

---

# OPNFV Functest Documentation

*Release master*

**Functest <[opnfv-tech-discuss@lists.opnfv.org](mailto:opnfv-tech-discuss@lists.opnfv.org)>**

**Sep 14, 2020**



---

## Contents

---

<b>1</b>	<b>functest</b>	<b>3</b>
1.1	functest package . . . . .	3
<b>2</b>	<b>Indices and tables</b>	<b>31</b>
	<b>Python Module Index</b>	<b>33</b>
	<b>Index</b>	<b>35</b>



Contents:



## 1.1 functest package

### 1.1.1 Subpackages

#### functest.core package

#### Submodules

#### functest.core.cloudify module

Cloudify testcase implementation.

```
class functest.core.cloudify.Cloudify(**kwargs)
    Bases: functest.core.singlevm.SingleVm2
```

Cloudify Orchestrator Case.

```
cloudify_archive = '/home/opnfv/functest/images/cloudify-docker-manager-community-19.0
```

```
cloudify_container = 'docker-cfy-manager:latest'
```

```
create_server_timeout = 600
```

```
execute()
```

Deploy Cloudify Manager.

```
filename = '/home/opnfv/functest/images/ubuntu-16.04-server-cloudimg-amd64-disk1.img'
```

```
flavor_disk = 40
```

```
flavor_ram = 4096
```

```
flavor_vcpus = 2
```

```
kill_existing_execution(dep_name)
```

kill existing execution

**ports = [80, 443, 5671, 53333]**

**prepare ()**

Create the security group and the keypair

It can be overridden to set other rules according to the services running in the VM

Raises: Exception on error

**put\_private\_key ()**

Put private keypair in manager

**ssh\_connect\_loops = 12**

**upload\_cfy\_plugins (yaml, wgn)**

Upload Cloudify plugins

**username = 'ubuntu'**

`functest.core.cloudify.get_execution_id (client, deployment_id)`

Get the execution id of a env preparation.

network, security group, fip, VM creation

`functest.core.cloudify.wait_for_execution (client, execution, logger, timeout=3600)`

Wait for a workflow execution on Cloudify Manager.

## functest.core.singlevm module

Ease deploying a single VM reachable via ssh

It offers a simple way to create all tenant network resources + a VM for advanced testcases (e.g. deploying an orchestrator).

**class** `functest.core.singlevm.SingleVm1 (**kwargs)`

Bases: `functest.core.singlevm.VmReady1`

Deploy a single VM reachable via ssh (scenario1)

It inherits from TenantNetwork1 which creates all network resources and completes it by booting a VM attached to that network.

It ensures that all testcases inheriting from SingleVm1 could work without specific configurations (or at least read the same config data).

**check\_console\_loop = 6**

**check\_console\_regex = ' login: '**

**clean ()**

Clean the resources.

It can be overridden if resources must be deleted after running the test case.

**connect (vm1)**

Connect to a virtual machine via ssh

It first adds a floating ip to the virtual machine and then establishes the ssh connection.

Returns: - (fip, ssh) - None on error

**create\_floating\_ip\_timeout = 120**



**execute ()**

Say hello world via ssh

It can be overridden to execute any command.

Returns: echo exit codes

**prepare ()**

Create the security group and the keypair

It can be overridden to set other rules according to the services running in the VM

Raises: Exception on error

**run (\*\*kwargs)**

Boot the new VM

Here are the main actions: - add a new ssh key - boot the VM - create the security group - execute the right command over ssh

Returns: - TestCase.EX\_OK - TestCase.EX\_RUN\_ERROR on error

**ssh\_connect\_loops = 6**

**ssh\_connect\_timeout = 1**

**username = 'cirros'**

**class** `functest.core.singlevm.SingleVm2 (**kwargs)`

Bases: `functest.core.singlevm.SingleVm1`

Deploy a single VM reachable via ssh (scenario2)

It creates new user/project before creating and configuring all tenant network resources and vms required by advanced testcases.

It ensures that all testcases inheriting from SingleVm2 could work without specific configurations (or at least read the same config data).

**clean ()**

Clean the resources.

It can be overridden if resources must be deleted after running the test case.

**class** `functest.core.singlevm.VmReady1 (**kwargs)`

Bases: `functest.core.tenantnetwork.TenantNetwork1`

Prepare a single VM (scenario1)

It inherits from TenantNetwork1 which creates all network resources and prepares a future VM attached to that network.

It ensures that all testcases inheriting from SingleVm1 could work without specific configurations (or at least read the same config data).

**boot\_vm (name=None, \*\*kwargs)**

Boot the virtual machine

It allows booting multiple machines for the child testcases. It forces the same configuration for all subtest-cases.

Returns: vm

Raises: exception on error

**check\_regex\_in\_console** (*name, regex=' login: ', loop=6*)

Wait for specific message in console

Returns: True or False on errors

**clean** ()

Clean the resources.

It can be overridden if resources must be deleted after running the test case.

**clean\_orphan\_security\_groups** ()

Clean all security groups which are not owned by an existing tenant

It lists all orphan security groups in use as debug to avoid misunderstanding the testcase results (it could happen if cloud admin removes accounts without cleaning the virtual machines)

**count\_active\_hypervisors** ()

Count all hypervisors which are up.

**count\_hypervisors** ()

Count hypervisors.

**create\_flavor** (*name=None*)

Create flavor

It allows creating multiple flavors for the child testcases. It forces the same configuration for all subtestcases.

Returns: flavor

Raises: exception on error

**create\_flavor\_alt** (*name=None*)

Create flavor

It allows creating multiple alt flavors for the child testcases. It forces the same configuration for all subtestcases.

Returns: flavor

Raises: exception on error

**create\_server\_timeout** = 180

**extra\_alt\_properties** = {}

**extra\_properties** = {}

**filename** = '/home/opnfv/functest/images/cirros-0.4.0-x86\_64-disk.img'

**filename\_alt** = '/home/opnfv/functest/images/cirros-0.4.0-x86\_64-disk.img'

**flavor\_alt\_disk** = 1

**flavor\_alt\_extra\_specs** = {}

**flavor\_alt\_ram** = 1024

**flavor\_alt\_vcpus** = 1

**flavor\_disk** = 1

**flavor\_extra\_specs** = {}

**flavor\_ram** = 512

**flavor\_vcpus** = 1

```
image_alt_format = 'qcow2'
```

```
image_format = 'qcow2'
```

```
publish_image (name=None)
```

Publish image

It allows publishing multiple images for the child testcases. It forces the same configuration for all sub-testcases.

Returns: image

Raises: exception on error

```
publish_image_alt (name=None)
```

Publish alternative image

It allows publishing multiple images for the child testcases. It forces the same configuration for all sub-testcases.

Returns: image

Raises: exception on error

```
run (**kwargs)
```

Boot the new VM

Here are the main actions: - publish the image - create the flavor

Returns: - TestCase.EX\_OK - TestCase.EX\_RUN\_ERROR on error

```
visibility = 'private'
```

```
class functest.core.singlevm.VmReady2 (**kwargs)
```

Bases: *functest.core.singlevm.VmReady1*

Deploy a single VM reachable via ssh (scenario2)

It creates new user/project before creating and configuring all tenant network resources, flavors, images, etc. required by advanced testcases.

It ensures that all testcases inheriting from SingleVm2 could work without specific configurations (or at least read the same config data).

```
clean ()
```

Clean the resources.

It can be overridden if resources must be deleted after running the test case.

## functest.core.tenantnetwork module

Ease deploying tenant networks

It offers a simple way to create all tenant network resources required by a testcase (including all Functest ones):

- TenantNetwork1 selects the user and the project set as env vars
- TenantNetwork2 creates a user and project to isolate the same resources

This classes could be reused by more complexed scenarios (Single VM)

```
class functest.core.tenantnetwork.NewProject (cloud, case_name, guid)
```

Ease creating new projects/users

**clean ()**  
Remove projects/users

**create ()**  
Create projects/users

**get\_environ ()**  
Get new environ

**class** `functest.core.tenantnetwork.TenantNetwork1 (**kwargs)`  
Bases: `xtesting.core.testcase.TestCase`

Create a tenant network (scenario1)

It creates and configures all tenant network resources required by advanced testcases (subnet, network and router).

It ensures that all testcases inheriting from TenantNetwork1 could work without network specific configurations (or at least read the same config data).

**allow\_no\_fip = False**

**cidr = '192.168.120.0/24'**

**clean ()**  
Clean the resources.

It can be overridden if resources must be deleted after running the test case.

**create\_network\_resources ()**  
Create all tenant network resources

It creates a router which gateway is the external network detected. The new subnet is attached to that router.

Raises: exception on error

**static get\_default\_role (cloud, member='Member')**  
Get the default role

It also tests the role in lowercase to avoid possible conflicts.

**static get\_external\_network (cloud)**  
Return the configured external network name or the first retrieved external network name

**static get\_public\_auth\_url (cloud)**  
Get Keystone public endpoint

**run (\*\*kwargs)**  
Run the test case.

It allows running TestCase and getting its execution status.

The subclasses must override the default implementation which is false on purpose.

The new implementation must set the following attributes to push the results to DB:

- result,
- start\_time,
- stop\_time.

**Args:** kwargs: Arbitrary keyword arguments.

**shared\_network = False**

**class** `functest.core.tenantnetwork.TenantNetwork2` (\*\*kwargs)

Bases: `functest.core.tenantnetwork.TenantNetwork1`

Create a tenant network (scenario2)

It creates new user/project before creating and configuring all tenant network resources required by a testcase (subnet, network and router).

It ensures that all testcases inheriting from TenantNetwork2 could work without network specific configurations (or at least read the same config data).

**clean** ()

Clean the resources.

It can be overridden if resources must be deleted after running the test case.

## Module contents

### functest.opnfv\_tests package

#### Subpackages

#### functest.opnfv\_tests.openstack package

#### Subpackages

#### functest.opnfv\_tests.openstack.api package

#### Submodules

#### functest.opnfv\_tests.openstack.api.connection\_check module

Verify the connection to OpenStack Services

**class** `functest.opnfv_tests.openstack.api.connection_check.ConnectionCheck` (\*\*kwargs)

Bases: `xtesting.core.testcase.TestCase`

Perform simplest queries

**func\_list** = ['get\_network\_extensions', 'list\_aggregates', 'list\_domains', 'list\_endpoints']

**run** (\*\*kwargs)

Run all read operations to check connections

## Module contents

### functest.opnfv\_tests.openstack.cinder package

#### Submodules

#### functest.opnfv\_tests.openstack.cinder.cinder\_test module

CinderCheck testcase.

**class** `functest.opnfv_tests.openstack.cinder.cinder_test.CinderCheck` (\*\*kwargs)

Bases: `functest.core.singlevm.SingleVm2`

CinderCheck testcase implementation.

Class to execute the CinderCheck test using 2 Floating IPs to connect to the VMs and one data volume

**clean** ()

Clean the resources.

It can be overridden if resources must be deleted after running the test case.

**execute** ()

Execute CinderCheck testcase.

