



# **OPNFV FUNCTEST configuration/installation guide**

*Release arno.2015.1.0 (34315d7)*

**OPNFV**

January 08, 2016



## CONTENTS

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Prerequisites</b>	<b>3</b>
2.1	Docker installation . . . . .	3
2.2	Connectivity to OPNFV management network . . . . .	3
2.3	External network on SUT . . . . .	4
<b>3</b>	<b>High level architecture</b>	<b>5</b>
<b>4</b>	<b>Manual Installation</b>	<b>7</b>
4.1	Focus on the OpenStack credentials . . . . .	9
4.2	Additional Options . . . . .	9
<b>5</b>	<b>Integration in CI</b>	<b>11</b>
<b>6</b>	<b>Configuration</b>	<b>13</b>
<b>7</b>	<b>Errors</b>	<b>15</b>
<b>8</b>	<b>References</b>	<b>17</b>



## INTRODUCTION

**\*\* DOCUMENT IS IN PROGRESS FOR BRAHMAPUTRA \*\***

This document describes how to install and configure Functest in OPNFV.



## PREREQUISITES

The installation of the OPNFV solution is out of scope of this document but can be found in XXX. In the rest of the document the OPNFV solution would be considered as the System Under Test (SUT).

**Several prerequisites are needed for functest:**

1. docker must be installed on the Jumphost
2. a connectivity from the jumphost to the SUT management network is needed
3. an external network must be configured

### 2.1 Docker installation

Log on your jumphost then install docker (e.g. for Ubuntu):

```
curl -sSL https://get.docker.com/ | sh
```

Add your user to docker group to be able to run commands without sudo:

```
sudo usermod -aG docker <your_user>
```

**References:**

- [Ubuntu](#)
- [RHEL](#)

### 2.2 Connectivity to OPNFV management network

Some of the Functest tools need to have access to the OpenStack management network of the controllers [1].

For this reason and besides the public network, you need to configure an interface on the management network as well in your jumphost (or wherever you want to deploy the Functest docker to run the tests).

Example:

```
The OPNFV Fuel installation uses VLAN tagged 300 and subnet 192.168.1.0/24 as Openstack Management network.
Supposing that eth1 is the physical interface with access to that subnet:
$ ip link add name eth1.300 link eth1 type vlan id 300
$ ip link set eth1.300 up
$ ip addr add 192.168.1.66/24 dev eth1.300
```

