



# OpenStack Tackerによる free5GCのLCM操作デモの紹介

NTT ネットワークイノベーションセンタ (NIC)

小川 賢太郎

2021年7月13日 Open Mobile Network Infra Meetup #3

## 小川 賢太郎 (Slack: @Kentaro Ogawa)

### 2003年～

- NTT入社 ネットワークサービスシステム研究所(NS研)配属
- 汎用HWやオープンSWで構築するオープンアーキテクチャサーバ/ルータの研究開発に従事
- 並行してITU-TやIETF等での標準化活動や海外のキャリア/ベンダ/大学との共同実験を推進

### 2015年～

- 徐々にNFV技術領域にシフト
- キャリア向け仮想化サーバプラットフォームの研究開発に従事

### 2020年～

- キャリア仮想化基盤で運用可能なGeneric-VNFMとしてのOpenStack Tackerの開発に参画
- Tacker普及に向けたコミュニティの仲間作りを推進

## 【経緯①】Tacker普及活動用のデモ環境構築

- ⇒ Tackerに興味を持った人にいつでも見せられるデモを用意しよう！
- ⇒ TackerからLCMを制御するアプリは？手頃な5GC系のOSSはないか？
- ⇒ Meetup #1で同研究所の上酔尾さんのfree5GCに関する発表を見る
- ⇒ 協力を仰ぎfree5GCの資材や運用ノウハウの提供を受けてデモ完成！
- ⇒ OpenStackの公式YouTubeチャンネルにデモビデオをアップ  
<https://www.youtube.com/watch?v=JK0Jp9yA1Kw>

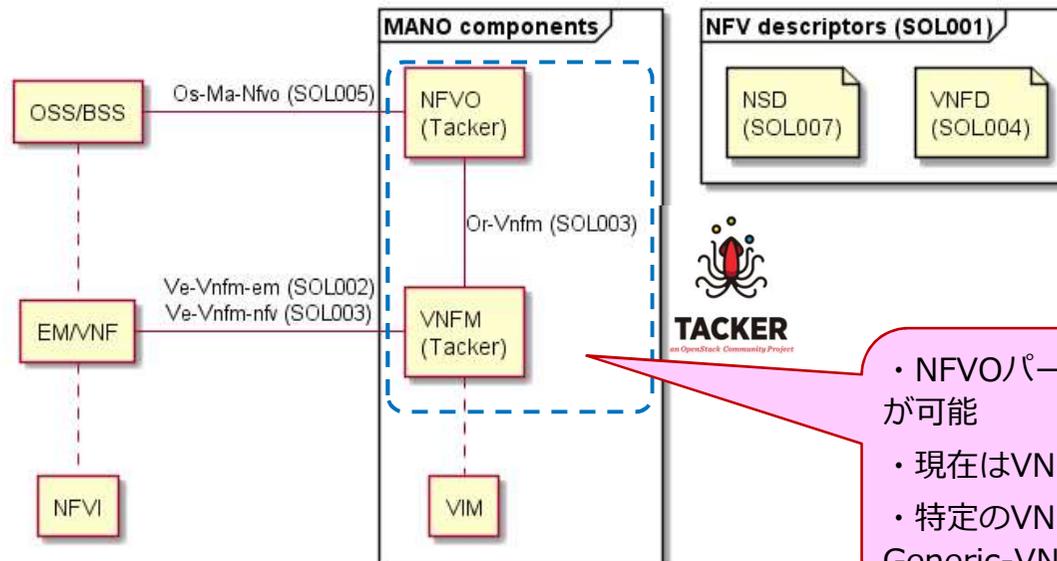
## 【経緯②】omni-jpへのTacker紹介

- ⇒ Meetup #2でKDDI 辻さんのTacker紹介発表を見る
- ⇒ 流れに乗ってTacker+free5GCのデモを紹介しよう！

## 【発表内容】

- Tackerによりfree5GCのLCM制御を行うケースの機能構成/NW構成例
- free5GCのLCM制御の動作概要
- デモ環境構築を通して見えたfree5GCの改善点

- TackerはMANOのコンポーネントとして振る舞い、VNFを動作させるための仮想化基盤を提供
  - キャリアNWのインフラ基盤として要求される各種機能のサポート
    - VNFのライフサイクル管理やパッケージ管理機能
    - 物理レイヤを考慮した性能最適化 (anti-affinity, SR-IOVなど)
  - ETSI SOL準拠のREST API
  - OpenStack VIMおよびKubernetes VIM



- NFVOパートとVNFMパートの分離が可能
- 現在はVNFMパートの開発に注力
- 特定のVNFをターゲットとはせず、Generic-VNFMを目指している

# G-VNFMとしてのTackerのメリット

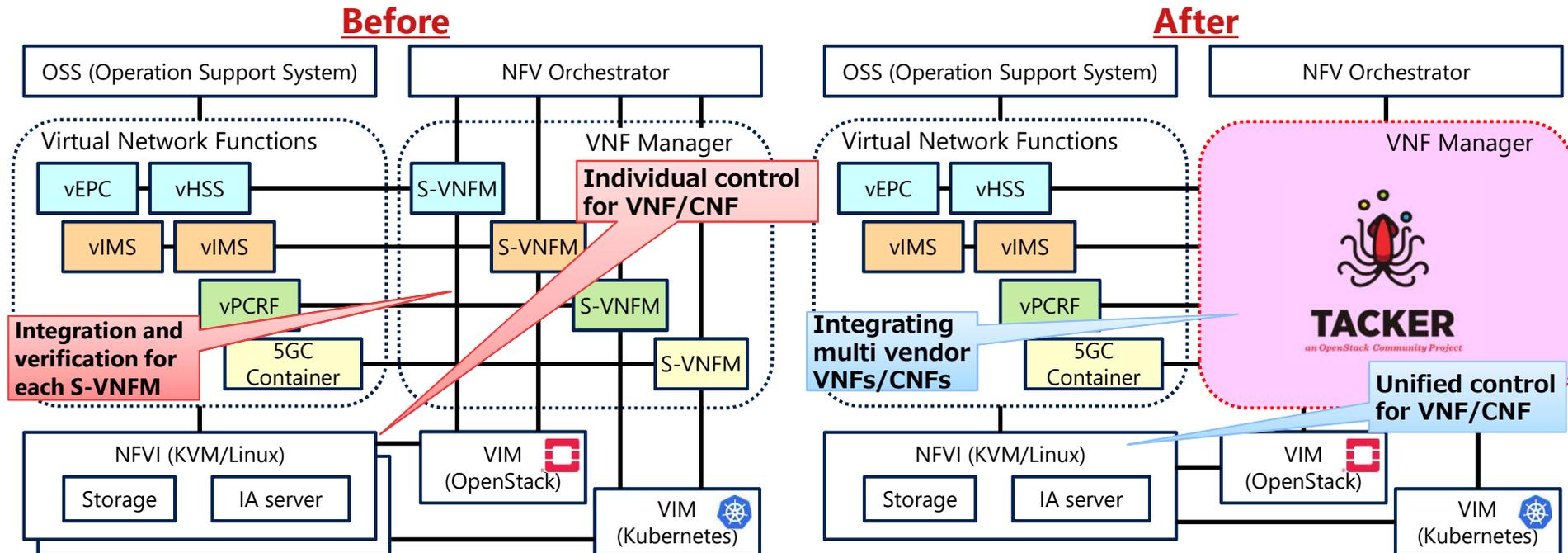


【課題】 VNFベンダはVNFM(Specific-VNFM)をバインドして提供するケースが多く、VNF増加に伴い、結合試験パターンが増加、障害時の切分け長期化、運用者の負担増。