Sets up the OpenStack keypair, router, security group, and VM instance objects then validates cinder.  
:return: the exit code from the super.execute() method

**prepare** ()

Create the security group and the keypair

It can be overridden to set other rules according to the services running in the VM

Raises: Exception on error

**volume\_timeout** = 60

### Module contents

#### `functest.opnfv_tests.openstack.patrole` package

#### Submodules

#### `functest.opnfv_tests.openstack.patrole.patrole` module

**class** `functest.opnfv_tests.openstack.patrole.patrole.Patrole` (\*\*kwargs)

Bases: `functest.opnfv_tests.openstack.tempest.tempest.TempestCommon`

**configure** (\*\*kwargs)

Create all openstack resources for tempest-based testcases and write tempest.conf.

**run** (\*\*kwargs)

Boot the new VM

Here are the main actions: - publish the image - create the flavor

Returns: - TestCase.EX\_OK - TestCase.EX\_RUN\_ERROR on error

### Module contents

#### `functest.opnfv_tests.openstack.rally` package

#### Submodules

#### `functest.opnfv_tests.openstack.rally.rally` module

Rally testcases implementation.

```
class functest.opnfv_tests.openstack.rally.rally.RallyBase (**kwargs)
    Bases: functest.core.singlevm.VmReady2
```

Base class form Rally testcases implementation.

```
allow_no_fip = True
```

```
apply_blacklist (case_file_name, result_file_name)
    Apply blacklist.
```

```
blacklist_file = '/home/docs/checkouts/readthedocs.org/user_builds/functest-api/envs/s
```

```
build_task_args (test_name)
    Build arguments for the Rally task.
```

```
clean ()
    Cleanup of OpenStack resources. Should be called on completion.
```

```
static clean_rally_conf (rally_conf='/etc/rally/rally.conf')
    Clean Rally config
```

```
static clean_rally_logs (rally_conf='/etc/rally/rally.conf')
    Clean Rally config
```

```
concurrency = 4
```

```
static create_rally_deployment (environ=None)
    Create new rally deployment
```

```
excl_func ()
    Exclude functionalities.
```

```
static excl_scenario ()
    Exclude scenario.
```

```
static export_task (file_name, export_type='html')
    Export all task results (e.g. html or xunit report)
```

**Raises:** subprocess.CalledProcessError: if Rally doesn't return 0

**Returns:** None

```
static file_is_empty (file_name)
    Determine is a file is empty.
```

```
static get_task_id (tag)
    Get task id from command rally result.
```

**Parameters** *tag* –

**Returns** *task\_id* as string

```
static get_verifier_deployment_id ()
    Returns deployment id for active Rally deployment
```

```
static in_iterable_re (needle, haystack)
    Check if given needle is in the iterable haystack, using regex.
```

**Parameters**

- **needle** – string to be matched
- **haystack** – iterable of strings (optionally regex patterns)

**Returns** True if needle is equal to any of the elements in haystack, or if a nonempty regex pattern in haystack is found in needle.

```
is_successful ()
    The overall result of the test.

iterations_amount = 10

prepare_run (**kwargs)
    Prepare resources needed by test scenarios.

prepare_task (test_name)
    Prepare resources for test run.

rally_aar4_patch_path = '/home/docs/checkouts/readthedocs.org/user_builds/functest-api/
rally_conf_path = '/etc/rally/rally.conf'
rally_dir = '/home/docs/checkouts/readthedocs.org/user_builds/functest-api/envs/stable
rally_scenario_dir = '/home/docs/checkouts/readthedocs.org/user_builds/functest-api/env
run (**kwargs)
    Run testcase.

run_task (test_name)
    Run a task.

run_tests (**kwargs)
    Execute tests.

shared_network = True

stests = ['authenticate', 'glance', 'cinder', 'gnocchi', 'heat', 'keystone', 'neutron']
support_dir = '/home/docs/checkouts/readthedocs.org/user_builds/functest-api/envs/stable
task_dir = u'/home/opnfv/functest/data/rally/task'

static task_succeed (json_raw)
    Parse JSON from rally JSON results.

    Parameters json_raw -
    Returns Bool

task_timeout = 3600

temp_dir = u'/home/opnfv/functest/data/rally/task/var'

template_dir = '/home/docs/checkouts/readthedocs.org/user_builds/functest-api/envs/stable
tenants_amount = 3

static update_keystone_default_role (rally_conf='/etc/rally/rally.conf')
    Set keystone_default_role in rally.conf

static update_rally_logs (res_dir, rally_conf='/etc/rally/rally.conf')
    Print rally logs in res dir

users_amount = 2

static verify_report (file_name, uuid, export_type='html')
    Generate the verifier report (e.g. html or xunit report)

    Raises: subprocess.CalledProcessError: if Rally doesn't return 0

    Returns: None

visibility = 'public'
```



```

    volume_service_type = 'volumev3'
    volume_version = 3
class functest.opnfv_tests.openstack.rally.rally.RallyFull (**kwargs)
    Bases: functest.opnfv_tests.openstack.rally.rally.RallyBase
    Rally full testcase implementation.
    task_timeout = 7200
class functest.opnfv_tests.openstack.rally.rally.RallyJobs (**kwargs)
    Bases: functest.opnfv_tests.openstack.rally.rally.RallyBase
    Rally OpenStack CI testcase implementation.
    apply_blacklist (case_file_name, result_file_name)
        Apply blacklist.
    build_task_args (test_name)
        Build arguments for the Rally task.
    clean ()
        Cleanup of OpenStack resources. Should be called on completion.
    prepare_run (**kwargs)
        Create resources needed by test scenarios.
    prepare_task (test_name)
        Prepare resources for test run.
    stests = ['neutron']
    task_timeout = 7200
class functest.opnfv_tests.openstack.rally.rally.RallySanity (**kwargs)
    Bases: functest.opnfv_tests.openstack.rally.rally.RallyBase
    Rally sanity testcase implementation.

```

## Module contents

### functest.opnfv\_tests.openstack.refstack package

#### Submodules

#### functest.opnfv\_tests.openstack.refstack.refstack module

Refstack testcase implementation.

```

class functest.opnfv_tests.openstack.refstack.refstack.Refstack (**kwargs)
    Bases: functest.opnfv_tests.openstack.tempest.tempest.TempestCommon
    Refstack testcase implementation class.
    generate_test_list (**kwargs)
        Generate test list based on the test mode.

```

## Module contents

### functest.opnfv\_tests.openstack.shaker package

#### Submodules

### functest.opnfv\_tests.openstack.shaker.shaker module

**Shaker** wraps around popular system network testing tools like iperf, iperf3 and netperf (with help of flent). Shaker is able to deploy OpenStack instances and networks in different topologies. Shaker scenario specifies the deployment and list of tests to execute.

```
class functest.opnfv_tests.openstack.shaker.shaker.Shaker (**kwargs)
    Bases: functest.core.singlevm.SingleVm2

    Run shaker full+perf I2 and I3

    check_console_loop = 12

    check_requirements ()
        Check the requirements of the test case.

        It can be overridden on purpose.

    clean ()
        Clean the resources.

        It can be overridden if resources must be deleted after running the test case.

    create_server_timeout = 300

    execute ()

        Returns:

        • 0 if success
        • 1 on operation error

    filename = '/home/opnfv/functest/images/shaker-image-1.3.0+stretch.qcow2'

    flavor_disk = 3

    flavor_ram = 512

    flavor_vcpus = 1

    port = 9000

    prepare ()
        Create the security group and the keypair

        It can be overridden to set other rules according to the services running in the VM

        Raises: Exception on error

    quota_cores = -1

    quota_instances = -1

    shaker_timeout = '3600'

    ssh_connect_loops = 12

    username = 'debian'
```

## Module contents

### functest.opnfv\_tests.openstack.tempest package

#### Submodules

#### functest.opnfv\_tests.openstack.tempest.tempest module

Tempest testcases implementation.

```

class functest.opnfv_tests.openstack.tempest.tempest.TempestCommon (**kwargs)
    Bases: functest.core.singlevm.VmReady2

    TempestCommon testcases implementation class.

    apply_tempest_blacklist (black_list)
        Exclude blacklisted test cases.

    static backup_tempest_config (conf_file, res_dir)
        Copy config file to tempest results directory

    check_extensions ()
        Check the mandatory network extensions.

    check_requirements ()
        Check the requirements of the test case.

        It can be overridden on purpose.

    check_services ()
        Check the mandatory services.

    clean ()
        Cleanup all OpenStack objects. Should be called on completion.

    static clean_rally_conf (rally_conf='etc/rally/rally.conf')
        Clean Rally config

    configure (**kwargs)
        Create all openstack resources for tempest-based testcases and write tempest.conf.

    static configure_tempest_update_params (tempest_conf_file, image_id=None, flavor_id=None, compute_cnt=1, image_alt_id=None, flavor_alt_id=None, admin_role_name='admin', cidr='192.168.120.0/24', domain_id='default')

        Add/update needed parameters into tempest.conf file

    static configure_verifier (deployment_dir)
        Execute rally verify configure-verifier, which generates tempest.conf

    static create_verifier ()
        Create new verifier

    filename_alt = '/home/opnfv/functest/images/cirros-0.4.0-x86_64-disk.img'

    generate_test_list (**kwargs)
        Generate test list based on the test mode.

```

**static get\_verifier\_deployment\_dir** (*verifier\_id, deployment\_id*)

Returns Rally deployment directory for current verifier

**static get\_verifier\_id** ()

Returns verifier id for current Tempest

**static get\_verifier\_repo\_dir** (*verifier\_id*)

Returns installed verifier repo directory for Tempest

**static get\_verifier\_result** (*verif\_id*)

Retrieve verification results.

**is\_successful** ()

The overall result of the test.

**parse\_verifier\_result** ()

Parse and save test results.

**static read\_file** (*filename*)

Read file and return content as a stripped list.

**run** (*\*\*kwargs*)

Boot the new VM

Here are the main actions: - publish the image - create the flavor

Returns: - TestCase.EX\_OK - TestCase.EX\_RUN\_ERROR on error

**run\_verifier\_tests** (*\*\*kwargs*)

Execute tempest test cases.

**shared\_network = True**

**tempest\_blacklist = '/home/docs/checkouts/readthedocs.org/user\_builds/functest-api/envs/**

**tempest\_conf\_yaml = '/home/docs/checkouts/readthedocs.org/user\_builds/functest-api/envs/**

**tempest\_custom = '/home/docs/checkouts/readthedocs.org/user\_builds/functest-api/envs/**

**tempest\_public\_blacklist = '/home/docs/checkouts/readthedocs.org/user\_builds/functest-**

**update\_compute\_section** ()

Update compute section in tempest.conf

**update\_default\_role** (*rally\_conf='/etc/rally/rally.conf'*)

Detect and update the default role if required

**update\_network\_section** ()

Update network section in tempest.conf

**update\_rally\_regex** (*rally\_conf='/etc/rally/rally.conf'*)

Set image name as tempest img\_name\_regex

**update\_scenario\_section** ()

Update scenario section in tempest.conf

**static update\_tempest\_conf\_file** (*conf\_file, rconfig*)

Update defined paramters into tempest config file

**update\_validation\_section** ()

Update validation section in tempest.conf

**visibility = 'public'**

```

class functest.opnfv_tests.openstack.tempest.tempest.TempestHeat (**kwargs)
    Bases: functest.opnfv_tests.openstack.tempest.tempest.TempestCommon

    Tempest Heat testcase implementation class.

    clean ()
        Cleanup all OpenStack objects. Should be called on completion.

    configure (**kwargs)
        Create all openstack resources for tempest-based testcases and write tempest.conf.

    filename_alt = '/home/opnfv/functest/images/Fedora-Cloud-Base-30-1.2.x86_64.qcow2'

    flavor_alt_disk = 4

    flavor_alt_ram = 512

    flavor_alt_vcpus = 1

class functest.opnfv_tests.openstack.tempest.tempest.TempestHorizon (**kwargs)
    Bases: functest.opnfv_tests.openstack.tempest.tempest.TempestCommon

    Tempest Horizon testcase implementation class.

    configure (**kwargs)
        Create all openstack resources for tempest-based testcases and write tempest.conf.

