Resource Management

Resource Management:

- Proposed name for the project: Promise
- Proposed name for the repository: promise
- Project Categories: (Requirements)

Project description:

Operators require to reserve resources for future usages. Such events could be predicted traffic increase due to concert, natural disasters etc. However, at present, OpenStack cannot reserve resources for future usage. OpenStack requires instant instantiation of Virtual Machines (VMs) in order to occupy resources intended to be reserved. Blazar can reserve compute resources for future by keeping the VMs in shelved mode. However, such reserved resources can also be used for scaling out rather than new VM instantiation. Blazar do not support network and storage resource reservation yet.

Fig. 1 shows the Resource Reservation for future usage scenario. Providing a start time to allocate the reserved resources is one approach to solve this problem. VMs won’t be instantiated until the start time arrives. This also provides a resource guarantee for the period expressed by start and end time. In Fig. 1, User/Client represents the requestor and the consequent consumer of the resources who will own the VMs to be instantiated on those resources. In ETSI NFV terminology, User/Client would be the NFV Orchestrator (NFVO) and VNF Manager (VNFM).

Along with Resource Reservation, this Project includes Capacity Management where the VIM performs the resource pool capacity management and interacts with NFVO to exchange capacity information.

Scope:

- Describe the problem being solved by project
  - Based on traffic as well as load prediction or any other operational events, an Operator requires to reserve resources for future usage. OpenStack cannot reserve resources for future usage. In this project, this problem will be solved.

- Specify any interface/API specification proposed
  - Or-Vi, Vi-Vnfm reference points of ETSI NFV E2E Architecture framework will be detailed to enable resource reservation for future usage along with resource capacity management.

- Identify a list of features and functionality will be developed.
  - VIM (e.g. OpenStack) shall be able to reserve resources (compute, network, storage) for future usage
  - VIM (e.g. OpenStack) northbound interfaces (Or-Vi, Vi-Vnfm) shall be able to receive and respond to such resource reservation request and the consequent resource allocation, and capacity management on behalf of NFVI. In this regard, the following interfaces shall be defined taking GS NFV-MAN 001 V0.6.4 (2014-10) into account:
    - Virtualised resources catalogue management (GS NFV-MAN 001: Chapter 7.3.1)
    - Virtualised resources capacity management (GS NFV-MAN 001: Chapter 7.3.2)
    - Virtualised resources management (GS NFV-MAN 001: Chapter 7.3.3)

- Identify what is in or out of scope. So during the development phase, it helps reduce discussion.
  - In scope:
- Resource reservation
  - Set start time and end time for allocation
  - Increase/decrease reserved resources
- Resource capacity management
  - Add/remove resource pools
  - Discovery of available resource capacity on resource providers
  - Monitoring of available resource capacity on resource providers
  - Increase/decrease available resource capacity
- VIM northbound interfaces (see GS NFV-MAN 001: Chapter 7.3.1, 7.3.2, 7.3.3)
  - Receive/Reply resource reservation requests
  - Receive/Reply resource capacity pool requests
  - Allocate reserved resources when start time arrives
  - Terminate allocated resources before end time arrives
- Out of scope
  - Optimization of resource usage efficiency
  - Algorithm to avoid waste of resources reserved for future usage

Describe how the project is extensible in future
- It can be extended to resource management based on policy
  - e.g. VNF mapping based on DPDK support, affinity rules etc.
  - this project focuses on realizing the reservation/capacity management feature, not on determining policy for such features

Testability: (optional, Project Categories: Integration & Testing)
N/A

Documentation: (optional, Project Categories: Documentation)
N/A

Dependencies:
- Identify similar projects is underway or being proposed in OPNFV or upstream project
  - Copper (Virtual Infrastructure Deployment Policies)
- Identify any open source upstream projects and release timeline.
  - StormForge - https://github.com/stormstack/stormforge
    - Plan is to align release timelines.
  - Stormify - https://github.com/stormstack/stormify
    - Plan is to align release timelines.
- Are there any external fora or standard development organization dependencies. If possible, list and informative and normative reference specifications.
  - ETSI NFV MANO GS
  - ETSI NFV INF GSs
- If project is an integration and test, identify hardware dependency.
  - This project shall not have any hardware dependency

Committers and Contributors:
- Name of and affiliation of the maintainer:
- Names and affiliations of the committers:
  - Ashiq Khan (DOCOMO: khan@nttdocomo.com)
  - Peter Lee (ClearPath: plee@clearpathnet.com)
  - Ryota Mibu (NEC: r-mibu@cq.jp.neclab.com)
  - Ravi Chunduru (ClearPath: rchunduru@clearpathnet.com)
  - Carlos Goncalves (NEC: Carlos.Goncalves@neclab.eu)
  - Serge Manning (Sprint: Serge.Manning@sprint.com)
  - Dirk Kutscher (NEC: dirk.kutscher@neclab.eu)
  - Siva Busa (ClearPath: SBusa@clearpathnet.com)
  - Hironobu Maeda (ClearPath: HMaeda@clearpathnet.com)
  - Arturo Martin De Nicolas (Ericsson: arturo.martin-de-nicolas@ericsson.com)
  - Murali Birru (murali.yadav@calsoftlabs.com)
- Any other contributors: TBD

Planned deliverables
- Describe the project release package as OPNFV or open source upstream projects.
  - Specification of the requirements shall be completed by March 2015
  - Includes API Docs and Northbound Interface specification Docs
  - Reference implementation of the requirements specification shall be completed by March 2015
  - StormForge open source software will be contributed to OPNFV with complete functional implementation of the N/B API specifications covering Resource Capacity management and Resource Reservation service
  - Stormify open source software will be contributed to OPNFV with complete functional implementation of data model abstractions with dynamic generation of REST/JSON N/B API specifications
pkgcloud open source software will be integrated to provide library abstraction to OpenStack APIs.

The reference implementation will be designed to operate as a plug-in to OpenStack that can run independently from the underlying OpenStack deployment. Current primary target OpenStack compatibility is Juno but may also be compatible to older OpenStack releases (compatibility report will be provided along with the code contribution).

- Description of potential use cases (i.e. different ways of realizing resource reservation/capacity management)
- If project deliverables have multiple dependencies across other project categories, described linkage of the deliverables.
- N/A

Proposed Release Schedule:

- When is the first release planned?
  - March, 2015.
- Will this align with the current release cadence
  - Yes.