⇒VNFMは機能面での差別化要素が少なくETSI標準化も進んでいるので、ETSI標準に準拠したTackerに統一することで導入/運用コストを削減。

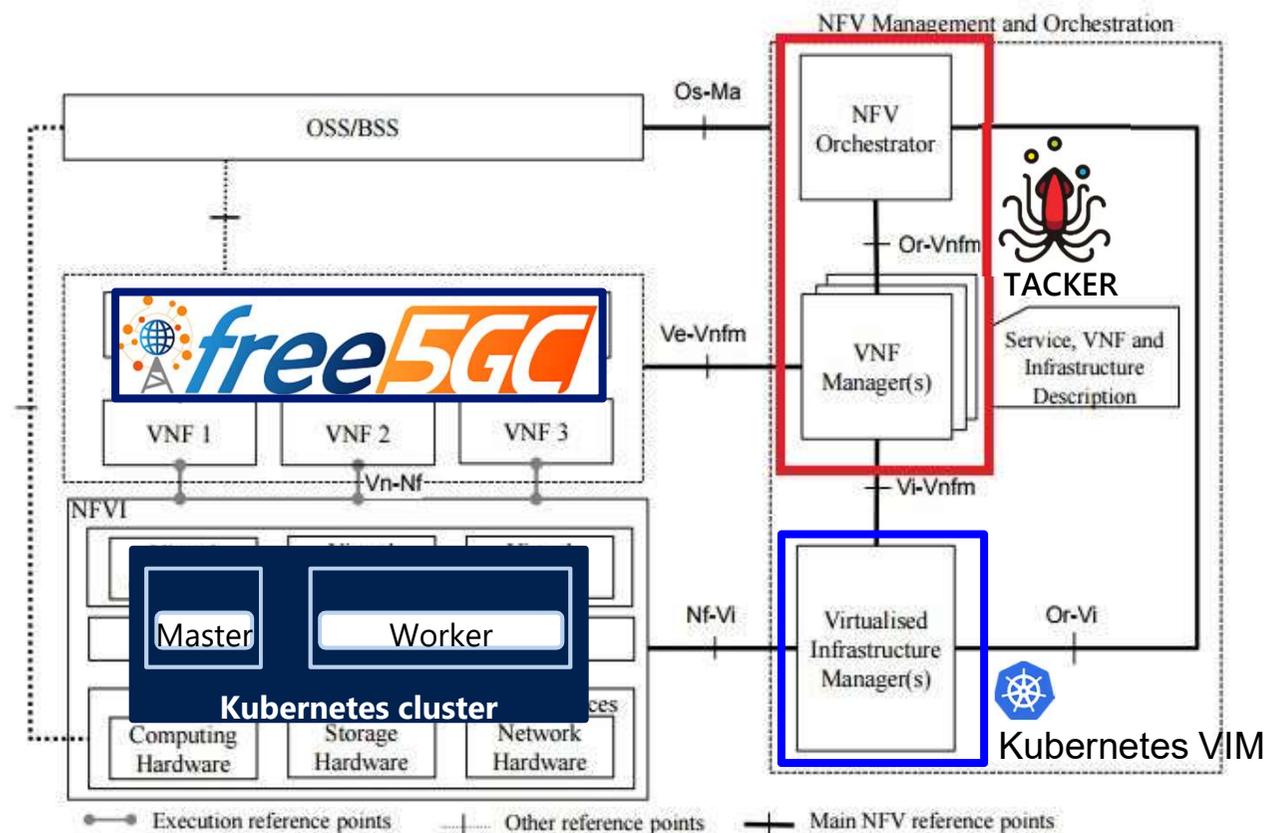
【課題】 NFVの次のフェーズとしてコンテナ化が進行しているが、過渡期においてはVM/コンテナ基盤を個別に運用する負担増。

⇒TackerによりVM/コンテナ基盤を統一オペレーションで制御、基盤リソース共有。



# デモの目的

本デモンストレーションでは、OpenStack TackerによりKubernetes cluster上にコンテナアプリ (free5GC) がデプロイされ、ライフサイクル管理 (Instantiate, Scale, Heal) が制御される様子を見せる。



