

# Casablanca-M1-CLI Release planning

- 1 [Overview](#)
- 2 [Scope](#)
  - 2.1 [Use Cases](#)
    - 2.1.1 [Epics](#)
    - 2.1.2 [Stories](#)
  - 2.2 [Longer term road map](#)
- 3 [Release Deliverable](#)
- 4 [Sub-Components](#)
- 5 [ONAP Dependencies](#)
  - 5.1 [Platform Maturity](#)
- 6 [Architecture](#)
  - 6.1 [High level architecture diagram](#)
    - 6.1.1 [CLI Components](#)
    - 6.1.2 [Sample YAML CLI template](#)
  - 6.2 [API Incoming Dependencies](#)
  - 6.3 [CLI Outgoing Dependencies](#)
  - 6.4 [Third Party Products Dependencies](#)
- 7 [Testing and Integration Plans](#)
- 8 [Gaps](#)
- 9 [Known Defects and Issues](#)
- 10 [Risks](#)
- 11 [Resources](#)
- 12 [Release Milestone](#)
- 13 [Team Internal Milestone](#)
- 14 [Documentation, Training](#)
- 15 [Other Information](#)
  - 15.1 [Vendor Neutral](#)
  - 15.2 [Free and Open Source Software](#)

## Overview

Project Name	Enter the name of the project
Target Release Name	Casablanca
Project Lifecycle State	incubation
Participating Company	Huawei

## Scope

- Standardise Open Command Specification (OCS) standard.
- Add VNFSDK commands (Stretch goal)
- Meed CII bading stanadard

## Use Cases

As a user/operator, i should be able to perform the ONAP operations from OCC/Linux console, required for Casablanca release.

## Minimum Viable Product




Standardize Open Command Specification (OCS) standard.

Support and add support for VNFSDK

## 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

Key	Summary	T	Created	Updated	Due	Assignee	Reporter	P	Status	Resolution
CLI-124	Hyper links are missing		Aug 17, 2018	Oct 11, 2018		Unassigned	None		CLOSED	Done
CLI-123	Command profile		Aug 13, 2018	Sep 20, 2018		Unassigned	None		CLOSED	Done
CLI-122	Enable Casablanca support		Aug 01, 2018	Nov 23, 2018		Unassigned	None		CLOSED	Done
CLI-121	EOL Amsterdam support		Aug 01, 2018	Sep 17, 2018		Unassigned	None		CLOSED	Done
CLI-120	CII badging		Jun 28, 2018	Sep 19, 2018		Unassigned	None		CLOSED	Done
CLI-118	Support VNFSDK		Jun 27, 2018	Sep 20, 2018		Unassigned	None		CLOSED	Done
CLI-68	Add CLI support for VNF-SDK		Nov 07, 2017	Sep 20, 2018		Unassigned	None		CLOSED	Done

7 issues

Stories

Key	Summary	T	Created	Updated	Due	Assignee	Reporter	P	Status	Resolution
CLI-124	Hyper links are missing		Aug 17, 2018	Oct 11, 2018		Unassigned	None		CLOSED	Done
CLI-123	Command profile		Aug 13, 2018	Sep 20, 2018		Unassigned	None		CLOSED	Done
CLI-122	Enable Casablanca support		Aug 01, 2018	Nov 23, 2018		Unassigned	None		CLOSED	Done
CLI-121	EOL Amsterdam support		Aug 01, 2018	Sep 17, 2018		Unassigned	None		CLOSED	Done
CLI-120	CII badging		Jun 28, 2018	Sep 19, 2018		Unassigned	None		CLOSED	Done
CLI-118	Support VNFSDK		Jun 27, 2018	Sep 20, 2018		Unassigned	None		CLOSED	Done
CLI-68	Add CLI support for VNF-SDK		Nov 07, 2017	Sep 20, 2018		Unassigned	None		CLOSED	Done

7 issues

Longer term road map

As CLI is critical in devops environment, in future, CLI will be extended to use in VNF boot scripts and Integration projects to make the ONAP integration point smoother and easier.