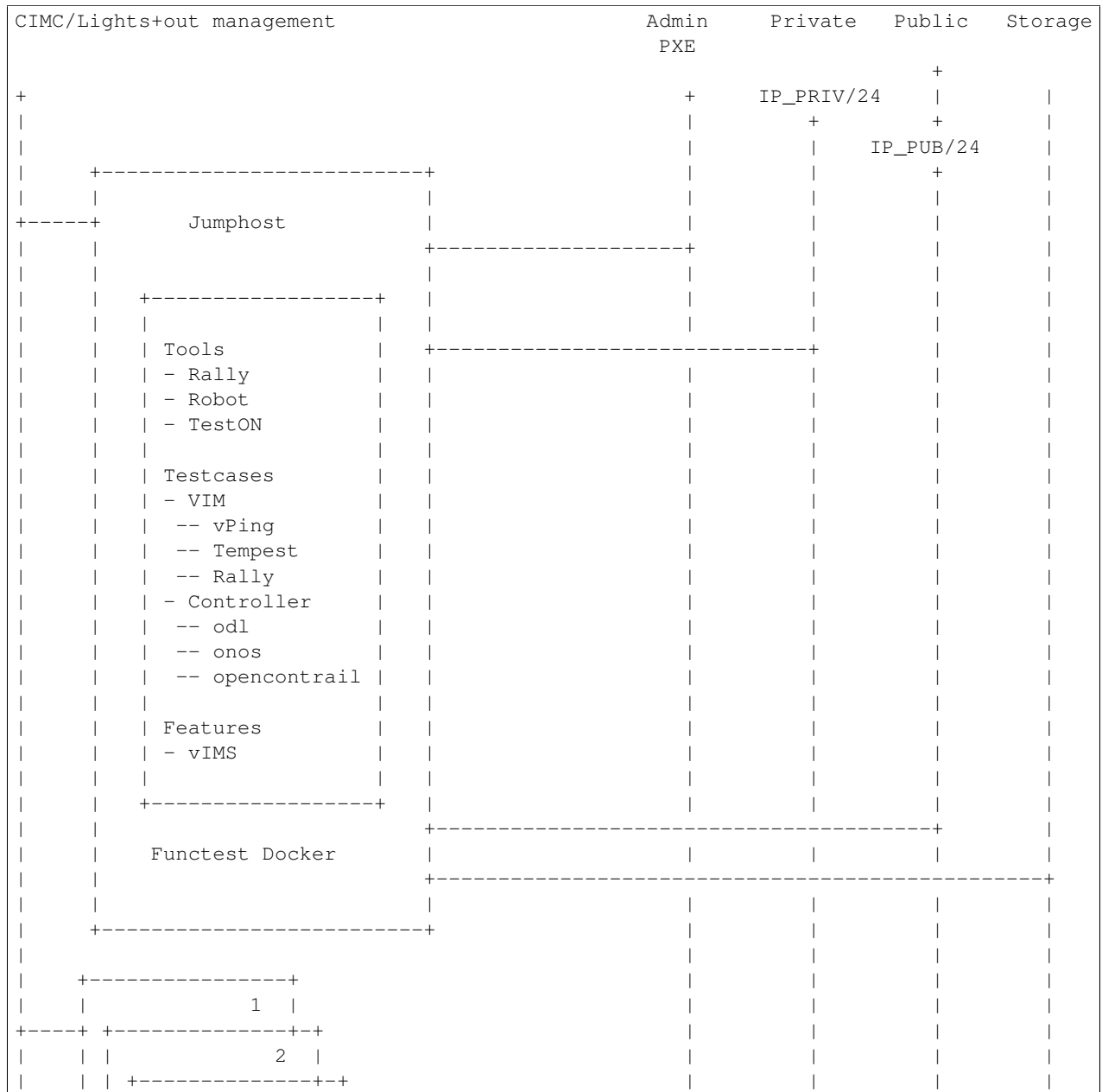
## 2.3 External network on SUT

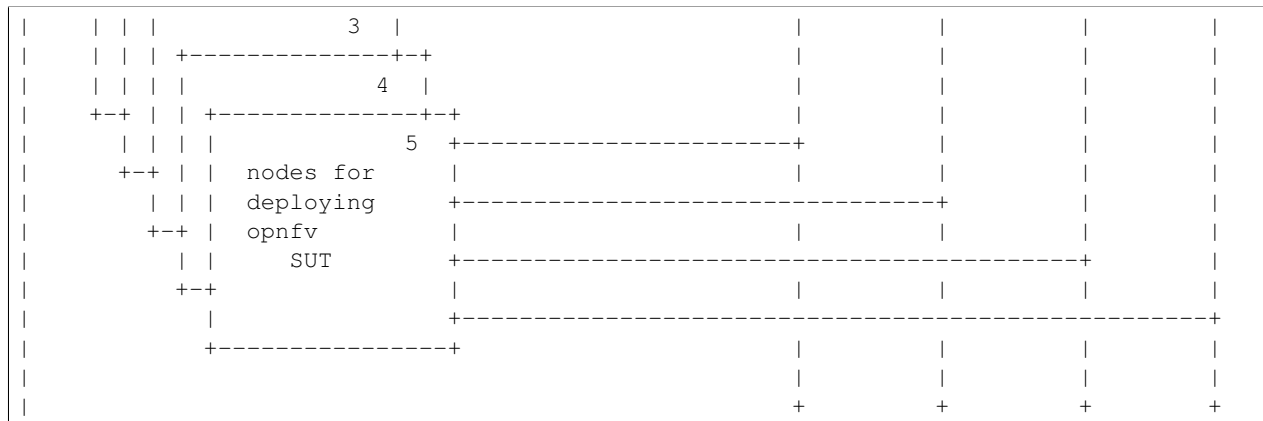
Some of the tests on VIM need an external network. You may perform functest without an external network but some of the tests will fail. A default floating ip pool must be configured in nova.conf.



