Services Pre-requisites/Requirement

DCAE platform is microservice centric, the services (collectors and analytics interfacing with NF events) themselves can be build independently and integrated with DCAE with minimal work. However as ONAP itself requiring carrier grade level, all components has strict S3P goals and LF process to comply.

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. Normally before the functionality 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.

Pre-requisites

For community members looking to contribute new service under ONAP DCAE project, below steps outline general steps to be followed by the service owners.

- Review new service proposal with architecture team and project team. If new collectors, this typically requires review with ARC and VNFREQ team as this introduces new interface between xNF and DCAE.
- Identify usecase this service will be targeted under (optional).
- Capture external and internal dependencies and api consumed/provided (this could be documented in wiki to start, but swagger spec required by M2)
- Repository creation (PTL will submit request to RM, who will coordinate with TSC/LF support)
- Work with PTL to scope the service and EPIC/US creations and release/sprints to target around M1

Contribution Guidelines

1. Setup required Jenkins job (under ci-management repo) for building artifacts/docker images, sonar-scans, CLM. 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.
2. Push (seed) code into gerrit on the chosen repo (Reference this ONAP WiKi page for details of configuring for using Gerrit: Configuring Gerrit)
   a. Ensure complaince with all NFR's (Non-Functionrequirement)
   b. Leverage DCAE common sdk for config retrieval/dmaap pub & sub etc (DCAESDKIntegration)
   c. License text included in each file. Apache 2 for coding files; CC4 for others.
3. Ensure all committers and usecase owners/leads are looped into gerrit submission for review
4. Once change is merged, review CLM/coverty scan report and address all CRITICAL/HIGH License/security issues identified (**TSC MUST HAVE)**
   a. Reports will be under https://nexus-iq.wl.linuxfoundation.org/assets/index.html (access is restricted; work with committers to obtain the report)
5. ONAP Deployment Integration
6. DCAE MOD integration support (OPTIONAL from KOHN)
   a. Every component to be onboarded into DCAE, should prepare a component spec (a.k.a spec) - which is meta data represented in json describing the component configuration model. The spec file should be added into component repo (under <repo><component>/dpo /spec directory). For more info on component spec, refer documentation under RTD
7. Add CSIT test (How to guide Creating a CSIT Test). This can be done under "integration" repo or within component repo itself (see dcaegen2 /services/pm-mapper or dcaegen2/collectors/datafile)
8. Documentation
9. Demo

Non-Functional requirement

All new contribution MUST be complaint with Global requirements and approved "best Practice" requirements. Following list key NFR's

- Support Java11 (or higher) or Python 3.x (as recommended by SECCOM for targetted release) **(TSC MUST HAVE)**
- Code Coverage - Min 80% **(DCAE MUST HAVE)**
- Logging compliance to Best Practice requirement - Jakarta Best Practice Proposal for Standardized Logging Fields - v2 (old spec - ~~https://wiki.onap.org/pages/viewpage.action?pageId=28378955~~)
- Log to be written into file **and stdout (TSC MUST HAVE)**

- If API's are exposed, need to conform to ONAP API Common Versioning Strategy (CVS) Guidelines & OpenAPI spec/swagger file to be defined and shared with impacted project by M2.
- Containers must have no more than one main process **(TSC MUST HAVE)**
- Containers must crash properly when a failure occurs **(TSC MUST HAVE)**
- No hardcoded password in the container (should be made configurable and set at deployment time and sourced using CBS apis') **(TSC MUST HAVE)**
- Use ONAP-integration supported base image

Note: All above are common ONAP requirements for any new contributions. TSC MUST HAVE are **required for passing release candidate.**

ONAP Deployment Integration

All ONAP component deployments are through helm-charts maintained under OOM repository; refer to https://docs.onap.org/projects/onap-dcaegen2/en/latest/sections/installation_oom.html for OOM/dcae chart structure.

New DCAE Microservice chart contribution should go under https://git.onap.org/oom/tree/kubernetes/dcaegen2-services; all DCAE component charts should leverage oom/kubernetes/dcaegen2-services/common/dcaegen2-services-common templates. Refer to following link - https://docs.onap.org/projects/onap-dcaegen2/en/latest/sections/dcaeservice_helm_template.html for details on the supported features via this template.

With Kohn release, all DCAE components will be enabled for daily/weekly deployments. This is controlled by override here - https://git.onap.org/oom/tree/kubernetes/onap/resources/overrides/onap-all.yaml. New components being added under ONAP, should update this yaml for enabling automated daily/weekly deployment.

DCAE SDK Integration

With Jakarta release, Consul and ConfigBindingService interface has been deprecated from DCAE. All Microservice configuration are resolved through files mounted via Configmap created part of dcae-services helm chart deployment. CBS SDK library are available within DCAE which can be used by DCAE Microservices for configuration retrieval. For details on the API - refer CBS SDK Java Library .

Corresponding CBS library available also for python components - Python Modules

Additional DACE SDK/libraries is also available for DMAAP interface; for more info refer Java Library

Its strongly recommended to use DCAE SDK library for consistency across DCAE services

Documentation

DCAE WIKI

The project wiki space (https://wiki.onap.org/display/DW/DCAE+Documentation) can be used to documents general design about the components itself; can serve the community to know about the component itself and point to other repo/release documentation.

README file

Each of service component repository must include a README instruction on how the repo should cloned, build locally and executed.

Release documentation

All DCAE components release specific documentation are maintained as source under dcaegen2 repository. his repo has documentation build setup and generated RTD under ONAP-RTD

Service component related RST can be added under separate directory under dcaegen2 repository. Post merge/build - corresponding RTD will be located under DCAE-Service component page of ONAP-RTD

See **Resource** at the bottom for more information on RTD.

Demo

To be able to certify the component for release, the MS owner should present demo to project team (and integration team) using onap Jenkins build images/container and manual deployment via blueprint under DCAE platform. This demo should be ideally completed around M4 deadline to meet release timeline.

Demo Guidelines

- Each demo is expected to be under <20 min with ~5 min Q&A
- Demo scope to include following

    a. Info on ONAP deployment dependencies/pre-requisite

b. Walkthrough component Helm-charts configuration
c. Deployment Demo (though helm)
d. Deployment Validation (health check/logs)
e. Functional Flow simulation (using scripts/curl for mocking up feed)
f. Corresponding validation (logs and/or dmaap)
g. Documentation references

- Planned/Future Updates
    - List Features/stories/bugs (with JIRA's) deferred to next release

Resources

JJB - https://wiki.onap.org/display/DW/Using+Standard+Jenkins+Job+%28JJB%29+Templates

CSIT - Creating a CSIT Test

Documentation - 2017-09-19 Documentation Tutorial, Further information regarding documentation can also be found here: http://onap.readthedocs.io/en/latest/index.html.

DCAE JIRA - https://jira.onap.org/secure/RapidBoard.jspa?rapidView=49&view=planning.nodetail