Yardstick

Architecture & status
June 2015
Hans Feldt

Recap

- A *framework* to test *non functional characteristics* of an NFV Infrastructure as perceived from an application
- Different aspects:
  - Network
  - Storage
  - Compute
  - Security?
- Verify quality of service

Framework requirements

- Cloud: OpenStack Juno or later
- Orchestrate deployment of an application
- Support placement groups (affinity)
- Measure some aspect from “inside” the application
- Get the output and store it
  - Can be used for post processing external to yardstick
- Analyze data
  - Simple SLA success/failure built in
- Possible to use in CI activities
  - Continuously measure performance and capture degradation early
  - Work in progress

Framework overview

- Inspired by rally
- Written in python
- Command line tool *yardstick*
- Run on a system (laptop) with cloud connectivity
- Benchmark task described in a configuration file
- Application resources deployed using the cloud orchestration service
- Runs test scripts inside VMs using SSH
- ...

Concepts

- Scenario
  - Type/class of measurement for example Ping, Pktgen, (Iperf, LmBench, ...)
- Context
  - The set of cloud resources used by a benchmark (scenario)
  - Simplified Heat template (context is converted into a Heat template)
  - Deployed into a stack using Heat
  - Context - 1:1 - Stack
- Runner
  - Logic that determines *how* the test is run
  - Number of iterations, input value stepping, duration etc
  - Runs in a subprocess
- SLA
  - Some limit to be verified (specific to scenario), for example max latency
  - Action to take: assert, monitor etc
- Benchmark task configuration file
  - Specifies scenarios(s), runner(s) and context(s)
  - Input to yardstick command

“Hello world” task example

```
schema: "yardstick:task:0.1"

scenarios:
-
  type: Ping
  host: client.demo
  target: 8.8.8.8
  runner:
    type: Duration
    duration: 60
    interval: 1
```

```
context:
  name: demo
  image: cirros-0.3.3
  flavor: m1.tiny
  user: cirros
  servers:
    client:
      floating_ip: true
  networks:
    test:
      cidr: '10.0.1.0/24'
      external_network: "net04_ext"
```

Example usage:

```
$ export OS_AUTH_URL=...
$ yardstick ping.yaml
```

Benchmark Samples

- Deploy a single VM and ping an Internet server
- Deploy two VMs on the ***same*** network and let the client VM ping the server VM
- Deploy two VMs on ***different*** networks and let the client VM ping the server VM
- Deploy two VMs on the ***same*** network and let the client VM ping the server VM using an OpenStack Heat HOT template
- Using the same context(stack), run several benchmarks in serial or parallel
- Iperf3, pktgen

Other content

- Image building support
  - yardstick-img-modify tool to build an image (using qemu-nbd)
  - Uses ubuntu server cloud image as base and adds required packages
- Unit & style testing
  - Same setup as OpenStack projects
  - Tox, mock, flake8, etc
  - run_tests.sh script at top for hookup with gerrit gate test
- Documentation
  - TBD; generated from reStructuredText (rst) files

Backlog

- https://jira.opnfv.org/browse/YARDSTICK/

Ideas/Future

- Stimuli
  - An external script configured in a benchmark task config
  - Runs single shot after some time or periodically to generate some infrastructure event, examples:
    - Instance live migration
    - Interface down/up
    - https://jira.opnfv.org/browse/YARDSTICK-44
- Service chaining; framework support
- Built in simple visualization using some plot tool
- Database backend for storing results. Rally has it
- Plugins for scenarios, runners. See rally