

SDNC : Beijing : 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](#)
- 16 [Charter Compliance](#)
- 17 [Release Key Facts](#)

Overview

Project Name	Enter the name of the project
Target Release Name	Beijing
Project Lifecycle State	Incubation
Participating Company	AT&T, Huawei, ZTE, China Mobile, AMDOCS, Intel, Orange, Coriant, Bell Canada, Tech Mahindra, Brocade

Scope

What is this release trying to address?

The SDN-C project provides a global network controller, built on the Common Controller SDK, which manages, assigns and provisions network resources. As a "global" controller, the SDN-C project is intended to run as one logical instance per enterprise, with potentially multiple geographically diverse virtual machines / docker containers in clusters to provide high availability. The project also will support the ability to invoke other local SDN controllers, including third party SDN controllers.

In the Beijing release, the SDN-C project will be used to manage, assign and provision network resources for the Beijing release use cases, listed in the Use Cases section below.

Use Cases

The use cases supported in the Beijing release are:

- Virtual Domain Name Server (vDNS)
- Virtual Firewall (vFW)
- Virtual Voice over LTE (vVoLTE)
- Virtual Customer Premise Equipment (vCPE)



























Minimum Viable Product

The Minimum Viable Product for Beijing is the set of capabilities needed to support the use cases listed above.

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
SDNC-276	Code coverage for northbound project		Mar 19, 2018	Jun 13, 2018		Dan Timoney	None		CLOSED	Done
SDNC-240	Upgrade to OpenDaylight Nitrogen		Jan 17, 2018	May 02, 2018		Dan Timoney	Dan Timoney		CLOSED	Done
SDNC-237	Support VNF Change Management		Jan 12, 2018	Jul 11, 2018		Dan Timoney	None		CLOSED	Done
SDNC-227	Upgrade to OpenDaylight Nitrogen release		Dec 13, 2017	May 02, 2018		Dan Timoney	Dan Timoney		CLOSED	Done
SDNC-222	Implement changes to improve usability /programmability of platform		Dec 13, 2017	Jul 11, 2018		Dan Timoney	Dan Timoney		CLOSED	Done
SDNC-221	Implement process/code changes necessary to conform to ONAP project standards		Dec 13, 2017	Jun 05, 2018		Dan Timoney	Dan Timoney		CLOSED	Done
SDNC-220	Implement process/code changes necessary to improve platform scalability		Dec 13, 2017	Jun 05, 2018		Dan Timoney	Dan Timoney		CLOSED	Done
SDNC-219	Implement process/code changes necessary to measure and improve platform stability		Dec 13, 2017	Jun 05, 2018		Dan Timoney	Dan Timoney		CLOSED	Done
SDNC-218	Implement code changes necessary to measure and improve performance		Dec 13, 2017	Jul 11, 2018		Dan Timoney	Dan Timoney		CLOSED	Done
SDNC-217	Implement process/code changes identified as part of CII Badging process		Dec 13, 2017	Jun 05, 2018		Dan Timoney	Dan Timoney		CLOSED	Done
SDNC-209	Improve code metrics		Dec 04, 2017	May 02, 2018		Dan Timoney	Dan Timoney		CLOSED	Done
SDNC-124	Multi-site High-available - Manual Failover		Oct 13, 2017	Jun 05, 2018		Unassigned	None		CLOSED	Done
SDNC-23	Support local high availability configuration		Jun 29, 2017	Jun 05, 2018		Dan Timoney	Dan Timoney		CLOSED	Done

[13 issues](#)

