CLAMP R6 - M1 Release Planning

1. Overview
2. Scope
   - 2.1 What is this release trying to address?
   - 2.2 Use Cases
   - 2.3 Minimum Viable Product
   - 2.4 Functionality
     - 2.4.1 Epics
     - 2.4.2 Stories
   - 2.5 Longer term Roadmap
3. Release Deliverables
4. Sub-Components
5. Architecture
   - 5.1 High level architecture diagram
     - 5.1.1 Architecture
   - 5.2 Platform Maturity
   - 5.3 API Incoming Dependencies
   - 5.4 API Outgoing Dependencies
   - 5.5 Third Party Products Dependencies
6. Testing and Integration Plans
7. Gaps
8. Known Defects and Issues
9. Risks
10. Resources
11. Release Milestone
12. Team Internal Milestone
13. Documentation, Training
14. Other Information
   - 14.1 Vendor Neutral
   - 14.2 Free and Open Source Software
15. Charter Compliance
16. Release Key Facts

Overview

<table>
<thead>
<tr>
<th>Project Name</th>
<th>Enter the name of the project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target Release Name</td>
<td>Frankfurt Release</td>
</tr>
<tr>
<td>Project Lifecycle State</td>
<td>Incubation. (Refer to ONAP Charter, section 3.3 Project Lifecycle for further information)</td>
</tr>
<tr>
<td>Participating Company</td>
<td>AT&amp;T, Ericsson, Huawei, ... (TBC)</td>
</tr>
</tbody>
</table>

Scope

What is this release trying to address?

CLAMP want to enhance the flexibility and ease of introducing new Control loop by:

1. Implementing a new Control Loop creation flow: Self Serve Control Loop.
2. Add Tosca policy-model support for Operational Policies definitions.
3. Add integration to CDS for Actor/Action selection.
4. Move from SearchGuard to OpenDistro
5. Document current upgrade component strategy (TSC must have)
6. SECCOM Perform Software Composition Analysis - Vulnerability tables (TSC must have)
7. SECCOM Password removal from OOM HELM charts (TSC must have)
8. SECCOM HTTPS communication vs. HTTP (TSC must have) - CLAMP has already moved to HTTPS so no work is required here

<table>
<thead>
<tr>
<th>Scope</th>
<th>Priority</th>
<th>Committer</th>
<th>Resources</th>
<th>Epic</th>
<th>Dependencies</th>
</tr>
</thead>
</table>
Use Cases

The existing use cases are still going to be supported and additional use cases will be supported for the Frankfurt Release (as defined by the Control loop sub committee and TSC)

Minimum Viable Product

The minimum viable product that we aim to reach within R6 is to have the CLAMP application El Alto(R5) features at least running with, the new Operational policy-model and the new additional self-serve Control Loop flow working.

Functionalities

List the functionalities that this release is committing to deliver by providing a link to JIRA Epics and Stories. In the JIRA Priority field, specify the priority (either High, Medium, Low). The priority will be used in case de-scoping is required. Don’t assign High priority to all functionalities.

Epics

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>T</th>
<th>Created</th>
<th>Updated</th>
<th>Due</th>
<th>Assignee</th>
<th>Reporter</th>
<th>P</th>
<th>Status</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLAMP-834</td>
<td>Release Candidate 1 Integration and Test</td>
<td>🔥</td>
<td>Apr 21, 2020</td>
<td>May 05, 2020</td>
<td>Apr 30, 2020</td>
<td>Gervais-Martial Ngueko</td>
<td>David McBride</td>
<td></td>
<td>CLOSED</td>
<td>Done</td>
</tr>
<tr>
<td>CLAMP-657</td>
<td>Release Candidate 0 Integration and Test</td>
<td>🔥</td>
<td>Mar 01, 2020</td>
<td>Apr 03, 2020</td>
<td>Mar 26, 2020</td>
<td>Gervais-Martial Ngueko</td>
<td>David McBride</td>
<td></td>
<td>CLOSED</td>
<td>Done</td>
</tr>
<tr>
<td>CLAMP-623</td>
<td>SECCOM or related work</td>
<td>🔥</td>
<td>Jan 21, 2020</td>
<td>Apr 27, 2020</td>
<td></td>
<td>Gervais-Martial Ngueko</td>
<td>Gervais-Martial Ngueko</td>
<td></td>
<td>CLOSED</td>
<td>Done</td>
</tr>
</tbody>
</table>
### Stories

