



**GT-FENDE: Ecosistema Federado para Oferta,
Distribuição e Execução de Funções Virtualizadas de
Rede**

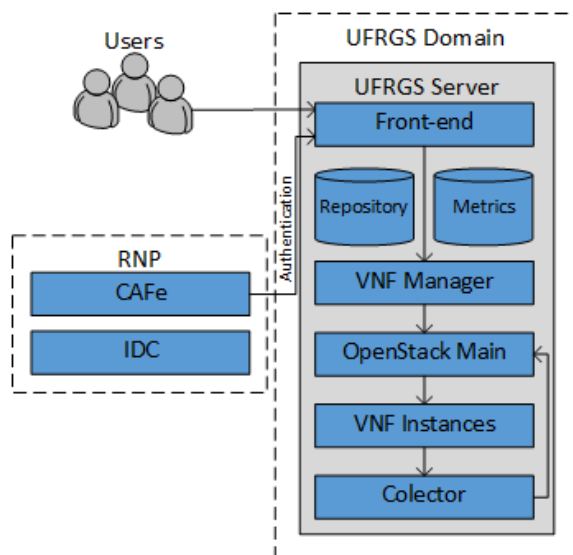
Manual de Instalação e Configuração do Protótipo

Leonardo da Cruz Marcuzzo, Muriel Figueredo Franco, Giovanni Venâncio de Souza, Cassiano Andrei Dias da Silveira Schneider, Lucas Bondan, Carlos Raniery Paula dos Santos

28/02/2018

1. Introdução

Este documento tem por objetivo descrever os passos de instalação da plataforma FENDE, a qual é composta por 3 elementos principais: VNF Server, VNF Manager e Front-end. A plataforma pode ser configurada para operar de duas maneiras distintas: (i) local, com todos os elementos rodando em um mesmo equipamento e (ii) remota, com o front-end coordenando os demais elementos remotamente. As Figuras 1 e 2 abaixo ilustram os dois diferentes modos



operação da plataforma FEND

Figura 1: Configuração local

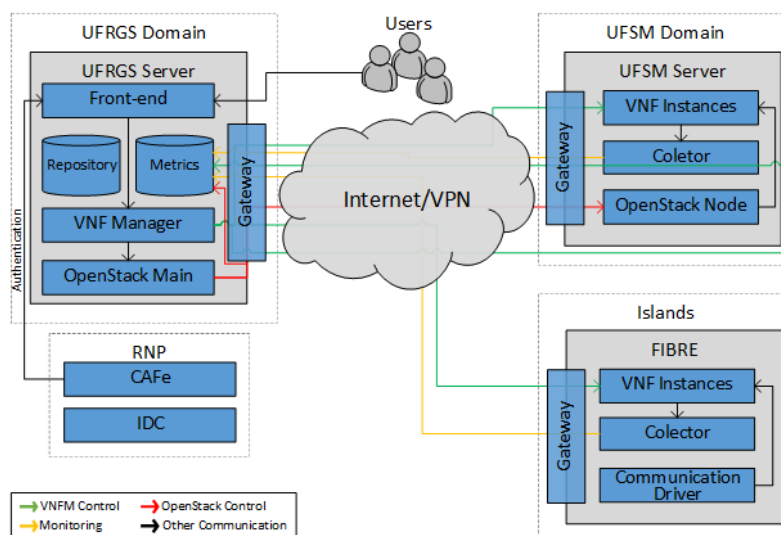


Figura 2: Configuração remota

Nas subseções seguintes estão descritos todos os passos para a instalação e configuração da plataforma, considerando os dois modos de operação. Uma vez instalados e configurados, a plataforma pode ser acessada através do navegador web, através do endereço definido para a plataforma. Para mais informações sobre a arquitetura da plataforma FENDE, pode-se consultar o Relatório Técnico 3 - Avaliação dos resultados do protótipo.

2. VNF Server e VNF Manager

Para a correta operação da plataforma, o VNF Manager deve ser instalado na máquina que será utilizada para a instanciação das VNFs. O VNF Manager utiliza o plugin Tacker do OpenStack para o gerenciamento à nível de recursos. O Tacker fornece uma API para a configuração básica de máquinas virtuais em um ambiente virtualizado. Complementarmente, é utilizada a API disponibilizada pelo VNF Server adotado, chamado Click-On-OSv, para a gerenciamento de funções à nível de software. Além disso, foi desenvolvido um submódulo do VNFM responsável por integrar ambas as APIs e gerenciar de maneira completa o ciclo de vida das VNFs.