Release Deliverable

Deliverable Name	Deliverable Description
------------------	-------------------------

CLI Nexus zip archive	ZIP archive used to install the CLI manually
CLI docker container	Docker container to run the ONAP command and will integrate into demo environment
Documentation	Developer guide User guide Release notes Architecture open-cli-schema-1.0 specification

## Sub-Components

NIL

## ONAP Dependencies

1. VNFSDK
  - a. API to conduct tests, verification and market places

## Platform Maturity

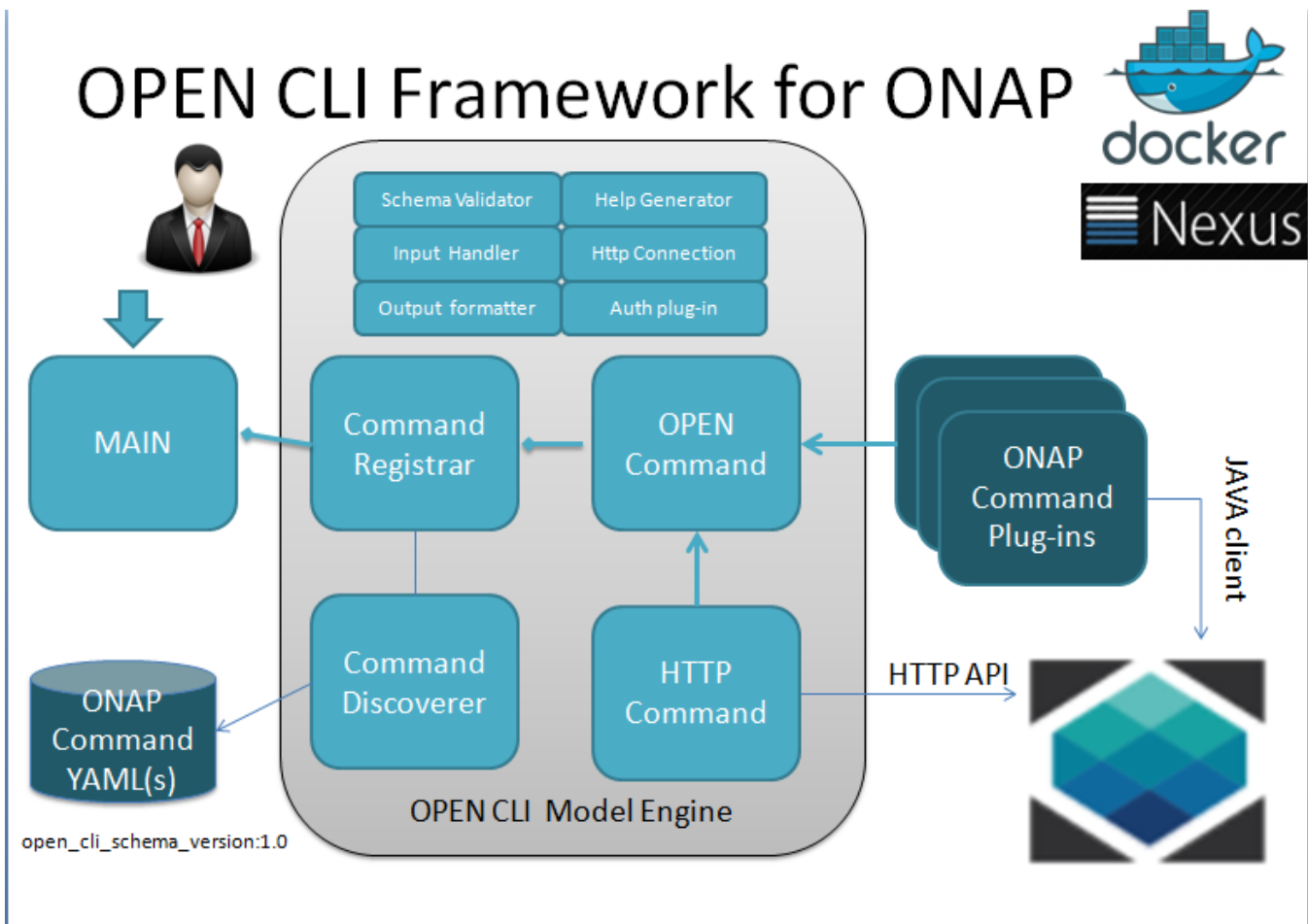
Referring to [CII Badging Security Program](#) and [Platform Maturity Requirements](#), fill out the table below by indicating the actual level , the targeted level for the current release and the evidences on how you plan to achieve the targeted level.

