DCAE Contribution and Development

The current ONAP DCAE project is also known as DCAEGEN2. This document describes how to develop for DACEGEN2 sub-components. Topics that are generic across ONAP are not covered by this document, Please reference the overall ONAP wiki age for more information.

Overview

Typically DCAE sub-components are developed individually under their own ONAP Gerrit/Git repos. The initiation of a new repo generally starts during the release planning time, which usually starts around the code freeze time of the previous ONAP release. Normally before the function freeze time (M2), projects that are planned to be included in a release are determined, and infrastructure items such as repo creation for new projects are completed.

DCAEGEN2 Code Repo Structure

https://gerrit.onap.org/r/#/admin/projects/?filter=dcaegen2

Requesting New Repo

Pre-requisite - The new MS/component must be approved by ARC/TSC and request will be made by PTL

To request a new component:

1. Add the repo details under the Resources and Repositories (Deprecated) page.
2. Communicate with ONAP Release Manager and Linux Foundation Helpdesk.

Jenkins Job Definition for new repo

ONAP uses Jenkins based CICD tool chain. However, contributors are only given read access to the Jenkins servers. All jobs are created by automatic generation from JJB definitions. As a DCAE contributor, to create new Jenkins job will require a new JJB file being created under the jjb/dcaegen2 directory of the ci-managerment project.

The status of Jenkins jobs can be viewed at http://jenkins.onap.org.

Initial Seed Code Contribution

Reference this ONAP WiKi page for details of configuring for using Gerrit: Configuring Gerrit.

Code Requirements

1. License text included in each file. Apache 2 for coding files; CC4 for others.
    a. The Copyright line for contributing organization inserted or updated reflecting the contribution year.
2. A LICENSE.txt file placed at the root of the repo to provide umbrella coverage.
3. Unit testing coverage > 53% for R4
    a. For Java and Python code, coverage reporting: http://sonar.onap.org.
    b. For other languages, coverage reporting done in language native method.
4. CLM reporting free of critical vulnerabilities for both security and licensing. CLM reports available at: https://nexus-iq.wl.linuxfoundation.org/assets/index.html

Documentation

Documentation that is targeting users of ONAP is to be placed under the docs directory of the dcaegen2 project in RST format. Documentation contributed here will be automatically built into web pages at http://onap.readthedocs.io.

Further information regarding documentation can be found here: http://onap.readthedocs.io/en/latest/index.html.

Artifact Requirements

There are two levels of artifacts that will be built from ONAP source codes:

1. Software component
    a. jar, war, wheel/pypi, wagon, etc.
2. Docker container

It is generally expected that each functional entity of DCAE project to be available and packaged into Docker images. Docker versioning guideline can be found at Independent Versioning and Release Process#StandardizedDockerTagging.

CSIT testing

CSIT is an integrated automatic testing practice of ONAP for container level testing. It is expected that component contributors to implement CSIT testing plans for their components so that basic component level integrity can be tested. More information regarding CSIT testing can be found at: Creating a CSIT Test.

Bug reporting

Bugs are tracked using the ONAP JIRA system available under the DCAE project: https://jira.onap.org/secure/RapidBoard.jspa?rapidView=49&view=planning.nodetail