Sendo assim, para a correta operação do VNF Manager, uma imagem do VNF Server deve estar disponível para ser instanciada pelo VNF Manager. Tanto a imagem do VNF Server quando os códigos do VNF Manager estão disponíveis nos repositórios oficiais do projeto. A seguir, seguem os passos a serem realizados para instalação e configuração do VNF Manager com a imagem do VNF Server.

Instalação do OpenStack e Tacker

1. Clone da Marketplace
`git clone https://git.rnp.br/gt-fende/web-frontend.git`
2. Criar usuário *stack* e adicionar aos sudoers
`$ sudo adduser stack;`
`$ sudo -i`
`$ echo "stack ALL=(ALL) NO PASSWD: ALL" >> /etc/sudoers`
2. Login no usuário *stack*
`$ su - stack`
3. Clone do devstack na versão *stable/pike*
`$ git clone https://git.openstack.org/openstack-dev/devstack -b stable/pike`
4. Criar o arquivo de configuração do devstack na pasta raiz
`$ cd devstack; touch local.conf`
5. Copiar o seguinte conteúdo para o arquivo *local.conf*. É necessário alterar a variável *HOST_IP* para o IP da máquina que estará executando o OpenStack (máquina local):
`[[local|localrc]]`

```
enable_plugin networking-sfc https://git.openstack.org/openstack/networking-sfc stable/pike
```

```
enable_plugin barbican https://git.openstack.org/openstack/barbican stable/pike
```

```
enable_plugin heat https://git.openstack.org/openstack/heat stable/pike
```

```
enable_plugin mistral https://git.openstack.org/openstack/mistral stable/pike
```

```
enable_plugin tacker https://git.openstack.org/openstack/tacker stable/pike
```

```
USE_BARBICAN=True
```

```
ENABLE_HTTPD_MOD_WSGI_SERVICES=True
```

```
KEYSTONE_USE_MOD_WSGI=True
```

```
ADMIN_PASSWORD=devstack
```

```
DATABASE_PASSWORD=devstack
```

```
RABBIT_PASSWORD=devstack
```

```
SERVICE_PASSWORD=devstack
```

```
HOST_IP=x.y.z.w
```

```
LOGFILE=$DEST/logs/stack.sh.log
```

```
LOGDAYS=2
```

```
SWIFT_HASH=66a3d6b56c1f479c8b4e70ab5c2000f5
SWIFT_REPLICAS=1
SWIFT_DATA_DIR=$DEST/data
[[post-config]$NOVA_CONF]]
[[libvirt]
live_migration_uri = qemu+ssh://stack@%s/system
cpu_mode = host-passthrough
virt_type = kvm
```

6. Execução do devstack:

```
$ ./stack.sh
```

A instalação pode levar cerca de 1 hora para terminar. Assim que for concluída, exporte as variáveis de ambiente:

```
$ source openrc nfv_user nfv
```

7. Os seguintes comandos configuram a imagem do Click-on-OSv no OpenStack:

- Criar a imagem click-on-osv. O parâmetro file deve apontar para o arquivo da imagem:

```
$ openstack image create click-on-osv --disk-format qcow2 --container-format bare --file click-on-osv.img --public --min-disk 11 --min-ram 512
```

- Adicionar o conjunto de instruções SSSE3:

```
$ openstack image set --property
```

```
CIM_PASD_InstructionSetExtensionName=x86:SSSE3 click-on-osv
```

8. Por fim, execute o agente *em_server*:

```
$ nohup web-frontend/vnfm/em/em_server.py &
```

3. Front-end

O protótipo da plataforma FENDE considera três diferentes atores: (i) Desenvolvedores (*Developers*), os quais podem solicitar a inserção de suas VNFs na marketplace da plataforma; (ii) Revisores (*Reviewers*), responsáveis por analisar as solicitações dos desenvolvedores e adicionar ou remover VNFs dos catálogos; e (iii) Clientes (ou *Tenants*), os quais podem adquirir e implantar as VNFs disponibilizadas pelos desenvolvedores e aceitas pelos revisores. Para gerenciamento e interação com os diferentes usuários, foi desenvolvido um Front-end web utilizando Django Framework e Javascript. O Front-end tem a responsabilidade de receber as ações designadas pelos diferentes atores e traduzí-las em comandos para os demais elementos da plataforma. A seguir, seguem os passos para instalação e configuração do front-end, já considerando uma instalação válida do VNF Manager para ser utilizada pelo Front-end.