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>T</th>
<th>Created</th>
<th>Updated</th>
<th>Due</th>
<th>Assignee</th>
<th>Reporter</th>
<th>P</th>
<th>Status</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLAMP-556</td>
<td>Functionality and API Freeze</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CLOSED</td>
<td>Done</td>
</tr>
<tr>
<td>CLAMP-520</td>
<td>Frankfurt release planning milestone</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CLOSED</td>
<td>Done</td>
</tr>
<tr>
<td>CLAMP-514</td>
<td>support self serve control Loop for Frankfurt release</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CLOSED</td>
<td>Done</td>
</tr>
<tr>
<td>CLAMP-492</td>
<td>CLAMP to support any policy type without additional coding</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CLOSED</td>
<td>Done</td>
</tr>
<tr>
<td>CLAMP-490</td>
<td>get list of action for a given closed loop from CDS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CLOSED</td>
<td>Done</td>
</tr>
</tbody>
</table>

10 issues
Another Key long term goal is to provide a better user experience by having more flexibility to add micro-service without code development.

At a higher level, CLAMP is about supporting and managing the broad operational life cycle of VNFs/VMs and ultimately ONAP components itself. It will offer the ability to configure, test, deploy and update control loop automation - both closed and open. Automating these functions would represent a significant saving on operational costs compared to traditional methods.

Another Key long term goal is to reach a common platform for managing control loops within ONAP: Indicate at a high level the longer term roadmap. This is to put things into the big perspective.

The long term goal is to reach a common platform for managing control loops within ONAP:

CLAMP is a platform for designing and managing control loops. It is used to setup a control loop, configure it with specific parameters for a particular network service, then deploying and undeploying it. Once deployed, the user can also update the loop with new parameters during runtime, as well as suspending and restarting it.

It interacts with other systems to deploy and execute the closed loop. For example, it receives CSAR service package from SDC, associating it with the VF resource. It requests from DCAE the instantiation of microservices to manage the closed loop flow. Further, it creates and updates multiple policies in the Policy Engine that define the control loop flow.

The ONAP CLAMP platform abstracts the details of these systems under the concept of a control loop model. The setup of a control loop and its management is represented by a workflow in which all relevant system interactions take place. This is essential for a self-service model of creating and managing control loops, where no low-level user interaction with other components is required.

At a higher level, CLAMP is about supporting and managing the broad operational life cycle of VNFs/VMs and ultimately ONAP components itself. It will offer the ability to configure, test, deploy and update control loop automation - both closed and open. Automating these functions would represent a significant saving on operational costs compared to traditional methods.

Another Key long term goal is to provide a better user experience by having more flexibility to add micro-service without code development.
A Dashboard has been introduced to allow the user to get a quick overview of the status and events of running control loops.

Release Deliverables

Indicate the outcome (Executable, Source Code, Library, API description, Tool, Documentation, Release Note...) of this release.

<table>
<thead>
<tr>
<th>Deliverable Name</th>
<th>Deliverable Description</th>
<th>Deliverable location</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLAMP Docker container</td>
<td>Docker images available on nexus3</td>
<td>Nexus3 docker registry</td>
</tr>
<tr>
<td>Source Code</td>
<td>Code of the Designer and run time of CLAMP</td>
<td>CLAMP git repository</td>
</tr>
<tr>
<td>Deployment scripts</td>
<td>Scripts that can be used to help with the container instantiation and configuration</td>
<td>CLAMP git repository</td>
</tr>
<tr>
<td>Property Files</td>
<td>Properties files that can be used to tune the configuration of CLAMP depending on the environment</td>
<td>CLAMP git repository</td>
</tr>
</tbody>
</table>

Sub-Components

...