```

## Module contents

### functest.opnfv\_tests.openstack.vmtip package

#### Submodules

### functest.opnfv\_tests.openstack.vmtip.vmtip module

**VMTP** is a small python application that will automatically perform ping connectivity, round trip time measurement (latency) and TCP/UDP throughput measurement for the following East/West flows on any OpenStack deployment:

- VM to VM same network (private fixed IP, flow #1)
- VM to VM different network using fixed IP (same as intra-tenant L3 fixed IP, flow #2)
- VM to VM different network using floating IP and NAT (same as floating IP inter-tenant L3, flow #3)

```

class functest.opnfv_tests.openstack.vmtip.vmtip.Vmtip (**kwargs)
    Bases: functest.core.singlevm.VmReady2

    Class to run Vmtip as an OPNFV Functest testcase

    check_requirements ()
        Check the requirements of the test case.

        It can be overridden on purpose.

    clean ()
        Clean the resources.

        It can be overridden if resources must be deleted after running the test case.

    create_network_resources ()
        Create router

```

It creates a router which gateway is the external network detected.

Raises: exception on error

**create\_server\_timeout** = 300

**filename** = '/home/opnfv/functest/images/ubuntu-14.04-server-cloudimg-amd64-disk1.img'

**flavor\_disk** = 0

**flavor\_ram** = 2048

**flavor\_vcpus** = 1

**generate\_keys** ()

Generate Keys

Raises: Exception on error

**run** (\*\*kwargs)

Boot the new VM

Here are the main actions: - publish the image - create the flavor

Returns: - TestCase.EX\_OK - TestCase.EX\_RUN\_ERROR on error

**run\_vmtop** ()

Run Vmtop and generate charts

Raises: Exception on error

**ssh\_retry\_timeout** = 240

**write\_config** ()

Write vmtop.conf

Raises: Exception on error

## Module contents

### functest.opnfv\_tests.openstack.vping package

#### Submodules

### functest.opnfv\_tests.openstack.vping.vping\_ssh module

vPingSSH testcase.

**class** `functest.opnfv_tests.openstack.vping.vping_ssh.VPingSSH` (\*\*kwargs)

Bases: `functest.core.singlevm.SingleVm2`

VPingSSH testcase implementation.

Class to execute the vPing test using a Floating IP to connect to one VM to issue the ping command to the second

**clean** ()

Clean the resources.

It can be overridden if resources must be deleted after running the test case.

**execute ()**

Ping the second VM

Returns: ping exit codes

**prepare ()**

Create the security group and the keypair

It can be overridden to set other rules according to the services running in the VM

Raises: Exception on error

**functest.opnfv\_tests.openstack.vping.vping\_userdata module**

vping\_userdata testcase.

**class** `functest.opnfv_tests.openstack.vping.vping_userdata.VPingUserData (**kwargs)`

Bases: `functest.core.singlevm.VmReady2`

Class to execute the vPing test using userdata and the VM's console

**clean ()**

Clean the resources.

It can be overridden if resources must be deleted after running the test case.

**run (\*\*kwargs)**

Sets up the OpenStack VM instance objects then executes the ping and validates. :return: the exit code from the super.execute() method

**Module contents****Module contents****functest.opnfv\_tests.sdn package****Subpackages****functest.opnfv\_tests.sdn.odl package****Submodules****functest.opnfv\_tests.sdn.odl.odl module**

Define classes required to run ODL suites.

It has been designed for any context. But helpers are given for running test suites in OPNFV environment.

**Example:** `$ python odl.py`

**class** `functest.opnfv_tests.sdn.odl.odl.ODLParser`

Parser to run ODL test suites.

**parse\_args (argv=None)**

Parse arguments.

It can call `sys.exit` if arguments are incorrect.

**Returns:** the arguments from cmdline

**class** `functest.opnfv_tests.sdn.odl.odl.ODLTests` (\*\*kwargs)

Bases: `xtesting.core.robotframework.RobotFramework`

ODL test runner.

`basic_suite_dir = u'/src/odl_test/csit/suites/integration/basic'`

`default_suites = [u'/src/odl_test/csit/suites/integration/basic', u'/src/odl_test/csit`

`neutron_suite_dir = u'/src/odl_test/csit/suites/openstack/neutron'`

`odl_test_repo = u'/src/odl_test'`

`odl_variables_file = u'/src/odl_test/csit/variables/Variables.robot'`

**run** (\*\*kwargs)

Run suites in OPNFV environment

It basically checks env vars to call main() with the keywords required.

**Args:** kwargs: Arbitrary keyword arguments.

**Returns:** EX\_OK if all suites ran well. EX\_RUN\_ERROR otherwise.

**run\_suites** (suites=None, \*\*kwargs)

Run the test suites

It has been designed to be called in any context. It requires the following keyword arguments:

- odlusername,
- odlpassword,
- osauthurl,
- neutronurl,
- osusername,
- osprojectname,
- ospassword,
- odlip,
- odlwebport,
- odlrestconfport.

**Here are the steps:**

- set all RobotFramework\_variables,
- create the output directories if required,
- get the results in output.xml,
- delete temporary files.

**Args:** kwargs: Arbitrary keyword arguments.

**Returns:** EX\_OK if all suites ran well. EX\_RUN\_ERROR otherwise.

**classmethod** `set_robotframework_vars` (odlusername='admin', odlpassword='admin')

Set credentials in csit/variables/Variables.robot.

**Returns:** True if credentials are set. False otherwise.



```
functest.opnfv_tests.sdn.odl.odl.main()
    Entry point
```

## Module contents

## Module contents

### functest.opnfv\_tests.vnf package

#### Subpackages

### functest.opnfv\_tests.vnf.epc package

#### Submodules

### functest.opnfv\_tests.vnf.epc.juju\_epc module

Juju testcase implementation.

```
class functest.opnfv_tests.vnf.epc.juju_epc.JujuEpc (**kwargs)
```

```
    Bases: functest.core.singlevm.SingleVm2
```

```
    Abot EPC deployed with JUJU Orchestrator Case
```

```
check_app (name='abot-epc-basic', status='active')
```

```
    Check application status.
```

```
cidr = '192.168.120.0/24'
```

```
clean ()
```

```
    Clean created objects/functions.
```

```
deploy_orchestrator ()
```

```
    Create network, subnet, router
```

```
    Bootstrap juju
```

```
deploy_vnf ()
```

```
    Deploy ABOT-OAI-EPC.
```

```
execute ()
```

```
    Prepare testcase (Additional pre-configuration steps).
```

```
filename = '/home/opnfv/functest/images/ubuntu-16.04-server-cloudimg-amd64-disk1.img'
```

```
filename_alt = '/home/opnfv/functest/images/ubuntu-14.04-server-cloudimg-amd64-disk1.i
```

```
flavor_alt_disk = 10
```

```
flavor_alt_ram = 4096
```

```
flavor_alt_vcpus = 1
```

```
flavor_disk = 10
```

```
flavor_ram = 2048
```

```
flavor_vcpus = 1
```

```
juju_timeout = '4800'
```

**publish\_image\_alt** (*name=None*)

Publish alternative image

It allows publishing multiple images for the child testcases. It forces the same configuration for all sub-testcases.

Returns: image

Raises: exception on error

**test\_vnf** ()

Run test on ABoT.

**username** = 'ubuntu'

`functest.opnfv_tests.vnf.epc.juju_epc.process_abot_test_result` (*file\_path*)

Process ABoT Result

`functest.opnfv_tests.vnf.epc.juju_epc.sig_test_format` (*sig\_test*)

Process the signaling result to have a short result

`functest.opnfv_tests.vnf.epc.juju_epc.update_data` (*obj*)

Update Result data

## Module contents

### functest.opnfv\_tests.vnf.ims package

#### Submodules

#### functest.opnfv\_tests.vnf.ims.clearwater module

Ease testing any Clearwater deployment

```
class functest.opnfv_tests.vnf.ims.clearwater.ClearwaterTesting (case_name,  
bono_ip,  
ellis_ip)
```

vIMS clearwater base usable by several orchestrators

**availability\_check** (*signup\_code='secret', two\_numbers=False*)

Create one or two numbers

**run\_clearwater\_live\_test** (*public\_domain, signup\_code='secret'*)

Run the Clearwater live tests

It first runs dnsmasq to reach clearwater services by FQDN and then the Clearwater live tests. All results are saved in `ims_test_output.txt`.

**Returns:**

- a dict containing the overall results
- None on error

#### functest.opnfv\_tests.vnf.ims.cloudify\_ims module

CloudifyIms testcase implementation.

```

class functest.opnfv_tests.vnf.ims.cloudify_ims.CloudifyIms (**kwargs)
    Bases: functest.core.cloudify.Cloudify

    Clearwater vIMS deployed with Cloudify Orchestrator Case.

    check_requirements ()
        Check the requirements of the test case.

        It can be overridden on purpose.

    clean ()
        Clean created objects/functions.

    cop_wgn = 'https://github.com/cloudify-cosmo/cloudify-openstack-plugin/releases/download/
    cop_yaml = 'https://github.com/cloudify-cosmo/cloudify-openstack-plugin/releases/download/
    deploy_vnf ()
        Deploy Clearwater IMS.

    execute ()
        Deploy Cloudify Manager.

        network, security group, fip, VM creation

    filename_alt = '/home/opnfv/functest/images/ubuntu-14.04-server-cloudimg-amd64-disk1.i
    flavor_alt_disk = 3
    flavor_alt_ram = 1024
    flavor_alt_vcpus = 1
    quota_port = 50
    quota_security_group = 20
    quota_security_group_rule = 100

    test_vnf ()
        Run test on clearwater ims instance.

```

### functest.opnfv\_tests.vnf.ims.heat\_ims module

HeatIms testcase implementation.

```

class functest.opnfv_tests.vnf.ims.heat_ims.HeatIms (**kwargs)
    Bases: functest.core.singlevm.VmReady2

    Clearwater vIMS deployed with Heat Orchestrator Case.

    clean ()
        Clean created objects/functions.

    create_network_resources ()
        Create all tenant network resources

        It creates a router which gateway is the external network detected. The new subnet is attached to that
        router.

        Raises: exception on error

    deploy_vnf ()
        Deploy Clearwater IMS.