## 1. 構成

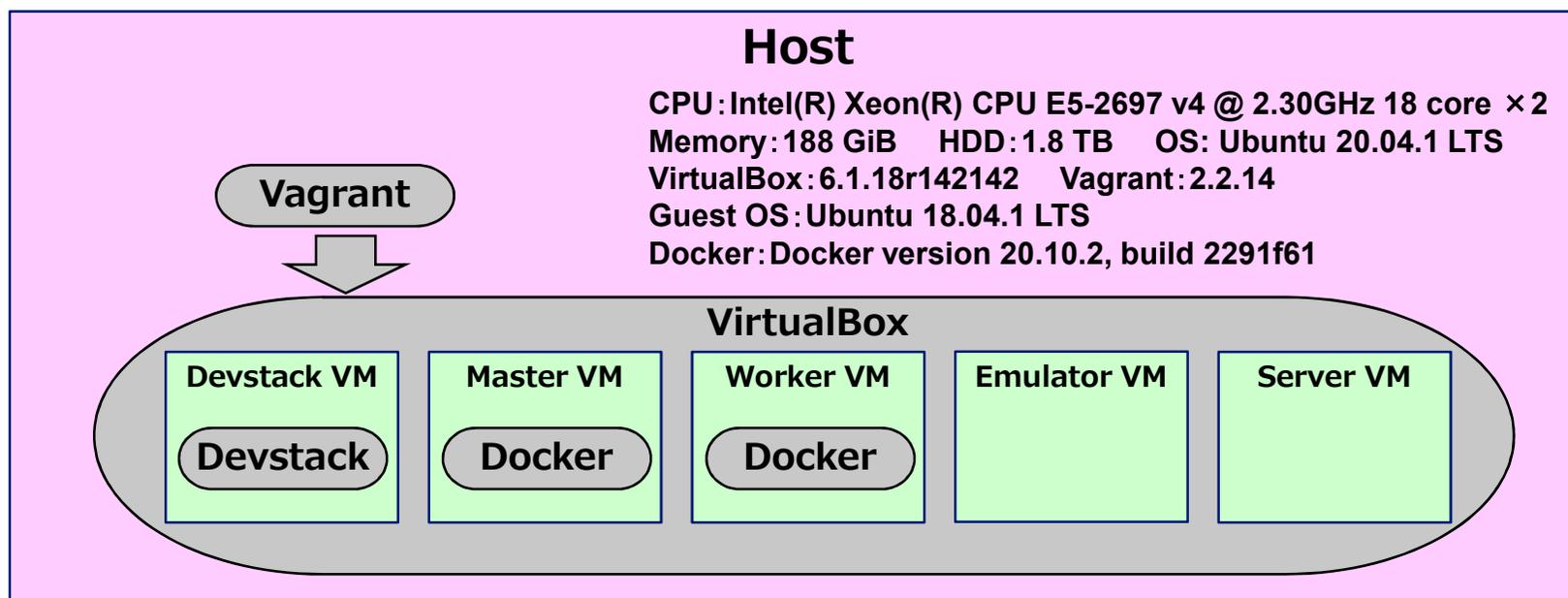
**Tacker** (W 版) とコンテナ上のアプリ free5GC

## 2. 動作

**Tacker** による free5GC の操作

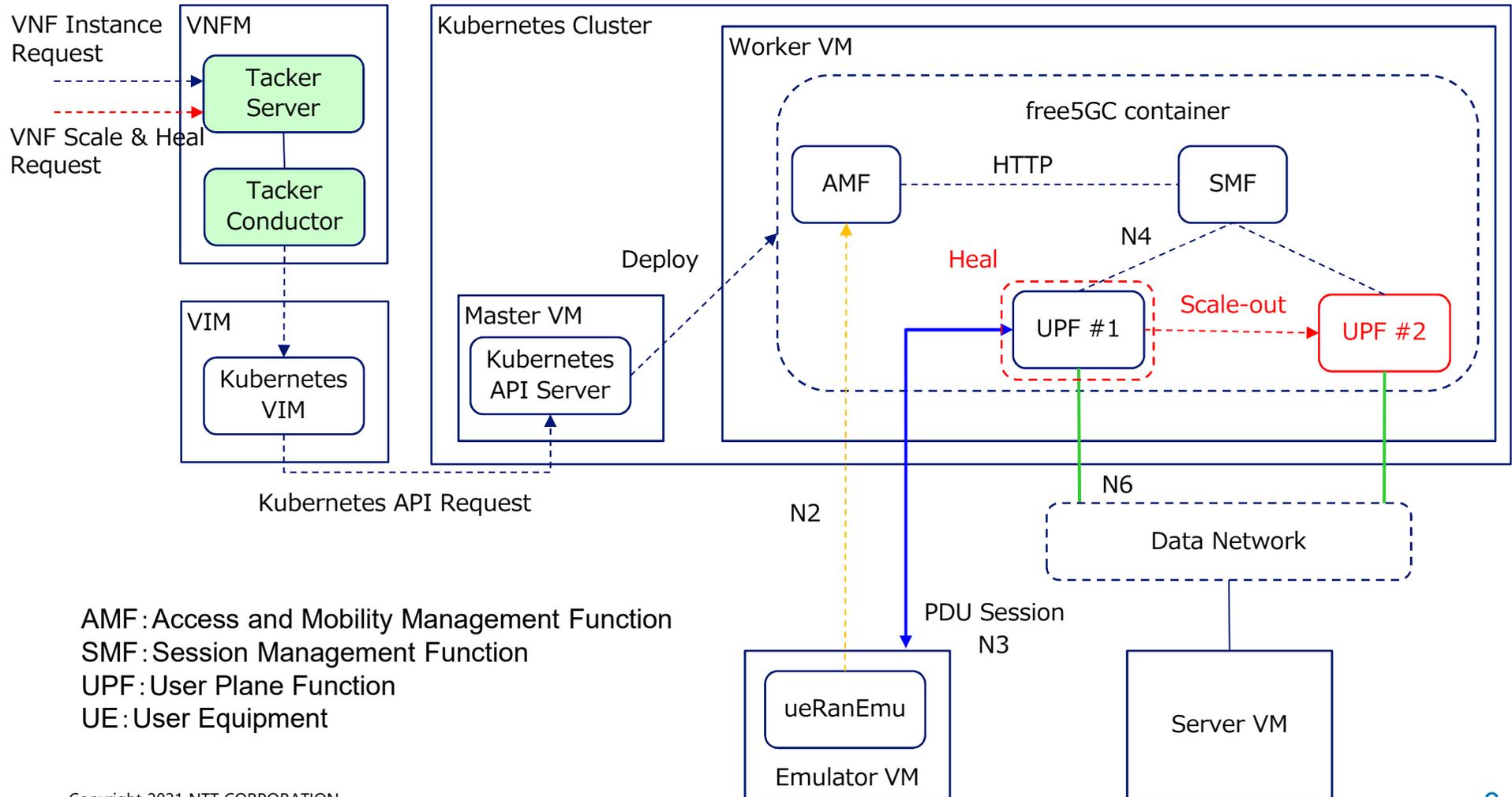
- A) free5GC の **Instantiate**
- B) free5GC の UPF プロセスの **Scale**
- C) free5GC の UPF プロセスの **Heal**

- 1台のサーバ上にデモ環境を構築
  - 仮想化ソフトはVirtualBoxを使用
  - フロントエンドツールとしてVagrantを使用し、5つのVMを構築
  - 1つのVMにDevstackを使用しOpenStack/Tackerを構築
  - 2つのVMにDockerを使用しfree5GC用のKubernetes clusterを構築
  - 残り2つのVMはUE/RANエミュレータおよび外部Webサーバに使用



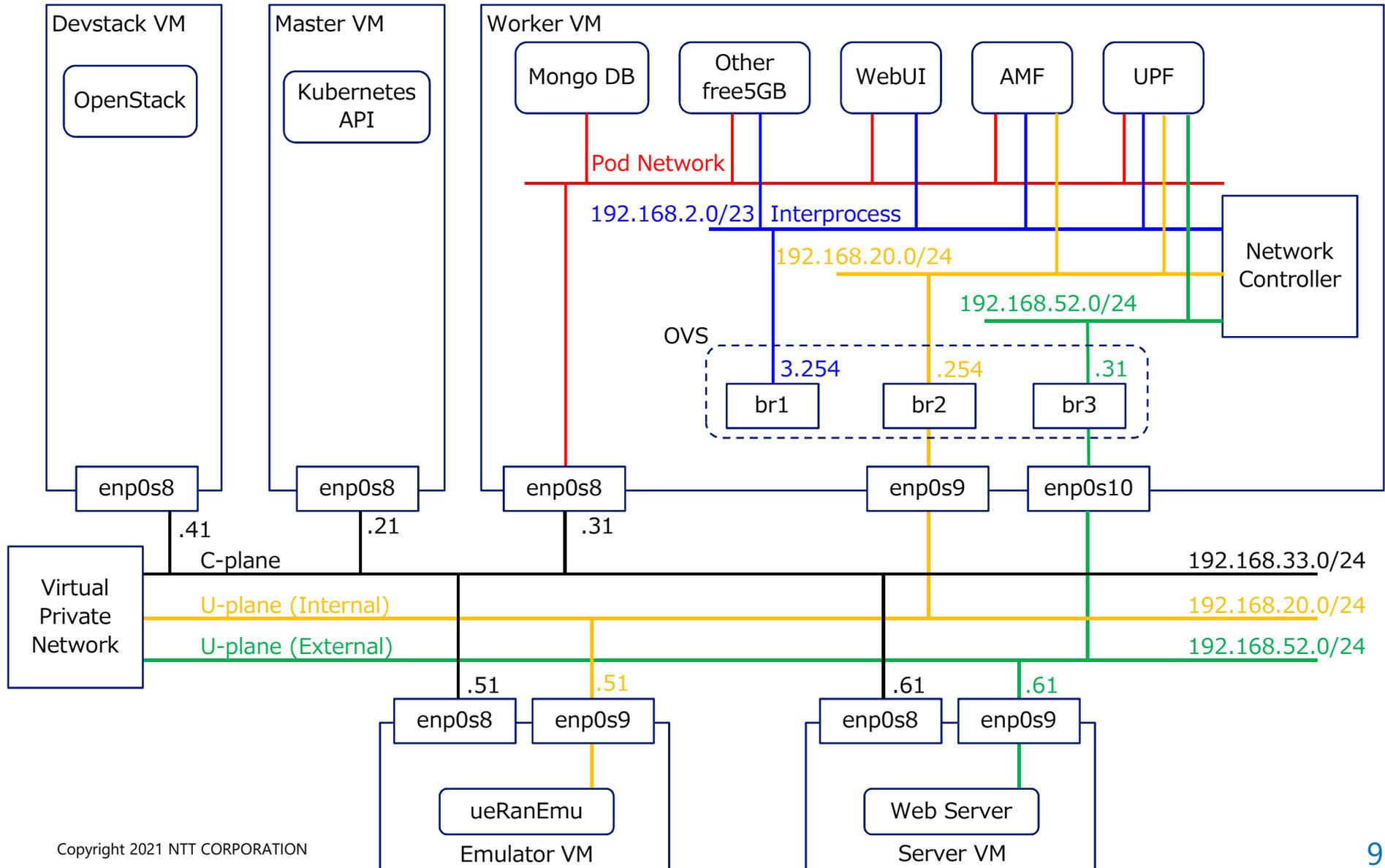
# 機能構成図

デモ環境の機能構成を下記に示す。TackerからのリクエストによりKubernetes cluster上でfree5GCプロセスの生成、制御が行われる。  
UEエミュレータはUPF経由で外部サーバと通信する。



