

# MultiCloud Jakarta Release Planning

The content of this template is expected to be fill out for M1 Release Planning Milestone.

- 1 [Overview](#)
- 2 [Scope](#)
  - 2.1 [What is this release trying to address?](#)
  - 2.2 [Requirements](#)
  - 2.3 [Minimum Viable Product](#)
  - 2.4 [Functionalities](#)
    - 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](#)
  - 5.2 [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](#)

## Overview

Project Name	Enter the name of the project
Target Release Name	Jakarta
Project Lifecycle State	Mature
Participating Company	Orange, Wind River

## Scope

### What is this release trying to address?

- 1, Align to S3P and security requirements and Architecture.
- 2, Enhance MultiCloud support to CNF orchestration

### Requirements

Functional requirements

REQ-890 - Getting issue details...

STATUS

Non-functional requirements (TSC must have)

Requirement with risk:

## Minimum Viable Product

Enhanced MultiCloud Plugin for k8s

## 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

MULTICLOUD-890 - Getting issue details...

STATUS

## Stories

## Longer term roadmap

Provide a mediation layer between ONAP and underlying infrastructures, including openstack, k8s, public clouds, etc..

## Release Deliverables

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

Deliverable Name	Deliverable Description
Docker Images	Docker images for all Multi-VIM/Cloud components
Documentations	Developer and user documentations for all Multi-VIM/Cloud components
Maven Artifacts	Maven Artifacts for all Multi-VIM/Cloud components
Release Note	Release Note cover all Multi-VIM/Cloud components
Source Code	Source code for all Multi-VIM/Cloud components

## Sub-Components

List all sub-components part of this release.  
Activities related to sub-components must be in sync with the overall release.

Please see the INFO.yaml files associated with each repo as the authoritative sources of information. <https://gerrit.onap.org/r/admin/repos/q/filter:multicloud>

## Architecture

### High level architecture diagram

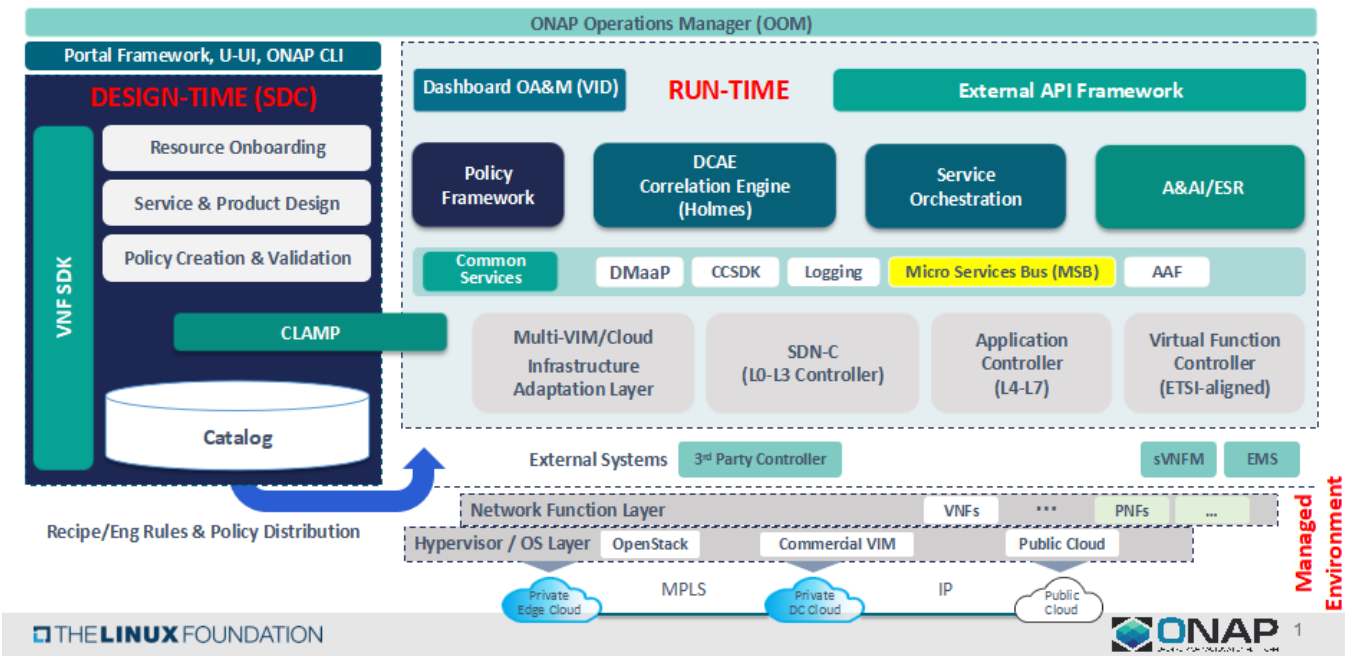
At that stage within the Release, the team is expected to provide more Architecture details describing how the functional modules are interacting.