```

```
execute ()
    Prepare Tenant/User
    network, security group, fip, VM creation
filename = '/home/opnfv/functest/images/ubuntu-14.04-server-cloudimg-amd64-disk1.img'
flavor_disk = 3
flavor_ram = 1024
flavor_vcpus = 1
parameters = {'private_mgmt_net_cidr': '192.168.100.0/24', 'private_mgmt_net_gateway'
quota_port = 50
quota_security_group = 20
quota_security_group_rule = 100
run (**kwargs)
    Deploy and test clearwater
    Here are the main actions: - deploy clearwater stack via heat - test the vnf instance
    Returns: - TestCase.EX_OK - TestCase.EX_RUN_ERROR on error
test_vnf ()
    Run test on clearwater ims instance.
```

## Module contents

### functest.opnfv\_tests.vnf.router package

#### Subpackages

#### functest.opnfv\_tests.vnf.router.test\_controller package

#### Submodules

#### functest.opnfv\_tests.vnf.router.test\_controller.function\_test\_exec module

vrouter function test execution module

```
class functest.opnfv_tests.vnf.router.test_controller.function_test_exec.FunctionTestExec (
    vrouter function test execution class
config_reference_vnf (target_vnf, reference_vnf, test_kind)
config_target_vnf (target_vnf, reference_vnf, test_kind)
logger = <logging.Logger object>
result_check (target_vnf, reference_vnf, test_kind, test_list)
run (target_vnf, reference_vnf_list, test_info, test_list)
```

## Module contents

### functest.opnfv\_tests.vnf.router.vnf\_controller package

#### Submodules

#### functest.opnfv\_tests.vnf.router.vnf\_controller.checker module

vrouter test result check module

```

class functest.opnfv_tests.vnf.router.vnf_controller.checker.Checker
    vrouter test result check class

    static load_check_rule (rule_file_dir, rule_file_name, parameter)

    logger = <logging.Logger object>

    static regexp_information (response, rules)

```

#### functest.opnfv\_tests.vnf.router.vnf\_controller.command\_generator module

command generator module for vrouter testing

```

class functest.opnfv_tests.vnf.router.vnf_controller.command_generator.CommandGenerator
    command generator class for vrouter testing

    static command_create (template, parameter)

    static load_template (template_dir, template)

    logger = <logging.Logger object>

```

#### functest.opnfv\_tests.vnf.router.vnf\_controller.ssh\_client module

ssh client module for vrouter testing

```

class functest.opnfv_tests.vnf.router.vnf_controller.ssh_client.SshClient (ip_address,
                                                                 user,
                                                                 pass-
                                                                 word=None,
                                                                 key_filename=None)

    ssh client class for vrouter testing

    close ()

    connect (time_out=10, retrycount=10)

    static error_check (response, err_strs=None)

    logger = <logging.Logger object>

    send (cmd, prompt, timeout=10)

```

#### functest.opnfv\_tests.vnf.router.vnf\_controller.vm\_controller module

vm controll module

```
class functest.opnfv_tests.vnf.router.vnf_controller.vm_controller.VmController (util_info)
    vm controll class

    command_create_and_execute (ssh, test_cmd_file_path, cmd_input_param, prompt_file_path)

    command_execute (ssh, command, prompt)

    command_gen_from_template (command_file_path, cmd_input_param)

    command_list_execute (ssh, command_list, prompt)

    config_vm (vm_info, test_cmd_file_path, cmd_input_param, prompt_file_path)

    connect_ssh_and_config_vm (vm_info, test_cmd_file_path, cmd_input_param,
                               prompt_file_path)

    logger = <logging.Logger object>
```

### functest.opnfv\_tests.vnf.router.vnf\_controller.vnf\_controller module

vrouter controll module

```
class functest.opnfv_tests.vnf.router.vnf_controller.vnf_controller.VnfController (util_info)
    vrouter controll class

    config_vnf (source_vnf, destination_vnf, test_cmd_file_path, parameter_file_path, prompt_file_path)

    logger = <logging.Logger object>

    output_check_result_detail_data (res_data_list)

    result_check (target_vnf, reference_vnf, check_rule_file_path_list, parameter_file_path,
                  prompt_file_path)
```

## Module contents

### Submodules

#### functest.opnfv\_tests.vnf.router.cloudify\_vrouter module

vrouter testcase implementation.

```
class functest.opnfv_tests.vnf.router.cloudify_vrouter.CloudifyVrouter (**kwargs)
    Bases: functest.core.cloudify.Cloudify
```

vrouter testcase deployed with Cloudify Orchestrator.

```
check_console_loop = 12
```

```
check_requirements ()
    Check the requirements of the test case.
```

It can be overridden on purpose.

```
clean ()
    Clean the resources.
```

It can be overridden if resources must be deleted after running the test case.

```
cop_wgn = 'https://github.com/cloudify-cosmo/cloudify-openstack-plugin/releases/download'
```

```
cop_yaml = 'https://github.com/cloudify-cosmo/cloudify-openstack-plugin/releases/download'
```

```

deploy_vnf()
execute()
    Deploy Cloudify Manager. network, security group, fip, VM creation
filename_alt = '/home/opnfv/functest/images/vyos-1.1.8-amd64.qcow2'
flavor_alt_disk = 3
flavor_alt_ram = 1024
flavor_alt_vcpus = 1
test_vnf()

```

### functest.opnfv\_tests.vnf.router.utilvnf module

Utility module of vrouter testcase

```

class functest.opnfv_tests.vnf.router.utilvnf.Utilvnf
    Utility class of vrouter testcase

    static convert_functional_test_result (result_data_list)
get_address (server_name, network_name)
get_blueprint_outputs (cfy_manager_ip, deployment_name)
get_blueprint_outputs_networks (cfy_manager_ip, deployment_name)
get_blueprint_outputs_vnfs (cfy_manager_ip, deployment_name)
get_mac_address (server_name, network_name)
static get_reference_vnf_list (vnf_info_list)
static get_target_vnf (vnf_info_list)
static get_test_scenario (file_path)
static get_vnf_info (vnf_info_list, vnf_name)
get_vnf_info_list (cfy_manager_ip, topology_deploy_name, target_vnf_name)
logger = <logging.Logger object>
output_test_result_json()
request_vm_delete (vnf_info_list)
set_credentials (cloud)
write_result_data (result_data)

```

### functest.opnfv\_tests.vnf.router.vrouter\_base module

vrouter testing base class module

```

class functest.opnfv_tests.vnf.router.vrouter_base.VrouterOnBoardingBase (util,
                                                                    util_info)
    vrouter testing base class

    function_test_vrouter (target_vnf_name, test_info)
        function test execution

```

```
get_vnf_info_list (target_vnf_name)  
test_vnf ()  
    vrouter test execution
```

## Module contents

## Module contents

## Module contents

## functest.utils package

## Submodules

## functest.utils.config module

```
class functest.utils.config.Config
```

```
    fill ()  
    patch_file (patch_file_path)
```

## functest.utils.constants module

## functest.utils.env module

```
functest.utils.env.get (env_var)  
functest.utils.env.string ()
```

## functest.utils.functest\_utils module

```
functest.utils.functest_utils.convert_dict_to_ini (value)  
    Convert dict to oslo.conf input  
functest.utils.functest_utils.convert_ini_to_dict (value)  
    Convert oslo.conf input to dict  
functest.utils.functest_utils.convert_ini_to_list (value)  
    Convert list to oslo.conf input  
functest.utils.functest_utils.convert_list_to_ini (value)  
    Convert list to oslo.conf input  
functest.utils.functest_utils.execute_command (cmd, info=False, error_msg="", verbose=True, output_file=None)  
functest.utils.functest_utils.execute_command_raise (cmd, info=False, error_msg="", verbose=True, output_file=None)
```



`functest.utils.functest_utils.get_nova_version (cloud)`

Get Nova API microversion

Returns:

- Nova API microversion
- None on operation error

`functest.utils.functest_utils.get_openstack_version (cloud)`

Detect OpenStack version via Nova API microversion

It follows [MicroversionHistory](#).

Returns:

- OpenStack release
- Unknown on operation error

`functest.utils.functest_utils.get_parameter_from_yaml (parameter, yfile)`

Returns the value of a given parameter in file.yaml parameter must be given in string format with dots Example: `general.openstack.image_name`

`functest.utils.functest_utils.list_services (cloud)`

Search Keystone services via `$OS_INTERFACE`.

It mainly conforms with [Shade](#) but allows testing vs public endpoints. It's worth mentioning that it doesn't support keystone v2.

**Returns** a list of `munch.Munch` containing the services description

**Raises** `OpenStackCloudException` if something goes wrong during the openstack API call.

`functest.utils.functest_utils.search_services (cloud, name_or_id=None, filters=None)`

Search Keystone services via `$OS_INTERFACE`.

It mainly conforms with [Shade](#) but allows testing vs public endpoints. It's worth mentioning that it doesn't support keystone v2.

**Parameters**

- **name\_or\_id** – Name or id of the desired service.
- **filters** – a dict containing additional filters to use. e.g. `{'type': 'network'}`.

**Returns** a list of `munch.Munch` containing the services description

**Raises** `OpenStackCloudException` if something goes wrong during the openstack API call.

## Module contents

### 1.1.2 Module contents



## CHAPTER 2

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



### f

functest, 29  
 functest.core, 9  
 functest.core.cloudify, 3  
 functest.core.singlevm, 4  
 functest.core.tenantnetwork, 7  
 functest.opnfv\_tests, 28  
 functest.opnfv\_tests.openstack, 19  
 functest.opnfv\_tests.openstack.api, 9  
 functest.opnfv\_tests.openstack.api.connection\_check, 9  
 functest.opnfv\_tests.openstack.cinder, 10  
 functest.opnfv\_tests.openstack.cinder.cinder\_test, 9  
 functest.opnfv\_tests.openstack.patrole, 10  
 functest.opnfv\_tests.openstack.patrole.patrole, 10  
 functest.opnfv\_tests.openstack.rally, 13  
 functest.opnfv\_tests.openstack.rally.rally, 10  
 functest.opnfv\_tests.openstack.refstack, 14  
 functest.opnfv\_tests.openstack.refstack.refstack, 13  
 functest.opnfv\_tests.openstack.shaker, 15  
 functest.opnfv\_tests.openstack.shaker.shaker, 14  
 functest.opnfv\_tests.openstack.tempest, 17  
 functest.opnfv\_tests.openstack.tempest.tempest, 15  
 functest.opnfv\_tests.openstack.vmtop, 18  
 functest.opnfv\_tests.openstack.vmtop.vmtop, 17  
 functest.opnfv\_tests.openstack.vping, 19  
 functest.opnfv\_tests.openstack.vping.vping\_ssh, 18  
 functest.opnfv\_tests.openstack.vping.vping\_userdata, 19  
 functest.opnfv\_tests.sdn, 21  
 functest.opnfv\_tests.sdn.odl, 21  
 functest.opnfv\_tests.sdn.odl.odl, 19  
 functest.opnfv\_tests.vnf, 28  
 functest.opnfv\_tests.vnf.epc, 22  
 functest.opnfv\_tests.vnf.epc.juju\_epc, 21  
 functest.opnfv\_tests.vnf.ims, 24  
 functest.opnfv\_tests.vnf.ims.clearwater, 22  
 functest.opnfv\_tests.vnf.ims.cloudify\_ims, 22  
 functest.opnfv\_tests.vnf.ims.heat\_ims, 23  
 functest.opnfv\_tests.vnf.router, 28  
 functest.opnfv\_tests.vnf.router.cloudify\_vrouter, 26  
 functest.opnfv\_tests.vnf.router.test\_controller, 25  
 functest.opnfv\_tests.vnf.router.test\_controller.functest, 24  
 functest.opnfv\_tests.vnf.router.utilvnf, 27  
 functest.opnfv\_tests.vnf.router.vnf\_controller, 26  
 functest.opnfv\_tests.vnf.router.vnf\_controller.check, 25  
 functest.opnfv\_tests.vnf.router.vnf\_controller.com, 25  
 functest.opnfv\_tests.vnf.router.vnf\_controller.ssh, 25  
 functest.opnfv\_tests.vnf.router.vnf\_controller.vmtop, 25  
 functest.opnfv\_tests.vnf.router.vnf\_controller.vnf, 26