# 論理ネットワーク構成図

デモ環境の論理ネットワーク構成を下記に示す。



# Tacker による free5GC の操作



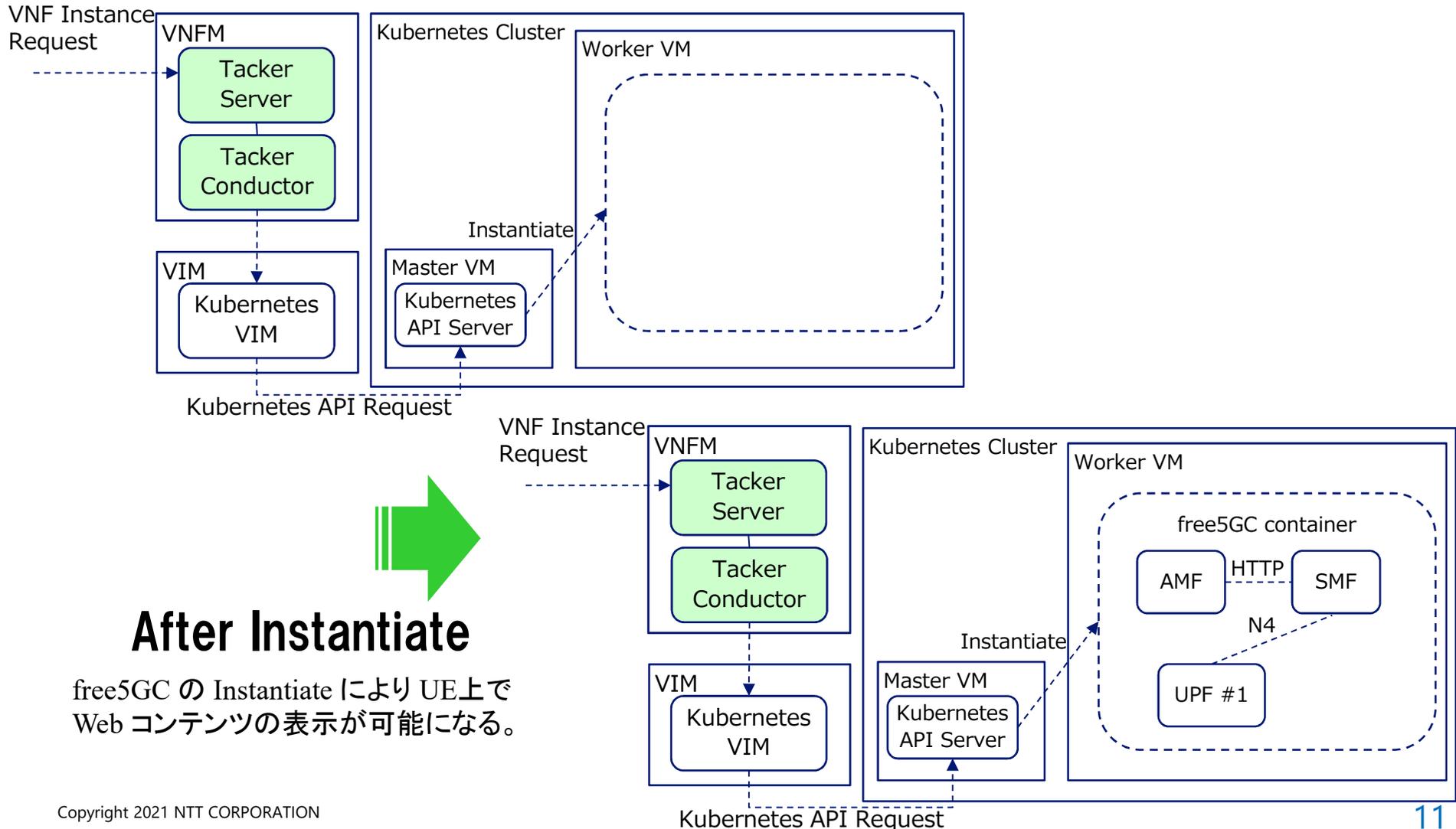
**Tacker** のコマンドライン実行により free5GC の操作を行う

- A) free5GC の **Instantiate**
- B) free5GC の UPF プロセスの **Scale**
- C) free5GC の UPF プロセスの **Heal**

# Tacker 操作 A) free5GC の Instantiate (1/2)

Tacker 操作により free5GC の各プロセスを **Instantiate** する。

**Before Instantiate** この時点では UE への Web サーバのレスポンスが 404 not found である。



# Tacker 操作 A) free5GC の Instantiate (2/2)

Tacker 操作により free5GC 各プロセスを **Instantiate** する。

(1) VNF Create を実行する

Tacker 操作画面

(2) VNF Instantiate を実行する

```
$ openstack vnflcm create a8bb9888-7c63-4293-a95e-2ef102ae1d3d
```

Field	Value
ID	4e521b9b-292e-44ab-89bd-c9cd135fac9c
Instantiation State	NOT_INSTANTIATED
Links	{ "self": { "href": "/vnflcm/v1/vnf_instances/4e521b9b-292e-44ab-89bd-c9cd135fac9c" }, "instantiate": { "href": "/vnflcm/v1/vnf_instances/4e521b9b-292e-44ab-89bd-c9cd135fac9c/instantiate" } }
VNF Instance Description	None
VNF Instance Name	vnf-4e521b9b-292e-44ab-89bd-c9cd135fac9c
VNF Product Name	Sample VNF
VNF Provider	Company
VNF Software Version	1.0
VNFD ID	a8bb9888-7c63-4293-a95e-2ef102ae1d3d
VNFD Version	1.0
vnfPkgId	

```
$ openstack vnflcm instantiate 4e521b9b-292e-44ab-89bd-c9cd135fac9c /var/tmp/cnf_free5gc/inst_param.yaml  
Instantiate request for VNF Instance 4e521b9b-292e-44ab-89bd-c9cd135fac9c has been accepted.
```

