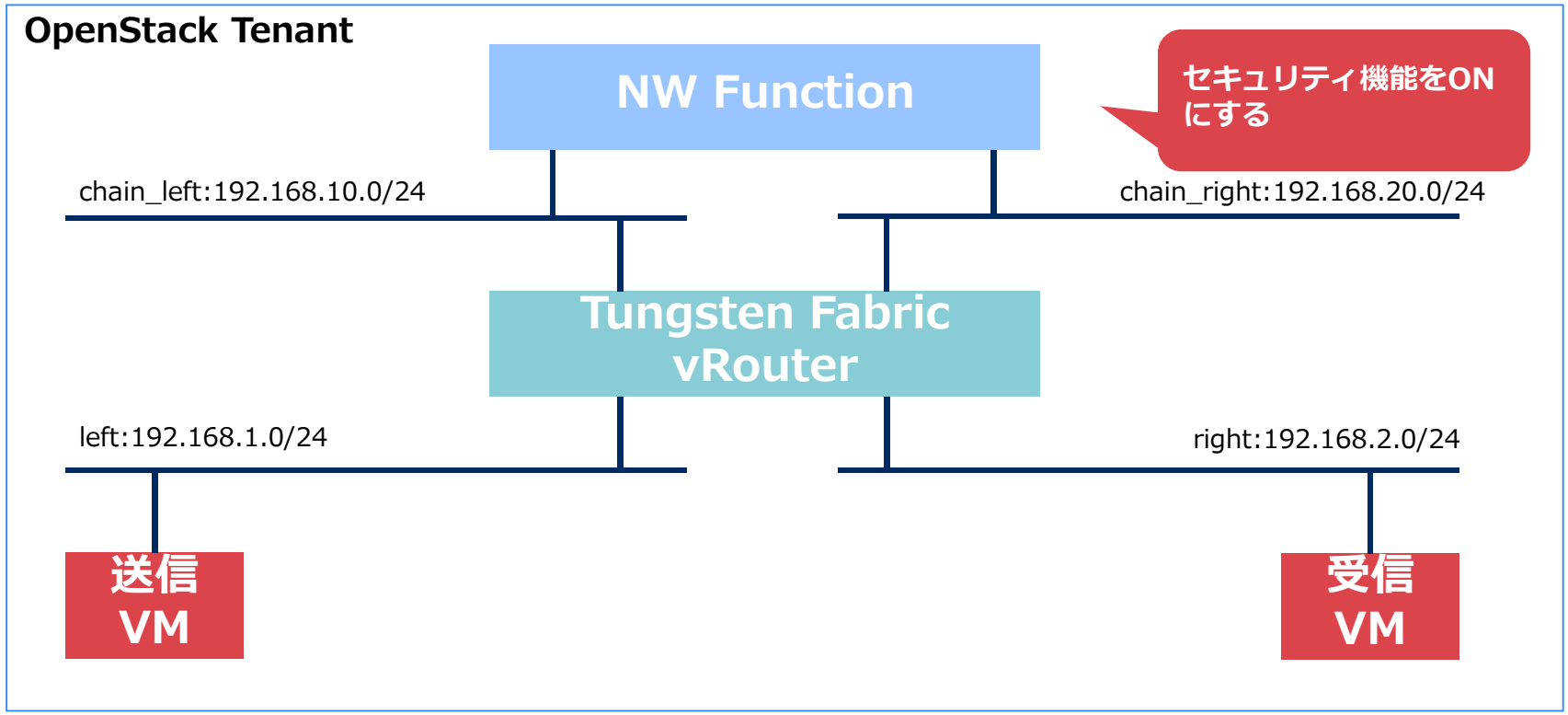
- OpenStackのテナント上に送信VM/受信VM/NW Functionを作成
- 送信VM/受信VM/NW Functionは、Tungsten Fabric vRouterにより接続される