[functest.opnfv\\_tests.vnf.router.vrouter\\_base](#),  
27

[functest.utils](#), 29

[functest.utils.config](#), 28

[functest.utils.constants](#), 28

[functest.utils.env](#), 28

[functest.utils.functest\\_utils](#), 28

**A**

allow\_no\_fip (func`test`.core.tenantnetwork.TenantNetwork1 (func`test`.core.singlevm.SingleVm1 attribute), 4  
     attribute), 8  
 allow\_no\_fip (func`test`.opnfv\_tests.openstack.rally.rally.RallyBase (func`test`.opnfv\_tests.openstack.shaker.shaker.Shaker  
     attribute), 11  
 apply\_blacklist () (func`test`.opnfv\_tests.openstack.rally.rally.RallyBase (func`test`.opnfv\_tests.vnf.router.cloudify\_vrouter.CloudifyVrouter  
     method), 11  
 apply\_blacklist () (func`test`.opnfv\_tests.openstack.rally.rally.RallyJobs (func`test`.core.singlevm.SingleVm1 attribute), 4  
     method), 13  
 apply\_tempest\_blacklist () (func`test`.opnfv\_tests.openstack.tempest.tempest.TempestCommon (func`test`.opnfv\_tests.openstack.tempest.tempest.TempestCommon  
     method), 15  
     method), 15  
 availability\_check () (func`test`.opnfv\_tests.vnf.ims.clearwater.ClearwaterTesting (func`test`.core.singlevm.VmReady1 method), 5  
     method), 22  
     method), 22

**B**

backup\_tempest\_config () (func`test`.opnfv\_tests.openstack.tempest.tempest.TempestCommon (func`test`.opnfv\_tests.openstack.tempest.tempest.TempestCommon  
     static method), 15  
     method), 15  
 basic\_suite\_dir (func`test`.opnfv\_tests.sdn.odl.odl.ODLTests (func`test`.opnfv\_tests.openstack.vmtp.vmtp.Vmtp  
     attribute), 20  
     method), 17  
 blacklist\_file (func`test`.opnfv\_tests.openstack.rally.rally.RallyBase (func`test`.opnfv\_tests.vnf.ims.cloudify\_ims.CloudifyIms  
     attribute), 11  
     method), 23  
 boot\_vm () (func`test`.core.singlevm.VmReady1 (func`test`.opnfv\_tests.vnf.router.cloudify\_vrouter.CloudifyVrouter  
     method), 5  
     method), 26  
 build\_task\_args () (func`test`.opnfv\_tests.openstack.rally.rally.RallyBase (func`test`.opnfv\_tests.openstack.tempest.tempest.TempestCommon  
     method), 11  
     method), 15  
 build\_task\_args () (func`test`.opnfv\_tests.openstack.rally.rally.RallyJobs Checker (class in func`test`.opnfv\_tests.vnf.router.vnf\_controller.checker),  
     method), 13  
     25

**C**

check\_app () (func`test`.opnfv\_tests.vnf.epc.juju\_epc.JujuEpc (func`test`.core.tenantnetwork.TenantNetwork1 at-  
     method), 21  
     tribute), 8  
     tribute), 21  
 CinderCheck (class in func`test`.opnfv\_tests.openstack.cinder.cinder\_test),

9		cloudify_archive (functest.core.cloudify.Cloudify attribute), 3
clean () (functest.core.singlevm.SingleVm1 method), 4		cloudify_container (functest.core.cloudify.Cloudify attribute), 3
clean () (functest.core.singlevm.SingleVm2 method), 5		CloudifyIms (class in functest.opnfv_tests.vnf.ims.cloudify_ims), 22
clean () (functest.core.singlevm.VmReady1 method), 6		CloudifyVrouter (class in functest.opnfv_tests.vnf.router.cloudify_vrouter), 26
clean () (functest.core.singlevm.VmReady2 method), 7		CinderCheckCreate () (functest.opnfv_tests.vnf.router.vnf_controller.command_create_and_execute () (functest.opnfv_tests.vnf.router.vnf_controller.vnf_controller.VmController) method), 26
clean () (functest.core.tenantnetwork.NewProject method), 7		command_execute () (functest.opnfv_tests.vnf.router.vnf_controller.vnf_controller.VmController) method), 26
clean () (functest.core.tenantnetwork.TenantNetwork1 method), 8		Common_from_template () (functest.opnfv_tests.vnf.router.vnf_controller.vnf_controller.VmController) method), 26
clean () (functest.core.tenantnetwork.TenantNetwork2 method), 9		command_list_execute () (functest.opnfv_tests.vnf.router.vnf_controller.vnf_controller.VmController) method), 26
clean () (functest.opnfv_tests.openstack.cinder.cinder_test.CinderCheckCreate () (functest.opnfv_tests.vnf.router.vnf_controller.command_create_and_execute () (functest.opnfv_tests.vnf.router.vnf_controller.vnf_controller.VmController) method), 26		CPingSSHGenerator (class in functest.opnfv_tests.vnf.router.vnf_controller.command_generator) method), 25
clean () (functest.opnfv_tests.openstack.rally.rally.RallyBase command_create_and_execute () (functest.opnfv_tests.vnf.router.vnf_controller.vnf_controller.VmController) method), 26		CPingUserdata (class in functest.opnfv_tests.vnf.router.vnf_controller.command_generator) method), 25
clean () (functest.opnfv_tests.openstack.rally.rally.RallyJobs command_execute () (functest.opnfv_tests.vnf.router.vnf_controller.vnf_controller.VmController) method), 26		concurrency (functest.opnfv_tests.openstack.rally.rally.RallyBase attribute), 11
clean () (functest.opnfv_tests.openstack.shaker.shaker.Shaker (functest.opnfv_tests.vnf.router.vnf_controller.vnf_controller.VmController) method), 26		Config (class in functest.utils.config), 28
clean () (functest.opnfv_tests.openstack.tempest.tempest.TempestCommon_from_template () (functest.opnfv_tests.vnf.router.vnf_controller.vnf_controller.VmController) method), 26		config_reference_vnf () (functest.opnfv_tests.vnf.router.test_controller.function_test_executor) method), 24
clean () (functest.opnfv_tests.openstack.tempest.tempest.TempestHeader method), 26		config_target_vnf () (functest.opnfv_tests.vnf.router.test_controller.function_test_executor) method), 24
clean () (functest.opnfv_tests.openstack.vmt.vmt.Vmt command_list_execute () (functest.opnfv_tests.vnf.router.vnf_controller.vnf_controller.VmController) method), 26		config_vm () (functest.opnfv_tests.vnf.router.vnf_controller.vnf_controller.VmController) method), 26
clean () (functest.opnfv_tests.openstack.vping.vping_ssh.VPingSSHGenerator (class in functest.opnfv_tests.vnf.router.vnf_controller.command_generator) method), 25		config_vnf () (functest.opnfv_tests.vnf.router.vnf_controller.vnf_controller.VmController) method), 26
clean () (functest.opnfv_tests.openstack.vping.vping_userdata.VPingUserdata (class in functest.opnfv_tests.vnf.router.vnf_controller.command_generator) method), 25		configure () (functest.opnfv_tests.openstack.patrole.patrole.Patrole) method), 10
clean () (functest.opnfv_tests.vnf.epc.juju_epc.JujuEpc concurrency (functest.opnfv_tests.openstack.rally.rally.RallyBase attribute), 11		configure () (functest.opnfv_tests.openstack.tempest.tempest.TempestCommon) (functest.opnfv_tests.openstack.tempest.tempest.TempestHeader) method), 15
clean () (functest.opnfv_tests.vnf.epc.juju_epc.JujuEpc Config (class in functest.utils.config), 28		configure () (functest.opnfv_tests.openstack.tempest.tempest.TempestHeader) method), 17
clean () (functest.opnfv_tests.vnf.ims.cloudify_ims.CloudifyIms config_reference_vnf () (functest.opnfv_tests.vnf.router.test_controller.function_test_executor) method), 24		configure () (functest.opnfv_tests.openstack.tempest.tempest.TempestHeader) method), 17
clean () (functest.opnfv_tests.vnf.ims.heat_ims.HeatIms config_target_vnf () (functest.opnfv_tests.vnf.router.test_controller.function_test_executor) method), 24		configure_tempest_update_params () (functest.opnfv_tests.openstack.tempest.tempest.TempestCommon) method), 15
clean () (functest.opnfv_tests.vnf.router.cloudify_vrouter.CloudifyVrouter (class in functest.opnfv_tests.vnf.router.cloudify_vrouter) method), 26		configure_verifier () (functest.opnfv_tests.openstack.tempest.tempest.TempestCommon) method), 15
clean_orphan_security_groups () (functest.core.singlevm.VmReady1 method), 6		Cloudify (class in functest.core.cloudify), 3
clean_orphan_security_groups () (functest.core.singlevm.VmReady1 method), 6		
clean_rally_conf () (functest.opnfv_tests.openstack.rally.rally.RallyBase static method), 11		
clean_rally_conf () (functest.opnfv_tests.openstack.rally.rally.RallyJobs static method), 11		
clean_rally_conf () (functest.opnfv_tests.openstack.tempest.tempest.TempestCommon static method), 15		
clean_rally_logs () (functest.opnfv_tests.openstack.rally.rally.RallyBase static method), 11		
ClearwaterTesting (class in functest.opnfv_tests.vnf.ims.clearwater), 22		
close () (functest.opnfv_tests.vnf.router.vnf_controller.ssh_client.SshClient) method), 15		
Cloudify (class in functest.core.cloudify), 3		



static method), 15

connect () (functest.core.singlevm.SingleVm1 method), 4

connect () (functest.opnfv\_tests.vnf.router.vnf\_controller.ssh\_client.SshClient timeout method), 25

connect\_ssh\_and\_config\_vm () (functest.opnfv\_tests.vnf.router.vnf\_controller.vnf\_controller.VmController timeout method), 26