Indicate where your project fit within the [ONAP Architecture diagram](#).

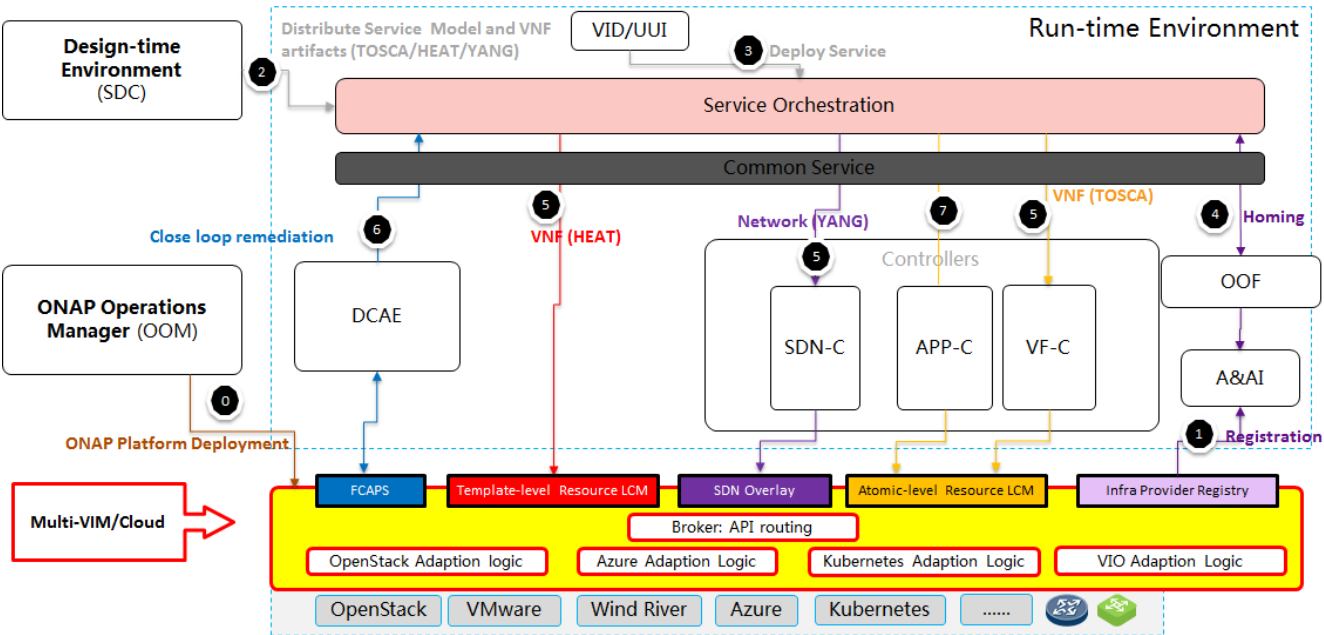
Block and sequence diagrams showing relation within the project as well as relation with external components are expected.