検証方法

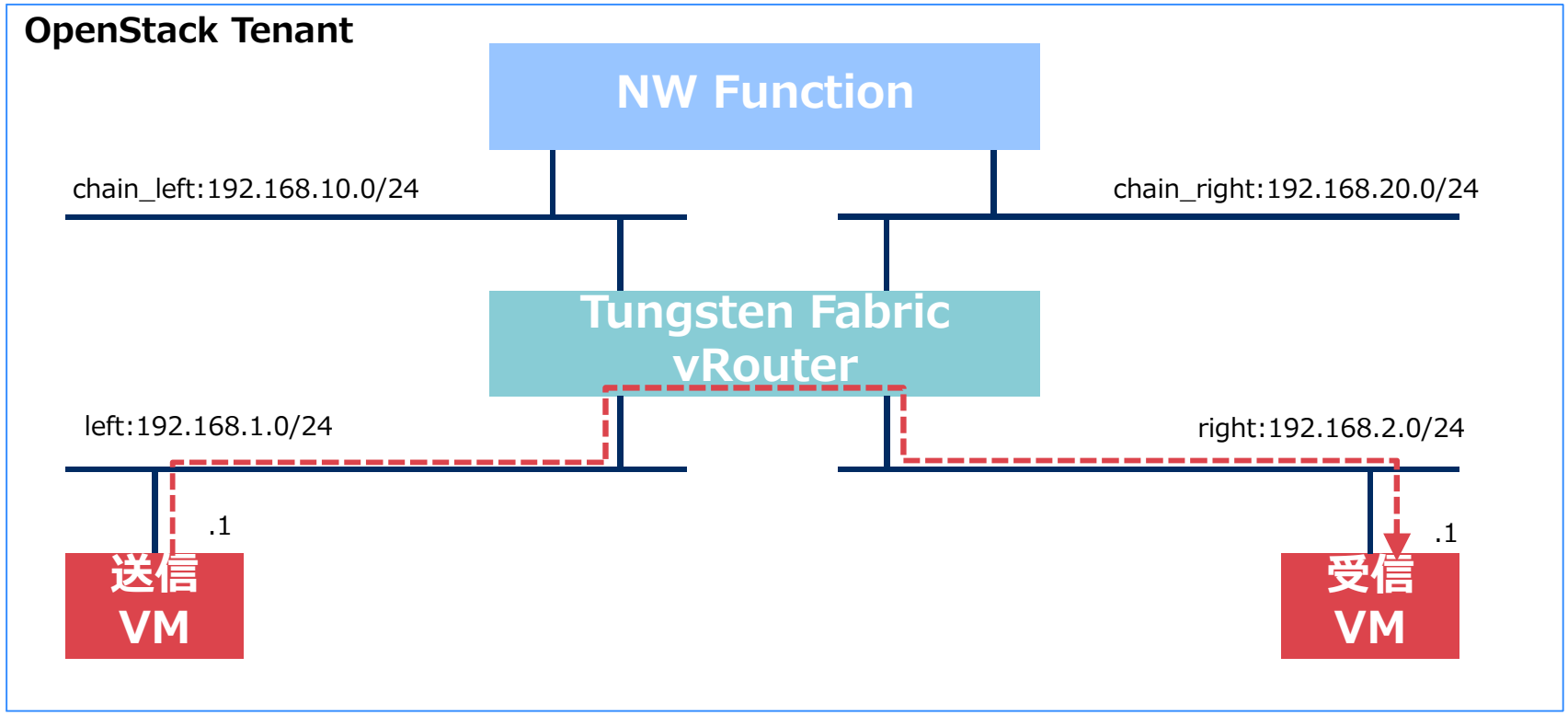
① NW Functionのセキュリティ機能を有効化する

- パケットキャプチャ、WAF機能、DDoS対策機能を有効化する
