

APPC Beijing M1 Release Planning Template

DRAFT PROPOSAL FOR COMMENTS

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



Info

Use the "Copy" and "Move" options (available under the ..., top right of this page) to duplicate this template into your project wiki.
Use the Wiki to document the release plan. Don't provide PowerPoint.
Use as much diagrams and flow charts as you need, directly in the wiki, to convey your message.

- 1 [Overview](#)
- 2 [Scope](#)
 - 2.1 [What is this release trying to address?](#)
 - 2.2 [Use Cases](#)
 - 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 [ONAP Dependencies](#)
- 6 [Architecture](#)
 - 6.1 [High level architecture diagram](#)
 - 6.2 [Platform Maturity](#)
 - 6.3 [API Incoming Dependencies](#)
 - 6.4 [API Outgoing Dependencies](#)
 - 6.5 [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	Beijing Release
Project Lifecycle State	Incubation. Refer to ONAP Charter, section 3.3 Project Lifecycle for further information
Participating Company	AT&T, Intel, Tech Mahindra

Scope

What is this release trying to address?

The Beijing Release will be focusing on Platform Maturity as a priority. Additional LCM actions will be added to the scope of APIs supported. These additional LCM actions have already been developed and validated by AT&T and will be contributed to ONAP.

More specifically, APPC will focus on the following areas in Beijing.

- Remove references and use of openecomp and replace with onap
- Address Platform Maturity requirements to the extent possible to comply with minimum requirements requested for Beijing. Further details below in Platform Maturity table.
- Support AAF (Authentication and Authorization Framework) for API access

- Dependency on AAF project to provide feature (**AAF-91**) to enable AAF security on the web server level (jetty level). AAF has accepted the story for Beijing.
- Upgrade ODL version to Nitrogen (driven by CCSDK dependency)
- Replace MySQL with MariaDB (driven by CCSDK/SDNC dependency)
- Increase Code Coverage to 50%
- Provide support for the following new LCM actions:
 - Following Actions in support of In-place software upgrade
 - QuiesceTraffic
 - ResumeTraffic
 - UpgradeSoftware
 - UpgradePreCheck
 - UpgradePostCheck
 - UpgradeBackup
 - UpgradeBackout
 - Additional LCM actions including:
 - ~~ActionCancel~~ (will not be part of Beijing Release)
 - ActionStatus
 - AttachVolume
 - DetachVolume
- Contribute CDT Tool - an APP-C Design Tool enabling VNF owners to create templates and other artifacts used by APP-C Configure actions (used to apply a post-instantiation configuration) as well as other life cycle commands.
- Documentation updates for Beijing
- Support new LCM action for ConfigScaleOut

Use Cases

Describe the use case this release is targeted for (better if reference to customer use case).

The use cases supported in Amsterdam release will continue to be supported as part of regression assuming all other components do likewise.

APPC contribute to two use cases as part of the functional requirements.

- **Manual** Config Scale Out will be supported..
- **Change Management** did not flag APPC has impacted because the use case they are looking at is focusing on an L3 VNF, so that would be the pervue of SDNC to do the in place software upgrade; however, if it was an L4-L7 type VNF, this would be APPC executing action. APPC will be adding capability to support an in-place software upgrade. We will contribute the code, but don't have a specific use case planned to exercise it.
-

Minimum Viable Product

Describe the MVP for this release.

- Northbound REST Layer - This layer is responsible for interfacing with APPC clients such as SO, SDC, etc... via YANG-based API vis REST HTTP, message bus (DMaaP) and exposes APPC APIs enabling performance of lifecycle operations.
- APPC Provider - responsible for validating the YANG Remote Processing Call (RPC) input and for rejected when malformed. Upon successful validation, the APPC Provider call the Dispatcher to continue the request processing.
- APPC Dispatcher - responsible for processing requests received by the Request Handler and, if valid, selects the correct Directed Graph (DG) workflow for execution. Upon DG execution completion, the Dispatcher informs the initiator of results of the execution and updated data in AAI.
- Service Logic Interpreter (SLI) - responsible for executing the DGs. The Dispatcher invokes the SLI framework to execute a specific DG based on the input action. The SLI executes the DG and the sub-DG and returns a success or failure response to the Dispatcher.
- AAI Interface - APPC uses AAI as a source of VNF data, including status, topology, and operational data. It also makes updates to AAI as a result of a lifecycle operation.
- Southbound Layer - plugin architecture, uses several adapters to communicate to VNFs
 - Restconf adaptor
 - Netconf
 - SSH (XML/CLI)
 - IaaS - Interface as a Service, which is part of the OpenDaylight platform, which provides the framework for APPC
 - Chef
 - Ansible
- Components that operate across all the APPC modules, including Logging, Security, KPI Service, data access service for access to internal DB (MySQL in R1, MariaDB planned in R2)

