OPNFV User Guide

Release arno.2015.1.0 (ce17ebe)

OPNFV

August 19, 2016

CONTENTS

CHAPTER

ONE

COLORADO 1.0

1.1 Abstract

OPNFV is a collaborative project aimed at providing a variety of virtualisation deployments intended to host applications serving the networking and carrier industry. This document provides guidance and instructions for using platform features designed to support these applications, made available in the Brahmaputra release of OPNFV.

This document is not intended to replace or replicate documentation from other open source projects such as OpenStack or OpenDaylight, rather highlight the features and capabilities delivered through the OPNFV project.

1.2 Overview

OPNFV provides a variety of virtual infrastructure deployments designed to host virtualised network functions (VNFs). This guide intends to help users of the platform leverage the features and capabilities delivered by the OPNFV project.

OPNFV Continuous Integration builds, deploys and tests combinations of virtual infrastructure components in what are defined as scenarios. A scenario may include components such as OpenStack, OpenDaylight, OVS, KVM etc. where each scenario will include different source components or configurations. Scenarios are designed to enable specific features and capabilities in the platform that can be leveraged by the OPNFV user community.

1.2.1 OPNFV Features

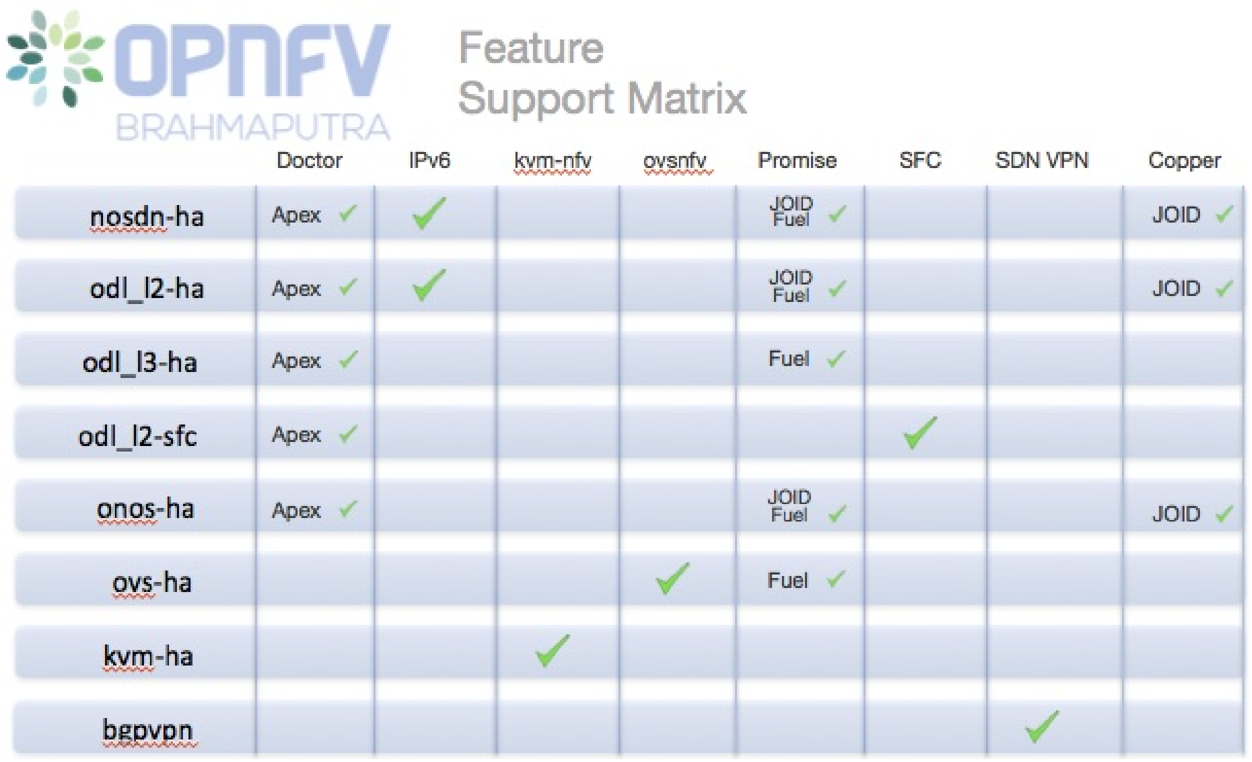
Each OPNFV scenario provides unique features and capabilities, it is important to ensure you have a scenario deployed on your infrastructure that provides the right capabilities for your needs before working through the user guide.