  - 今回は全てを一気に有効化する訳ではなく、単品の機能で有効化



検証方法

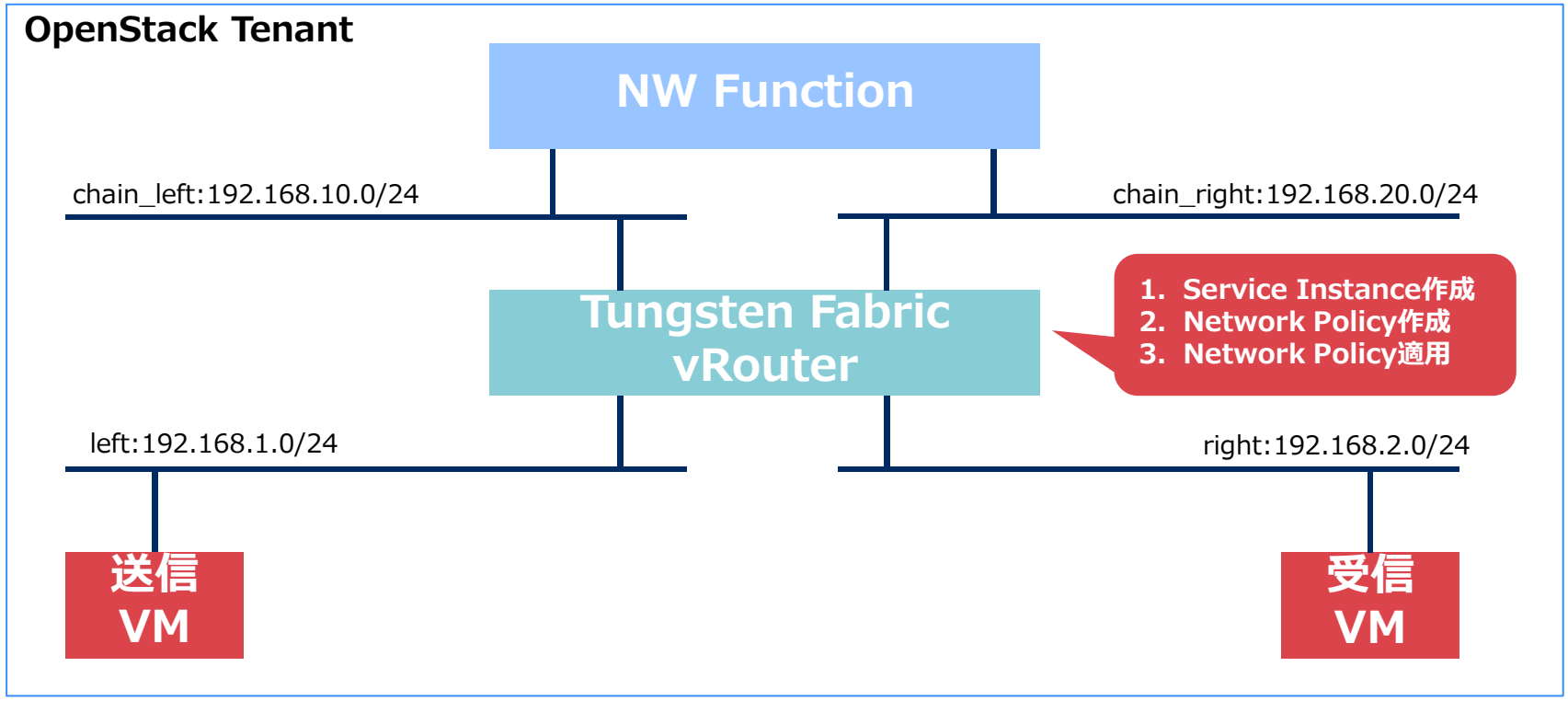
② 送信VM(192.168.1.1)から受信VM(192.168.2.1)へアクセスを実施