## HIGH LEVEL ARCHITECTURE

The high level architecture of Functest within OPNFV can be described as follow:





All the components needed by Functest are in the docker file. Thanks to docker, there is no more library issues.

**This docker file will:**

- prepare the environment for functional testing
- retrieve SUT credentials
- perform the tests
- push the results into the OPNFV test result database

This docker file can be integrated into CI or deployed independantly of the CI. A description of the Brahmautra testcases can be retrieved in the Functest user guide [2]

Please note that the Functest container has been designed for OPNFV, however, it would be possible to adapt it to any VIM+controller environment as most of the testcases are upstream testcases.

## MANUAL INSTALLATION

Pull the Functest Docker image from the hub:

```
$ docker pull opnfv/functest
```

Check that the image is available:

```
$ docker images
```

### Run the docker container giving the environment variables

- `INSTALLER_TYPE`. Possible values are “apex”, “compass”, “fuel”, “joid” or “foreman” (Arno).
- `INSTALLER_IP`. each installer has its installation strategy. Functest may need to know the IP of the installer to retrieve the credentials (e.g. usually “10.20.0.2” for fuel, not needed for joid, “172.30.10.73” for foreman...)

The minimum command to create the Functest docker file can be described as follow:

```
docker run -it -e "INSTALLER_IP=10.20.0.2" -e "INSTALLER_TYPE=fuel" opnfv/functest:latest_stable /bin
```

Optionnaly, it is possible to precise the container name through the option `-name`:

```
docker run --name "CONTAINER_NAME" -it -e "INSTALLER_IP=10.20.0.2" -e "INSTALLER_TYPE=fuel" opnfv/fun
```

It is also possible to indicate the path of the OpenStack creds using `-v`:

```
docker run -it -e "INSTALLER_IP=10.20.0.2" -e "INSTALLER_TYPE=fuel" -v <path_to_your_local_creds_fi
```

Your local file will be paste within the container under `/home/opnfv/functest/conf/openstack.creds` and used by the different test suites.

Once run you shall be inside the docker container and ready to run Functest.

Inside the container, you must have the following arborescence:

```
`-- home
  |-- opnfv
  |   |-- functest
  |   |   |-- conf
  |   |   |-- data
  |   |   |-- results
  |   |-- repos
  |       |-- bgpvpn
  |       |-- functest
  |       |-- odl_integration
  |       |-- rally
  |       |-- releng
  |       |-- vims-test
```

Basically the container includes:

- Functest directory to store the configuration (the OpenStack creds are paste in /home/opnpgb/functest/conf), the data (images neede for test for offline testing), results (some temporary artifacts may be stored here)
- Repositories: the functest repository will be used to prepare the environnement, run the tests. Other repositories are used for the installation of the tooling (e.g. rally) and/or the retrieval of feature projects scenarios (e.g. bgpvpn)

The arborescence under the functest repo can be described as follow:

```
.
|-- INFO
|-- LICENSE
|-- commons
|   |-- ims
|   |-- mobile
|   `-- traffic-profile-guidelines.rst
|-- docker
|   |-- Dockerfile
|   |-- common.sh
|   |-- prepare_env.sh
|   |-- requirements.pip
|   `-- run_tests.sh
|-- docs
|   |-- configguide
|   |-- functest.rst
|   |-- images
|   `-- userguide
`-- testcases
    |-- Controllers
    |-- VIM
    |-- __init__.py
    |-- config_functest.py
    |-- config_functest.yaml
    |-- functest_utils.py
    |-- functest_utils.pyc
    |-- vIMS
    `-- vPing
```

We may distinguish 4 different folders:

- commons: it is a folder dedicated to store traffic profile or any test inputs that could be reused by any test project
- docker: this folder includes the scripts that will be used to setup the environment and run the tests
- docs: this folder includes the user and installation/configuration guide
- testcases: this folder includes the scripts required by Functest internal test cases

Firstly run the script to install functest environment:

```
$ ${repos_dir}/functest/docker/prepare_env.sh
```

NOTE: `${repos_dir}` is a default environment variable inside the docker container, which points to /home/opnfv/repos

Run the script to start the tests:

```
$ ${repos_dir}/functest/docker/run_tests.sh
```

NOTE: This will run ALL the tests by default, see [2] for details

## 4.1 Focus on the OpenStack credentials

The OpenStack creds are needed to test the VIM. They can be retrieved in several ways:

- using the `-v` option when creating
- sourcing them manually once connected on the docker, once log perform a `source <openstack_creds>`
- automatically retrieved via a script (CI)

If not source you will get an error during the test environnement preparation.