This user guide outlines how to work with key components and features in the platform, each feature description section will indicate the scenarios that provide the components and configurations required to use it.

Each scenario provides a set of platform capabilities and features that it supports. It is possible to identify which features are provided by reviewing the scenario name, however not all features and capabilities are discernible from the name itself.

Brahmaputra feature support matrix

The following table provides an overview of the available scenarios and supported features in the Brahmaputra release of OPNFV.


OPNFV BRAHMAPUTRA Feature Support Matrix

	Doctor	IPv6	kvm-nfv	ovsnfv	Promise	SFC	SDN VPN	Copper
nosdn-ha	Apex ✓	✓			JOID Fuel ✓			JOID ✓
odl_l2-ha	Apex ✓	✓			JOID Fuel ✓			JOID ✓
odl_l3-ha	Apex ✓				Fuel ✓			
odl_l2-sfc	Apex ✓					✓		
onos-ha	Apex ✓				JOID Fuel ✓			JOID ✓
ovs-ha				✓	Fuel ✓			
kvm-ha			✓					
bgpvpn							✓	

The table above provides an overview of which scenarios will support certain feature capabilities. The table does not indicate if the feature or scenario has limitations. Refer to the Configuration Guide for details on the state of each scenario and further information.

Feature development in the Brahmaputra release often consisted of the development of specific requirements and the further integration and validation of those requirements. This results in some features only being supported on the platform when a specific scenario, providing the capabilities necessary to run the feature, is deployed.

Scenario Naming

In OPNFV, scenarios are identified by short scenario names. These names follow a scheme that identifies the key components and behaviours of the scenario, the rules for scenario naming are as follows:

> os-[controller]-[feature]-[mode]-[option]

For example: *os-nosdn-kvm-noha* provides an OpenStack based deployment using neutron including the OPNFV enhanced KVM hypervisor.

The [feature] tag in the scenario name describes the main feature provided by the scenario. This scenario may also provide support for features, such as advanced fault management, which are not apparent in the scenario name. The following section describes the features available in each scenario.

For details on which scenarios are best for you and how to install and configure them on your infrastructure the OPNFV Configuration Guide provides a valuable reference.

The user guide will describe how to enable and utilise features and use cases implemented and tested on deployed OPNFV scenarios. For details of the use cases and tests that have been run you should check the validation procedures section of the OPNFV Configuration Guide. This will provide information about the specific use cases that have been validated and are working on your deployment.

1.2.2 General usage guidelines

The user guide for OPNFV features and capabilities provide step by step instructions for using features that have been configured according to the installation and configuration instructions.

This guide is structured in a manner that will provide usage instructions for each feature in its own section. Start by identifying the feature capability you would like to leverage, then read through the relevant user guide section to understand how to work with the feature. The combination of platform features, if available in a given scenario and not otherwise indicated, should operate according to the documentation. Dependencies between features will be highlighted in the user guide text.

You may wish to use the platform in a manner that the development teams have not foreseen, or exercise capabilities not fully validated on the platform. If you experience issues leveraging the platform for the uses you have envisioned, the OPNFV user mailing list provides a mechanism to establish a dialog with the community to help you overcome any issues identified.

It may be that you have identified a bug in the system, or that you are trying to execute a use case that has not yet been implemented. In either case it is important for OPNFV to learn about it as we are in essence a development project looking to ensure the required capabilities for our users are available.

1.3 Using common platform components

This section outlines basic usage principals and methods for some of the commonly deployed components of supported OPNFV scenario's in Brahmaputra. The subsections provide an outline of how these components are commonly used and how to address them in an OPNFV deployment. The components derive from autonomous upstream communities and where possible this guide will provide direction to the relevant documentation made available by those communities to better help you navigate the OPNFV deployment.

1.3.1 Common VIM components

Brahmaputra OpenStack User Guide

OpenStack is a cloud operating system developed and released by the OpenStack project. OpenStack is used in OPNFV for controlling pools of compute, storage, and networking resources in a Pharos compliant infrastructure.

OpenStack is used in Brahmaputra to manage tenants (known in OpenStack as projects), users, services, images, flavours, and quotas across the Pharos infrastructure. The OpenStack interface provides the primary interface for an operational Brahmaputra deployment and it is from the "horizon console" that an OPNFV user will perform the majority of administrative and operational activities on the deployment.

OpenStack references

The OpenStack user guide provides details and descriptions of how to configure and interact with the OpenStack deployment. This guide can be used by lab engineers and operators to tune the OpenStack deployment to your liking.