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> <p><b>NOTE:</b></p> <p>Performance test is not performed in amsterdam though it performant, which is best observed during the usage.</p> <p>so current level is marked 0. In this release, we will add benchmarking test cases to show the level 1</p>
Stability	1	1		<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> <p><b>NOTE:</b></p> <p>Test cases are not added to test the stability and benchmark it. so current level is marked 0. Otherwise, its very stable at level 1 or more.</p> <p>will address the test cases in this release.</p>
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	1	1		<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>

## Architecture

### High level architecture diagram



### CLI Components

- Open Command** – Models the most of the requirements of CLI and it's the CLI schema engine, understands the OPEN-CLI 1.0 schema and make it as Command
- Command Plug-in** : Provides extensibility support to implement any commands which depends on java api
- HTTP Command** : An special command plug-in provided by framework to implement the Command on top of REST API without writing any java code. (only YAML template is sufficient)
- Command Discoverer** – Discovers the available CLI templates (YAML file created by using OPEN-CLI 1.0 schema) placed under OPEN\_CLI\_HOME directory or its sub-directories and register them into Command Registrar.
- Command Registrar** – Maintains the map of command name vs actual command executable
- Main** – Provides the interactive/direct command mode to run the commands from Linux OS console

### Sample YAML CLI template

Following sample YAML shows the YAML file used to create the microservice in Open-O using CLI

## openo microservice-create

```
open_cli_schema_version: 1.0
name: microservice-create
description: Register microservice into Open-O
service:
  name: msb
  version: v1

parameters:
- name: service-name
  description: Open-O service name
  type: string
  short_option: x
  long_option: service-name
  is_optional: false
- name: service-version
  description: Open-O service version
  type: string
  short_option: y
  long_option: service-version
  is_optional: false
- name: service-url
  description: Open-O service base url
  type: url
  short_option: r
  long_option: service-url
  is_optional: false
- name: node-ip
  description: Open-O service running node IP
  type: string
- name: node-port
  description: Open-O service running node port
  type: string
- name: create-or-update
  description: Open-O service create or update
  type: bool
  default_value: true
results:
  direction: portrait
  attributes:
    - name: name
      description: Open-O service name
      scope: short
      type: string
    - name: version
      description: Open-O service version
      scope: short
      type: string
    - name: url
      description: Open-O service base url
      scope: short
      type: url
    - name: status
      description: Open-O service status
      scope: short
      type: long
    - name: nodes
      description: Open-O service running nodes
      scope: long
      type: string
    - name: location
      description: Open-O service location
      scope: long
      type: url
http:
  request:
    uri: /services
    method: POST
    body: '{"serviceName":"${service-name}","version":"${service-version}","url":"${service-url}","protocol":'
```

```
REST", "visualRange": "1", "lb_policy": "hash", "nodes": [{ "ip": "${node-ip}", "port": "${node-port}", "ttl": 0 } ] }'
  headers:
  queries:
    createOrUpdate: ${create-or-update}
  success_codes:
    - 201
    - 200
  result_map:
    name: ${b{$.serviceName}}
    version: ${b{$.version}}
    url: ${b{$.url}}
    status: ${b{$.status}}
    nodes: ${b{$.nodes[*].ip}}:${b{$.nodes[*].port}}
    location: ${h{Location}}
```

## API Incoming Dependencies

To provide the required commands for each of the ONAP services, this project will depends on the REST API provided by every other services such as SO, AAI, etc

