VES Home

Welcome to the VNF Event Stream (VES) Project Home

This project was approved May 31, 2016 based upon the VNF Event Stream project proposal.

Project description:

Objective:

This project will develop OPNFV platform support for VNF event streams, in a common model and format intended for use by NFV Service Providers (SPs), e.g. in managing VNF health and lifecycle. The project's goal is to enable a significant reduction in the effort to develop and integrate VNF telemetry-related data into automated VNF management systems, by promoting convergence toward a common event stream format and collection system.

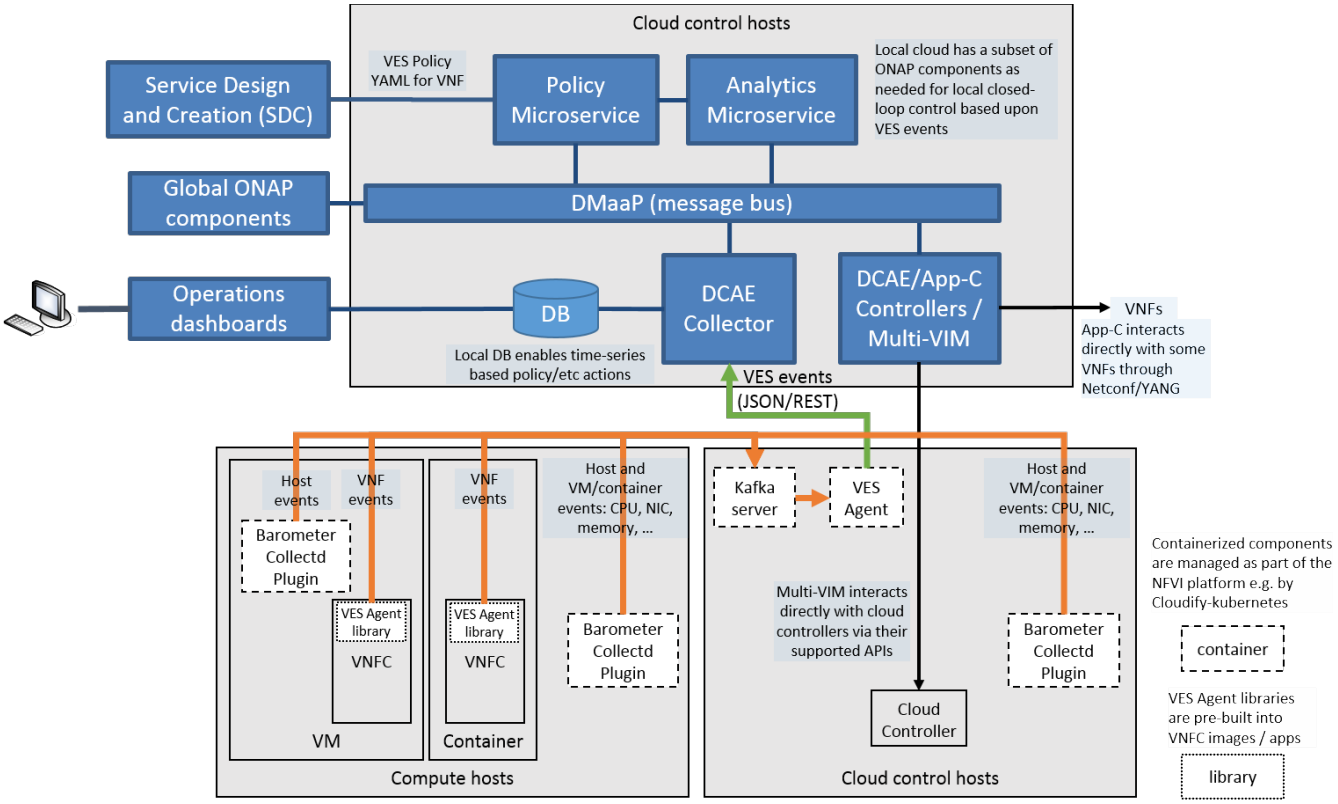
The VES doc source, code, and tests are available at:

- OPNFV github (generally updated with 30 minutes of merged commits)
- OPNFV gitweb
- To clone from the OPNFV repo, see the instructions at the Gerrit project page

Powerpoint intro to the project: OPNVF VES.pptx. A demo of the project (vHello_VES Demo) was first presented at OpenStack Barcelona (2016), and updated for the OPNFV Summit 2017 (VES ONAP demo - see below for more info).

The following diagram illustrates the concept and scope for the VES project, which includes:

- From ONAP
  - a Common Event Data Model for the "VNF Event Stream", with report "domains" covering e.g. heartbeat, measurements (e.g. resource usage), and events (e.g. faults)
  - A YAML-based "VES VNF artifact" that can be used by VNF developers to describe what the VNF produces in VES format, and the significance (e.g. recommended processing/reaction) for those events or events reported from the VNF host. The YAML artifact is used in the creation/configuration of policy microservices that help lifecycle-manage the VNF.
  - VES Agent library enabling VNFs to directly support VES for delivering application-specific measurements and events
  - Various other ONAP components that support the broader goals and end-to-end operation of a closed-loop lifecycle management platform for VNFs. Shown below are examples of these components (generically/abstractly represented here) that might be deployed in a local cloud to enable local closed-loop control as necessary. These include:
    - The DCAE Collector, which receives events in the VES format and publishes them on the ONAP message bus, as well as saves them in a local time-series database.
    - The local message bus ("Data Movement as a Platform" DMaaP) for ONAP components
    - Various DCAE microservices that implement the closed-loop control or other actions (e.g. analytics analysis) based upon the VES events.
    - Various ONAP components that interface with cloud controllers and application controllers for requested policy microservice actions, through the Multi-VIM layer of ONAP. See Multi-VIM FCAPS VES for R1.docx for a concept document addressing these functions.
- From OPNFV
  - The Barometer guide to using the Collectd plugin which publishes host measurements and events over a local Kafka bus, with subject "collectd"
  - The Barometer VES Plugin (ves_app.py) which consumes the collectd Kafka events and forwards them in VES format to a pre-configured VES/DCAE Collector
  - The VES test collector and other utilities as needed to create an emulated ONAP environment for testing.
  - VES scripts to deploy the local ONAP components as needed for real closed loop tests etc.
  - VES demos that illustrate the VES framework in operation.
  - VES tests that support certification of test VNFs, NFV platforms, and real VNFs as VES compliant for a key subset of closed loop control use cases.



VES Roadmap:

The latest project overview and roadmap forward is described in the talk from the 2017 OPNFV Summit: VNF Event Streaming Onboarding Telemetry Policies.pptx

As of the OPNFV Fraser release, the VES project will focus on:

- creation of test tools and tests to enable OPNFV reference platform distributions and VNFs to be validated against the VES schema, and verified as interoperable with the related elements of ONAP
- development/integration of additional VES framework features in advance of upstreaming to ONAP, or cross-community testing of features as they are developed in ONAP
    - new transport protocols for the VES Agent to VES Collector interface, e.g. GRPC + Protocol Buffers
    - VES configuration management, including the concept of "VES Controller"
        - The VES Controller is responsible for configuring the VES source (e.g. a host, VM/container, or app/VNFC) per static and dynamic needs of various tenants for monitoring of elements (VES data "sources") in their scope, e.g. the infrastructure provider (focused on the data center and cloud platform/servers), service provider (focused on services supported by a collection of VNFs), or application controller (focused on specific VNFs)
        - The VES Controller exposes an API via which tenants can deliver YAML formatted input files defining specific objectives of the tenants for monitoring elements in their scope, e.g. hosts, VMs, containers, VNFs, and future elements in potential VES scope e.g. other data center elements such as switches
        - The VES YAML will be used to customize the default VES configuration for those elements, via a simple JSON or YAML formatted message transported to the source via VES-inband messages (transported from the VES Controller to the monitoring source) or an out-of-band framework such as Consul
    - new VES data model features
    - VES Collector support for harmonization of VES collected data into a generic model for ONAP timeseries data, as defined in the ONAP Multi-VIM project, e.g. similar in concept/approach to the YAML-based data translator used by the Barometer VES Agent (ves_app.py)

See Project Planning for the current release project plan.

