OPNFV Platform Overview document

Release arno.2015.1.0 (deee452)

OPNFV

February 15, 2016

CONTENTS

CHAPTER

ONE

INTRODUCTION

The OPNFV project provides different kinds of output to its users:

1. **Target Software Platform**

   This provides the software which will run as NFVI and VIM in an actual NFV deployment It is an integrated solution of several other open source projects, e.g. OpenStack, Linux, ODL.

2. **Deployment Tools**

   So called installers help the user deploy target software on his hardware. OPNFV provides multiple options to do this.

3. **Test Cases and Framework**

   The target software platform can be verified and evaluated using these testcases. Their goal is to show that the deployed platform is usable to run VNFs.

4. **Documentation**

   OPNFV provides the necessary documents describing target software platform, deployment tools, tests, etc. in their architecture and usage.

5. **Requirements**

   OPNFV community works on requirements of open source projects used in OPNFV to make these projects better suitable for NFV telco carrier use cases. These requirements are described in requirement documents and also forwarded to the “upstream” projects in the format required by these projects.

6. **Community Labs**

   OPNFV creates lab environments not only for development of OPNFV, but also to help build the NFV ecosystem. OPNFV labs follow a defined structure and configuration. Some of the labs have their dedicated tasks in the development environment, some of the labs will be provided for open use.

OPNFV Releases

OPNFV bundles the target software, installers, documentation, test cases and lab description to releases and provides documentation describing the scope and features provided.

Since OPNFV releases contain different options (e.g. for installers, SDN controllers, etc.) We define a set of scenarios, that is a set of components, combinations and associated configuration. This set of scenarios shows which combinations of components and features are tested as part of the OPNFV release.

This overview document introduces these components and scenarios on a high level and points you to more detailed documentation. It describes the OPNFV Brahmaputra release.

OPNFV Requirement documents typically describe requirements that will be implemented in later releases of OPNFV. Thus they are not part of a release package.

Also community labs are independent of releases. Only the lab description is included in the release and describes the requirements of a lab to successfully run Brahmaputra deployments.

CHAPTER

TWO

TARGET SOFTWARE PLATFORM

2.1 Software architecture

This section will provide information which upstream projects, versions and components are integrated in the Brahmaputra release

2.1.1 OpenStack

OPNFV uses OpenStack as cloud management system. Brahmaputra is based on OpenStack Liberty Release. It comprises the following sub-projects and modules:

- Nova (Compute)
- Neutron (Network)
- Cinder (Block Storage)
- Swift (Object Storage)
- Ceilometer (Telemetry)
- Keystone (Identity)
- Glance (Image Service)
- Heat (Orchestration)
- etc.

Some of the sub-projects are not deployed in all scenarios. Besides target software, also deployment and test framework use OpenStack components (Fuel, Tempest, Rally)

2.1.2 Operating System

OPNFV uses Linux on all target machines. Depending on the installers, different distributions are supported.

Ubuntu 14 supported by Fuel, Compass and Joid installers CentOS 7 supported by Apex and Compass

2.1.3 SDN Controllers

OPNFV Brahmaputra release supports three different SDN controllers:

- OpenDaylight
- ONOS

- OpenContrail

Depending on the SDN controller you are using, the featureset will vary.

OpenDaylight

Editor's note: We need a high level paragraph here and a description of how we use ODL.

ONOS

Editors note: We need a high level paragraph here and a description of how we use ONOS, especially the relation of ONOS and ONOSFW project's integration and features.

OpenContrail

Editors note: We need a high level paragraph here and a description of how we use OpenContrail, including its vRouter capabilities.

2.1.4 Data Plane

2.1.5 Other Components

2.2 Deployment Architecture

OPNFV starts with a typical configuration with 3 controller nodes running OpenStack, SDN, etc. and a minimum of 2 compute nodes for deployment of VNFs. A detailed description of this 5 node configuration can be found in pharos documentation.

The 3 controller nodes allow to provide an HA configuration. The number of compute nodes can be increased dynamically after the initial deployment.

OPNFV can be deployed on bare metal or in a virtual environment, where each of the hosts is a virtual machine and provides the virtual resources using nested virtualization.

The initial deployment is done using a so-called "jumphost". This server (either bare metal or virtual) is first installed with the installer program that then installs OpenStack and other components on the controller nodes and compute nodes. See the installer documentation for more details.

Editors note: In a second level of detail, describe how software is distributed over the 3 controller nodes, compute nodes and other hardware.

In Brahmaputra, the following scenarios are supported:

2.3 Dynamic View

Editors note: we might skip this section completely for Brahmaputra.

Or we provide rather short statements. In later versions, we have to describe which software is involved in which way during:

- VNF Life Cycle (onboarding, instantiate, scaling): we can reference to other documents
- Hardware Life Cycle (mainly how to add compute nodes, but also other cases)

- ...

CHAPTER

THREE

DEPLOYMENT TOOLS

Brahmaputra provides 4 different installers.

The installers will deploy the target platform onto a set of virtual or bare metal servers according to the configuration files. After the deployment, it doesn't matter which of the installers had been used to deploy the target scenario.

3.1 Apex

Apex is an OPNFV Installation tool based on RDO Manager that deploys OPNFV using the RDO Project OpenStack Distribution. RDO manager is a Triple-O based installation tool. Triple-O is an image based life cycle deployment tool that is a member of the OpenStack Big Tent Governance.

Apex uses Centos on all target platforms and can deploy all SDN controllers.

3.2 Compass

Compass is an installer project based on open source project Compass, which provides automated deployment and management of OpenStack and other distributed systems. It can be considered as what the LiveCD to a single box for a pool of servers – bootstrapping the server pool.

Compass is based on Ansible. It can deploy Ubuntu or Centos as target operating system and ODL and ONOS as SDN controllers.

3.3 Fuel

Editors note: Just a high level intro and link to the main fuel documents.

3.4 Joid

Editors note: Just a high level intro and link to the main joid documents.

CHAPTER

FOUR

TESTCASES AND FRAMEWORK

Editors note: Just a high level description about the different types of tests and the role of yardstick as central framework.