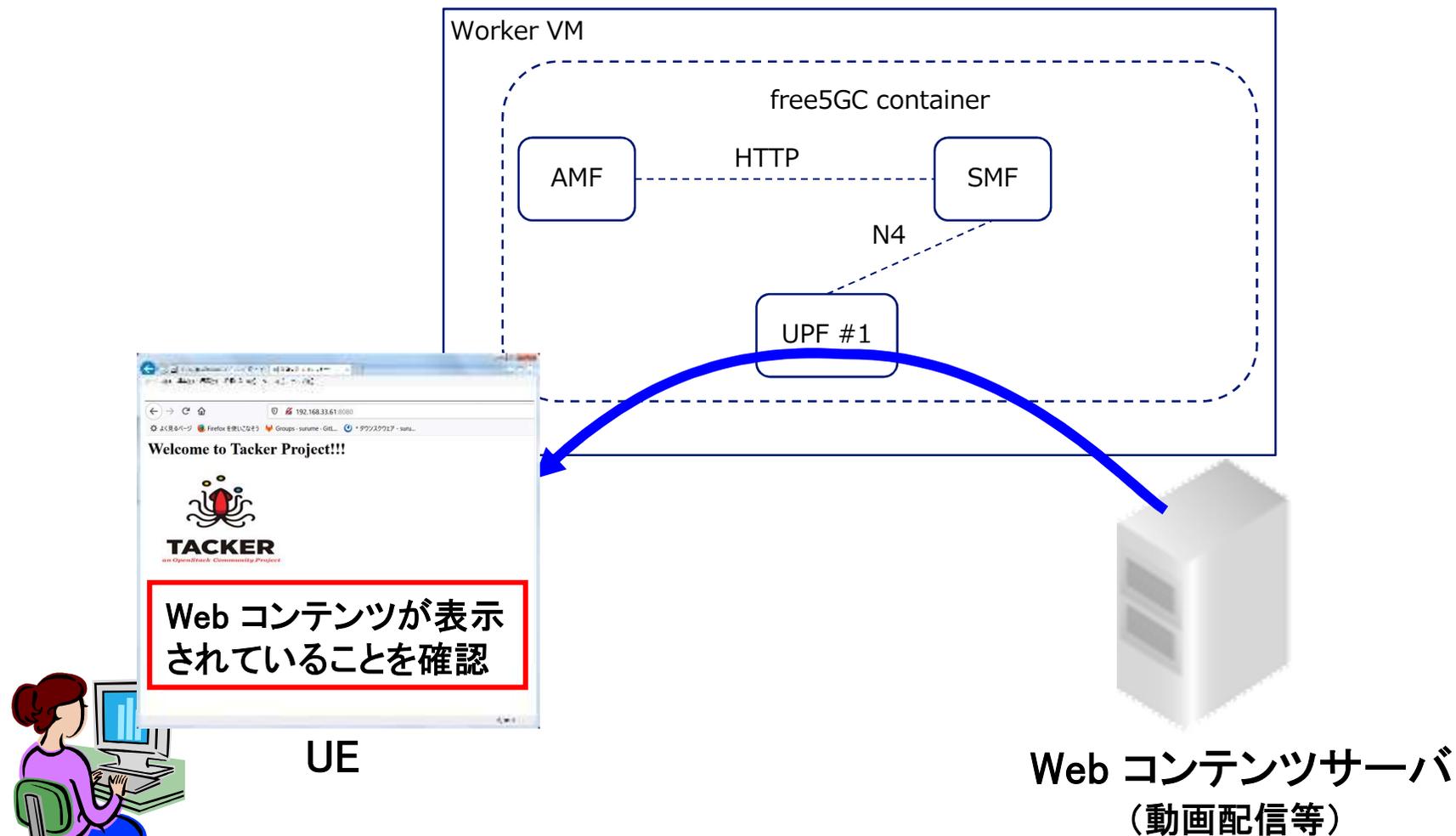
(3) 正常性を確認する  
VNF に付与された  
ID を確認

```
$ openstack vnflcm show aa17a2ae-ddd1-4a3a-8814-ab0f44fc3140
```

Field	Value
ID	aa17a2ae-ddd1-4a3a-8814-ab0f44fc3140
Instantiated Vnf Info	{ "flavourId": "simple", "vnfState": "STARTED", "scaleStatus": { "aspectId": "upf_aspect", "scaleLevel": 0 } }

# free5GC の動作正常性の確認

Tacker から Instantiate された free5GC が正常に動作していることを確認するため、UPF経由で取得したWeb コンテンツをUE上で表示させる。



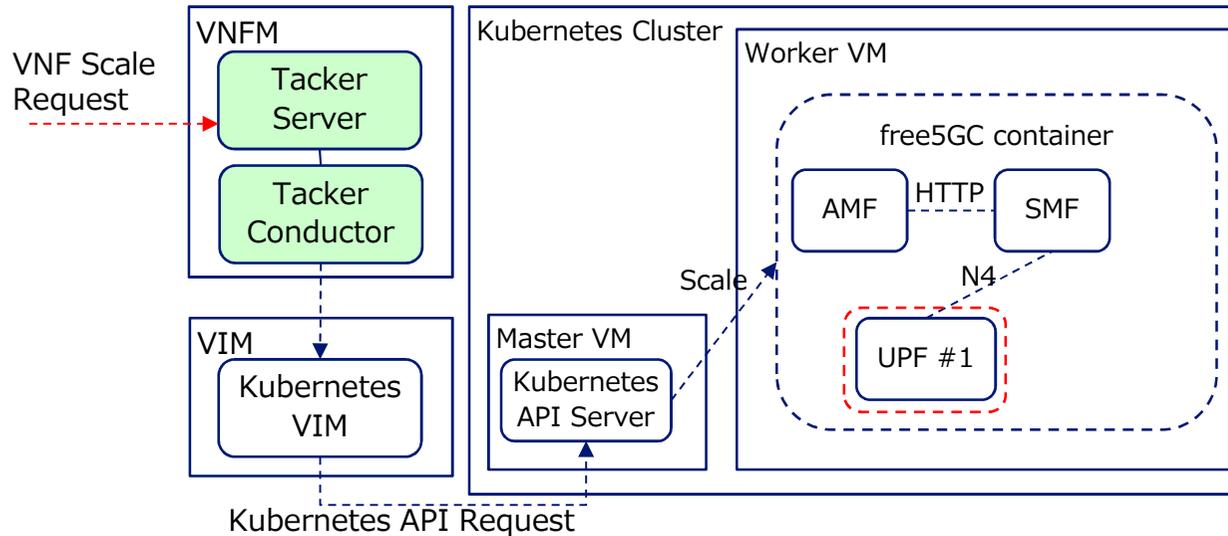
# Tacker 操作 B) free5GC UPF の Scale (1/2)



