

# Enhancement on PNF S/W Upgrade using Ansible

## Table of Contents

- [Table of Contents](#)
- [Scenario Scope In Frankfurt](#)
- [Procedure](#)
  - [Pre-conditions](#)
  - [Upgrade procedure](#)
  - [Sequence Diagram](#)
- [Impacts](#)
  - [API Impacts Details](#)
- [Test Status](#)
- [Preparation](#)
  - [Detailed Description of Integration Testcases and Results](#)

## Scenario Scope In Frankfurt

This scenario will **complete the E2E procedure of PNF in-place SW upgrade** case started from Casablanca (evolved in Dublin). The E2E procedure is VID SO Controller(SDNC/CCSDK)Ansible Server EMS PNF.

- Trigger by VID to upgrade specific PNF instance's software
- Enhance SO to support PNF S/W upgrade workflow execution
- Add SO BBs to do precheck/downloadNESw/activateNESw/UpdateAAI/postcheck using SDNC client
- Support LCM API for downloadNESw and activateNESw actions
- Provide ansible playbooks for downloadNESw and activateNESw
- Enhance EMS emulator for integration test

Since this scenario started from Casablanca, it is helpful to find more details completed in past releases with following link.

- In Casablanca: [5G - PNF Software Update](#)
- In Dublin: [5G - PNF SW Upgrade \(Casablanca carry-over items\)](#)

## Procedure

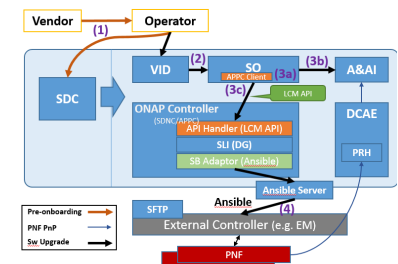
### Pre-conditions

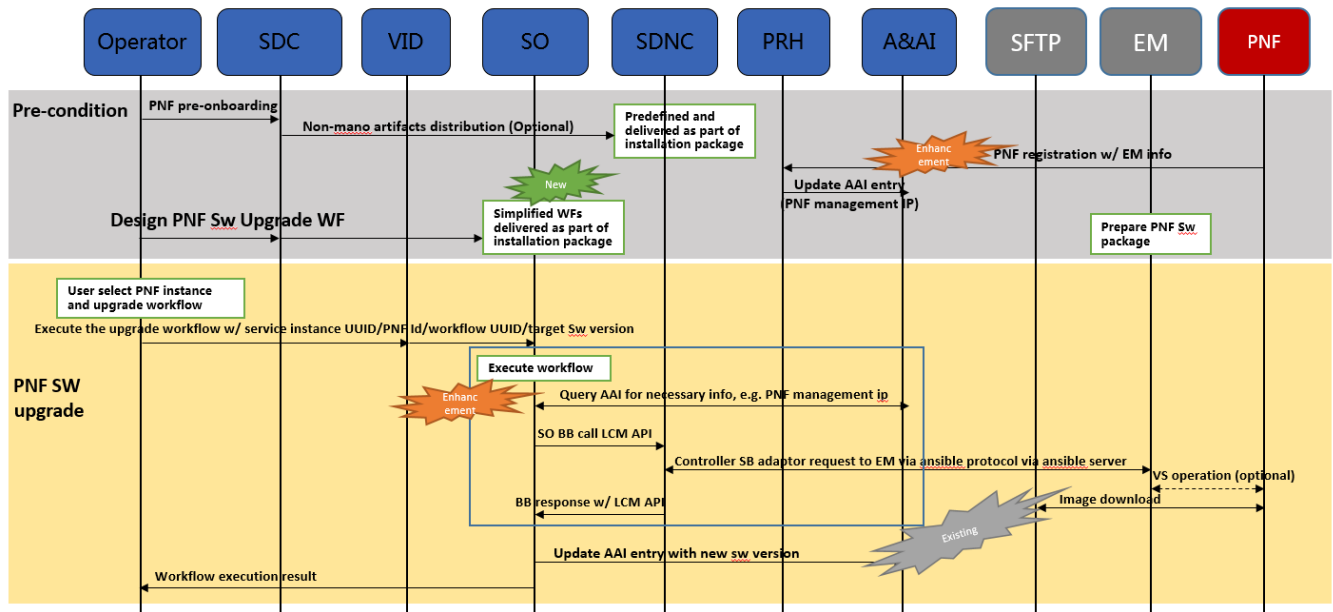
- ONAP is ready to use, while min components are VID/SO/SDNC/CCSDK.
- PNF PnP is completed, and EMS ipaddress is stored in AAI within PNF Registration step, as ipAddressV4Oam.
- EMS (Emulator) is running, and SFTP (can be with EMS) is ready.

