**Project Name:**

* Proposed name for the project: *OSCAR* (Open Source Carrier grade ARchitecture)
* Proposed name for the repository: repo-oscar
* Project Categories: (Integration & Testing, Collaborative Development)

**Project description:**

Project “OSCAR” provides a platform that can be used to assemble and test a base set of infrastructure components for the OPNFV. A few example VNFs will be provided to validate the integrity and the overall functionality of the OPNFV implementation platform. The goals of the Oscar Project is closely aligned with what the OPNFV technical steering committee (TSC) is envisioning for the first release. This is fast track the integration of the core open source network components, learn from their differences and commonality of deployment experience, and feed that back into producing a more flexible implementation framework.

The project targets an installation on a virtual environment based on Ubuntu/Trusty as the base operating system and distribution. Oscar provides a solution to automatically install and configure the required components using existing installer and configuration tools and perform a set of basic system level tests (i.e. test whether OpenStack, OpenContrail, network components like vRouter, etc are operational, tests whether a set of VNFs can be deployed/removed on the compute node and more). All components of the project shown in the table below have already been tested and the overall functionality of the aggregate software components has been validated. As such, this platform should serve as a framework to fast track the continuous integration of various components for the first release (targeted for March 2015).

**Scope:**

The scope of the project is captured in the table shown below. As the table indicates all the features and functionality are part of the first release of the OPNFV. Nearly all these features have been tested and integrated within the OpenContrail, an open source platform under the Apache 2.0 license. Various POCs and pre-production version of the OpenContrail have been demonstrated in numerous lab and production systems.

|  |  |  |
| --- | --- | --- |
| **Technology Area** | **Sub-Area** | **Technology Proposed** |
| **Virtual Infrastructure Orchestration** | Compute Orchestration | Openstack Juno or later |
| Storage Orchestration  | Openstack Juno or later - Ceph based Distributed storage |
| Network Orchestration  | Openstack Juno or later - OpenContrail SDN Controller |
| **Server OS** |   | Ubuntu - Trusty |
| **Server Hypervisor** |   | KVM-Qemu |
| **Virtual Network Device on Server** |   | OpenContrail - vRouter |
| **Support for Physical Appliance/Baremetal Server** |   | VTEP termination on ToR switch using OVSDB  |
| **Virtual Infrastructure Availability** | Orchestration Controller Availability | High Availability with Active-Active-Active mechanism |
| SDN Controller Availability | High Availability with Active-Active-Active mechanism |
| **Physical Infrastructure Design**  | Server-Network Connectivity | Reference Cluster Design with connectivity specification |
| Network Gateway | Network Gateway interface to Internet and L3VPN |
| **Physical Infrastructure Configuration/Imaging**  | Server Imaging/Configuration  | Cobbler based imaging; Puppet based configuration |
| Network Device Imaging/Configuration | Netconf based device configuration management |
| **Service Orchestration**  | VNF Initiation | Heat Template based Virtual Network and Service Chain creation |
| VNF Configuration | Individual EMS based VNF configuration |
| **Service Scaling**  | Horizontal Scaling of Service  | API based horizontal scaling of services |
| Vertical Scaling of Service | On-demand resource augmentation of VNF |
| **Traffic Steering**  | Traffic steering through transparent Services | API based creation of transparent (bump-in-the-wire) Service Chain between two networks |
| Traffic steering through Services with L3 processing | API based creation of L3 processed Service Chain between two networks  |
| Traffic steering through multiple virtual services | API based creation of multiple virtualized services between two networks |
| Traffic steering through virtual and physical services | API and netconf based traffic steering through Virtualized and Physical Appliance |
| **User Interface** | Creation of Service Chains | GUI or API based orchestration of Service Chains |
| Operation and Management of Cluster | GUI or API based Operation and Management of Cluster |
| CLI Interface | Standard CLI based Operations |
| **Operability** | Infrastructure Resources Monitoring  | CPU, Mem, NIC, vCPU, vMem, vNIC, Virtual Network Traffic, Environment Variables, events |
| Service Monitoring | Service Status Monitoring |
| Diagnostic | Endpoint reachability testing |
| **Data Collection & Analytics**  | Log Collection | Service Logs & Syslogs |
| Flow Record | 1:1 flow record collection |
| Packet Capture | API driven on-demand full packet capture of any flow |
| Flow Path  | Correlate overlay and underlay data to trace flow path  |

**Targeted Test Cases:**

* OSCAR provides automated testing tools for installation of various components as well as health check for individual components. This includes (but not limited to):
* OpenStack health check,
* OpenContrail health check.
* Life cycle management and automated system-level testing: CRUD operations for multiple instances of VNFs plus life cycle management of associated network services.

**VNFs Served by OSCAR:**

* FW/SRX
* vFW/vSRX (Firefly)
* vPE
* vCDN
* vCPE
* vMME (vMCG)
* 3rd Party VNFs

**Testability: ''(optional, Project Categories: Integration & Testing)''**

Information regarding testing and integration including interoperability, scalability, high availability are provided in the above table. Any additional information for quality assurance and test resources will be available if necessary.

**Documentation: ''(optional, Project Categories: Documentation)''**

All API-related documents will be available in a timely manner. Detailed description of the Functional Architecture (building blocks, reference points, interfaces and protocols, work flow diagrams, etc.) will be provided during the development, integration and testing processes.

**Dependencies:**

* The OSCAR project relies on the following open source projects:
* OpenStack Juno release: various components including Nova, Neutron, Ceilometer, Heat, etc.,
* OpenContrail: vRouter and other components,
* Installer: Cobbler,
* Configuration & Management: Puppet,
* QEMU/KVM,
* Linux/Ubuntu distribution.

**Committers and Contributors:**

* Parantap Lahiri (plahiri@juniper.net)
* Parviz Yegani (pyegani@juniper.net)
* Raghavendra Mallya (rmallya@juniper.net)
* Kalyanjeet Gogoi (kjgogoi@juniper.net)
* More committers and contributors will be added later.

**Planned deliverables**

Project Oscar:

* Installs scripts for automated deployment of the test setup to a bare metal environment,
* Test scripts to allow for automated system level testing and component health check of all network functions, software components, etc in both virtual and physical environments,
* Heat-based orchestration template for delivered sample VNFs as well as other vendors’ VNFs.

**Proposed Release Schedule:**

* The first release is targeted for March 2015.
* The project aligns with the current release cadence.