Architecture

High level architecture diagram

*At that stage within the Release, the team is expected to provide more Architecture details describing how the functional modules are interacting.*

*Block and sequence diagrams showing relation within the project as well as relation with external components are expected.*

*Anyone reading this section should have a good understanding of all the interacting modules.*

### Architecture

- Below we show how the CLAMP application fits into ONAP. The red figure below shows the CLAMP application components. There is a design portion and an operations component.
CLAMP is separated in 3 areas, which are currently (in seed code) both supported by a single application:

1. **Design/Setup Time** (Cockpit/UI to Configure the received templates)
   a. SDC will distribute a CSAR, for a service, the part of the CSAR that CLAMP will use are:
      i. the Control Loop flow Templates(e.g. blueprint) are defined in DCAE-D(sub-component of SDC) and distributed to CLAMP by SDC. The templates format is **TOSCA**. The **blueprint is also pushed** by SDC, to DCAE platform orchestration engine.
      ii. The policy-models defining the DCAE µS used inside the blueprint. Note that policy-engine will also receive this SDC distribution and so should be also aware of those policy-models.
   b. **policies (configuration and operational policies)** are pushed/provisioned towards the Policy Component of ONAP. (those policies will be triggered by DCAE during Closed Loop operations).
      i. The DCAE team needs to provide models to Policy team in order for the Configuration policy to be built.
   c. DCAE (in Frankfurt release), plan to introduce the DCAE-DESIGNER to (this is part of the new Self Serve Control Loop initiative):
      i. design control loop flow (sequences of µS)
      ii. expose an API for CLAMP to retrieve the list of existing flow, so that CLAMP can use it to create new Control Loop's
   d. CDS (in Frankfurt release):
      i. expose by an API (or through CSAR distribution via SDC), the list of actors and corresponding actions, that CLAMP can then configure as part of an Operationnal policy configuration

2. **Run time** (DCAE-Policy, grabbing events and triggering policies based actions)
   a. the triggering to deploy (and then effectively start the closed loop) a blueprint will be manual (via CLAMP cockpit) an automatic deployment based on an event will come in future release.
   b. The CLAMP cockpit will support the following action at runtime:
      i. start (start the provisioned Closed Loop on DCAE)
      ii. stop (stop a provisioned Closed Loop on DCAE)

3. **Dashboard** (ELK based)
a. CLAMP also provides (as a separate components) an ELK stack (with specific configurations for the elk components) that listen to Control Loop events published on DMAAP on specific dmaap topics.

CLAMP will thus control the typical following control loop flow within ONAP:

![Control Loop Flow Diagram]

**Platform Maturity**

Please check the centralized wiki page: Frankfurt Release Platform Maturity

**API Incoming Dependencies**

List the API this release is expecting from other ONAP component(s) releases. Prior to Release Planning review, Team Leads must agreed on the date by which the API will be fully defined. The API Delivery date must not be later than the **release API Freeze date**.

Prior to the delivery date, it is a good practice to organize an API review with the API consumers.

<table>
<thead>
<tr>
<th>API Name</th>
<th>API Description</th>
<th>API Definition Date</th>
<th>API Delivery Date</th>
<th>API Definition link (i.e. swagger)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Same as previous release</td>
<td>SDC Client.jar library provided by SDC team) used to get service template (describing control loop flow) and blueprint id (to know which blueprint has been distributed to DCAE for this Control Loop template)</td>
<td>Already available</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Same as previous release</td>
<td>API exposed by Policy to create/update guard policies (used for scale out use case operational policies)</td>
<td></td>
<td>Already available</td>
<td></td>
</tr>
<tr>
<td>Same as previous release</td>
<td>API exposed by Policy to create/update policies</td>
<td></td>
<td>Already available</td>
<td></td>
</tr>
<tr>
<td>Same as previous release</td>
<td>API exposed by policy to get a policy-model given the policy-model-id</td>
<td>ongoing</td>
<td>TBD</td>
<td></td>
</tr>
<tr>
<td>Same as previous release</td>
<td>API exposed by Policy to create polidy pdp group</td>
<td>ongoing</td>
<td>TBD</td>
<td></td>
</tr>
<tr>
<td>Same as previous release</td>
<td>API exposed by DCAE to start/stop a Closed Loop</td>
<td></td>
<td>Already available</td>
<td></td>
</tr>
<tr>
<td>Same as previous release</td>
<td>API exposed by DCAE to trigger the deployment/undeployment of a Control Loop template</td>
<td></td>
<td>Already available</td>
<td></td>
</tr>
<tr>
<td>Same as previous release</td>
<td>API exposed by DCAE to get status of a CLAMP deployed µS</td>
<td></td>
<td>Already available</td>
<td></td>
</tr>
<tr>
<td>Same as previous release</td>
<td>API exposed by DCAE to get status of all µS</td>
<td>ongoing</td>
<td>TBD</td>
<td></td>
</tr>
<tr>
<td></td>
<td>API exposed by DCAE to get the list of Control Loop Flow available to use by CLAMP</td>
<td>ongoing</td>
<td>TBD</td>
<td></td>
</tr>
</tbody>
</table>
API Outgoing Dependencies

API this release of CLAMP is delivering to other ONAP Component(s) releases.