Preparação e configuração inicial antes de rodar a versão local:

1. Acessar o diretório raiz:

```
$ cd /var/www/fende
```

2. Acessar a pasta scripts

```
$ cd scripts
```

3. Satisfazer todas as dependências e instalar os pacotes necessários:

```
$ sudo pip install -r requirements.txt
```

4. Criar base de dados experimental para a versão local e realiza as configurações necessárias no sistema

```
$ ./createdb
```

Para rodar uma versão local:

1. Acessar o diretório raiz:
`$ cd /var/www/fende`
2. Acessar a pasta de scripts:
`$ cd scripts`
3. Executar o script para rodar o django-server:
`$ sudo ./runserver`

Para fazer o deployment de uma versão remota utilizando o servidor apache:

1. Acessar a pasta do projeto
`$ cd /var/www/fende`
2. Acessar a pasta /docs/Apache_deploy
`$ cd /docs/Apache_deploy`
3. Copiar todos os arquivos .conf para a pasta sites-available do apache
`$ cp *.conf /etc/apache2/sites-available`
4. Acessar instalação do shibboleth
`$ cd shibboleth-installation`
5. Executar o [script install.md](#)
6. Retornar para a pasta Apache_deploy
`$ cd ..`
7. Copiar a pasta sp para /var/www
`$ cp -R sp /var/www`
8. Retornar para a pasta raiz do projeto
`$ cd ..`
9. Acessar a pasta de scripts
`$ cd scripts`
10. Executar o comando para satisfazer as dependências e instalar os pacotes necessários para a execução do django
`$ sudo pip install -r requirements.txt #(Satisfaz todas a dependências e instala os pacotes necessários)`
11. Criar uma base de dados para experimentação
`$./createdb`

Configurando o VNFM no front-end:

1. Faça uma cópia do arquivo *tacker.conf.example* em *web-frontend/vnfm/* e altere o parâmetro *url* para o IP da máquina que está executando o OpenStack e a senha para *devstack*. Essa cópia deve ter o nome *tacker.conf*.
2. Por fim, altere o arquivo *web-frontend/vnfm/em_client.py* e altere o IP para o IP da máquina que está executando o OpenStack.

Visualização de Métricas

A visualização das métricas de utilização das VNFs instanciadas (e.x., CPU, memória, uso de rede) é feita através do Grafana. Os dados originados de cada uma das VNFs instanciadas são coletadas e formatadas por um coletor local. Este coletor recebe uma lista de VNFs do VNFM e utiliza a API do VNF Server para recuperar as métricas, enviando-as para um banco de dados temporal. Desta forma, é necessário que o coletor esteja executando nos servidores em que VNFs são instanciadas. O banco de dados utilizado é o InfluxDB, que deve estar rodando junto com o Grafana e o Web-Frontend. Assim, o coletor também deve ser capaz de se conectar ao banco de dados. No Grafana, uma dashboard com templates deve ser criada, para que os gráficos possam ser exibidos no Web-Frontend de forma dinâmica. A seguir são dadas instruções de instalação e configuração dos serviços.

a. InfluxDB

O InfluxDB é o banco de dados temporal responsável por armazenar as métricas das VNFs que estão sendo executadas. Sua instalação deve ser feita no mesmo servidor onde o Web-Frontend foi instalado. Estas instruções são para uma distribuição Ubuntu versão 16.04 (fonte: <https://docs.influxdata.com/influxdb/v1.4/introduction/installation/>).

```
1. Adicionar repositórios e instalar o serviço:
$ curl -sL https://repos.influxdata.com/influxdb.key | sudo apt-key add -
$ echo "deb https://repos.influxdata.com/ubuntu xenial stable" | sudo
tee /etc/apt/sources.list.d/influxdb.list
$ sudo apt-get update && sudo apt-get install influxdb
$ sudo service influxdb start
```

```
2. Configurar usuário:
$ influx -host gt-fende.inf.ufrgs.br
Connected to http://gt-fende.inf.ufrgs.br:8086 version 1.4.2
InfluxDB shell version: 1.4.2
//Como https nao esta configurado, este password estara em plaintext nas
requisicoes
> CREATE USER coletor WITH PASSWORD 'gtfende' WITH ALL PRIVILEGES
> SHOW USERS
user admin
-----
coletor true
```

```
3. Habilitar e testar autenticação:
$ sudo service influxdb stop
//Em /etc/influxdb/influxdb.conf, em [http], colocar a linha
auth-enabled = true
//Reiniciar o servico e testar
$ sudo service influxdb start
$ influx -host gt-fende.inf.ufrgs.br
Connected to http://gt-fende.inf.ufrgs.br:8086 version 1.4.2
InfluxDB shell version: 1.4.2
>SHOW USERS
ERR: unable to parse authentication credentials
Warning: It is possible this error is due to not setting a database.
Please set a database with the command "use <database>".
//Com usuario autenticado
$ influx -host gt-fende.inf.ufrgs.br -username coletor -password gtfende
Connected to http://gt-fende.inf.ufrgs.br:8086 version 1.4.2
InfluxDB shell version: 1.4.2
> SHOW USERS
user admin
-----
```

```
coletor true
```