### Upgrade procedure

- (1) In PNF pre-onboarding, operator delivers the PNF packages, including necessary ansible artifacts, to SDC. [playbook distribution is out of scope in this release]
- (2) In Sw Upgrade runtime, operator initiates the command, via VID or timer, to SO. (Before that, operator should design the upgrade workflow, or use the existing one).
- (3) SO executes the Sw upgrade task, like A&AI retrieval, and sends LCM requests to the controller.
- (4) Ansible Adaptor forwards requests to EMS via ansible server.

### Sequence Diagram





## Impacts

Project/Component	Story
SO	Support generic PNF workflow design
SO	Support upgrade sw version at PNF instance level
SO	Align with SO building block for PNF Upgrade procedure <ul style="list-style-type: none"> <li>1) Create new SO BB – downloadNESw</li> <li>2) Create new SO BB – activateNESw</li> <li>3) Create new SO BB – update AAI</li> <li>4) Update SO BB – preCheck and postCheck</li> </ul>
SO	Support different controller decision points in SO
SDNC/CCSDK	Support LCM API for downloadNESw and activateNESw actions
SDNC/CCSDK	Provide ansible playbooks for downloadNESw and activateNESw
VNFRQTS	PNF Software Upgrade with ansible with EM
Integration	<ul style="list-style-type: none"> <li>1) EMS simulator extension</li> <li>2) Integration document revision for PNF Sw Upgrade UC</li> </ul>

## API Impacts Details

### - LCM API of CCSDK:

LCM API Abbre.	HTTP Method	URI	Yang Model Section	Request Example	Response Example

UpgradePreCheck*	POST	/operations/ LCM: upgrade-pre-check	<pre> rpc upgrade-pre-check {     description "An operation to check that the VNF has the correct software version needed for a software upgrade.";     input {         uses common-header;         leaf action {             type             action;             mandatory true;         }         uses action-identifiers;         leaf payload {             type             payload;             mandatory true;         }         output {             uses common-header;             uses status;             leaf payload {                 type                 payload;             }             mandatory true;         }     } } </pre>	<pre> {   "input": {     "action": "UpgradePreCheck",     "action-identifiers": {       "pnf-name": "5gDU0001"     },     "common-header": {       "api-ver": "2.00",       "flags": {         "force": "FALSE",         "mode": "NORMAL",         "ttl": 65000       },       "originator-id": "MSO",       "request-id": "a1171d12-2ae9-496b-a2df-99784572914d",       "sub-request-id": "a1171d12-2ae9-496b-a2df-99784572914d",       "timestamp": "2019-03-15T09:07:58.255Z"     },     "payload": "{ \"ipaddress-v4-oam\": \"192.168.35.83\", \"playbook-name\": \"ansible_huawei_precheck\", \"oldSwVersion\": \"v1\", \"targetSwVersion\": \"v2\", \"ruleName\": \"r001\", \"additionalData\": \"{}\"}"   } } </pre>	<p>400: Success</p> <pre> {   "output": {     "common-header": {       "api-ver": "2.00",       "originator-id": "MSO",       "request-id": "a1171d12-2ae9-496b-a2df-99784572914d",       "sub-request-id": "a1171d12-2ae9-496b-a2df-99784572914d"     },     "payload": "{ \"result\": \"Success\"}",     "status": {       "code": 400,       "message": "Ansible Request f0f45105-1a51-4592-a31b-f178d5eb4638 finished with Result = success, Message = FINISHED"     }   } } </pre>
------------------	------	-------------------------------------	--	---	---