ConnectionCheck (class in create\_server\_timeout functest.opnfv\_tests.openstack.api.connection\_check), 9

convert\_dict\_to\_ini () (in module functest.utils.functest\_utils), 28

convert\_functional\_test\_result () (functest.opnfv\_tests.vnf.router.util.vnf.UtilVnf static method), 27

convert\_ini\_to\_dict () (in module functest.utils.functest\_utils), 28

convert\_ini\_to\_list () (in module functest.utils.functest\_utils), 28

convert\_list\_to\_ini () (in module functest.utils.functest\_utils), 28

cop\_wgn (functest.opnfv\_tests.vnf.ims.cloudify\_ims.CloudifyIms attribute), 23

cop\_wgn (functest.opnfv\_tests.vnf.router.cloudify\_vrouter.CloudifyVrouter attribute), 26

cop\_yaml (functest.opnfv\_tests.vnf.ims.cloudify\_ims.CloudifyIms method), 23

cop\_yaml (functest.opnfv\_tests.vnf.router.cloudify\_vrouter.CloudifyVrouter attribute), 26

count\_active\_hypervisors () (functest.core.singlevm.VmReady1 method), 6

count\_hypervisors () (functest.core.singlevm.VmReady1 method), 6

create () (functest.core.tenantnetwork.NewProject method), 8

create\_flavor () (functest.core.singlevm.VmReady1 method), 6

create\_flavor\_alt () (functest.core.singlevm.VmReady1 method), 6

create\_floating\_ip\_timeout (functest.core.singlevm.SingleVm1 attribute), 4

create\_network\_resources () (functest.core.tenantnetwork.TenantNetwork1 method), 8

create\_network\_resources () (functest.opnfv\_tests.openstack.vmtptest.Vmtp method), 17

create\_network\_resources () (functest.opnfv\_tests.vnf.ims.heat\_ims.HeatIms method), 23

create\_rally\_deployment () (functest.opnfv\_tests.openstack.rally.rally.RallyBase static method), 11

create\_server\_timeout (functest.core.cloudify.Cloudify attribute), 3

create\_server\_timeout (functest.core.singlevm.VmReady1 attribute), 6

create\_server\_timeout (functest.opnfv\_tests.openstack.shaker.shaker.Shaker attribute), 14

create\_server\_timeout (functest.opnfv\_tests.openstack.vmtptest.Vmtp attribute), 18

create\_verifier () (functest.opnfv\_tests.openstack.tempest.tempest.TempestCommon static method), 15

## D

default\_suites (functest.opnfv\_tests.sdn.odl.odl.ODLTests attribute), 20

deploy\_orchestrator () (functest.opnfv\_tests.vnf.epc.juju\_epc.JujuEpc method), 21

deploy\_vnf () (functest.opnfv\_tests.vnf.epc.juju\_epc.JujuEpc method), 21

deploy\_vnf () (functest.opnfv\_tests.vnf.ims.cloudify\_ims.CloudifyIms method), 23

deploy\_vnf () (functest.opnfv\_tests.vnf.ims.heat\_ims.HeatIms method), 23

deploy\_vnf () (functest.opnfv\_tests.vnf.router.cloudify\_vrouter.CloudifyVrouter method), 26

## E

error\_check () (functest.opnfv\_tests.vnf.router.vnf\_controller.ssh\_client.SshClient static method), 25

excl\_func () (functest.opnfv\_tests.openstack.rally.rally.RallyBase method), 11

excl\_scenario () (functest.opnfv\_tests.openstack.rally.rally.RallyBase static method), 11

execute () (functest.core.cloudify.Cloudify method), 3

execute () (functest.core.singlevm.SingleVm1 method), 4

execute () (functest.opnfv\_tests.openstack.cinder.cinder\_test.CinderCheck method), 10

execute () (functest.opnfv\_tests.openstack.shaker.shaker.Shaker method), 14

execute () (functest.opnfv\_tests.openstack.vping.vping\_ssh.VPingSSH method), 18

execute () (functest.opnfv\_tests.vnf.epc.juju\_epc.JujuEpc method), 21

execute () (functest.opnfv\_tests.vnf.ims.cloudify\_ims.CloudifyIms method), 23

execute () (functest.opnfv\_tests.vnf.ims.heat\_ims.HeatIms method), 23

execute () (functest.opnfv\_tests.vnf.router.cloudify\_vrouter.CloudifyVrouter method), 27

execute\_command() (in module flavor\_alt\_ram (functest.opnfv\_tests.vnf.epc.juju\_epc.JujuEpc attribute), 21  
 functest.utils.functest\_utils), 28

execute\_command\_raise() (in module flavor\_alt\_ram (functest.opnfv\_tests.vnf.ims.cloudify\_ims.CloudifyIm attribute), 23  
 functest.utils.functest\_utils), 28

export\_task() (functest.opnfv\_tests.openstack.rally.rally.RallyBase flavor\_alt\_ram (functest.opnfv\_tests.vnf.router.cloudify\_vrouter.Cloud attribute), 27  
 static method), 11

extra\_alt\_properties flavor\_alt\_vcpus (functest.core.singlevm.VmReady1 attribute), 6

extra\_properties (functest.core.singlevm.VmReady1 flavor\_alt\_vcpus (functest.opnfv\_tests.openstack.tempest.tempest.Tem attribute), 17  
 attribute), 6

flavor\_alt\_vcpus (functest.opnfv\_tests.vnf.epc.juju\_epc.JujuEpc attribute), 21

**F**

file\_is\_empty() (functest.opnfv\_tests.openstack.rally.rally.RallyBase flavor\_alt\_vcpus (functest.opnfv\_tests.vnf.ims.cloudify\_ims.Cloudify attribute), 23  
 static method), 11

filename (functest.core.cloudify.Cloudify attribute), 3

filename (functest.core.singlevm.VmReady1 attribute), 6

filename (functest.opnfv\_tests.openstack.shaker.shaker.Shaker attribute), 3

filename (functest.opnfv\_tests.openstack.vmtptest.vmtptest.Vmtptest flavor\_disk (functest.core.cloudify.Cloudify attribute), 6  
 attribute), 18

filename (functest.opnfv\_tests.vnf.epc.juju\_epc.JujuEpc flavor\_disk (functest.core.singlevm.VmReady1 attribute), 6  
 attribute), 21

filename (functest.opnfv\_tests.vnf.ims.heat\_ims.HeatIm flavor\_disk (functest.opnfv\_tests.openstack.shaker.shaker.Shaker attribute), 14  
 attribute), 24

filename\_alt (functest.core.singlevm.VmReady1 at- flavor\_disk (functest.opnfv\_tests.openstack.vmtptest.vmtptest.Vmtptest attribute), 18  
 tribute), 6

filename\_alt (functest.opnfv\_tests.openstack.tempest.tempest.Tem flavor\_disk (functest.opnfv\_tests.vnf.epc.juju\_epc.JujuEpc attribute), 21  
 attribute), 15

filename\_alt (functest.opnfv\_tests.openstack.tempest.tempest.Tem flavor\_disk (functest.opnfv\_tests.vnf.ims.heat\_ims.HeatIm attribute), 24  
 attribute), 17

filename\_alt (functest.opnfv\_tests.vnf.epc.juju\_epc.JujuEpc flavor\_disk (functest.opnfv\_tests.vnf.ims.heat\_ims.HeatIm attribute), 21  
 attribute), 21

filename\_alt (functest.opnfv\_tests.vnf.ims.cloudify\_ims.CloudifyIm flavor\_disk (functest.opnfv\_tests.vnf.ims.heat\_ims.HeatIm attribute), 24  
 attribute), 23

filename\_alt (functest.opnfv\_tests.vnf.router.cloudify\_vrouter.CloudifyV flavor\_extra\_specs (functest.core.singlevm.VmReady1 attribute), 6  
 router attribute), 27

fill() (functest.utils.config.Config method), 28

flavor\_alt\_disk (functest.core.singlevm.VmReady1 flavor\_ram (functest.core.cloudify.Cloudify attribute), 3  
 attribute), 6

flavor\_alt\_disk (functest.opnfv\_tests.openstack.tempest.tempest.Tem flavor\_ram (functest.core.singlevm.VmReady1 attribute), 6  
 attribute), 17

flavor\_alt\_disk (functest.opnfv\_tests.vnf.epc.juju\_epc.JujuEpc flavor\_ram (functest.opnfv\_tests.vnf.epc.juju\_epc.JujuEpc attribute), 21  
 attribute), 21

flavor\_alt\_disk (functest.opnfv\_tests.vnf.ims.cloudify\_ims.CloudifyIm flavor\_ram (functest.opnfv\_tests.vnf.ims.heat\_ims.HeatIm attribute), 24  
 attribute), 23

flavor\_alt\_disk (functest.opnfv\_tests.vnf.router.cloudify\_vrouter.CloudifyV flavor\_vcpus (functest.core.cloudify.Cloudify attribute), 3  
 router attribute), 27

flavor\_alt\_extra\_specs flavor\_vcpus (functest.opnfv\_tests.openstack.vmtptest.vmtptest.Vmtptest attribute), 18  
 (functest.core.singlevm.VmReady1 attribute), 6

flavor\_alt\_ram (functest.core.singlevm.VmReady1 flavor\_vcpus (functest.opnfv\_tests.vnf.epc.juju\_epc.JujuEpc attribute), 21  
 attribute), 6

flavor\_alt\_ram (functest.opnfv\_tests.openstack.tempest.tempest.Tem flavor\_vcpus (functest.opnfv\_tests.vnf.ims.heat\_ims.HeatIm attribute), 24  
 attribute), 17

func\_list (*functest.opnfv\_tests.openstack.api.connection\_check.ConnectionCheck* attribute), 9

func\_test\_connection\_check (*functest.opnfv\_tests.vnf.epc.juju\_epc* (module), 21)

functest (module), 29

functest.core (module), 9

functest.core.cloudify (module), 3

functest.core.singlevm (module), 4

functest.core.tenantnetwork (module), 7

functest.opnfv\_tests (module), 28

functest.opnfv\_tests.openstack (module), 19

functest.opnfv\_tests.openstack.api (module), 9

functest.opnfv\_tests.openstack.api.connection\_check (module), 9

functest.opnfv\_tests.openstack.cinder (module), 10

functest.opnfv\_tests.openstack.cinder.cinder\_test\_controller (module), 9

functest.opnfv\_tests.openstack.patrole (module), 10

functest.opnfv\_tests.openstack.patrole.patrole (module), 10

functest.opnfv\_tests.openstack.rally (module), 13

functest.opnfv\_tests.openstack.rally.rally (module), 10

functest.opnfv\_tests.openstack.refstack (module), 14

functest.opnfv\_tests.openstack.refstack.refstack (module), 13

functest.opnfv\_tests.openstack.shaker (module), 15

functest.opnfv\_tests.openstack.shaker.shaker (module), 14

functest.opnfv\_tests.openstack.tempest (module), 17

functest.opnfv\_tests.openstack.tempest.tempest (module), 15

functest.opnfv\_tests.openstack.vmtop (module), 18

functest.opnfv\_tests.openstack.vmtop.vmtop (module), 17

functest.opnfv\_tests.openstack.vping (module), 19

functest.opnfv\_tests.openstack.vping.vping\_ssh (module), 18

functest.opnfv\_tests.openstack.vping.vping\_userdata\_keys () (*functest.opnfv\_tests.openstack.vmtop.vmtop.Vmtop* method), 18

functest.opnfv\_tests.sdn (module), 21

functest.opnfv\_tests.sdn.odl (module), 21

functest.opnfv\_tests.sdn.odl.odl (module), 19

functest.opnfv\_tests.vnf (module), 28

functest.opnfv\_tests.vnf.epc (module), 22

functest.opnfv\_tests.vnf.ims (module), 24

functest.opnfv\_tests.vnf.ims.clearwater (module), 22

functest.opnfv\_tests.vnf.ims.cloudify\_ims (module), 22

functest.opnfv\_tests.vnf.ims.heat\_ims (module), 23

functest.opnfv\_tests.vnf.router (module), 28

functest.opnfv\_tests.vnf.router.cloudify\_vrouter (module), 26

functest.opnfv\_tests.vnf.router.test\_controller (module), 25

functest.opnfv\_tests.vnf.router.test\_controller.function\_test\_exec (class in *functest.opnfv\_tests.vnf.router.test\_controller.function\_test\_exec*), 24

functest.opnfv\_tests.vnf.router.utilvnf (module), 27

functest.opnfv\_tests.vnf.router.vnf\_controller (module), 26

functest.opnfv\_tests.vnf.router.vnf\_controller.check\_ssh () (method), 25

functest.opnfv\_tests.vnf.router.vnf\_controller.combo () (method), 25

functest.opnfv\_tests.vnf.router.vnf\_controller.ssh () (method), 25

functest.opnfv\_tests.vnf.router.vnf\_controller.vmtop () (method), 25

functest.opnfv\_tests.vnf.router.vnf\_controller.vnf () (method), 26

functest.opnfv\_tests.vnf.router.vrouter\_base (class in *functest.opnfv\_tests.vnf.router.vrouter\_base.VrouterOnBoarding* method), 27

functest.utils (module), 29

functest.utils.config (module), 28

functest.utils.constants (module), 28

functest.utils.env (module), 28

functest.utils.functest\_utils (module), 28

function\_test\_vrouter () (method), 27

FunctionTestExec (class in *functest.opnfv\_tests.vnf.router.test\_controller.function\_test\_exec*), 24