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
-----	---------	---	---------	---------	-----	----------	----------	---	--------	------------

APPC-431	Support Manual ScaleOut of a VNF		Jan 19, 2018	Jul 10, 2019	Unassigned	None		CLOSED	Done
APPC-415	Scalability Test		Jan 17, 2018	May 31, 2018	Unassigned	None		CLOSED	Done
APPC-414	Resiliency Support & Testing		Jan 17, 2018	May 30, 2018	Unassigned	None		CLOSED	Done
APPC-413	Stability Testing		Jan 17, 2018	May 30, 2018	Unassigned	None		CLOSED	Done
APPC-406	ODL & DB for Beijing		Jan 15, 2018	May 30, 2018	Unassigned	None		CLOSED	Done
APPC-376	Security - R2		Jan 08, 2018	May 30, 2018	Unassigned	None		CLOSED	Done
APPC-356	New LCM Action and component evolution		Dec 15, 2017	Feb 07, 2018	Unassigned	None		CLOSED	Done
APPC-342	Code Coverage for Beijing (R2) Release		Dec 13, 2017	Jul 10, 2019	Unassigned	None		CLOSED	Done
APPC-308	Documentation Updates for Beijing Release (R2)		Nov 02, 2017	Jun 04, 2018	Unassigned	None		CLOSED	Done
APPC-305	Platform Hardening - R2		Nov 02, 2017	May 20, 2018	Unassigned	None		CLOSED	Done
APPC-187	APPC Configuration Design Tool (CDT) - GUI		Sep 01, 2017	May 30, 2018	Unassigned	None		CLOSED	Done

11 issues

Stories

Key	Summary	T	Created	Updated	Due	Assignee	Reporter	P	Status	Resolution
APPC-852	Add junit tests for Request.java		Apr 13, 2018	Apr 26, 2018		Unassigned	None		CLOSED	Done
APPC-850	Unit tests for Application Controller Config: onap/appc/requesthandler/impl classes		Apr 11, 2018	May 05, 2018		Unassigned	None		CLOSED	Done
APPC-847	APPC Resiliency Documentation		Apr 10, 2018	May 17, 2018		Unassigned	None		CLOSED	Done
APPC-846	APPC Resiliency Testing		Apr 10, 2018	May 08, 2018		Unassigned	None		CLOSED	Done
APPC-845	Add junit test for PoolException.java		Apr 10, 2018	Apr 13, 2018		Unassigned	None		CLOSED	Done
APPC-843	Add junit test for PoolExtensionException.java		Apr 10, 2018	Apr 13, 2018		Unassigned	None		CLOSED	Done
APPC-842	Add junit test for PoolDrainedException.java		Apr 10, 2018	Apr 13, 2018		Unassigned	None		CLOSED	Done
APPC-836	JUNIT updates for APPC : onap/appc/common/utis UnmodifiableProperties		Apr 09, 2018	May 05, 2018		Unassigned	None		CLOSED	Done
APPC-834	Add junit test for TimedOutException.java		Apr 06, 2018	Apr 09, 2018		Unassigned	None		CLOSED	Done
APPC-832	JUnit test for Application Controller Config: onap/appc/ccadaptor		Apr 05, 2018	Apr 17, 2018		Unassigned	None		CLOSED	Done
APPC-831	Login for Test screen and couple of fixes on test screen		Apr 05, 2018	Apr 16, 2018		Unassigned	None		CLOSED	Done