※Tungsten Fabric vRouterには、あらかじめ left から rightのルートが登録されている物とする

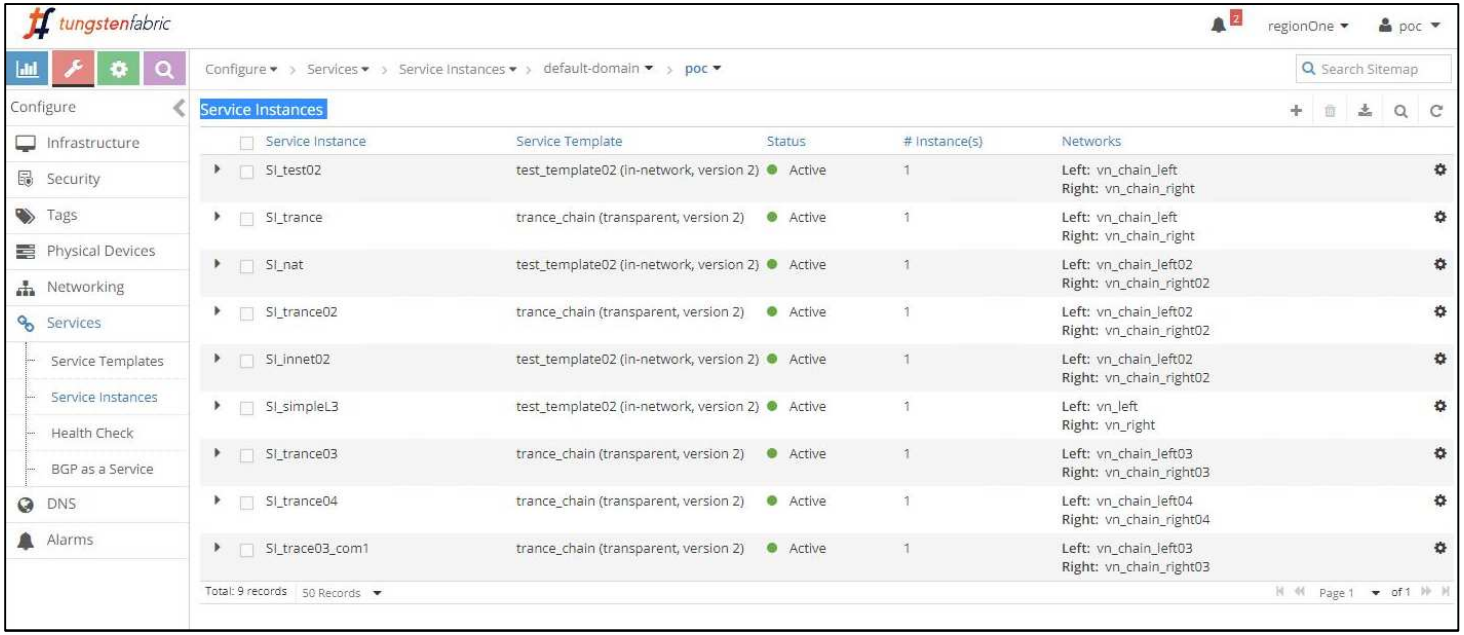
検証方法

③ Tungsten FabricにNW Functionを経由させる設定を投入して、経路を変更



【参考】Tungsten Fabricへの設定

1. NW FunctionをService Instanceとして登録



Name	任意の名称
Service Template	test_template(in-network, version 2)
Interface Type:left	chain_left
Interface Type:right	chain_right
Port Tuples:left	NW FunctionのVN_chain_leftに所属するIPアドレス
Port Tuples:right	NW FunctionのVN_chain_rightに所属するIPアドレス