Stories

Key	Summary	T	Created	Updated	Due	Assignee	Reporter	P	Status	Resolution
SDNC-1728	Upgrade sdnc /northbound to ODL Sulfur		Aug 03, 2022	Aug 03, 2022		Dan Timoney	Dan Timoney	=	IN PROGRESS	Unresolved
SDNC-1549	Add network/RAN slice CSIT testing		May 17, 2021	Apr 27, 2023		Unassigned	Dan Timoney	=	OPEN	Unresolved
SDNC-1230	Implement GR-API PNF RPCs		Jun 03, 2020	Apr 27, 2023		Dan Timoney	Dan Timoney	=	OPEN	Unresolved
SDNC-1228	Implement GR-API PNF support		Jun 03, 2020	Apr 27, 2023		Dan Timoney	Dan Timoney	=	OPEN	Unresolved
SDNC-1227	Implement GR-API allotted resources RPCs		Jun 03, 2020	Apr 27, 2023		Dan Timoney	Dan Timoney	=	OPEN	Unresolved
SDNC-1226	Define data model for GR-API allotted resources		Jun 03, 2020	Apr 27, 2023		Dan Timoney	Dan Timoney	=	OPEN	Unresolved
SDNC-1225	Implement GR-API allotted resources		Jun 03, 2020	Apr 27, 2023		Dan Timoney	Dan Timoney	=	OPEN	Unresolved
SDNC-594	Enable support for MP2MP services		Jan 23, 2019	Jul 15, 2020		Unassigned	None	=	OPEN	Unresolved
SDNC-438	Support multiple vGMUXes in vCPE use case		Sep 11, 2018	Mar 30, 2022		Dan Timoney	None	=	IN PROGRESS	Unresolved
SDNC-224	Migrate data from release N-1 to release N		Dec 13, 2017	Sep 09, 2020		Unassigned	Dan Timoney	=	IN PROGRESS	Unresolved

10 issues

Longer term roadmap

One critical long term objective for the SDN-C project is support for integration with other third party SDN Controllers (e.g. Open Contrail), well as integration with the SDN Agent project from Open-O. For the Beijing release, since our primary goal is to support the use cases identified above and to improve platform maturity, the degree to which we support such integration will be dictated by the needs of those use cases. However, we do want to bear in mind that such integration is critical and will be included in our release plans going forward.

Release Deliverables

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

Deliverable Name	Deliverable Description	Deliverable Location
SDNC Source Code	Source code for SDNC project	ONAP Gerrit
SDNC Maven Artifacts	Compiled code that can be referenced in other projects as maven dependencies	ONAP Nexus
SDNC Docker Containers	Docker containers associated with SDNC project: <ul style="list-style-type: none"> Controller container Database container Admin portal container 	ONAP Nexus
Documentation	User and developer guides	ONAP Wiki
SDNC CI/CD automation	Scripts to automate compilation and deployment of maven artifacts and docker containers	ONAP Gerrit ONAP Jenkins

Sub-Components

Subcomponents of each ONAP project may be found on the [Resources and Repositories \(Deprecated\)](#) page on this wiki. Please see the SDN-C section of that page for subcomponent list of SDN-C.

ONAP Dependencies

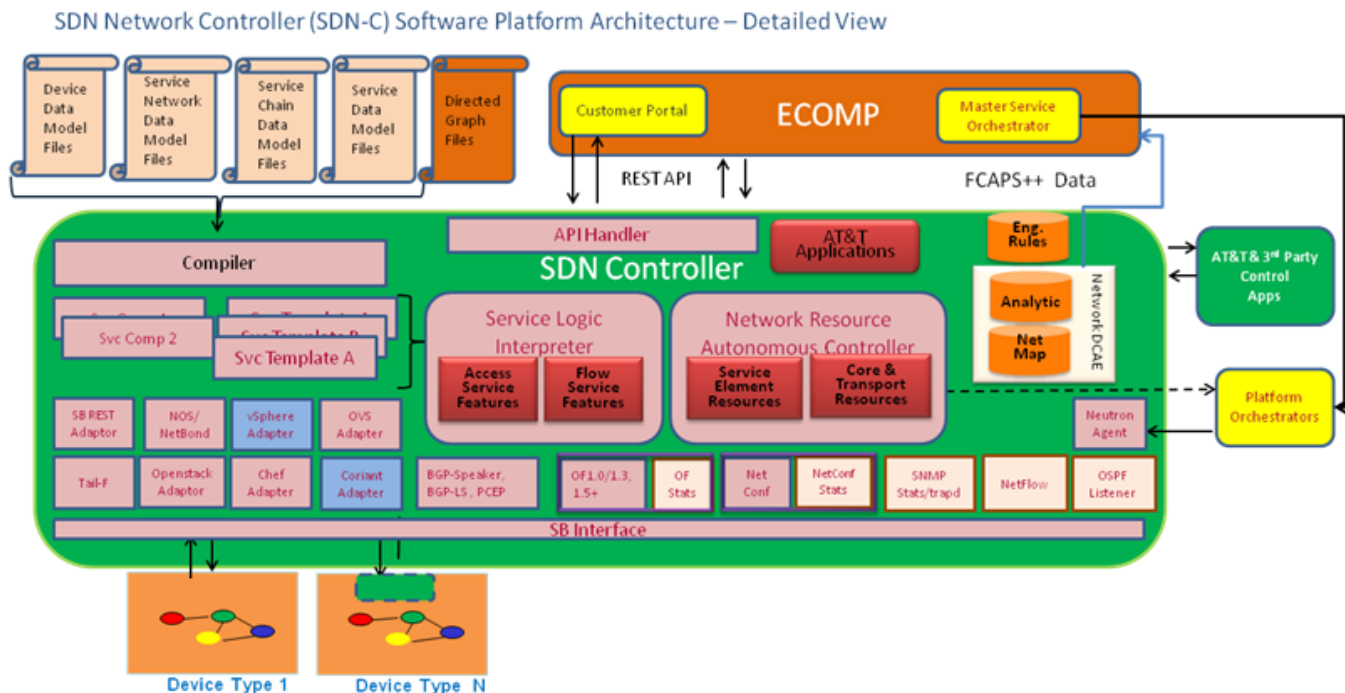
SDN-C depends on the following projects:

- Active and Available Inventory (A&AI)
- Common Controller SDK (CCSDK).
- Service Design and Creation (SDC)
- Data Movement as a Platform (DMaaP)
- Documentation
- Integration
- External API
- Modeling
- Multi VIM/Cloud
- Policy

Architecture

High level architecture diagram

The following diagram shows the high level architecture of SDNC:



The major architectural components of the SDN-C controller are:

- Device Data Models : Yang models that define interfaces to devices (virtual or physical) that the SDNC configures
- Service Network Data Models : Yang models that define data maintained within the SDNC about the network used by the set of services supported by this SDNC instance
- Service Chain Data Models : Yang models that define how services supported by an SDNC instance can be chained
- Service Data Models : Yang models that define data maintained within the SDNC for the set of services it supports
- Directed Graphs : programmable logic, updatable at run time with no restart, that define the behavior of the SDNC
- Service Logic Interpreter : module provided by CCSDK which allows platform to execute directed graphs
- API Handler : code (mostly generate from service Yang models) which implements RESTCONF API into SDNC. Most API handlers should follow the following pattern:
 - Call directed graph named after invoked RPC, passing RESTCONF RPC parameters as Java Properties object.
 - Return results from directed graph invocation as response to RESTCONF RPC
- Interface adaptors - code that allows directed graphs to invoke external interfaces

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	SDNC will perform 72 hour soak at component level. We assume that integration team will also do a 72 hour ONAP soak which includes SDNC.	<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	In Beijing, CCSDK will support a clustered OpenDaylight configuration in Kubernetes, as well as a clustered database, to allow for automated detection and recovery within a site. SDNC will use that configuration to meet this requirement. See SDN-C Clustering on Kubernetes for further details	<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	SDNC will improve test coverage to > 50%	<ul style="list-style-type: none"> 0 – none 1 – CII Passing badge + 50% Test Coverage + 50% test coverage 2 – CII Silver badge; internal communication encrypted; role-based access control and authorization for all calls 3 – CII Gold
Scalability	0	1	SDNC can be scaled either by adding additional OpenDaylight containers and/or database containers, or by deploying multiple instances of SDNC 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	SDNC will support ONAP standard logging.	<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	See readthedocs and wiki.	<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 release is expecting from other releases.

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)
A&AI : VNF	API used to read/write information about VNFs	Defined in seed code	Included in seed code	TBD
SDC : distribution	API used to distribute artifacts from SDC to subscribers	Defined in seed code	Included in seed code	TBD
DMaaP	API used to receive DHCP event notification	8/23/17	8/23/17	DMaaP API

API Outgoing Dependencies

API this release is delivering to other releases.

