
OVN Scale Test Documentation

Release 0.1

Lei Huang

April 12, 2016

1	Contents	3
1.1	Overview	3
1.2	Installation	3
1.3	Rally step-by-step	4
2	Indices and tables	9

OVN Scale Test is a plugin of openstack Rally.

OVN Scale Test is intended to provide the community with a OVN control plan scalability test tool that is capable of performing **specific, complicated** and **reproducible** test cases on **simulated** scenarios.

When something fails, performs slowly or doesn't scale, it's really hard to answer different questions on "what", "why" and "where" without a solid scalability testing framework.

For start using this tool, you need to have a [Rally](#) installed, the workflow is also similar to Rally's.

Contents

1.1 Overview

OVN scale test automates and unifies multi-node test environment deployment. It can be used as a basic tool for an *CI system* that would continuously improve its performance, stability and scalability.

1.2 Installation

Because **OVN Scale Test** is mostly a plugin of Rally, you should install Rally firstly, then install it on top of Rally.

1.2.1 Install Rally

Rally is dedicated to OpenStack now(this situation will be change soon, Rally developers are splitting Rally from OpenStack: [Rally Brainstorm](#)), **OVN Scale Test** makes some changes on Rally to skip OpenStack specific code, and these changes have not been pushed to Rally upstream. Hence you need to use a forked Rally from repo <https://github.com/l8huang/rally.git>, you can clone it and install it by running its installation script:

```
$ git clone https://github.com/l8huang/rally.git
$ cd rally
$ ./install_rally.sh
```

If you execute the script as regular user, Rally will create a new virtual environment in `~/rally/` and install in it, and will use sqlite as database backend. If you execute the script as root, Rally will be installed system wide. For more installation options, please refer to the [Rally installation](#) page.

Note: Rally requires Python version 2.7 or 3.4.

1.2.2 Install OVN Scale Test

After Rally is installed, you can install **OVN Scale Test** now. Get a copy of it from repo <https://github.com/openvswitch/ovn-scale-test.git>:

```
$ git clone https://github.com/openvswitch/ovn-scale-test.git
$ cd ovn-scale-test
$ ./install.sh
```

If installed successful, you can see:

```
=====
Installation of OVN scale test is done!
=====
```

In order to work with Rally you have to enable the virtual environment with the command:

```
. /home/<user>/rally/bin/activate
```

You need to run the above command on every new shell you opened before using Rally, but just once per session.

Information about your Rally installation:

```
* Method: virtualenv
* Virtual Environment at: /home/<user>/rally
* Configuration file at: /home/<user>/rally/etc/rally
* Samples at: /home/<user>/rally/samples
```

Run `./install.sh` with option `--help` to have a list of all available options:

```
$ ./install.sh --help
Usage: install.sh [options]

This script will install OVN scale test tool in your system.

Options:
  -h, --help                Print this help text
  -v, --verbose              Verbose mode
  -s, --system               Install system-wide.
  -d, --target DIRECTORY    Install Rally virtual environment into DIRECTORY.
                             (Default: /home/lhuang8/rally if not root).
  --url                      Git repository public URL to download Rally OVS from.
                             This is useful when you have only installation script and want to install Rally
                             from custom repository.
                             (Default: https://github.com/l8huang/rally-ovs.git).
                             (Ignored when you are already in git repository).
  --branch                   Git branch name or git tag (Rally OVS release) to install.
                             (Default: latest - master).
                             (Ignored when you are already in git repository).
  -y, --yes                  Do not ask for confirmation: assume a 'yes' reply
                             to every question.
  -p, --python EXE           The python interpreter to use. Default: /home/lhuang8/rally/bin/python
  --develop                  Install Rally with editable source code try.
                             (Default: false)
  --no-color                 Disable output coloring.
```

Notes: `--system` option is not supported yet.

1.3 Rally step-by-step

In the following tutorial, we will guide you step-by-step through different use cases that might occur in Rally, starting with the easy ones and moving towards more complicated cases.

1.3.1 Step 1. Setting up the environment

- *Setup a password-less SSH key(optional)*
- *Create a deployment in Rally*
- *Create ovs-sandboxes on farm nodes*

Setup a password-less SSH key(optional)

This step is optional. Rally needs root privilege for installing OVS on farm node, it's convenient to use a password-less SSH key to login as root and avoid type your password in config file.

1. Generate a password-less ssh key on rally node

```
$ ssh-keygen # just hit enter when ask for password
```

2. Copy the generated id_rsa.pub to all of other nodes, e.g:

```
$ ssh [username]@[hostname]
$ sudo mkdir /root/.ssh
$ sudo scp [username]@[rally-node]:/path/to/id_rsa.pub /root/.ssh
$ sudo cat .ssh/id_rsa.pub >> /root/.ssh/authorized_keys
```

3. Check if the password-less SSH key is setup properly, try ssh connect to other nodes from rally node:

```
$ ssh [username]@[hostname]
```

If success, you will get a bash prompt directly without password prompt

Create a deployment in Rally

You have to provide **OVN Scale Test** with a deployment it is going to run task. You can put the information about your hosts into a JSON configuration file, then `rally-ovs` is used to deploy a ovn controller node and one or more farm nodes.

Following config file creates a deployment with:

- one OVN controller node, runs an ovn-northd, a northbound ovsdb-server and a southbound ovsdb-server.
- two OVN sandbox farm node, runs ovs-sandboxes(each ovs-sandbox simulates one HV, consists of an ovn-controller, an ovs-vswitchd, and an ovsdb-server).