APPC-825	Unit test for RequestFailedException.java in onap/appc/adapters/iaas/impl/		Apr 03, 2018	Apr 09, 2018	Unassigned	None		CLOSED	Done
APPC-820	Unit tests for Application Controller: onap/appc/common/utills classes		Apr 02, 2018	May 05, 2018	Unassigned	None		CLOSED	Done
APPC-819	JUnit for CoreRegistry, TaskQueue, TaskQueueManager classes		Apr 02, 2018	Apr 17, 2018	Unassigned	None		CLOSED	Done
APPC-815	Add Junit for client-lib TimerServiceImpl.java		Mar 29, 2018	Apr 17, 2018	Unassigned	None		CLOSED	Done
APPC-809	JUnit tests for util.Time		Mar 29, 2018	Apr 10, 2018	Unassigned	None		CLOSED	Done
APPC-808	Add junit tests for dg common		Mar 28, 2018	Apr 10, 2018	Unassigned	None		CLOSED	Done
APPC-807	add junit for org.onap.appc.sdc.listener.Util.java		Mar 28, 2018	Apr 09, 2018	Unassigned	None		CLOSED	Won't Do
APPC-806	JUnit coverage for classes under onap.appc.flow.controller.data		Mar 28, 2018	Mar 29, 2018	Unassigned	None		CLOSED	Done
APPC-802	Provide unit test covarate for appc-common/src/main/java/org/onap/appc/logging/LoggingUtils		Mar 27, 2018	Apr 13, 2018	Unassigned	None		CLOSED	Done

Showing 20 out of 98 issues

Longer term roadmap

Indicate at a high level the longer term roadmap. This is to put things into the big perspective.

The long term road map is to achieve all the goals outlined in the approved project proposal; to be fully model and standards driven, be agnostics and make no assumptions about the network. Support configuration and lifecycle management of VNF/VNFC in a generic fashion so that on-boarding any new VNF/VNFC is just a matter of configuration and data. Longer term items include:

- Support different types of clouds, currently only support Openstack; Looking at re-architecting the southbound IaaS adapter to allow plugins for different cloud abstraction solutions (CDP-PAL, MultiCloud, etc...)
- Align to the controller architecture proposed as part of ONAP by the architecture team.

Release Deliverables

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

Deliverable Name	Deliverable Description	Deliverable Location
"App-c Image" Docker Container	Executable	Docker images available on nexus3
Java Source Code	The Java code for the main App-c components.	appc Git repository
Deployment Scripts	Linux shell scripts and Maven pom files used to generate the Docker containers.	appc/deployment Git repository
Directed Graph Xml Files (DGs)	Xml files define the directed graphs which are installed to database during startup and are used to determine actions taken by app-c	appc/deployment Git repository
Yang Model Files	Yang files are used to define the...	appc Git repository
Property Files	Property files are used to define values that may need to be changed depending on the environment app-c is run in.	appc Git repository

CDT tool	an APP-C Design Tool enabling VNF owners to create templates and other artifacts used by APP-C Configure actions (used to apply a post-instantiation configuration) as well as other life cycle commands	appc/cdt Git repository (NEW)
----------	--	-------------------------------

Sub-Components

List all sub-components part of this release.

Activities related to sub-components must be in sync with the overall release.

Sub-components are repositories and are consolidated in a single centralized place. Edit the [Release Components name for your project](#) in the centralized page.

ONAP Dependencies

List the other ONAP projects you depend on.

APPC depends on the the following components as part of the general ONAP architecture:

- **SDC**: Rest based interface exposed by SDC. APPC receives notifications from SDC on VNF information. SDC team provides an SDC Listener, which is used by APPC.
- **AAI**: APPC retrieves and updates VNF data from/to AAI.
- **DMaaP**: Message bus for communication with other components in the solution (SDC, DCAE, MSO, Portal, OOM)
- **CCSDK** - APPC currently gets ODL & DB package from CCSDK; CCSDK and APPC currently must align on ODL version.
- **AAF** - AAF is used for authentication of APIs
- **MultiVIM** - APPC can access Openstack via MultiVIM or CPD-PAL. MultiVIM is optional for APPC at this stage.

For the Beijing release, APPC has dependencies on the following three projects for specific deliverables:

- **CCSDK** - - Nitrogen ODL & MariaDB
- **AAF** - feature [AAF-91](#) - needed to address API level security
- **SO** - for manual scale out scenario