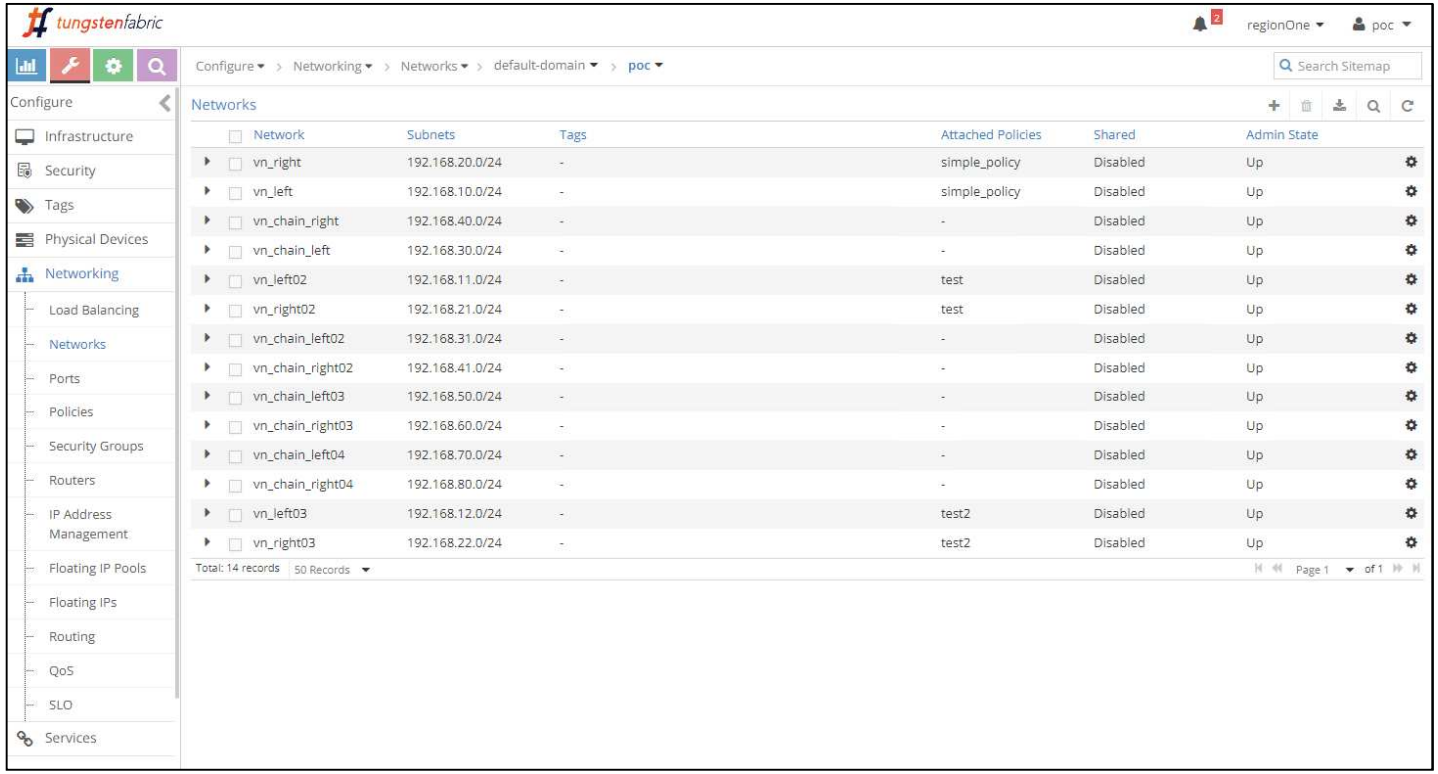
【参考】Tungsten Fabricへの設定

2. Network Policyを登録

		left ⇔ right		left ⇔ chain_left		left ⇔ chain_right		right ⇔ chain_right		right ⇔ chain_left	
Policy Rule	**Action**	PASS		PASS		PASS		PASS		PASS	
	Protocol	ANY		ANY		ANY		ANY		ANY	
	Source	left	right	left	chain_left	left	chain_right	right	chain_right	right	chain_left
	Ports	ANY		ANY		ANY		ANY		ANY	
	Destination	right	left	chain_left	left	chain_right	left	chain_right	right	chain_left	right
	Ports	ANY		ANY		ANY		ANY		ANY	
	Log	off		off		off		off		off	
	Service	on		off		on		off		on	
	Mirror	off		off		off		off		off	
	QoS	off		off		off		off		off	
	Service Instance	NW Function		-		NW Function		-		NW Function	

