Enhanced Service Assurance - Intel Platform with OPNFV* Barometer Project

Authors

John J Browne

Emma Collins

Jabir Kanhira Kadavathu

Krzysztof Kepka

Sunku Ranganath

Kamil Wiatrowski

1   Introduction

Service assurance is a key function of Communications Service Providers (CoSPs), who apply policies and processes to ensure that their services meet a pre-defined service quality level for an optimal subscriber experience. Using service assurance tools, CoSPs identify faults in the network and resolve issues quickly to minimize service downtime.

OPNFV* Barometer is an open source project (formerly known as Software Fastpath Service Quality Metrics) with the following capabilities:

- Ensure Intel® architecture platform metrics and events are accessible through industry standard interfaces.
- Demonstrate Intel® architecture platform technologies can be monitored, consumed, and actioned in real time.

The OPNFV* Barometer project has the following value propositions:

- Provides in-band and out-of-band telemetry and metrics to telemetry consuming northbound interfaces.
- Provides the most recent Intel® architecture platform metrics via collectd* plugins that are not available in a collectd release.
- Eases the industry adoption of Intel® architecture telemetry by providing reference deployments of telemetry with Time Series Databases (TSDB) and visualization tools.

This guide describes the steps to install OPNFV* Barometer and trigger the deployment of collectd, InfluxDB*, and Grafana* across a single/multi node setup, showcasing Intel® architecture differentiating metrics including, but not limited to, Intel® Resource Director Technology (Intel® RDT), Intelligent Platform Management Interface (IPMI), Mcelog, ovs_stats, ovs_events, hugepages, and intel_pmu.

This document is part of the Network Transformation Experience Kit, which is available at: https://networkbuilders.intel.com/

Table of Contents

Figures

Tables

1.1 Terminology

Table 1. Terminology

Abbreviation	Description
CoSP	Communications Service Provider
DPDK	Data Plane Development Kit
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol secure
IPMI	Intelligent Platform Management Interface
NFVI	Network Function Virtualization Infrastructure
RAS	Reliability, Availability, and Serviceability
RDT	Intel® Resource Director Technology (Intel® RDT)
SLA	Service Level Agreement
SNMP	Simple Network Management Protocol
SSH	Secure Shell
TSDB	Time Series Databases
VNF	Virtual Network Functions

1.2 Reference Documents

Table 2. Reference Documents

Reference	Source
Docker User Guide	https://opnfv-barometer.readthedocs.io/en/latest/release/userguide/docker.userguide.html
OPNFV* Barometer	https://wiki.opnfv.org/display/fastpath/Barometer+Home

2 Barometer Project

2.1 Barometer Overview

Platform telemetry and components to retrieve them are available in open source from the OPNFV Barometer project at: https://wiki.opnfv.org/display/fastpath/Barometer+Home which also contains the following overview:

The ability to monitor the Network Function Virtualization Infrastructure (NFVI) where VNFs are in operation is a key part of Service Assurance within an NFV environment. To enforce Service Level Agreements (SLAs) or to detect violations, faults or degradation in the performance of NFVI resources, monitoring is required so that events and relevant metrics are reported to higher level management and analytics systems. As fixed function appliances are replaced by virtualized appliances, the service levels, manageability, and service assurance needs to remain consistent or improve on what is available today. As such, the NFVI needs the capability to do the following:
1. Traffic monitoring and performance monitoring of the components that provide networking functionality to the Virtual Network Functions (VNFs), including: physical interfaces, Data Plane Development Kits (DPDKs), virtual switch interfaces and flows, as well as the virtual interfaces themselves and their status, etc.
2. Platform monitoring of components such as CPU, memory, load, cache, thermals, fan speeds, voltages and Reliability, Availability, and Serviceability (RAS) events, etc.

All of the statistics and events gathered must be collected in-service and must be capable of being reported by standard Telco mechanisms (for example, Simple Network Management Protocol (SNMP)), for potential enforcement or corrective actions. In addition, this information could be fed to analytics systems to enable failure prediction, and can also be used for intelligent workload placement.

2.2 Barometer Features

After you complete the installation, you will have the ability to gather Intel® architecture platform-specific metrics, store them in InfluxDB, and display them to a Grafana dashboard.

Barometer includes the following features:
- "*One-click install*" via Ansible* sets up the collectd* 5.8, InfluxDB, and Grafana containers in less than 10 minutes.
- Provides scalable means of deploying and collecting platform and infrastructure telemetry across multiple nodes.
- Default support of Intel® architecture differentiating metrics like Intel® RDT, IPMI, RAS, DPDK, and others with all their dependencies included.
- Validated support of "one-click install" and collectd containers across Ubuntu* 16.04, 18.04, and CentOS* 7.5.
- Provides large set of documentation updates for ease of use and out of the box metrics availability.

