OPNFV Functest Documentation

Release master

Functest <opnfv-tech-discuss@lists.opnfv.org>

Oct 26, 2018

Contents

Contents:

CHAPTER 1

functest

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.

create_server_timeout = 600

execute()
Deploy Cloudify Manager.

filename = '/home/opnfv/functest/images/cloudify-manager-premium-4.0.1.qcow2'

flavor_disk = 40

flavor_ram = 4096

flavor_vcpus = 2

ports = [80, 443, 5671, 53333]

prepare()
Create the security group and the keypair

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

Raises: Exception on error

ssh_connect_loops = 12

username = 'centos'

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).

clean()

Clean the resources.

It can be overriden 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 overriden to execute any command.

Returns: echo exit codes

prepare()

Create the security group and the keypair

It can be overriden 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 overriden 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 subtestcases.

Returns: vm

Raises: expection on error

check_regex_in_console(*name*, *regex=' login: '*, *loop=1*)
Wait for specific message in console

Returns: True or False on errors

clean()
Clean the resources.

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

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: expection 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: expection 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: expection 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: expection 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 overriden 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*)
Bases: object

Ease creating new projects/users

clean()
Remove projects/users

create()
Create projects/users

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).

cidr = '192.168.120.0/24'

clean()
Clean the resources.

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

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 overriden if resources must be deleted after running the test case.

Module contents

functest.opnfv_tests package

Subpackages

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**
Bases: object

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

Module contents

1.1.2 Module contents

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

f

Index