【参考】Tungsten Fabricへの設定

3. Network Policyを対象となるネットワークへ適用

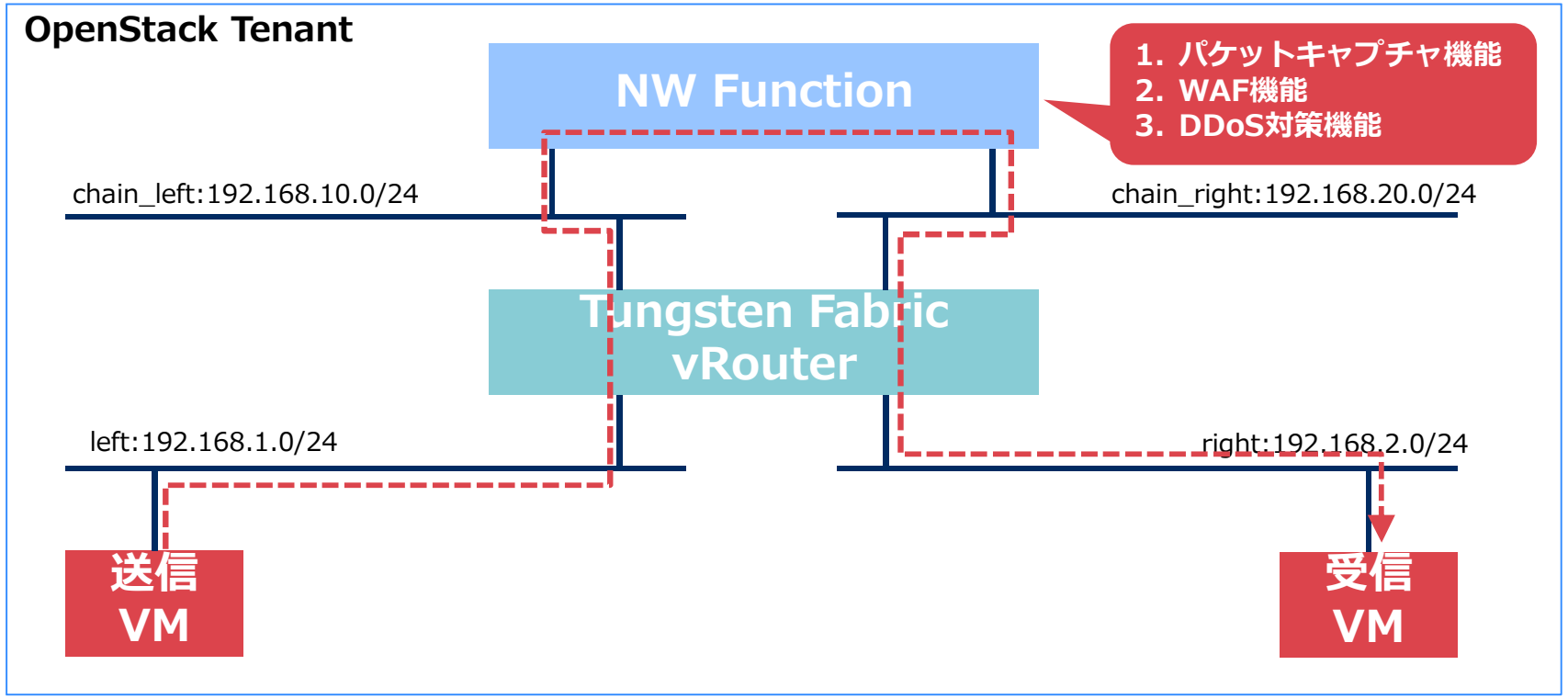


- パケットが経由する left / right / chain_left / chain_right の4つのネットワークへ適用

検証方法

④　受信VMでNW Functionのセキュリティ機能が動作している事を確認

1. パケットキャプチャ機能：受信VM宛のパケットをキャプチャできるか
2. WAF機能：受信VMへの不正アクセスをブロックできるか
3. DDoS対策機能：受信VMへのDDoS攻撃をブロックできるか



検証結果

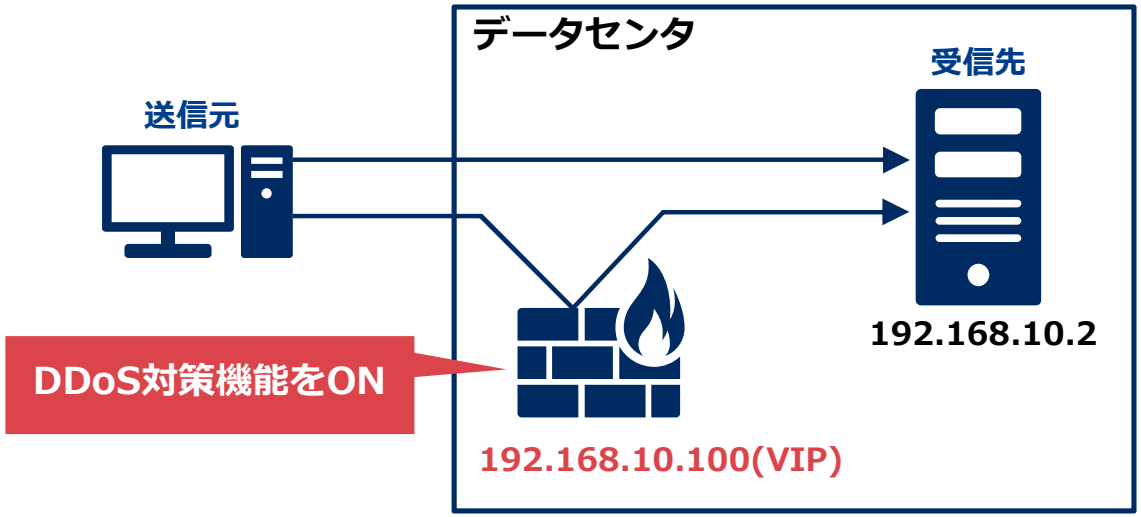
それぞれの仮説に対して、一定の効果と課題を抽出できた

#	仮説	検証結果
1	ユーザの運用が楽になる	•NWポリシーを設定するだけで、サービスチェイニング可能
		•ユーザが送信先を変更するケースがある【事象(1)】
2	クラウド事業者側の運用者の負荷を軽減できる	•これまで運用者が実施していたパケットキャプチャをサービスチェイニングで挟み込む事に成功
		•キャプチャしたパケットの詳細情報が消失する【事象(2)】
3	攻撃を受けたときの緊急回避を素早く行える	•WAFおよびDDoS機能をサービスチェイニングで挟み込む事に成功
		•マルチテナントを考えると不十分【事象(3)】

検証により発見した事象(1)

仮説1:ユーザの運用が楽になる

- 今回使用したNW Function(某社ロードバランサ)は、**セキュリティ機能を使用するには、ロードバランス設定を必ず有効化し、VIPを設定しないといけない仕様だった。**