Alignment with Upstream Projects

ONAP is the upstream home for key elements of the VES architecture, specifically:

- the VES data schema (ves_data_model.json)
- VES-specific monitoring agents and collectors, other than as developed by the Barometer project in OPNFV, or used in further VES testing /demos in OPNFV
    - for test/demo purposes, OPNFV VES will continue to develop "agent" code e.g. (e.g. evel_demo.c, a C-based test/demo agent for mapping source telemetry to the VES event format)
- the DCAE (Data Collection, Analytics, and Events) event collector
    - for test/demo purposes, OPNFV VES will continue develop "collector" code (e.g. monitor.py, a python-based test/demo client for receiving, validating, logging, and saving VES events into an influxdb database, for use as a datasource e.g. in Grafana or for closed-loop microservice testing)

- microservices that react in a closed-loop manner to VES events per policies for those events

Also leveraged by the current test (vHello_VES.sh) is a python-based collectd client (ves_plugin.py) developed in the Barometer project, which maps collectd source data to the VES event format. If/when ves_plugin.py will be upstreamed to ONAP DCAE is TBD.

Current Work Items

Item	Description
VES certification tools	Test tools and tests for certification of OPNFV reference platform distributions and VNFs as compliant with the VES schema and interoperable with ONAP for lifecycle management based upon VES events.
VES tests /demos	Development of VNF management test/demos based upon reference VNFs running on the OPNFV system. • vHello_ONAP demo for the OPNFV Summit 2017 • vHello_VES Demo for OpenStack Barcelona 2016
VES talks at events	OpenStack Seattle Days (Seattle, 9/30/16 - Friday after the ODL Summit) • "OPNFV VES: A Modeled Approach to Monitoring VNF Event Streams" (Bryan); see OPNFV_VES_OpenStack_Days_Seattle.pptx

Previous Work Items

Item	Description	Result
VES Agent	Architecture/design for VES Agent and other functions integrated with it in the bare metal host or VDU. See VES Agent.	Upstreamed to ONAP
Modeling Framework	Analysis of modeling frameworks in which to establish a base data model for VES. Options currently being analyzed (see sub-pages for details): • OpenConfig • ...	Upstreamed to ONAP
Operational Framework	Analysis of frameworks enabling functional tests/demos for VES-modeled telemetry. An operational framework may be related to a modeling framework, or independent. It provides APIs to interact with devices via models, and protocol bindings over which event streams are established and controlled. See Operational Framework.	Upstreamed to ONAP

Key Project Facts:

Project Name: VNF Event Stream (VES)
Repo name: VES
Lifecycle State: Incubation
Primary Contact: Bryan Sullivan, AT&T
Project Lead: Bryan Sullivan, AT&T
Jira Project Name: VNF Event Stream
Jira Project Prefix: VES
mailing list tag: VES
Committers:

- Bryan Sullivan, AT&T
- Aimee Ukasick, AT&T

Recent space activity

Bryan Sullivan

VES Home updated Jan 12, 2018 • view change

Testing updated Nov 20, 2017 • view change

Background created Nov 15, 2017

Certification created Nov 03, 2017

Project Plan updated Nov 03, 2017 • view change

Space contributors

- Bryan Sullivan (464 days ago)
- Aimee Ukasick (702 days ago)
- Maryam Tahhan (733 days ago)