Architecture

High level architecture diagram

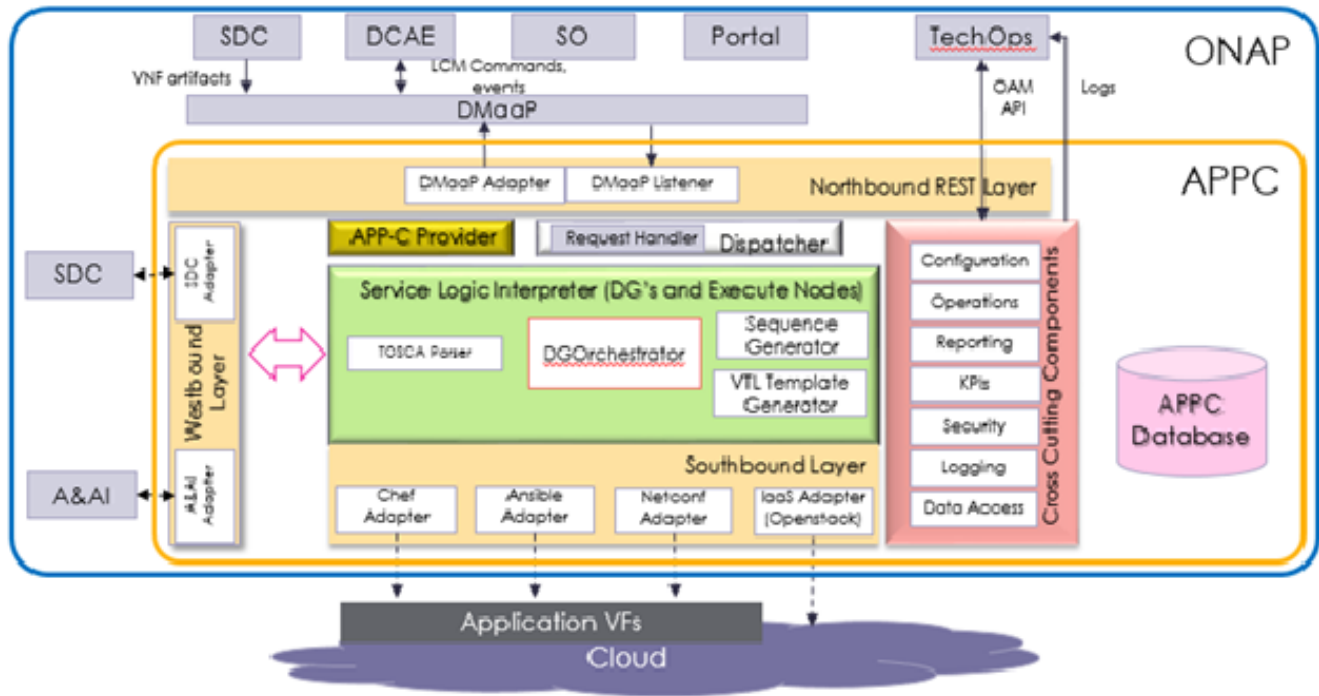
At that stage within the Release, the team is expected to provide more Architecture details describing how the functional modules are interacting.

Indicate where your project fit within the [ONAP Archiecture 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.

For details on the APPC architecture, refer to the [APPC User Guide](#).



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	0	0	Awaiting guidance from Benchmark subcommittee	<ul style="list-style-type: none"> 0 -- none 1 -- baseline performance criteria identified and measured 2 & 3 -- performance improvement plans created & implemented
Stability	0	1	APPC will do a 72 hr soak test We assume integration team will do a 72 hour soak test at the platform level.	<ul style="list-style-type: none"> 0 -- none 1 -- 72 hours component level soak w /random transactions 2 -- 72 hours platform level soak w /random transactions 3 -- 6 months track record of reduced defect rate
Resiliency	1	2	Clarification was provided by Jason Hunt that Level 1 is manual failure detection and recovery within a single site. APPC leverages CCSDK/SDNC distributions of OpenDaylight and DB. We will work closely with CCSDK/SDNC team to leverage what they are doing and not duplicate effort.	<ul style="list-style-type: none"> 0 -- none 1 -- manual failure and recovery (< 30 minutes) 2 -- automated detection and recovery (single site) 3 -- automated detection and recovery (geo redundancy)
Security	0	1	We will target 50% Sonar coverage for Beijing and complete the Passing badge Survey.	<ul style="list-style-type: none"> 0 -- none 1 -- CII Passing badge + 50% Test Coverage 2 -- CII Silver badge; internal communication encrypted; role-based access control and authorization for all calls 3 -- CII Gold

Scalability	1	1	APPC uses ODL distribution from CCSDK project. Scaling is a function of ODL capabilities - APPC can be scaled either by adding additional OpenDaylight containers and/or database containers, or by deploying multiple instances of APPC cluster.	<ul style="list-style-type: none"> 0 – no ability to scale 1 – single site horizontal scaling 2 – geographic scaling 3 – scaling across multiple ONAP instances
Manageability	1	1	APPC supports/integrates EELF.	<ul style="list-style-type: none"> 1 – single logging system across components; instantiation in < 1 hour 2 – ability to upgrade a single component; tracing across components; externalized configuration management
Usability	1	1	Documentation is already available on readthedocs . It will be updated for Beijing as needed.	<ul style="list-style-type: none"> 1 – user guide; deployment documentation; API documentation 2 – UI consistency; usability testing; tutorial documentation

• API Incoming Dependencies

List the API this project is expecting from other projects.