API Name	API Description	API Definition Date	API Delivery date	API Definition link (i.e.swagger)
Healthcheck	API used to verify that platform is available and healthy	Included in seed code	Delivered in seed code	TBD (requested Confluence OPEN API to be installed so this can be published on ONAP Wiki)
Generic VNF API	API used to request resources for VNFs. Will be deprecated in Beijing in favor of Generic Resource API.	Included in seed code	Delivered in seed code	TBD (requested Confluence OPEN API to be installed so this can be published on ONAP Wiki)
Generic Resource API	API used to request resources for VNFs. This API is a superset of the generic VNF API, which it replaces	Included in Amsterdam release	Delivered in Amsterdam	TBD (requested Confluence OPEN API to be installed so this can be published on ONAP Wiki)

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
OpenDaylight	OpenDaylight SDN controller platform	Nitrogen

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.

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
SDNC-469	Fix Cherrypy version		Sep 27, 2018	Sep 28, 2018		Unassigned	None	=	CLOSED	Won't Do
SDNC-435	SDNC uses the incorrect preloading data for vCPE vGW		Aug 28, 2018	Jan 03, 2019		Unassigned	None	=	CLOSED	Cannot Reproduce
SDNC-351	vnf-topology-operation-assign doesnt load		Jun 25, 2018	Jul 10, 2019		Unassigned	None	^	CLOSED	Done
SDNC-330	OOM: SDN-C should have clustering enabled by default		May 31, 2018	Jun 05, 2018		Unassigned	None	=	CLOSED	Done
SDNC-329	fix LCM DGs		May 29, 2018	Jun 05, 2018		Unassigned	None	=	CLOSED	Done
SDNC-327	SDNC failed to find vGW preload for vCPE		May 24, 2018	May 31, 2018		Unassigned	None	^	CLOSED	Done
SDNC-325	VF_MODEL fields like ecomp_generated_naming are null		May 21, 2018	Jun 07, 2018		Unassigned	None	^	CLOSED	Done
SDNC-323	vGW Preload not found		May 19, 2018	May 20, 2018		Dan Timoney	None	^	CLOSED	Done
SDNC-322	VF-Module-Assign object-path return & vnf-name xpath		May 19, 2018	May 31, 2018		Dan Timoney	None	^	CLOSED	Done
SDNC-313	LCM does not retrieve status from ansible-server correctly		May 13, 2018	May 17, 2018		Dan Timoney	Dan Timoney	=	CLOSED	Done
SDNC-311	sdnc/features build fails		May 11, 2018	May 17, 2018		Dan Timoney	Dan Timoney	=	CLOSED	Done
SDNC-310	General cleanup of SDNC helm charts		May 10, 2018	May 11, 2018		Unassigned	None	=	CLOSED	Done
SDNC-309	ansible adapter properties file missing		May 10, 2018	May 16, 2018		Dan Timoney	Dan Timoney	^	CLOSED	Done
SDNC-308	OOM: Update UEB and DMaaP charts to use new message-router name		May 09, 2018	May 11, 2018		Unassigned	None	=	CLOSED	Done
SDNC-306	SDNC fails health check on HEAT deployment		May 07, 2018	May 09, 2018		Dan Timoney	None	=	CLOSED	Done
SDNC-305	No model found for VF module customization UUID		May 07, 2018	Aug 12, 2018		Unassigned	None	=	CLOSED	Done
SDNC-303	aaiclient.properties using wrong JKS		May 04, 2018	May 17, 2018		Dan Timoney	Dan Timoney	=	CLOSED	Done
SDNC-302	SDNC VNF-API ssl certificate error updating AAI		May 04, 2018	May 11, 2018		Unassigned	None	^	CLOSED	Done
SDNC-299	OOM deployment with 2+ MySQL instances fails when NFS provisioner disabled		May 02, 2018	May 16, 2018		Unassigned	None	=	CLOSED	Done
SDNC-298	SDNC aaiclient.properties should reference v13 A&AI api		May 02, 2018	May 11, 2018		Unassigned	None	=	CLOSED	Done

Showing 20 out of 42 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
Functional requirements are incompletely defined	Functional requirements that are not sufficiently defined for sizing by 1/18 will not accepted into Beijing release	To fill out

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.

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](#).