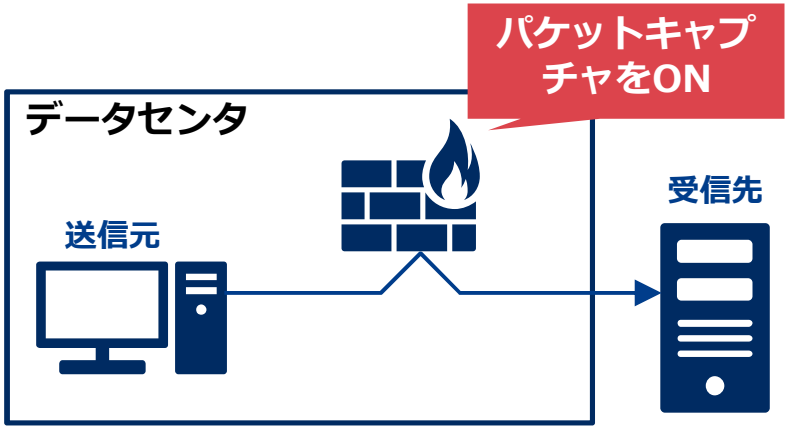
- サービスチェイニングOFF時の宛先：192.168.10.2
- サービスチェイニングON　時の宛先　：192.168.10.100

- 送信元から、この変化を意識させない為には、DNSのレコード更新やNATなどの対策が必要になる。
- 全自動のサービスチェイニングを提供する場合に工夫が必要。

検証により発見した事象(2)

仮説2:クラウド事業者側の運用者の負荷を軽減できる

- 今回使用した某社パケットキャプチャ用機器では、捕捉したパケットが**UDPに変換され「送信アドレス」や「宛先アドレス」が削除された。**



パケットキャプチャ結果

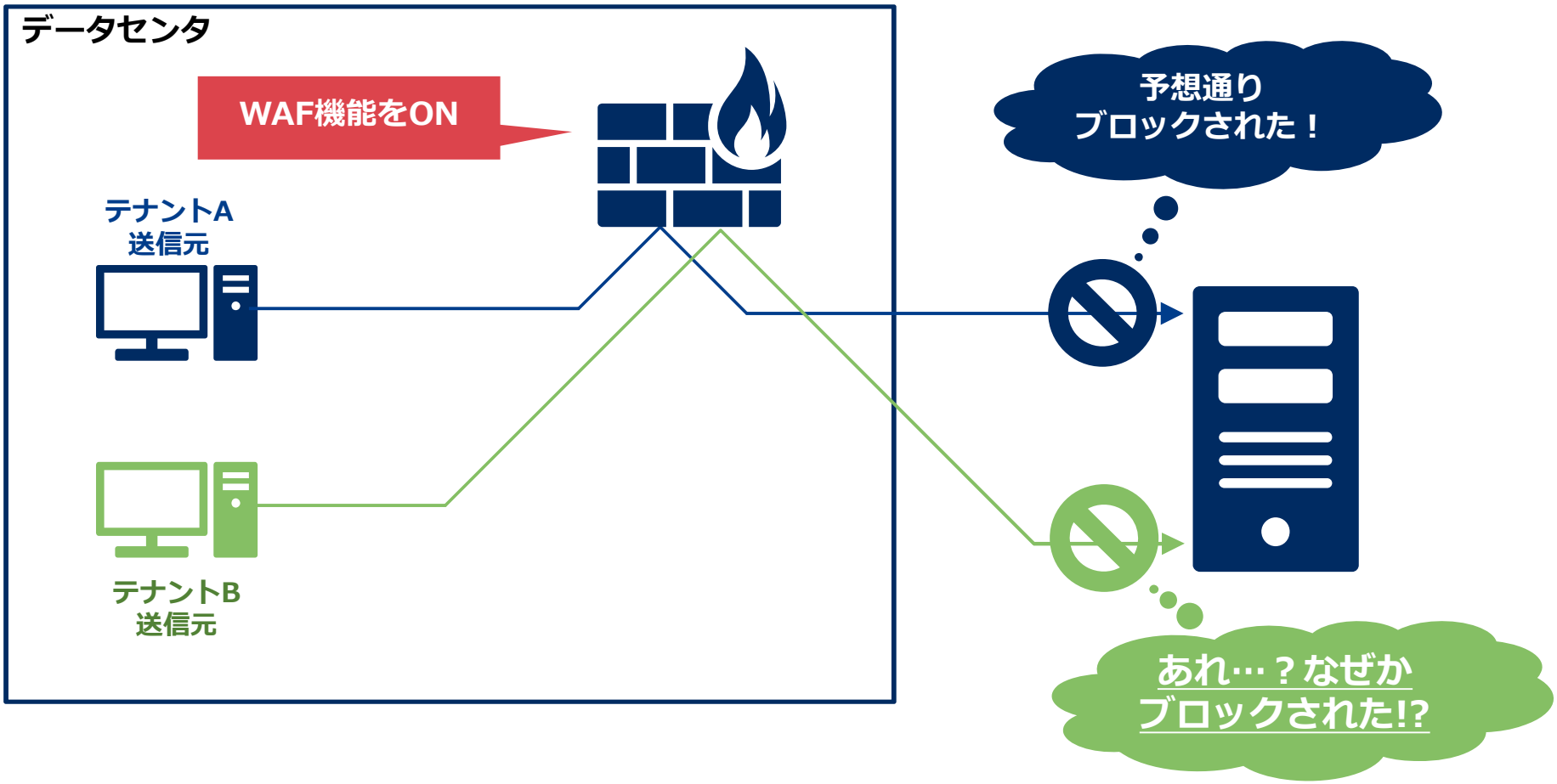
- Tungsten FabricのJuniper Headerフラグを「disabled」にすることで問題が解決した。