4. Criar banco para as VNFs:

```
$ influx-host gt-fende.inf.ufrgs.br -username coletor -password gtfende  
Connected to http://gt-fende.inf.ufrgs.br:8086 version 1.4.2  
InfluxDB shell version: 1.4.2  
CREATE DATABASE vnfs  
> SHOW DATABASES  
name: databases  
name  
----  
_internal  
vnfs
```

Após estes passos o banco de dados está instalado e pronto para receber os dados do coletor e enviá-los ao Grafana.

b. Grafana

O Grafana é o programa responsável por criar os gráficos para a visualização dos recursos das VNFs. A dashboard usada é dinâmica, de forma que VNFs instanciadas e deletadas são mostradas e apagadas, respectivamente, da visualização. Além disso, o dashboard criado pode gerar *iframes* para serem integrados diretamente com o Web-Frontend. Estas instruções são para uma distribuição Ubuntu versão 16.04 (fonte: <http://docs.grafana.org/installation/debian/>).

1. Adicionar repositórios e instalar o serviço:

```
$ echo "deb https://packagecloud.io/grafana/stable/debian/ jessie main" |  
sudo tee /etc/apt/sources.list.d/grafana.list  
$ curl https://packagecloud.io/gpg.key | sudo apt-key add -  
//Instalar  
$ sudo apt-get update  
$ sudo apt-get install grafana  
$ sudo systemctl daemon-reload  
$ sudo /bin/systemctl enable grafana-server  
$ sudo systemctl start grafana-server  
$ sudo systemctl status grafana-server
```

2. O serviço por padrão executa na porta 3000. Através da interface web, logar como admin (senha admin) e trocar a senha.

3. Após, parar o serviço e habilitar acesso anônimo:

```
//Em /etc/grafana/grafana.ini, descomentar as linhas  
[auth.anonymous]  
// enable anonymous access  
enabled = true  
// specify organization name that should be used for unauthenticated users  
org_name = GT-FENDE  
// specify role for unauthenticated users  
org_role = Viewer
```

4. Adicionar o datasource.

Na interface web, selecionar Data Sources, e “Add New Datasource”. Colocar um nome, definir como padrão e tipo InfluxDB, com o endereço de acordo com o definido no InfluxDB. Marcar “Basic Auth” e colocar o usuário e senha do banco. Selecionar como Database “vnfs” e deixar os campos de usuário da database em branco.

5. Importar a dashboard de template, que está disponível no git do projeto.

c. Coletor

O coletor é utilizado para recuperar, formatar e enviar ao banco as métricas de uso das VNFs. Ele deve ser executado no mesmo servidor onde as VNFs serão executadas.

1. No código, `vnf_collector.py`, configurar os seguintes parâmetros:

```
INFLUX_IP = 'gt-fende.inf.ufrgs.br' // De acordo com o endereço do banco
INFLUX_PORT = 8086 //Porta padrão
INFLUX_DATABASE = 'vnfs'
INFLUX_USER='coletor'
INFLUX_PASSWORD='gtfende'
```

2. Na função `main`, definir o nome do servidor, e o tempo de coleta e atualização das VNFs (por padrão coleta a cada 5 segundos e atualização a cada 10 segundos).
3. Iniciar o serviço com o seguinte comando:

```
$ nohup web-frontend/vnfm/em/vnf_collector.py &
```