Listing 1.1: samples/deployments/ovn-multihost.json

```
{
  "type": "OvnMultihostEngine",
  "controller": {
    "type": "OvnSandboxControllerEngine",
    "deployment_name": "ovn-controller-node",
    "ovs_repo": "https://github.com/openvswitch/ovs.git",
    "ovs_branch": "master",
    "ovs_user": "rally",
    "net_dev": "eth1",
    "controller_cidr": "192.168.10.10/16",
    "provider": {
      "type": "OvsSandboxProvider",
      "credentials": [
        {
          "host": "192.168.20.10",
```

```
        "user": "root"}
    ]
  },
  "nodes": [
    {
      "type": "OvnSandboxFarmEngine",
      "deployment_name": "ovn-farm-node-0",
      "ovs_repo" : "https://github.com/openvswitch/ovs.git",
      "ovs_branch" : "master",
      "ovs_user" : "rally",
      "provider": {
        "type": "OvsSandboxProvider",
        "credentials": [
          {
            "host": "192.168.20.20",
            "user": "root"}
        ]
      }
    }
  ]
}
```

Notes:

- Replace ‘username’ with a suitable username
- “controller_cidr” is a private address, ovn controller node’s ovsdb-server will listen on this ip. The IP address will be added to “net_dev” as a IP alias.
- With config file, one ovn controller node and two farm nodes will be deployed.

Run rally-ovs to create the deployment:

```
$ ./rally/bin/activate
$ rally-ovs deployment create --file ovn-scalability-test/deployments/ovn-multihost.json --name ovn-rally
...
+-----+-----+-----+-----+
| uuid                                | created_at                | name                | status                |
+-----+-----+-----+-----+
| 320115a1-0613-47a5-91f3-fc0a29a86e64 | 2016-04-12 12:47:54.144207 | ovn-multihost      | deploy->finish       |
+-----+-----+-----+-----+
Using deployment: 320115a1-0613-47a5-91f3-fc0a29a86e64
```

After this command executed successfully, the ovn controller node has an running ovn-northd, northbound ovsdb-server and southbound ovsdb-server now, but the two farm nodes have no ovs-sandboxes running on them, you need use task command to create ovs-sandboxes.

Notes: the command used here is rally-ovs, not rally

Run rally-ovs deployment with option --help to have a list of all available options:

```
$ rally-ovs deployment --help
usage: rally-ovs deployment [-h] {config,create,destroy,list,recreate,use} ...
Set of commands that allow you to manage ovs deployments.
Commands:
  config      Display configuration of the deployment.
  create      Create new deployment.
  destroy     Destroy existing deployment.
  list        List existing deployments.
```

```

recreate    Destroy and create an existing deployment.
use         Set active deployment.
optional arguments:
-h, --help  show this help message and exit

```

Create ovs-sandboxes on farm nodes

Now that we have a working and registered deployment, we can create ovs-sandboxes to simulate HVs. Let's create some ovs-sandboxes on farm node 0 as a example

Listing 1.2: samples/tasks/scenarios/ovn-sandbox/create_sandbox.json

```

{
  "version": 2,
  "title": "Create sandbox",
  "subtasks": [{
    "title": "Create sandbox on farm 0",
    "workloads": [{
      "name": "OvnSandbox.create_sandbox",
      "args": {
        "sandbox_create_args": {
          "farm": "ovn-farm-node-0",
          "amount": 3,
          "batch": 10,
          "start_cidr": "192.168.64.0/16",
          "net_dev": "eth1",
          "tag": "ToR1"
        }
      }
    }],
    "runner": {"type": "serial", "times": 1},
    "context": {
      "ovn_multihost": { "controller": "ovn-controller-node" }
    }
  }]
}

```

To start a Rally task, run the task start command:

```

$ rally-ovs task start samples/tasks/scenarios/ovn-sandbox/create_sandbox.json
-----
Preparing input task
-----

Input task is:
<Your task config here>
...
Task config is valid :)
-----

Task 41dd9197-6c74-4b74-a081-58b866b40de0: started
-----

Benchmarking... This can take a while...
...

Task 41dd9197-6c74-4b74-a081-58b866b40de0: finished
-----

```

```
test scenario OvnSandbox.create_sandbox
args position 0
```

Response Times (sec)								
action	min	median	90%ile	95%ile	max	avg	success	count
sandbox.create_sandbox	0.935	0.935	0.935	0.935	0.935	0.935	100.0%	1
total	0.956	0.956	0.956	0.956	0.956	0.956	100.0%	1

Load duration: 1.08041000366

Full duration: 1.10536003113

HINTS:

* To plot HTML graphics with this data, run:

```
rally task report 41dd9197-6c74-4b74-a081-58b866b40de0 --out output.html
```

* To generate a JUnit report, run:

```
rally task report 41dd9197-6c74-4b74-a081-58b866b40de0 --junit --out output.xml
```

* To get raw JSON output of task results, run:

```
rally task results 41dd9197-6c74-4b74-a081-58b866b40de0
```

1.3.2 Step 2. Rally input task format

- *Basic input task syntax*

Basic input task syntax

Rally comes with a really great collection of plugins and in most real-world cases you will use multiple plugins to test your OpenStack cloud. Rally makes it very easy to run **different test cases defined in a single task**.

Rally has two version of task format, in this tutorial we use v2.

To do so, use the following syntax:

Indices and tables

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