## 4.2 Additional Options

In case you need to provide different configuration parameters to Functest (e.g. commit IDs or branches for the repositories, ...) copy the `config_functest.yaml` from the repository to your current directory and run docker with a volume:

```
$ wget https://git.opnfv.org/cgit/functest/plain/testcases/config_functest.yaml
$ cmd1 = "/home/opnfv/repos/functest/docker/prepare_env.sh"
$ cmd2 = "/home/opnfv/repos/functest/docker/run_tests.sh"
$ docker run -t -e "INSTALLER_TYPE=fuel" -e "INSTALLER_IP=10.20.0.2" opnfv/functest \
-v $(pwd)/config_functest.yaml:/home/opnfv/functest/conf/config_functest.yaml \
"${cmd1} && ${cmd2}"
```



## INTEGRATION IN CI

In CI we use the docker file and execute commande within the container from Jenkins.

Docker creation in set-functest-env builder [3]:

```
envs="INSTALLER_TYPE=${INSTALLER_TYPE} -e INSTALLER_IP=${INSTALLER_IP} -e NODE_NAME=${NODE_NAME}"
[...]
docker pull opnfv/functest:latest_stable
cmd="docker run -id -e $envs ${labconfig} ${sshkey} ${res_volume} opnfv/functest:latest_stable /bin/ls"
echo "Functest: Running docker run command: ${cmd}"
${cmd}
docker ps -a
sleep 5
container_id=$(docker ps | grep 'opnfv/functest:latest_stable' | awk '{print $1}' | head -1)
echo "Container ID=${container_id}"
if [ -z ${container_id} ]; then
    echo "Cannot find opnfv/functest container ID ${container_id}. Please check if it is existing."
    docker ps -a
    exit 1
fi
echo "Starting the container: docker start ${container_id}"
docker start ${container_id}
sleep 5
docker ps
if [ $(docker ps | grep 'opnfv/functest:latest_stable' | wc -l) == 0 ]; then
    echo "The container opnfv/functest with ID=${container_id} has not been properly started. Exiting."
    exit 1
fi
cmd="${FUNCTEST_REPO_DIR}/docker/prepare_env.sh"
echo "Executing command inside the docker: ${cmd}"
docker exec ${container_id} ${cmd}
```

Test execution in functest-all builder [3]:

```
echo "Functest: run $FUNCTEST_SUITE_NAME"
cmd="${FUNCTEST_REPO_DIR}/docker/run_tests.sh --test $FUNCTEST_SUITE_NAME ${flag}"
container_id=$(docker ps -a | grep opnfv/functest | awk '{print $1}' | head -1)
docker exec $container_id $cmd
```

Docker clean in functest-cleanup builder [3]:

```
echo "Cleaning up docker containers/images..."
# Remove previous running containers if exist
if [[ ! -z $(docker ps -a | grep opnfv/functest) ]]; then
echo "Removing existing opnfv/functest containers..."
docker ps | grep opnfv/functest | awk '{print $1}' | xargs docker stop
docker ps -a | grep opnfv/functest | awk '{print $1}' | xargs docker rm
```

```
fi

# Remove existing images if exist
if [[ ! -z $(docker images | grep opnfv/functest) ]]; then
echo "Docker images to remove:"
docker images | head -1 && docker images | grep opnfv/functest
image_tags=$(docker images | grep opnfv/functest | awk '{print $2}')
for tag in "${image_tags[@]}; do
    echo "Removing docker image opnfv/functest:$tag..."
    docker rmi opnfv/functest:$tag
done
fi
```



## CONFIGURATION

Everything is preconfigured in the docker file. So except the credentials, there is no specific configuration needed. it is possible to customize the list of tests, see [2] for details.



---

CHAPTER  
**SEVEN**

---

**ERRORS**



## REFERENCES

OPNFV main site: [opnfvmain](#).

OPNFV functional test page: [opnfvfunctest](#).

IRC support chan: [#opnfv-testperf](#)