Once you have configured OpenStack to your purposes, or the Brahmaputra deployment meets your needs as deployed, an operator, or administrator, will find the best guidance for working with OpenStack in the OpenStack administration guide.

Connecting to the OpenStack instance

Once familiar with the basic of working with OpenStack you will want to connect to the OpenStack instance via the Horizon Console. The Horizon console provide a Web based GUI that will allow you operate the deployment. To do this you should open a browser on the JumpHost to the following address and enter the username and password:

> http://{Controller-VIP}:80/index.html> username: admin password: admin

Other methods of interacting with and configuring OpenStack,, like the REST API and CLI are also available in the Brahmaputra deployment, see the OpenStack administration guide for more information on using those interfaces.

1.3.2 Common SDN components

OpenDaylight User Guide

OpenDaylight is an SDN controller platform developed and released by the OpenDaylight project. The OpenDaylight controller is installed and configured in OPNFV as the networking component of a variety of OPNFV NVFi scenarios using the neutron ODL device driver as an integration point toward OpenStack.

OpenDaylight runs within a JVM and is installed in OPNFV within a container and integrated with OpenStack. The OpenDaylight instance can be configured through the OpenStack Horizon interface, or accessed directly from the OPNFV Jumphost. The Brahmaputra release of OPNFV integrates the latest Beryllium release.

OpenDaylight references

For an overview of the OpenDaylight controller a good reference is the Getting Started Guide. For more detailed information about using the platform the OpenDaylight User Guide provides a good feature by feature reference.

It is important when working on your Brahmaputra deployment to be aware of the configured state of the OpenDaylight controller in the scenario you have deployed, installing an SFC scenario will for instance configure the OpenDaylight controller with the required SFC Karaf features in the OpenDaylight controller. Make sure you read the installation and configuration guide carefully to understand the state of the deployed system.

Connecting to the OpenDaylight instance

Once you are familiar with the OpenDaylight controller and its configuration you will want to connect to the OpenDaylight instance from the Jumphost. To do this you should open a browser on the JumpHost to the following address and enter the username and password:

> http://{Controller-VIP}:8181/index.html> username: admin password: admin

Other methods of interacting with and configuring the controller, like the REST API and CLI are also available in the Brahmaputra deployment, see the OpenDaylight User Guide for more information on using those interfaces.

It is important to be aware that when working directly on the OpenDaylight controller the OpenStack instance will not always be aware of the changes you are making to the networking controller. This may result in unrecoverable inconsistencies in your deployment.

ONOS User Guide

ONOS is an SDN controller platform developed and released by the ONOS project. The ONOS controller is installed and configured in OPNFV as the networking component of a variety of OPNFV NFVI scenarios.

ONOS runs within a JVM instance and is integrated with OpenStack via a Neutron ML2 plugin. The ONOS instance can be configured through the OpenStack Neutron interface, or through native ONOS tools from the OPNFV jumphost. The Brahmaputra release of OPNFV integrates the latest ONOS 1.4 (EMU) release version.

ONOS references

For an overview of the ONOS controller, please see User Guide. For more detailed information about the EMU version of ONOS, documentation is available on the ONOS download page.

Connecting to the ONOS instance

Once you are familiar with the ONOS controller and its configuration you will want to connect to the ONOS instance from the Jumphost. To do this you should open a browser on the JumpHost to the following address and enter the username and password:

> http://{Controller-VIP}:8282/index.html> username: karaf password: karaf

Other methods of interacting with and configuring the controller, like the REST API and CLI are also available in the Brahmaputra deployment, see the ONOS User Guide for more information on using those interfaces.

It is important to be aware that when working directly on the ONOS controller the OpenStack instance will not always be aware of the changes you are making to the networking controller. This may result in unrecoverable inconsistencies in your deployment.

If you have any questions or need further assistance, you may also direct your queries to *ONOSFW Forum <http://forum.onosfw.com>*

1.4 ONOS User Guide

ONOS is an SDN controller platform developed and released by the ONOS project. The ONOS controller is installed and configured in OPNFV as the networking component of a variety of OPNFV NFVI scenarios.

ONOS runs within a JVM instance and is integrated with OpenStack via a Neutron ML2 plugin. The ONOS instance can be configured through the OpenStack Neutron interface, or through native ONOS tools from the OPNFV jumphost. The Brahmaputra release of OPNFV integrates the latest ONOS 1.4 (EMU) release version.

1.4.1 ONOS references

For an overview of the ONOS controller, please see User Guide. For more detailed information about the EMU version of ONOS, documentation is available on the ONOS download page.