<table>
<thead>
<tr>
<th>API Name</th>
<th>API Description</th>
<th>API Definition Date</th>
<th>API Delivery date</th>
<th>API Definition link (i.e. swagger)</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Third Party Products Dependencies

Third Party Products mean products that are mandatory to provide services for your components. Development of new functionality in third party product may or not be expected.

List the Third Party Products (OpenStack, ODL, RabbitMQ, ElasticSearch, Crystal Reports, ...).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camel</td>
<td>framework to define routing and mediation rules</td>
<td>2.24.0</td>
</tr>
<tr>
<td>Docker</td>
<td>Container engine</td>
<td>1.17</td>
</tr>
<tr>
<td>MariaDB</td>
<td>database container</td>
<td>10.3.12</td>
</tr>
<tr>
<td>Spring boot</td>
<td>Spring boot Framework dependencies</td>
<td>2.1.5</td>
</tr>
</tbody>
</table>

In case there are specific dependencies (Centos 7 vs Ubuntu 16. Etc.) list them as well.

Testing and Integration Plans

Provide a description of the testing activities (unit test, functional test, automation,...) that will be performed by the team within the scope of this release.

Describe the plan to integrate and test the release deliverables within the overall ONAP system.

Confirm that resources have been allocated to perform such activities.

CLAMP will invest in CSIT tests to allow further integration testing, CLAMP already provided some tests as part of previous releases.

Gaps

This section is used to document a limitation on a functionality or platform support. We are currently aware of this limitation and it will be delivered in a future Release.

List identified release gaps (if any), and its impact.

<table>
<thead>
<tr>
<th>Gaps identified</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Testing/Integration</td>
<td>limited testing of final product</td>
</tr>
</tbody>
</table>

Known Defects and Issues

Please refer to Frankfurt Defect Status

Risks

List the risks identified for this release along with the plan to prevent the risk to occur (mitigation) and the plan of action in the case the risk would materialized (contingency).

Please update any risk on the centralized wiki page - Frankfurt Risks

Resources

Link toward the Resources Committed to the Release centralized page.
Release Milestone

The milestones are defined at the Release Level and all the supporting project agreed to comply with these dates.

Team Internal Milestone

This section may be used to document internal milestones that the team agreed on.

Also, in the case the team has made agreement with other team to deliver some artifacts on a certain date that are not in the release milestone, provide these agreements and dates in this section.

It is not expected to have a detailed project plan.

<table>
<thead>
<tr>
<th>Date</th>
<th>Project</th>
<th>Deliverable</th>
</tr>
</thead>
<tbody>
<tr>
<td>To fill out</td>
<td>sdc</td>
<td>sdc UI/UX SDK</td>
</tr>
</tbody>
</table>

Documentation, Training

- Highlight the team contributions to the specific document related to the project (Config guide, installation guide...).
- Highlight the team contributions to the overall Release Documentation and training asset.
- High level list of documentation, training and tutorials necessary to understand the release capabilities, configuration and operation.
- Documentation includes items such as:
  - Installation instructions
  - Configuration instructions
  - Developer guide
  - End User guide
  - Admin guide
  - ...

⚠️ Note
The Documentation project will provide the Documentation Tool Chain to edit, configure, store and publish all Documentation asset.

Other Information

Vendor Neutral

If this project is coming from an existing proprietary codebase, ensure that all proprietary trademarks, logos, product names, etc. have been removed. All ONAP deliverables must comply with this rule and be agnostic of any proprietary symbols.

Free and Open Source Software

FOSS activities are critical to the delivery of the whole ONAP initiative. The information may not be fully available at Release Planning, however, to avoid late refactoring, it is critical to accomplish this task as early as possible. List all third party Free and Open Source Software used within the release and provide License type (BSD, MIT, Apache, GNU GPL, ...). In the case non Apache License are found inform immediately the TSC and the Release Manager and document your reasoning on why you believe we can use a non Apache version 2 license.

Each project must edit its project table available at Project FOSS

Charter Compliance

The project team comply with the ONAP Charter.

Release Key Facts

Fill out and provide a link toward the centralized Release Artifacts.