Tacker 操作により free5GC の UPF プロセスを Scale する。

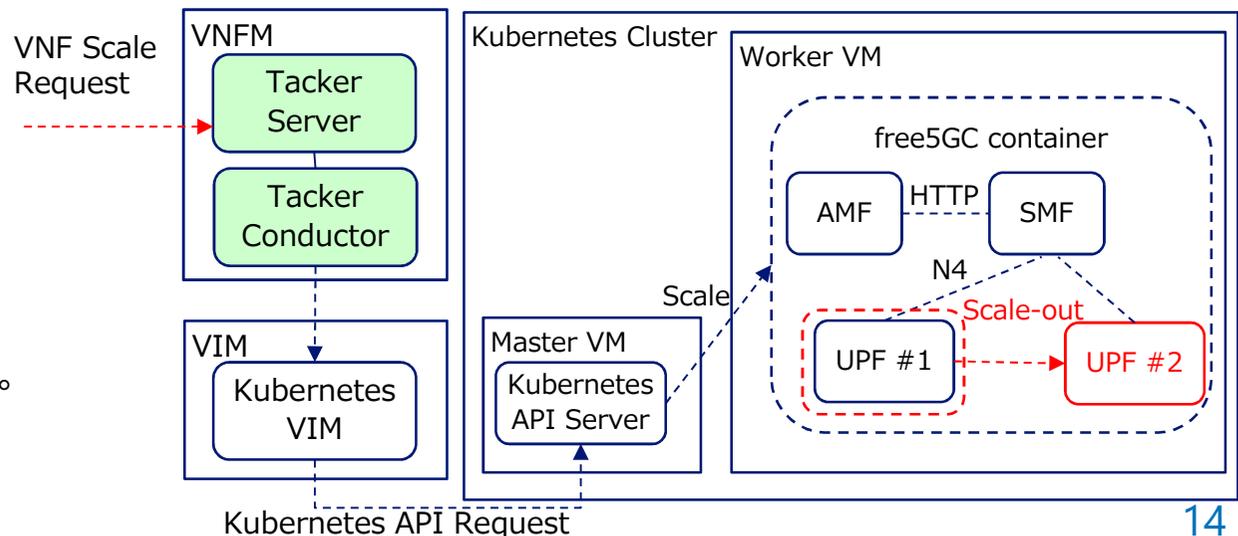
## Before Scale

この時点では UPF プロセスが一つである。



## After Scale

Scale により UPF プロセスが二つになる。



# Tacker 操作 B) free5GC UPF の Scale (2/2)



**Tacker 操作により free5GC の UPF プロセスを Scale する。**

Tacker 操作画面

(1) Scale 実行前) UPF プロセスが一つである

```
$ kubectl get deploy,pods -o wide
NAME READY UP-TO-DATE AVAILABLE AGE
deployment.apps/free5gc-amf-deployment 1/1 1 1 76s
deployment.apps/free5gc-ausf-deployment 1/1 1 1 77s
deployment.apps/free5gc-mongod 1/1 1 1 76s
deployment.apps/free5gc-nrf-deployment 1/1 1 1 76s
deployment.apps/free5gc-nssf-deployment 1/1 1 1 76s
deployment.apps/free5gc-pcf-deployment 1/1 1 1 76s
deployment.apps/free5gc-smf-deployment 1/1 1 1 76s
deployment.apps/free5gc-udm-deployment 1/1 1 1 77s
deployment.apps/free5gc-udr-deployment 1/1 1 1 76s
deployment.apps/free5gc-upf-deployment 1/1 1 1 76s
deployment.apps/free5gc-webui-deployment 1/1 1 1 77s

NAME READY STATUS RESTARTS AGE
pod/free5gc-amf-deployment-859965844-5glpx 1/1 Running 0 76s
pod/free5gc-ausf-deployment-846997ccdf-hacjq 1/1 Running 0 76s
pod/free5gc-mongodf-ddd76dd4c-q4p19 1/1 Running 0 76s
pod/free5gc-nrf-deployment-6cbc59dfd4-hrj2w 1/1 Running 0 76s
pod/free5gc-nssf-deployment-db9df66dc-nvn25 1/1 Running 0 76s
pod/free5gc-pcf-deployment-54f5855c68-9b7gc 1/1 Running 0 76s
pod/free5gc-smf-deployment-59c8dcc986-zfvxb 1/1 Running 0 75s
pod/free5gc-udm-deployment-d987fbbfd-924sg 1/1 Running 0 76s
pod/free5gc-udr-deployment-5b697f4b85-b79t5 1/1 Running 0 76s
pod/free5gc-upf-deployment-98d84679-w4rnq 1/1 Running 0 76s
pod/free5gc-webui-deployment-614714f4b0-tnkto 1/1 Running 0 76s
```

(3) Scale 実行後) UPF プロセスが二つに増えている

新規UPFプロセスの動作時間が他より短い

```
$ openstack vnflcm scale --type SCALE_OUT --aspect-id upf_aspect aa17a2ae-ddd1-4a3a-8814-ab0f44fc3140
scale request for vnfl Instance aa17a2ae-ddd1-4a3a-8814-ab0f44fc3140 has been accepted.
```

(2) UPF の Scale を実行する

```
o wide READY UP-TO-DATE AVAILABLE AGE
-deployment 1/1 1 1 6m59s
f-deployment 1/1 1 1 7m
godb 1/1 1 1 6m59s
-deployment 1/1 1 1 6m59s
f-deployment 1/1 1 1 6m59s
-deployment 1/1 1 1 6m59s
r-deployment 1/1 1 1 7m
-deployment 1/1 1 1 6m59s
-deployment 1/1 1 1 6m59s
ui-deployment 1/1 1 1 7m

READY STATUS RESTARTS AGE
859965844-5glpx 1/1 Running 0 6m59s
-846997ccdf-hacjq 1/1 Running 0 6m59s
-ddd76dd4c-q4p19 1/1 Running 0 6m59s
6cbc59dfd4-hrj2w 1/1 Running 0 6m59s
db9df66dc-nvn25 1/1 Running 0 6m59s
54f5855c68-9b7gc 1/1 Running 0 6m59s
59c8dcc986-zfvxb 1/1 Running 0 6m59s
d987fbbfd-924sg 1/1 Running 0 6m59s
5b697f4b85-b79t5 1/1 Running 0 6m59s
98d84679-w4rnq 1/1 Running 0 23s
-846997ccdf-hacjq 1/1 Running 0 6m59s
-ddd76dd4c-q4p19 1/1 Running 0 6m59s
6cbc59dfd4-hrj2w 1/1 Running 0 6m59s
db9df66dc-nvn25 1/1 Running 0 6m59s
54f5855c68-9b7gc 1/1 Running 0 6m59s
59c8dcc986-zfvxb 1/1 Running 0 6m59s
d987fbbfd-924sg 1/1 Running 0 6m59s
5b697f4b85-b79t5 1/1 Running 0 6m59s
98d84679-rkkxh 1/1 Running 0 23s
98d84679-w4rnq 1/1 Running 0 6m59s
614714f4b0-tnkto 1/1 Running 0 6m59s
```

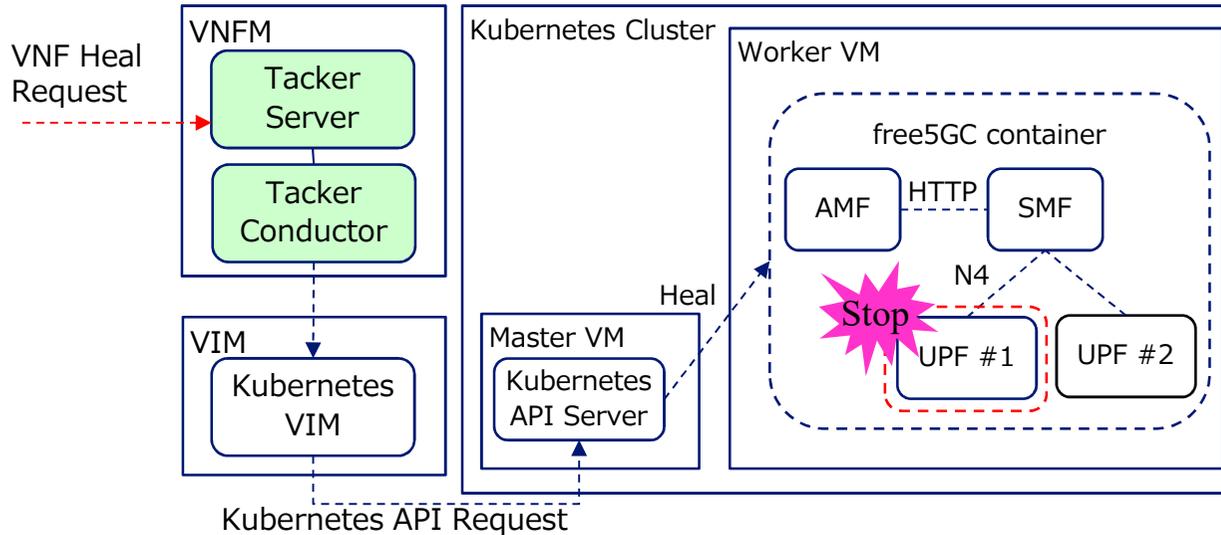
# Tacker 操作 C) free5GC UPF の Heal (1/2)