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

The Diagram below shows how Multi-VIM/Cloud fits into ONAP architecture in a high level view



The Diagram below depicts the interaction between Multi-VIM/Cloud and related ONAP components



## Platform Maturity

Referring to [CII Badging Security Program](#) and [Platform Maturity Requirements](#), fill out the table below by indicating the actual level , the targeted level for the current release and the evidences on how you plan to achieve the targeted level.

Area	Actual Level	Targeted Level for current Release	How, Evidences	Comments

Performance	1	1		<ul style="list-style-type: none"> <li>0 -- none</li> <li>1 – baseline performance criteria identified and measured</li> <li>2 &amp; 3 – performance improvement plans created &amp; implemented</li> </ul>
Stability	2	2		<ul style="list-style-type: none"> <li>0 – none</li> <li>1 – 72 hours component level soak w/random transactions</li> <li>2 – 72 hours platform level soak w/random transactions</li> <li>3 – 6 months track record of reduced defect rate</li> </ul>
Resiliency	2	2		<ul style="list-style-type: none"> <li>0 – none</li> <li>1 – manual failure and recovery (&lt; 30 minutes)</li> <li>2 – automated detection and recovery (single site)</li> <li>3 – automated detection and recovery (geo redundancy)</li> </ul>
Security	1	1		<ul style="list-style-type: none"> <li>0 – none</li> <li>1 – CII Passing badge + 50% Test Coverage</li> <li>2 – CII Silver badge; internal communication encrypted; role-based access control and authorization for all calls</li> <li>3 – CII Gold</li> </ul>
Scalability	1	1		<ul style="list-style-type: none"> <li>0 – no ability to scale</li> <li>1 – single site horizontal scaling</li> <li>2 – geographic scaling</li> <li>3 – scaling across multiple ONAP instances</li> </ul>
Manageability	2	2		<ul style="list-style-type: none"> <li>1 – single logging system across components; instantiation in &lt; 1 hour</li> <li>2 – ability to upgrade a single component; tracing across components; externalized configuration management</li> </ul>
Usability	2	2		<ul style="list-style-type: none"> <li>1 – user guide; deployment documentation; API documentation</li> <li>2 – UI consistency; usability testing; tutorial documentation</li> </ul>

## • API Incoming Dependencies

List the API this project is expecting from other projects.

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.

API Name	API Description	API Definition Date	API Delivery date	API Definition link (i.e.swagger)
<a href="#">eventListner/v5</a>	VES collector event publish APIs	N/A since It is ready now	N/A since It is ready now	<a href="https://wiki.onap.org/download/attachments/13599708/vs.html?version=1&amp;modificationDate=1503378662000&amp;api=v2">https://wiki.onap.org/download/attachments/13599708/vs.html?version=1&amp;modificationDate=1503378662000&amp;api=v2</a>

## • API Outgoing Dependencies

API this project is delivering to other projects.

API Name	API Description	API Definition Date	API Delivery date	API Definition link (i.e.swagger)
OpenStack Proxy endpoint	OpenStack APIs (align to newton release) with proxied endpoints exposed by each plugin (OpenStack, VIO)	Ready Since Amsterdam Release	Ready Since Amsterdam Release	<a href="https://onap.readthedocs.io/en/latest/submodules/multicloud/framework.git/docs/MultiCloud-APIv1-Specification.html">https://onap.readthedocs.io/en/latest/submodules/multicloud/framework.git/docs/MultiCloud-APIv1-Specification.html</a>