Prior to Release Planning review, Team Leads must agree 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)
SDC	REST API	Currently Available, but needs to be updated to use onap.org	TBD	Link toward the detailed API description
AAI	REST API	Currently Available	Currently Available	
CCSDK	OpenDayLight, SLI, AAI Client, dblib	TBD	TBD	
DMaaP	API to publish/subscribe to events sent for VNF/VM action requests.	Currently Available	Currently Available	DMaaP API
AAF	Application Authorization Framework	TBD	TBD	

• 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)
NB Interface	REST API	3/8/18	3/18/18	Link toward the detailed API description http://onap.readthedocs.io/en/latest/submodules/appc.git/docs/APPC%20LCM%20API%20Guide/APPC%20LCM%20API%20Guide.html

• 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
ODL	OpenDaylight controller platform	Nitrogen
Docker	Docker container host	1.12

MariaDB	data base container	TBD
---------	---------------------	-----

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.

- CSIT tests added as part of R1 will continue to be supported in R2
- Pairwise testing will be done in the WindRiver Dev lab similar to what was done in R1.
- Epics are created to track testing activities to address Platform Maturity items.

• 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
To fill out	To fill out

• 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
APPC-886	removed the property for configscaletout	🔴	May 07, 2018	May 08, 2018		Unassigned	None	⬆️	CLOSED	Done
APPC-558	ASDC-MESSAGE DG contains invalid xmlns www.openecomp.org	🔴	Jan 31, 2018	May 29, 2018		Unassigned	None	==	CLOSED	Done
APPC-324	appc-oam-bundle: failing on goal yang-maven-plugin	🔴	Nov 29, 2017	Jan 04, 2018		Unassigned	None	⬇️	CLOSED	Done
APPC-320	Bug Fixes in DG's for chef and ansible and encryptiontool	🔴	Nov 21, 2017	Dec 11, 2017		Unassigned	None	==	CLOSED	Done
APPC-319	APPC receives a 401 when authenticating directly to OpenStack	🔴	Nov 17, 2017	Nov 22, 2017		Unassigned	None	==	CLOSED	Done
APPC-318	Snapshot dependency for sdc was left in pom	🔴	Nov 17, 2017	Jan 24, 2018		Unassigned	None	==	CLOSED	Done
APPC-316	Null payload issue for Stop Application	🔴	Nov 09, 2017	Dec 11, 2017		Unassigned	None	⬇️	CLOSED	Done
APPC-315	appc-request-handler is giving error java.lang.NoClassDefFoundError	🔴	Nov 09, 2017	Feb 07, 2018		PRASHANT KUMAR	PRASHANT KUMAR	==	CLOSED	Done
APPC-312	APPC request is going to wrong request handler and rejecting request	🔴	Nov 04, 2017	Dec 11, 2017		PRASHANT KUMAR	PRASHANT KUMAR	==	CLOSED	Done
APPC-311	APPC-PROVIDER_LCM: HEALTHCHECK fails	🔴	Nov 03, 2017	Dec 11, 2017		Unassigned	None	==	CLOSED	Done
APPC-309	APPC LCM Provider URL missing in appc.properties	🔴	Nov 03, 2017	Aug 12, 2023		Unassigned	None	==	CLOSED	Done
APPC-307	Embed jackson-annotations dependency in appc-dg-common during run-time	🔴	Nov 02, 2017	Apr 26, 2018		Unassigned	None	⬇️	CLOSED	Done

APPC-276	Some Junit are breaking convention causing excessively long build		Oct 13, 2017	Dec 11, 2017	Unassigned	None	▼	CLOSED	Done
APPC-248	Missing code coverage due to compatibility issue between Jacoco and PowerMockue		Sep 28, 2017	Jun 15, 2018	Unassigned	None	▼	CLOSED	Done

[14 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
ODL upgrade to Nitrogen & DB to MariaDB - depends on CCSDK/SDNC projects	Accept risk	None
AAF delivery of AAF-91 in time to allow APPC to complete and test their work	Working closely with AAF team to understand their design approach	Turn AAF off for Beijing (same as in Amsterdam)

• 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
To fill out	To fill out	To fill out

• 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.

Documentation updates planned for Beijing release are tracked under Documentation Epic: [APPC-308](#)

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](#).