このフラグは一体何者でしょうか？

検証により発見した事象(3)

仮説3:攻撃を受けたときの緊急回避を素早く行える

- 今回使用したNW Function(某社ロードバランサ)は、**1つのネットワーク機器に複数テナント用の設定が投入できなかった。**
- **テナントAには有効な設定が、必ずしもテナントBに有効とは限らない。**



ディスカッションしたいこと

ディスカッションしたいこと

仮説1に対するテーマ

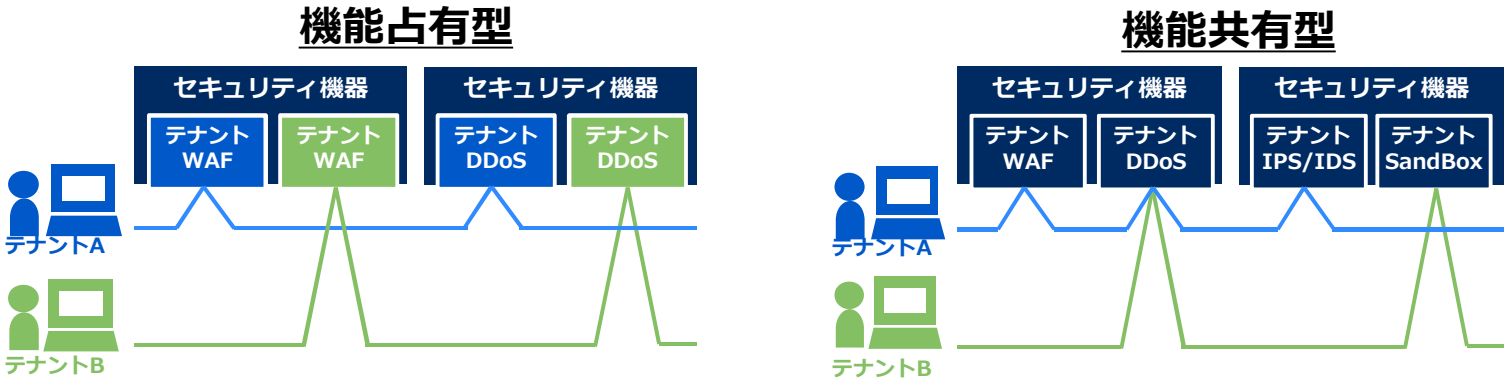
- DNSやNAT以外にも、宛先の変化を吸収できる様な方法は無いでしょうか？
  （あるいは宛先が変わらないネットワーク機器をご存知でしょうか？）

仮説2に対するテーマ

- 今回はパケットキャプチャを試してみましたが、サービスチェイニングによって運用者の負荷軽減に繋がる機能とは何でしょうか？

仮説3に対するテーマ

- マルチテナントを考えた時、共通機能だけでなく、テナント毎の設定をサービスチェイニングによって提供するには、どの様な方法が考えられますか？




Orchestrating a brighter world
NEC