1.4.2 Connecting to the ONOS instance

Once you are familiar with the ONOS controller and its configuration you will want to connect to the ONOS instance from the Jumphost. To do this you should open a browser on the JumpHost to the following address and enter the username and password:

> http://{Controller-VIP}:8282/index.html> username: karaf password: karaf

Other methods of interacting with and configuring the controller, like the REST API and CLI are also available in the Brahmaputra deployment, see the ONOS User Guide for more information on using those interfaces.

It is important to be aware that when working directly on the ONOS controller the OpenStack instance will not always be aware of the changes you are making to the networking controller. This may result in unrecoverable inconsistencies in your deployment.

If you have any questions or need further assistance, you may also direct your queries to *ONOSFW Forum <http://forum.onosfw.com>*

1.5 OpenDaylight User Guide

OpenDaylight is an SDN controller platform developed and released by the OpenDaylight project. The OpenDaylight controller is installed and configured in OPNFV as the networking component of a variety of OPNFV NVFi scenarios using the neutron ODL device driver as an integration point toward OpenStack.

OpenDaylight runs within a JVM and is installed in OPNFV within a container and integrated with OpenStack. The OpenDaylight instance can be configured through the OpenStack Horizon interface, or accessed directly from the OPNFV Jumphost. The Brahmaputra release of OPNFV integrates the latest Beryllium release.

1.5.1 OpenDaylight references

For an overview of the OpenDaylight controller a good reference is the Getting Started Guide. For more detailed information about using the platform the OpenDaylight User Guide provides a good feature by feature reference.

It is important when working on your Brahmaputra deployment to be aware of the configured state of the OpenDaylight controller in the scenario you have deployed, installing an SFC scenario will for instance configure the OpenDaylight controller with the required SFC Karaf features in the OpenDaylight controller. Make sure you read the installation and configuration guide carefully to understand the state of the deployed system.

1.5.2 Connecting to the OpenDaylight instance

Once you are familiar with the OpenDaylight controller and its configuration you will want to connect to the OpenDaylight instance from the Jumphost. To do this you should open a browser on the JumpHost to the following address and enter the username and password:

> http://{Controller-VIP}:8181/index.html> username: admin password: admin

Other methods of interacting with and configuring the controller, like the REST API and CLI are also available in the Brahmaputra deployment, see the OpenDaylight User Guide for more information on using those interfaces.

It is important to be aware that when working directly on the OpenDaylight controller the OpenStack instance will not always be aware of the changes you are making to the networking controller. This may result in unrecoverable inconsistencies in your deployment.

1.6 Brahmaputra OpenStack User Guide

OpenStack is a cloud operating system developed and released by the OpenStack project. OpenStack is used in OPNFV for controlling pools of compute, storage, and networking resources in a Pharos compliant infrastructure.

OpenStack is used in Brahmaputra to manage tenants (known in OpenStack as projects), users, services, images, flavours, and quotas across the Pharos infrastructure. The OpenStack interface provides the primary interface for an operational Brahmaputra deployment and it is from the “horizon console” that an OPNFV user will perform the majority of administrative and operational activities on the deployment.

1.6.1 OpenStack references

The OpenStack user guide provides details and descriptions of how to configure and interact with the OpenStack deployment. This guide can be used by lab engineers and operators to tune the OpenStack deployment to your liking.

Once you have configured OpenStack to your purposes, or the Brahmaputra deployment meets your needs as deployed, an operator, or administrator, will find the best guidance for working with OpenStack in the OpenStack administration guide.

1.6.2 Connecting to the OpenStack instance

Once familiar with the basic of working with OpenStack you will want to connect to the OpenStack instance via the Horizon Console. The Horizon console provide a Web based GUI that will allow you operate the deployment. To do this you should open a browser on the JumpHost to the following address and enter the username and password:

> http://{Controller-VIP}:80/index.html> username: admin password: admin

Other methods of interacting with and configuring OpenStack,, like the REST API and CLI are also available in the Brahmaputra deployment, see the OpenStack administration guide for more information on using those interfaces.

1.7 Using Brahmaputra Features

The following sections of the user guide provide feature specific usage guidelines and references. Providing users the necessary information to leveraging the features in the platform, some operation in this section may refer back to the guides in the general system usage section.

1.8 <Feature> capabilities and usage

Describe the specific capabilities and usage for <XYZ> feature.

1.8.1 <Feature and API usage guidelines and example>

Describe with examples how to use specfic features.