## G

get () (in module *functest.utils.env*), 28  
 get\_address () (*functest.opnfv\_tests.vnf.router.utilvnf.Utilvnf* method), 27  
 get\_blueprint\_outputs () (*functest.opnfv\_tests.vnf.router.utilvnf.Utilvnf* method), 27  
 get\_blueprint\_outputs\_networks () (*functest.opnfv\_tests.vnf.router.utilvnf.Utilvnf* method), 27  
 get\_blueprint\_outputs\_vnfs () (*functest.opnfv\_tests.vnf.router.utilvnf.Utilvnf* method), 27  
 get\_default\_role () (*functest.core.tenantnetwork.TenantNetwork1* static method), 8  
 get\_environ () (*functest.core.tenantnetwork.NewProject* method), 8  
 get\_execution\_id () (in module *functest.core.cloudify*), 4  
 get\_external\_network () (*functest.core.tenantnetwork.TenantNetwork1* static method), 8  
 get\_mac\_address () (*functest.opnfv\_tests.vnf.router.utilvnf.Utilvnf* method), 27  
 get\_nova\_version () (in module *functest.utils.functest\_utils*), 28  
 get\_openstack\_version () (in module *functest.utils.functest\_utils*), 29  
 get\_parameter\_from\_yaml () (in module *functest.utils.functest\_utils*), 29  
 get\_public\_auth\_url () (*functest.core.tenantnetwork.TenantNetwork1* static method), 8  
 get\_reference\_vnf\_list () (*functest.opnfv\_tests.vnf.router.utilvnf.Utilvnf* static method), 27  
 get\_target\_vnf () (*functest.opnfv\_tests.vnf.router.utilvnf.Utilvnf* static method), 27  
 get\_task\_id () (*functest.opnfv\_tests.openstack.rally.rally.RallyBase* static method), 11  
 get\_test\_scenario () (*functest.opnfv\_tests.vnf.router.utilvnf.Utilvnf* static method), 27  
 get\_verifier\_deployment\_dir () (*functest.opnfv\_tests.openstack.tempest.tempest.TempestCommon* static method), 15  
 get\_verifier\_deployment\_id () (*functest.opnfv\_tests.openstack.rally.rally.RallyBase* static method), 11  
 get\_verifier\_id () (*functest.opnfv\_tests.openstack.tempest.tempest.TempestCommon* static method), 16  
 get\_verifier\_repo\_dir () (*functest.opnfv\_tests.openstack.tempest.tempest.TempestCommon* static method), 16  
 get\_verifier\_result () (*functest.opnfv\_tests.openstack.tempest.tempest.TempestCommon* static method), 16  
 get\_vnf\_info () (*functest.opnfv\_tests.vnf.router.utilvnf.Utilvnf* static method), 27  
 get\_vnf\_info\_list () (*functest.opnfv\_tests.vnf.router.utilvnf.Utilvnf* method), 27  
 get\_vnf\_info\_list () (*functest.opnfv\_tests.vnf.router.vrouter\_base.VrouterOnBoarding* method), 27  
**H**  
 HeatIms (class in *functest.opnfv\_tests.vnf.ims.heat\_ims*), 23  
**I**  
 image\_alt\_format (*functest.core.singlevm.VmReady1* attribute), 6  
 image\_format (*functest.core.singlevm.VmReady1* attribute), 7  
 in\_iterable\_re () (*functest.opnfv\_tests.openstack.rally.rally.RallyBase* static method), 11  
 is\_successful () (*functest.opnfv\_tests.openstack.rally.rally.RallyBase* method), 11  
 is\_successful () (*functest.opnfv\_tests.openstack.tempest.tempest.TempestCommon* method), 16  
 iterations\_amount (*functest.opnfv\_tests.openstack.rally.rally.RallyBase* attribute), 12  
**J**  
 juju\_timeout (*functest.opnfv\_tests.vnf.epc.juju\_epc.JujuEpc* attribute), 21  
 JujuEpc (class in *functest.opnfv\_tests.vnf.epc.juju\_epc*), 21  
**K**  
 RallyBase  
 kill\_existing\_execution () (*functest.core.cloudify.Cloudify* method), 3  
**L**  
 list\_services () (in module *functest.utils.functest\_utils*), 29  
 load\_check\_rule () (*functest.opnfv\_tests.vnf.router.vnf\_controller.checker.Checker* static method), 25  
 load\_template () (*functest.opnfv\_tests.vnf.router.vnf\_controller.command* static method), 25  
 logger (*functest.opnfv\_tests.vnf.router.test\_controller.function\_test\_exec.I* attribute), 24

logger (*functest.opnfv\_tests.vnf.router.utilvnf.Utilvnf* attribute), 27  
 prepare () (*functest.core.cloudify.Cloudify* method), 4  
 prepare () (*functest.core.singlevm.SingleVm1* method), 5  
 logger (*functest.opnfv\_tests.vnf.router.vnf\_controller.checker.Checker* attribute), 25  
 prepare () (*functest.opnfv\_tests.openstack.cinder.cinder\_test.CinderChecker* method), 5  
 logger (*functest.opnfv\_tests.vnf.router.vnf\_controller.command\_generator.CommandGenerator* attribute), 25  
 prepare () (*functest.opnfv\_tests.openstack.shaker.shaker.Shaker* method), 14  
 logger (*functest.opnfv\_tests.vnf.router.vnf\_controller.ssh\_client.SshClient* attribute), 25  
 prepare () (*functest.opnfv\_tests.openstack.vping.vping\_ssh.VPingSSH* method), 14  
 logger (*functest.opnfv\_tests.vnf.router.vnf\_controller.vm\_controller.VmController* attribute), 26  
 prepare\_run () (*functest.opnfv\_tests.openstack.rally.rally.RallyBase* method), 13  
 logger (*functest.opnfv\_tests.vnf.router.vnf\_controller.vnf\_controller.VnfController* attribute), 26  
 prepare\_run () (*functest.opnfv\_tests.openstack.rally.rally.RallyJobs* method), 13  
 prepare\_task () (*functest.opnfv\_tests.openstack.rally.rally.RallyBase* method), 12  
 prepare\_task () (*functest.opnfv\_tests.openstack.rally.rally.RallyJobs* method), 13  
**M**  
 main () (in module *functest.opnfv\_tests.sdn.odl.odl*), 20  
**N**  
 neutron\_suite\_dir  
 (*functest.opnfv\_tests.sdn.odl.odl.ODLTests* attribute), 20  
 NewProject (class in *functest.core.tenantnetwork*), 7  
**O**  
 odl\_test\_repo (*functest.opnfv\_tests.sdn.odl.odl.ODLTests* attribute), 20  
 odl\_variables\_file  
 (*functest.opnfv\_tests.sdn.odl.odl.ODLTests* attribute), 20  
 ODLParser (class in *functest.opnfv\_tests.sdn.odl.odl*), 19  
 ODLTests (class in *functest.opnfv\_tests.sdn.odl.odl*), 20  
 output\_check\_result\_detail\_data ()  
 (*functest.opnfv\_tests.vnf.router.vnf\_controller.vnf\_controller.VnfController* method), 26  
 output\_test\_result\_json ()  
 (*functest.opnfv\_tests.vnf.router.utilvnf.Utilvnf* method), 27  
**P**  
 parameters (*functest.opnfv\_tests.vnf.ims.heat\_ims.HeatIms* attribute), 24  
 parse\_args () (*functest.opnfv\_tests.sdn.odl.odl.ODLParser* method), 19  
 parse\_verifier\_result ()  
 (*functest.opnfv\_tests.openstack.tempest.tempest.TempestCommon* method), 16  
 patch\_file () (*functest.utils.config.Config* method), 28  
 Patrole (class in *functest.opnfv\_tests.openstack.patrole.patrole*), 10  
 port (*functest.opnfv\_tests.openstack.shaker.shaker.Shaker* attribute), 14  
 ports (*functest.core.cloudify.Cloudify* attribute), 4  
**Q**  
 quota\_cores (*functest.opnfv\_tests.openstack.shaker.shaker.Shaker* attribute), 14  
 quota\_instances (*functest.opnfv\_tests.openstack.shaker.shaker.Shaker* attribute), 14  
 quota\_port (*functest.opnfv\_tests.vnf.ims.cloudify\_ims.CloudifyIms* attribute), 23  
 quota\_port (*functest.opnfv\_tests.vnf.ims.heat\_ims.HeatIms* attribute), 24  
 quota\_security\_group  
 (*functest.opnfv\_tests.vnf.ims.cloudify\_ims.CloudifyIms* attribute), 23  
 quota\_security\_group  
 (*functest.opnfv\_tests.vnf.ims.heat\_ims.HeatIms* attribute), 24  
 quota\_security\_group\_rule  
 (*functest.opnfv\_tests.vnf.ims.cloudify\_ims.CloudifyIms* attribute), 23  
 quota\_security\_group\_rule  
 (*functest.opnfv\_tests.vnf.ims.heat\_ims.HeatIms* attribute), 24  
**R**  
 rally\_aar4\_patch\_path  
 (*functest.opnfv\_tests.openstack.rally.rally.RallyBase* method), 13