VFC oriented APIs	abstract APIs for VFC which is inherited from OPENO for each plugin (OpenStack, VIO)	Ready Since Amsterdam Release	Ready Since Amsterdam Release	<a href="https://onap.readthedocs.io/en/latest/submodules/multicloud/framework.git/docs/MultiCloud-APIv1-Specification.html">https://onap.readthedocs.io/en/latest/submodules/multicloud/framework.git/docs/MultiCloud-APIv1-Specification.html</a>
OOF oriented APIs	Capacity check to filter out those underlying VIM/Cloud without adequate resources for VNF placement	Ready Since Beijing Release	Ready Since Beijing Release	<a href="https://onap.readthedocs.io/en/latest/submodules/multicloud/framework.git/docs/MultiCloud-APIv1-Specification.html">https://onap.readthedocs.io/en/latest/submodules/multicloud/framework.git/docs/MultiCloud-APIv1-Specification.html</a>
VIM/Cloud LCM API	Registry API is used to discover infrastructure's resources, including HPA and register them into AAI	Ready Since Amsterdam Release	Ready Since Amsterdam Release	<a href="https://onap.readthedocs.io/en/latest/submodules/multicloud/framework.git/docs/MultiCloud-APIv1-Specification.html">https://onap.readthedocs.io/en/latest/submodules/multicloud/framework.git/docs/MultiCloud-APIv1-Specification.html</a>
FCAPS configuration API	FCAPS configuration API is to provision VESagent with VES collector endpoint, metric to collect, event to report, etc.	Ready Since Beijing Release	Ready Since Beijing Release	<a href="https://onap.readthedocs.io/en/latest/submodules/multicloud/framework.git/docs/MultiCloud-APIv1-Specification.html">https://onap.readthedocs.io/en/latest/submodules/multicloud/framework.git/docs/MultiCloud-APIv1-Specification.html</a>
SO oriented APIs	Generic API for SO to adapt to various VIM/Cloud type with MultiCloud plugin	Ready Since Dublin Release	Ready Since Dublin Release	<a href="https://onap.readthedocs.io/en/latest/submodules/multicloud/framework.git/docs/MultiCloud-APIv1-Specification.html">https://onap.readthedocs.io/en/latest/submodules/multicloud/framework.git/docs/MultiCloud-APIv1-Specification.html</a>

## • 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, ...).

Name	Description	Version
memcached	Memcached is an in-memory key-value store for small chunks of arbitrary data (strings, objects)	1.4.33-1
rabbitmq-server	RabbitMQ is an implementation of AMQP, the emerging standard for high performance enterprise messaging. The RabbitMQ server is a robust and scalable implementation of an AMQP broker	3.6.6-1
uwsgi	uWSGI is a fast, self-healing and developer/sysadmin-friendly application container server	2.0.17
celery	Celery is an asynchronous task queue based on distributed message passing	4.0
keystoneauth1	library for authenticating to an OpenStack-based cloud	2.18.0
Django	A high-level Python Web framework that encourages rapid development and clean, pragmatic design.	1.9.6
djangorestframework	A powerful and flexible toolkit for building Web APIs	3.3.3
httplib2	A comprehensive HTTP client library, httplib2 supports many features left out of other HTTP libraries.	0.9.2
python-memcached	a Python interface to the memcached memory cache daemon	1.59
onappyllog	onap python logging library	1.0.6

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.

- Unit Test: 50% above coverage
- Functional test: CSIT to cover major functionalities
- Integration test: Support integration team to provide the end to end integration test.
- All the above should be automation tests run on the LF Jenkins Infrastructure.

## • 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.

Gaps identified	Impact
N/A	N/A

• Known Defects and Issues

Provide a link toward the list of all known project bugs.

• 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).

Risk identified	Mitigation Plan	Contingency Plan

• Resources

Please see the INFO.yaml files associated with each repo as the authoritative sources of information. <https://gerrit.onap.org/r/admin/repos/q/filter:multicloud>

• Release Milestone

The milestones are defined at the [Release Planning: Jakarta](#) and all the supporting project agreed to comply with these dates.

• Team Internal Milestone

This section is optional and may be used to document internal milestones within a project team or multiple project teams. For instance, 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, it is erecommended to provide these agreements and dates in this section.

It is not expected to have a detailed project plan.

Date	Project	Deliverable
To fill out	To fill out	To fill out

• Documentation, Training

- Highlight the team contributions to the specific document related to he 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.

Please refer to [Jakarta Documentation](#)

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](#).