Tacker 操作により free5GC の UPF プロセスを Heal する。

## Before Heal

UPF プロセス停止によりUE への Web サーバのレスポンスが 404 not found になる。

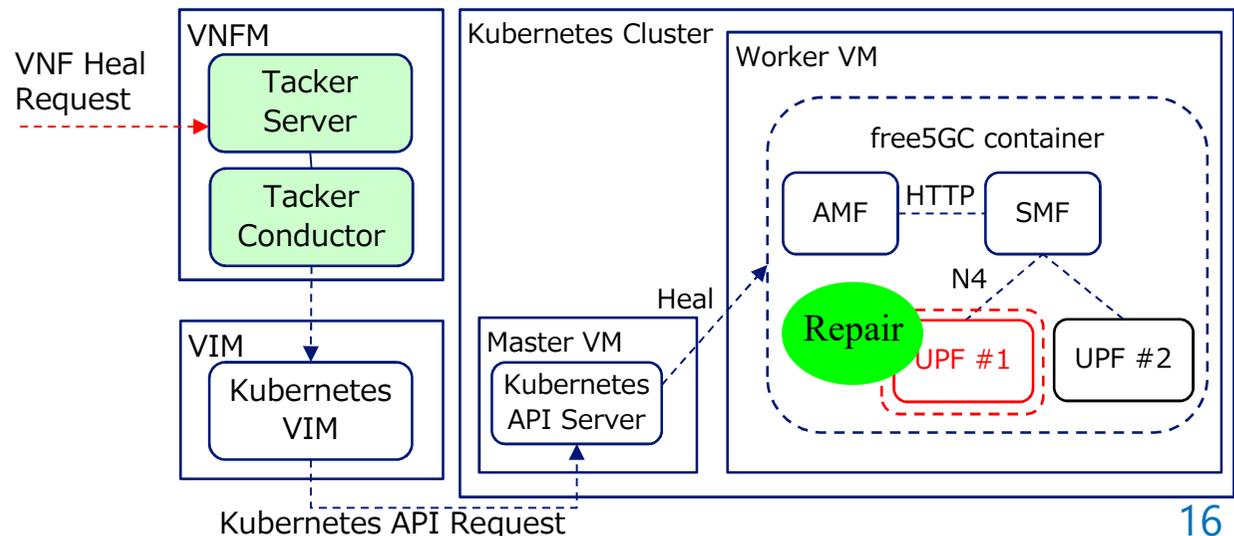


注) TackerによるHealはアプリレイヤの障害への対応を想定しており、KubernetesのAuto HealingによるPodの自動復旧とは異なる。



## After Heal

Heal により UPF プロセスが起動され 再度UE上でWeb コンテンツの表示が可能になる。



# Tacker 操作 C) free5GC UPF の Heal (2/2)



Tacker 操作により free5GC の UPF プロセスを Heal する。

(1) UPF #1 に Heal を実行する Tacker 操作画面

```
$ openstack vnflcm heal aa17a2ae-ddd1-4a3a-8814-ab0f44fc3140 --vnfc-instance ba76566c-458e-4597-8704-e07c3e5d8f41
```

(2) Heal の途中経過 対象 UPF のプロセス状態が Terminating

```
$ kubectl get deploy,pods -o wide
NAME READY UP-TO-DATE AVAILABLE AGE CONTACT
deployment.apps/free5gc-amf-deployment 1/1 1 1 14m myapp-ree5gc-amf-deployment
deployment.apps/free5gc-ausf-deployment 1/1 1 1 14m myapp-ree5gc-ausf-deployment
deployment.apps/free5gc-mongodb 1/1 1 1 14m free5gc-mongodb
deployment.apps/free5gc-nrf-deployment 1/1 1 1 14m myapp-ree5gc-nrf-deployment
deployment.apps/free5gc-nssf-deployment 1/1 1 1 14m myapp-ree5gc-nssf-deployment
deployment.apps/free5gc-pcf-deployment 1/1 1 1 14m myapp-ree5gc-pcf-deployment
deployment.apps/free5gc-smf-deployment 1/1 1 1 14m myapp-ree5gc-smf-deployment
deployment.apps/free5gc-udm-deployment 1/1 1 1 14m myapp-ree5gc-udm-deployment
deployment.apps/free5gc-udr-deployment 1/1 1 1 14m myapp-ree5gc-udr-deployment
deployment.apps/free5gc-upf-deployment 2/2 2 2 14m myapp-ree5gc-upf-deployment
deployment.apps/free5gc-webui-deployment 1/1 1 1 14m myapp-ree5gc-webui-deployment

NAME READY STATUS RESTARTS AGE
pod/free5gc-amf-deployment-859965844-5glpx 1/1 Running 0 14m
pod/free5gc-ausf-deployment-846997ccdf-hacjq 1/1 Running 0 14m
pod/free5gc-mongodb-ddd76dd4c-q4p19 1/1 Running 0 14m
pod/free5gc-nrf-deployment-6cbc59df-d4-hrj2w 1/1 Running 0 14m
pod/free5gc-nssf-deployment-db9df66dc-nvn25 1/1 Running 0 14m
pod/free5gc-pcf-deployment-54f5855c68-9b7gc 1/1 Running 0 14m
pod/free5gc-smf-deployment-59c8dcc986-zfvxb 1/1 Running 0 14m
pod/free5gc-udm-deployment-d987fbbfd-924sg 1/1 Running 0 14m
pod/free5gc-udr-deployment-5b697f4b85-b79t5 1/1 Running 0 14m
pod/free5gc-upf-deployment-98d84679-rkxh 1/1 Running 0 14m
pod/free5gc-upf-deployment-98d84679-vc52p 1/1 Running 0 14m
pod/free5gc-upf-deployment-98d84679-w4rnq 1/1 Terminating 0 14m
pod/free5gc-webui-deployment-61471f4d66-tnkt6 1/1 Running 0 14m
```