*attribute*), 12  
 rally\_conf\_path (*functest.opnfv\_tests.openstack.rally.rally.RallyBase* *attribute*), 12  
 rally\_dir (*functest.opnfv\_tests.openstack.rally.rally.RallyBase* *attribute*), 12  
 rally\_scenario\_dir (*functest.opnfv\_tests.openstack.rally.rally.RallyBase* *attribute*), 12  
 RallyBase (class in *functest.opnfv\_tests.openstack.rally.rally*), 10  
 RallyFull (class in *functest.opnfv\_tests.openstack.rally.rally*), 13  
 RallyJobs (class in *functest.opnfv\_tests.openstack.rally.rally*), 13  
 RallySanity (class in *functest.opnfv\_tests.openstack.rally.rally*), 13  
 read\_file () (*functest.opnfv\_tests.openstack.tempest.tempest.TempestCommon* *static method*), 16  
 Refstack (class in *functest.opnfv\_tests.openstack.refstack.refstack*), 13  
 regexp\_information () (*functest.opnfv\_tests.vnf.router.vnf\_controller.checker\_checker* *static method*), 25  
 request\_vm\_delete () (*functest.opnfv\_tests.vnf.router.utilvnf.Utilvnf* *method*), 27  
 result\_check () (*functest.opnfv\_tests.vnf.router.test\_controller.function\_test\_exec.FunctionTestExec* *method*), 24  
 result\_check () (*functest.opnfv\_tests.vnf.router.vnf\_controller.vnf\_controller.VnfController* *method*), 26  
 run () (*functest.core.singlevm.SingleVm1* *method*), 5  
 run () (*functest.core.singlevm.VmReady1* *method*), 7  
 run () (*functest.core.tenantnetwork.TenantNetwork1* *method*), 8  
 run () (*functest.opnfv\_tests.openstack.api.connection\_check.ConnectionCheck* *method*), 9  
 run () (*functest.opnfv\_tests.openstack.patrole.patrole.Patrole* *method*), 10  
 run () (*functest.opnfv\_tests.openstack.rally.rally.RallyBase* *method*), 12  
 run () (*functest.opnfv\_tests.openstack.tempest.tempest.TempestCommon* *method*), 16  
 run () (*functest.opnfv\_tests.openstack.vmtp.vmtp.Vmtp* *method*), 18  
 run () (*functest.opnfv\_tests.openstack.vping.vping\_userdata.VPingUserData* *method*), 19  
 run () (*functest.opnfv\_tests.sdn.odl.odl.ODLTests* *method*), 20  
 run () (*functest.opnfv\_tests.vnf.ims.heat\_ims.HeatIms* *method*), 24  
 run () (*functest.opnfv\_tests.vnf.router.test\_controller.function\_test\_exec.FunctionTestExec* *method*), 24  
 run\_clearwater\_live\_test () (*functest.opnfv\_tests.vnf.ims.clearwater.ClearwaterTesting* *method*), 22  
 run\_suites () (*functest.opnfv\_tests.sdn.odl.odl.ODLTests* *method*), 20  
 run\_task () (*functest.opnfv\_tests.openstack.rally.rally.RallyBase* *method*), 12  
 run\_tests () (*functest.opnfv\_tests.openstack.rally.rally.RallyBase* *method*), 12  
 run\_verifier\_tests () (*functest.opnfv\_tests.openstack.tempest.tempest.TempestCommon* *method*), 16  
 run\_vmtp () (*functest.opnfv\_tests.openstack.vmtp.vmtp.Vmtp* *method*), 18  
**S**  
 search\_services () (in *module* *functest.utils.functest\_utils*), 29  
 send () (*functest.opnfv\_tests.vnf.router.vnf\_controller.ssh\_client.SshClient* *method*), 25  
 set\_credentials () (*functest.opnfv\_tests.vnf.router.utilvnf.Utilvnf* *method*), 27  
 set\_test\_framework\_vars () (*functest.opnfv\_tests.sdn.odl.odl.ODLTests* *class method*), 20  
 Shaker (class in *functest.opnfv\_tests.openstack.shaker.shaker*), 14  
 shared\_network (*functest.opnfv\_tests.openstack.shaker.shaker.Shaker* *attribute*), 14  
 shared\_network (*functest.opnfv\_tests.openstack.shaker.shaker.Shaker* *attribute*), 8  
 shared\_network (*functest.opnfv\_tests.openstack.rally.rally.RallyBase* *attribute*), 12  
 shared\_network (*functest.opnfv\_tests.openstack.tempest.tempest.TempestCommon* *attribute*), 16  
 singlevm\_format () (in *module* *functest.opnfv\_tests.vnf.epc.juju\_epc*), 22  
 SingleVm1 (class in *functest.core.singlevm*), 4  
 SingleVm2 (class in *functest.core.singlevm*), 5  
 ssh\_connect\_loops (*functest.core.cloudify.Cloudify* *attribute*), 4  
 ssh\_connect\_loops (*functest.core.singlevm.SingleVm1* *attribute*), 5  
 ssh\_connect\_loops (*functest.opnfv\_tests.openstack.shaker.shaker.Shaker* *attribute*), 14  
 ssh\_connect\_timeout (*functest.core.singlevm.SingleVm1* *attribute*), 5  
 ssh\_retry\_timeout (*functest.opnfv\_tests.openstack.vmtp.vmtp.Vmtp* *attribute*), 18

SshClient (class in test\_vnf() (functest.opnfv\_tests.vnf.ims.cloudify\_ims.CloudifyIms  
 functest.opnfv\_tests.vnf.router.vnf\_controller.ssh\_client), method), 23  
 25 test\_vnf() (functest.opnfv\_tests.vnf.ims.heat\_ims.HeatIms  
 stests (functest.opnfv\_tests.openstack.rally.rally.RallyBase method), 24  
 attribute), 12 test\_vnf() (functest.opnfv\_tests.vnf.router.cloudify\_vrouter.CloudifyVro  
 stests (functest.opnfv\_tests.openstack.rally.rally.RallyJobs method), 27  
 attribute), 13 test\_vnf() (functest.opnfv\_tests.vnf.router.vrouter\_base.VrouterOnBoa  
 string() (in module functest.utils.env), 28 method), 28  
 support\_dir (functest.opnfv\_tests.openstack.rally.rally.RallyBase  
 attribute), 12

**U**  
 update\_compute\_section()  
 (functest.opnfv\_tests.openstack.tempest.tempest.TempestCommon  
 task\_dir (functest.opnfv\_tests.openstack.rally.rally.RallyBase method), 16  
 attribute), 12 update\_data() (in module  
 task\_succeed() (functest.opnfv\_tests.openstack.rally.rally.RallyBase functest.opnfv\_tests.vnf.epc.juju\_epc), 22  
 static method), 12 update\_default\_role()  
 task\_timeout (functest.opnfv\_tests.openstack.rally.rally.RallyBase functest.opnfv\_tests.openstack.tempest.tempest.TempestCommon  
 attribute), 12 method), 16  
 task\_timeout (functest.opnfv\_tests.openstack.rally.rally.RallyBase keystone\_default\_role()  
 attribute), 13 (functest.opnfv\_tests.openstack.rally.rally.RallyBase  
 task\_timeout (functest.opnfv\_tests.openstack.rally.rally.RallyJobs static method), 12  
 attribute), 13 update\_network\_section()  
 temp\_dir (functest.opnfv\_tests.openstack.rally.rally.RallyBase (functest.opnfv\_tests.openstack.tempest.tempest.TempestCommon  
 attribute), 12 method), 16  
 tempest\_blacklist update\_rally\_logs()  
 (functest.opnfv\_tests.openstack.tempest.tempest.TempestCommon functest.opnfv\_tests.openstack.rally.rally.RallyBase  
 attribute), 16 static method), 12  
 tempest\_conf\_yaml update\_rally\_regex()  
 (functest.opnfv\_tests.openstack.tempest.tempest.TempestCommon functest.opnfv\_tests.openstack.tempest.tempest.TempestCommon  
 attribute), 16 method), 16  
 tempest\_custom (functest.opnfv\_tests.openstack.tempest.tempest.TempestCommon update\_tempest\_connection()  
 attribute), 16 (functest.opnfv\_tests.openstack.tempest.tempest.TempestCommon  
 tempest\_public\_blacklist method), 16  
 (functest.opnfv\_tests.openstack.tempest.tempest.TempestCommon tempest\_conf\_file()  
 attribute), 16 (functest.opnfv\_tests.openstack.tempest.tempest.TempestCommon  
 TempestCommon (class in static method), 16  
 functest.opnfv\_tests.openstack.tempest.tempest), update\_validation\_section()  
 15 (functest.opnfv\_tests.openstack.tempest.tempest.TempestCommon  
 TempestHeat (class in method), 16  
 functest.opnfv\_tests.openstack.tempest.tempest), upload\_cfy\_plugins()  
 16 (functest.core.cloudify.Cloudify method),  
 TempestHorizon (class in 4  
 functest.opnfv\_tests.openstack.tempest.tempest), username (functest.core.cloudify.Cloudify attribute), 4  
 17 username (functest.core.singlevm.SingleVm1 attribute),  
 5  
 template\_dir (functest.opnfv\_tests.openstack.rally.rally.RallyBase  
 attribute), 12 username (functest.opnfv\_tests.openstack.shaker.shaker.Shaker  
 attribute), 14  
 TenantNetwork1 (class in attribute), 22  
 functest.core.tenantnetwork), 8 username (functest.opnfv\_tests.vnf.epc.juju\_epc.JujuEpc  
 TenantNetwork2 (class in attribute), 22  
 functest.core.tenantnetwork), 8 users\_amount (functest.opnfv\_tests.openstack.rally.rally.RallyBase  
 tenants\_amount (functest.opnfv\_tests.openstack.rally.rally.RallyBase attribute), 12  
 attribute), 12 Utilvnf (class in functest.opnfv\_tests.vnf.router.utilvnf),  
 test\_vnf() (functest.opnfv\_tests.vnf.epc.juju\_epc.JujuEpc 27  
 method), 22

## V

*verify\_report()* (*functest.opnfv\_tests.openstack.rally.rally.RallyBase*  
*static method*), 12  
*visibility* (*functest.core.singlevm.VmReady1* at-  
*tribute*), 7  
*visibility* (*functest.opnfv\_tests.openstack.rally.rally.RallyBase*  
*attribute*), 12  
*visibility* (*functest.opnfv\_tests.openstack.tempest.tempest.TempestCommon*  
*attribute*), 16  
*VmController* (class in  
*functest.opnfv\_tests.vnf.router.vnf\_controller.vnf\_controller*),  
 25  
*VmReady1* (class in *functest.core.singlevm*), 5  
*VmReady2* (class in *functest.core.singlevm*), 7  
*Vmtp* (class in *functest.opnfv\_tests.openstack.vmt.vmt*),  
 17  
*VnfController* (class in  
*functest.opnfv\_tests.vnf.router.vnf\_controller.vnf\_controller*),  
 26  
*volume\_service\_type*  
*(functest.opnfv\_tests.openstack.rally.rally.RallyBase*  
*attribute)*, 12  
*volume\_timeout* (*functest.opnfv\_tests.openstack.cinder.cinder\_test.CinderCheck*  
*attribute*), 10  
*volume\_version* (*functest.opnfv\_tests.openstack.rally.rally.RallyBase*  
*attribute*), 13  
*VPingSSH* (class in *functest.opnfv\_tests.openstack.vping.vping\_ssh*),  
 18  
*VPingUserdata* (class in  
*functest.opnfv\_tests.openstack.vping.vping\_userdata*),  
 19  
*VrouterOnBoardingBase* (class in  
*functest.opnfv\_tests.vnf.router.vrouter\_base*),  
 27

## W

*wait\_for\_execution()* (in module  
*functest.core.cloudify*), 4  
*write\_config()* (*functest.opnfv\_tests.openstack.vmt.vmt.Vmtp*  
*method*), 18  
*write\_result\_data()*  
*(functest.opnfv\_tests.vnf.router.util.vnf.Utilvnf*  
*method)*, 27