## CLI Outgoing Dependencies

NOTE: This project delivers CLI and not API

## Third Party Products Dependencies

Name	Description	Version
Docker	Docker container for CLI	Aligned with existing Docker version of ONAP
Ubuntu	For running CLI manually	14.04 / 16.04 32-bit/64-bit
Open JDK	For Java RTE	1.8

## 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.

## 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
NIL	NIL

## Known Defects and Issues

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

Key	Summary	T	Created	Updated	Due	Assignee	Reporter	P	Status	Resolution
CLI-475	Kohn Release Note not updated	🔴	Nov 30, 2022	Dec 05, 2022		Unassigned	Thomas Kulik	==	OPEN	Unresolved
CLI-466	Node port issue	🔴	Jun 28, 2022	Sep 19, 2022		Unassigned	None	^	OPEN	Unresolved
CLI-440	Sonar issues present in ONAP vnfsdk validation	🔴	Mar 20, 2022	Mar 20, 2022		Unassigned	None	^	OPEN	Not a Bug
CLI-438	issue in CLI docker image build	🔴	Mar 08, 2022	Mar 08, 2022		Unassigned	None	^	DELIVERED	Done
CLI-422	fix documentation	🔴	Oct 28, 2021	Nov 29, 2022		Unassigned	Thomas Kulik	^^	CLOSED	Done
CLI-376	update discovery,run robot scripts	🔴	Mar 29, 2021	Sep 19, 2022		Unassigned	None	==	OPEN	Unresolved
CLI-349	No Metadata in OnapCommand Info section	🔴	Jan 12, 2021	Jan 28, 2021		Unassigned	None	==	CLOSED	Done
CLI-348	Optional parameters with default values are lost after overriding selected parameter	🔴	Jan 12, 2021	Jan 28, 2021		Unassigned	None	==	CLOSED	Done
CLI-325	Using version 6.0.0 of CLI cause error in VNFSKD Marketplace	🔴	Oct 13, 2020	Nov 05, 2020		Unassigned	None	^^	CLOSED	Done
CLI-316	CLI certificates have bad owner	🔴	Sep 14, 2020	Sep 19, 2022		Unassigned	None	==	OPEN	Unresolved
CLI-315	use java 11 min recommended version	🔴	Sep 10, 2020	Oct 26, 2020		Unassigned	None	v	CLOSED	Done
CLI-267	Migrate Port 9090 -> 9443	🔴	Apr 20, 2020	Aug 05, 2020		Unassigned	None	==	CLOSED	Done
CLI-265	Add HTTPS certificate in OOM CLI charts	🔴	Apr 16, 2020	Apr 30, 2020		Unassigned	None	==	CLOSED	Done
CLI-264	Update SDC vf-model-add-artifact cli, add parameter artifactLabel	🔴	Apr 15, 2020	Apr 23, 2020		Unassigned	None	==	CLOSED	Done
CLI-263	There are some bugs on vnf-tosca-lcm.py script	🔴	Apr 15, 2020	Apr 23, 2020		Unassigned	None	==	CLOSED	Done
CLI-257	frankfurt release	🔴	Mar 09, 2020	Feb 23, 2022		Unassigned	None	==	CLOSED	Won't Do
CLI-255	Pods still run as root	🔴	Mar 02, 2020	Apr 21, 2020		Unassigned	None	^	CLOSED	Done
CLI-254	HTTP port open	🔴	Mar 02, 2020	Apr 21, 2020		Unassigned	None	^	CLOSED	Done
CLI-250	Sonar cloud migration	🔴	Feb 21, 2020	Mar 04, 2020		Unassigned	None	^	CLOSED	Done
CLI-229	Minor CLI bug fixes	🔴	Sep 30, 2019	Oct 14, 2019		Unassigned	None	==	CLOSED	Not a Bug

Showing 20 out of 88 issues

## 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
-----------------	-----------------	------------------

NONE		-
------	--	---

## Resources

Fill out [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 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 recommended to provide these agreements and dates in this section.

It is not expected to have a detailed project plan.

Date	Project	Deliverable
NIL	NIL	NIL

## 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](#).