2.3   Example Deployment Options

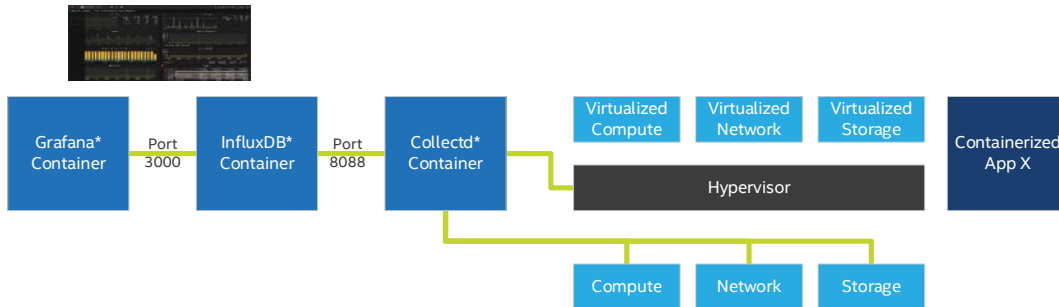Barometer can be deployed in either a single-node or multi-node configuration, as shown in the figures below.

Figure 1.   Single-Node Deployment

For single-node deployments, collectd, Influxdb, and Grafana containers are all on the same node.
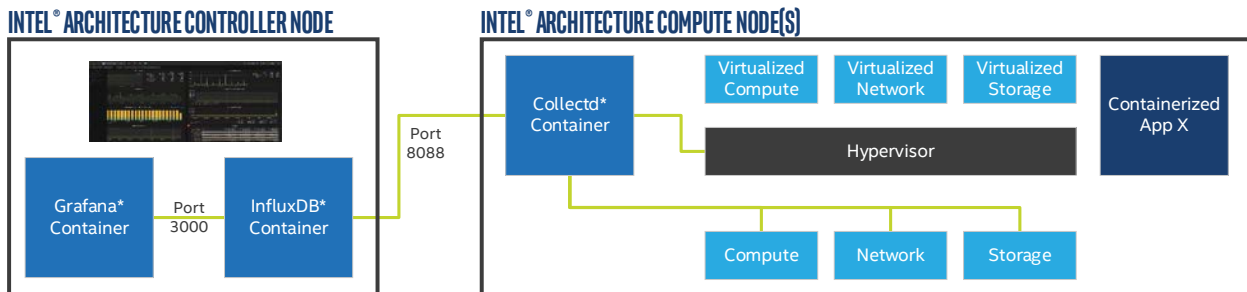

Figure 2.   Multi-node Deployment

A multi-node deployment assumes an N:1 ratio between Compute Node and Controller Node. On the compute node(s), collectd exports telemetry to the controller node. On the controller node, Influxdb and Grafana listen to the telemetry from collectd.

3   Installation Instructions

The following sections explain in detail how to install the Barometer solution.

3.1   Install Ansible*

If operating behind a proxy, refer to Section 4 for guidance on how to set up your proxy.

Note: Be aware of the following:
1. Sudo permissions or root access are required to install Ansible.
2. Ansible version needs to be 2.4 or higher, due to usage of import/include statements.

3.1.1   Install Ansible on Ubuntu*

The following steps have been verified with Ansible* 2.6.3 on Ubuntu* 16.04 and 18.04. To install Ansible 2.6.3 on Ubuntu, open a console window and enter the commands below:

```
$ sudo apt-get install python
$ sudo apt-get install python-pip
$ sudo pip install 'ansible==2.6.3'
```

3.1.2 Install Ansible on CentOS*

The following steps have been verified with Ansible 2.6.3 on CentOS* 7.5. To install Ansible 2.6.3 on CentOS, open a console window and enter the commands below:

```
$ sudo yum install python
$ sudo yum install epel-release
$ sudo yum install python-pip
$ sudo pip install 'ansible==2.6.3'
```

3.2 Clone barometer repository

Open a console window and enter the following commands to clone the Barometer repository:

```
$ git clone https://gerrit.opnfv.org/gerrit/barometer
$ cd barometer/docker/ansible
```

3.3 Edit inventory file

Edit the `$barometer_dir/docker/ansible/default.inv` inventory file and add the following hosts:

```
[collectd_hosts]
localhost

[collectd_hosts:vars]
install_mcelog=true
insert_ipmi_modules=true

[influxdb_hosts]
localhost

[grafana_hosts]
localhost

[prometheus_hosts]
#localhost

[kafka_hosts]
#localhost

[ves_hosts]
#localhost
```

Change `localhost` to different hosts wherever necessary. To change a host for kafka*, edit `kafka_ip_addr` in `./roles/config_files/vars/main.yml`.

Note: Hosts for influxdb and grafana are required only for collectd_service.yml.

Hosts for kafka and ves are required only for collectd_ves.yml.

3.4 Additional Plugin Dependencies

By default, Ansible tries to fulfill dependencies for the `mcelog` and `ipmi` plugins. For the `mcelog` plugin, Ansible installs the `mcelog` daemon. For the ipmi plugin, Ansible tries to insert `ipmi_devintf` and `ipmi_si` kernel modules. This behavior can be changed in the inventory file using the variables `install_mcelog` and `insert_ipmi_modules` as shown below. Both variables are independent:

```
[collectd_hosts:vars]
install_mcelog=false
insert_ipmi_modules=false
```

Note: If you are using Ubuntu 18.04, you must install the mcelog daemon manually before installing from Ansible scripts. The deb package is not available in the official Ubuntu 18.04 repository, which means that setting install_mcelog to true is ignored unless you manually install the mcelog daemon.