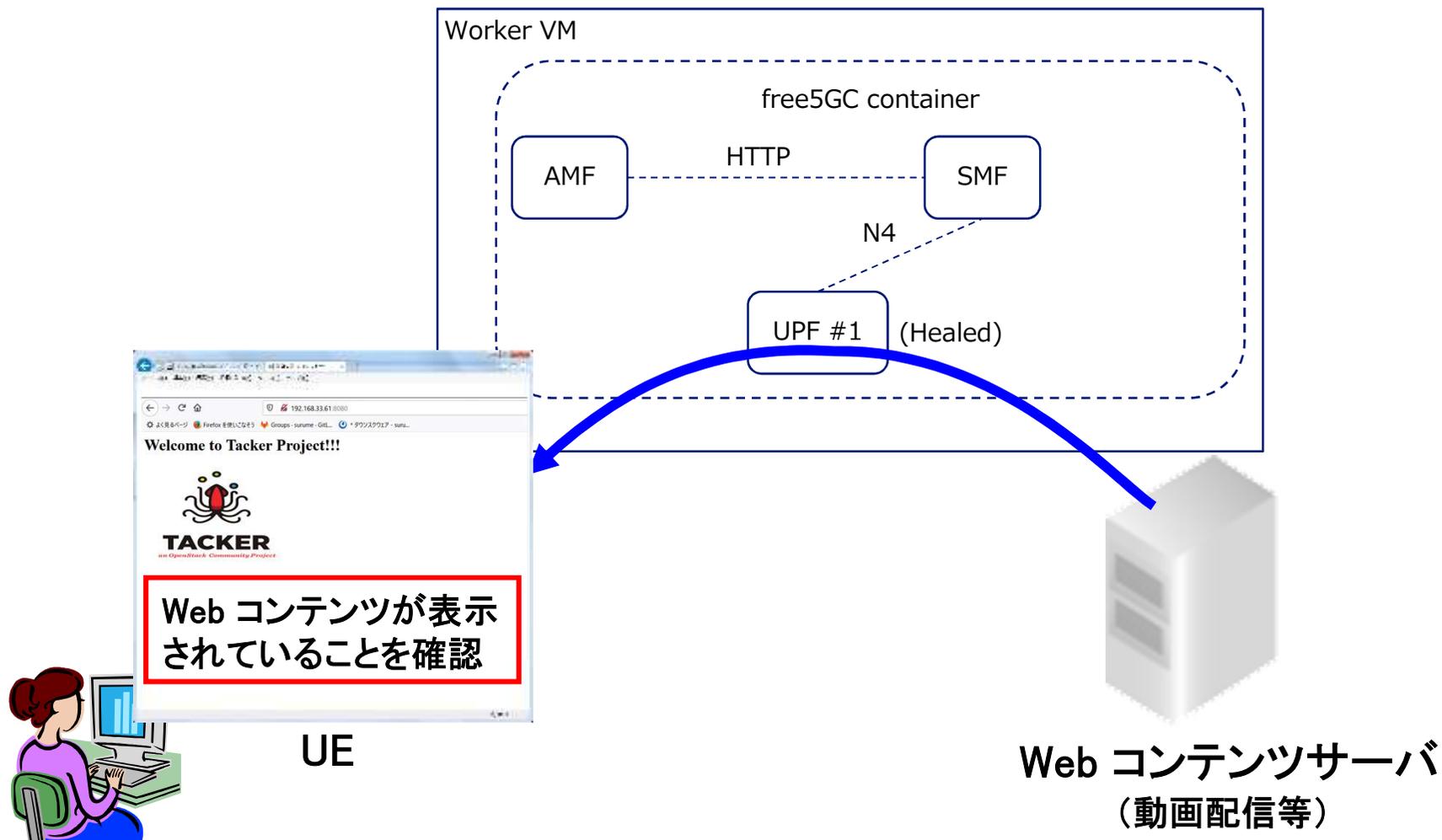
(3) Heal 完了後に新規 UPF プロセスが起動している  
新規UPFプロセスの動作時間が他より短い

```
$ kubectl get deploy,pods -o wide
NAME READY UP-TO-DATE AVAILABLE AGE CONTACT
deployment.apps/free5gc-amf-deployment 1/1 1 1 14m myapp-ree5gc-amf-deployment
deployment.apps/free5gc-ausf-deployment 1/1 1 1 14m myapp-ree5gc-ausf-deployment
deployment.apps/free5gc-mongodb 1/1 1 1 14m free5gc-mongodb
deployment.apps/free5gc-nrf-deployment 1/1 1 1 14m myapp-ree5gc-nrf-deployment
deployment.apps/free5gc-nssf-deployment 1/1 1 1 14m myapp-ree5gc-nssf-deployment
deployment.apps/free5gc-pcf-deployment 1/1 1 1 14m myapp-ree5gc-pcf-deployment
deployment.apps/free5gc-smf-deployment 1/1 1 1 14m myapp-ree5gc-smf-deployment
deployment.apps/free5gc-udm-deployment 1/1 1 1 14m myapp-ree5gc-udm-deployment
deployment.apps/free5gc-udr-deployment 1/1 1 1 14m myapp-ree5gc-udr-deployment
deployment.apps/free5gc-upf-deployment 2/2 2 2 14m myapp-ree5gc-upf-deployment
deployment.apps/free5gc-webui-deployment 1/1 1 1 14m myapp-ree5gc-webui-deployment

NAME READY STATUS RESTARTS AGE
pod/free5gc-amf-deployment-859965844-5glpx 1/1 Running 0 14m
pod/free5gc-ausf-deployment-846997ccdf-hacjq 1/1 Running 0 14m
pod/free5gc-mongodb-ddd76dd4c-q4p19 1/1 Running 0 14m
pod/free5gc-nrf-deployment-6cbc59df-d4-hrj2w 1/1 Running 0 14m
pod/free5gc-nssf-deployment-db9df66dc-nvn25 1/1 Running 0 14m
pod/free5gc-pcf-deployment-54f5855c68-9b7gc 1/1 Running 0 14m
pod/free5gc-smf-deployment-59c8dcc986-zfvxb 1/1 Running 0 14m
pod/free5gc-udm-deployment-d987fbbfd-924sg 1/1 Running 0 14m
pod/free5gc-udr-deployment-5b697f4b85-b79t5 1/1 Running 0 14m
pod/free5gc-upf-deployment-98d84679-rkxh 1/1 Running 0 14m
pod/free5gc-upf-deployment-98d84679-vc52p 1/1 Running 0 53s
pod/free5gc-upf-deployment-98d84679-w4rnq 1/1 Terminating 0 14m
pod/free5gc-webui-deployment-61471f4d66-tnkt6 1/1 Running 0 14m
```

# free5GC の動作正常性の確認

Tacker から Heal された free5GC の UPF プロセスが正常に動作していることを確認するため、**当該UPF経由で取得したWeb コンテンツをUE上で表示させる。**



【改善点①】SMFが起動時にのみUPFとPCFPセッションを確立するため、運用中にUPFをScale/Healした場合、新規構築したUPFとSMFの間でPCFPセッションが確立されず、SMFを再起動することで改めてUPFとのPCFPセッションを確立させる必要がある。

⇒運用中に新規構築したUPFとのPCFPセッションをSMFが自動確立してほしい。

【改善点②】UPFのIPアドレスはUPF設定用yamiファイルに直書きされるため、Scaleした際に同じIPアドレスを持つUPFが構築され、SMFとPCFPセッションが確立できない。

⇒UPFのIPアドレス指定方法について設定ファイル直書きだけでなく、DHCPサーバにアクセスして払い出しを受ける等の別手段も用意してほしい。

**上記改善点はfree5GCのプロセス構成が運用中に  
変化しない前提であれば不要のものだが、実運用を  
考慮すれば自動的なScale/Heal処理を可能にする  
作りにおいておくべきと考える。**

- Tacker公式ドキュメント:  
<https://docs.openstack.org/tacker/latest/user/index.html>
- Tackerソースコード:  
<https://releases.openstack.org/teams/tacker.html>
- Tackerデモビデオ:  
<https://www.youtube.com/watch?v=JK0Jp9yA1Kw>
- 今回紹介したデモの環境構築手順書や設定ファイルの共有も可能です。  
ご興味のある方はSlackでご連絡ください。

- Tackerからfree5GCのLCM制御を実行するデモを作成し、**TackerがモバイルネットワークインフラのMANOコンポーネントOSSとして適用可能**であることを確認しました。
- Tackerをすぐ使うための各種マテリアルは揃っているので、RAN系アプリの検証プラットフォーム等でお試しいただければと思います。