3.5 Configure SSH keys

Enter the following command to generate ssh keys if they are not present; if ssh keys already exist, skip to Section 3.6:

```
$ sudo ssh-keygen
```

Enter the following commands to copy the ssh key to all target hosts. It requires you to provide a root password. The following example is for localhost:

```
$ sudo -i
$ ssh-copy-id root@localhost
```

Enter the following command to verify that the key is added and password is not required to connect:

```
$ sudo ssh root@localhost
```

Note: Keys should be added to every target host. `[localhost]` is only used as an example. For multinode installation, keys need to be copied for each node in the format: `[collectd_hostname]`, `[influxdb_hostname]` etc.

3.6   Download and run the containers

The one-click installation features easy and scalable deployment of collectd, Influxdb, and Grafana containers using Ansible playbook. The following additional commands provide additional details:

Enter the following command to check the three containers are running:

```
$ sudo ansible-playbook -i default.inv collectd_service.yml
```

The output of `docker ps` should be similar to the following:

```
$ sudo docker ps
CONTAINER ID        IMAGE                       COMMAND             CREATED             STATUS
PORTS               NAMES
a033aeea180d        opnfv/barometer-grafana     "/run.sh"           9 days ago          Up 7
minutes                             bar-grafana
1bca2e4562ab        opnfv/barometer-influxdb    "/entrypoint.sh in..."   9 days ago          Up 7
minutes                             bar-influxdb
daeeb68ad1d5        opnfv/barometer-collectd    "/run_collectd.sh ..."   9 days ago          Up 7
minutes                             bar-collectd
```

Enter the following command to make changes while a container is running:

```
$ sudo docker exec -ti <CONTAINER ID> /bin/bash
```

Connect to `<host_ip>:3000` with a browser and log into Grafana with the username "admin" and password "admin". Refer to the *Grafana User Guide* (refer to Table 2) for details.

Figure 3.   Example of the Grafana UI

The collectd configuration files can be accessed directly on the target system in `/opt/collectd/etc/collectd.conf.d`. It can be used for manual changes or to enable/disable plugins. If the configuration has been modified, collectd must be restarted. Enter the following command to restart collectd:

```
$ sudo docker restart bar-collectd
```

Refer to Table 2, *Docker User Guide* for more information about Docker.

4   Proxy for Package Manager on Host

This step is only necessary if the host is behind an HTTP/HTTPS proxy. The Proxy URL must be set in a dedicated config file.

4.1   Proxy for CentOS*

For CentOS*, the config file is located in `/etc/yum.conf`. Open the config file and add the following line:

```
proxy=http://your.proxy.domain:1234
```

4.2   Proxy for Ubuntu*

For Ubuntu*, the config file is located in `/etc/apt/apt.conf`. Open the config file and add the following line:

```
Acquire::http::Proxy "http://your.proxy.domain:1234"
```

After updating the config file, enter the following command to update the apt mirrors:

```
$ sudo apt-get update
```

5   Proxy Environment Variables for Docker*/pip

This step is only necessary if the host is behind an HTTP/HTTPS proxy.

Configuring the proxy for a packaging system is not enough. Some proxy environment variables must be set in the system before Ansible* scripts can be started. Barometer configures the Docker* proxy automatically via an Ansible task as a part of the "one-click install" process.

You need to provide the proxy URL using common shell environment variables, and Ansible will automatically configure proxies for Docker to be able to fetch barometer images. Other components used by Ansible will also benefit from setting proxy variables properly in the system. For example, pip is used for downloading Python* dependencies.

Proxy variables used by the Ansible one-click install include:
*   `http_proxy`
*   `https_proxy`
*   `ftp_proxy`
*   `no_proxy`

The variables mentioned above must be visible for superuser, because most actions involving Ansible and Barometer installation require root privileges. Proxy variables are commonly defined in the `/etc/environment` file , but other locations are acceptable, as long as the variables can be seen by commands using `su`.

Sample proxy configuration in `/etc/environment`:

```
http_proxy=http://your.proxy.domain:1234
https_proxy=http://your.proxy.domain:1234
ftp_proxy=http://your.proxy.domain:1234
no_proxy=localhost
```

6   Summary

The disaggregation of hardware and software in an NFV environment can create management and service assurance challenges. Status, event, and other critical information from each is required for effective systems and service management. Intel platforms provide a rich and often unique set of metrics and telemetry to enhance management and operations for virtual networks.

The open source OPNFV* Barometer project makes it easy to deploy the components (collectd*, InfluxDB*, Grafana*, and others) needed to access the platform telemetry for ingestion into critical management systems at scale.

This guide has described how to install OPNFV* Barometer and trigger the deployment of collectd*, InfluxDB* and Grafana* across a single/multi node setup.