UpgradePostCheck"	/operations /LCM: upgrade -post-check	<pre> rpc upgrade-post-check {     description "An operation to check the VNF upgrade has been successful completed and all processes are running properly.";     input {         uses common-header;         leaf action {             type             action;         }         mandatory true;         uses action-identifiers;         leaf payload {             type             payload;         }         mandatory true;     }     output {         uses common-header;         uses status;         leaf payload {             type             payload;         }         mandatory true;     } } </pre>	<pre> {   "input": {     "action": "UpgradePostCheck",     "action-identifiers": {       "pnf-name": "5gDU0001"     },     "common-header": {       "api-ver": "2.00",       "flags": {         "force": "FALSE",         "mode": "NORMAL",         "ttl": 65000       },       "originator-id": "MSO",       "request-id": "a1171d12-2ae9-496b-a2df-99784572914d",       "sub-request-id": "a1171d12-2ae9-496b-a2df-99784572914d",       "timestamp": "2019-02-23T09:07:58.255Z"     },     "payload": "{\n\"ipaddress-v4-oam\": \"192.168.35.83\", \"playbook-name\": \"ansible_huawei_postcheck\", \"oldSwVersion\": \"v1\", \"targetSwVersion\": \"v2\", \"ruleName\": \"r102\", \"additionalData\": \"{}\"}"   } } </pre>	400: Success <pre> {   "output": {     "common-header": {       "api-ver": "2.00",       "originator-id": "MSO",       "request-id": "a1171d12-2ae9-496b-a2df-99784572914d",       "sub-request-id": "a1171d12-2ae9-496b-a2df-99784572914d"     },     "payload": "{\n\"result\": \"Success\"}",     "status": {       "code": 400,       "message": "Ansible Request c1c14dee-a80d-40b8-800a-9216a27c7d1c finished with Result = success, Message = FINISHED"     }   } } </pre>
-------------------	--	--	--	---

DownloadNESw	/operations /LCM: download-n-e-sw	<pre> rpc download-n-e-sw {     description "An operation to download NE software";     input {         uses common- header;         leaf action {             type action;  mandatory true;         }         uses action- identifiers;         leaf payload {             type payload;  mandatory true;         }     }     output {         uses common- header;         uses status;         leaf payload {             type payload;  mandatory true;         }     } } </pre>	<pre> {     "input": {         "action": "DownloadNESw",         "action-identifiers": {             "pnf-name": "5gDU0001"         },         "common-header": {             "api-ver": "2.00",             "flags": {                 "force": "FALSE",                 "mode": "NORMAL",                 "ttl": 65000             },             "originator-id": "MSO",             "request-id": "a1171d12-2ae9-496b-a2df-99784572914d",             "sub-request-id": "a1171d12-2ae9-496b-a2df-99784572914d",             "timestamp": "2019-10-19T10:20:16.125Z"         },         "payload": "{\n\"ipaddress-v4-oam\": \"192.168.35.83\", \"playbook-name\": \"ansible_huawei_downloadnesw\", \"swToBeDownloaded\": [{\n\"swLocation\": \"http://192.168.35.96:10080/ran_du_pkg1-v2.zip\", \"swFileSize\": 353, \"swFileCompression\": \"ZIP\", \"swFileFormat\": \"binary\"}]}\""}     } } </pre>	400: Success <pre> {     "output": {         "common- header": {             "api- ver": "2.00",  "originator-id": "MSO",  "request-id": "a1171d12-2ae9-496b-a2df-99784572914d",             "sub- request-id": "a1171d12-2ae9-496b-a2df-99784572914d"         },  "payload": "{\n\" result\": \" Success\"}",  "status": {  "code": 400,  "message": "Ansible Request 29cc5afd-082f-49be-bb52-3e96b178480c finished with Result = success, Message = FINISHED"         }     } } </pre>
--------------	---	--	---	--

ActivateNE Sw	/operations /LCM: activate-n-e-sw	<pre> rpc activate-n-e-sw {     description "An operation to activate NE software";     input {         uses common- header;         leaf action {             type action; mandatory true;         }         uses action- identifiers;         leaf payload {             type payload; mandatory true;         }     }     output {         uses common- header;         uses status;         leaf payload {             type payload; mandatory true;         }     } } </pre>	<pre> {   "input": {     "action": "ActivateNESw",     "action-identifiers": {       "pnf-name": "5gDU0001"     },     "common-header": {       "api-ver": "2.00",       "flags": {         "force": "FALSE",         "mode": "NORMAL",         "ttl": 65000       },       "originator-id": "MSO",       "request-id": "a1171d12-2ae9-496b-a2df-99784572914d",       "sub-request-id": "a1171d12-2ae9-496b-a2df-99784572914d",       "timestamp": "2019-10-19T10:25:28.116Z"     },     "payload": "{\n\"ipaddress-v4-oam\": \"192.168.35.83\", \"playbook-name\": \"ansible_huawei_activatenesw\", \"swVersionToBeActivated\": \"v2\""   } } </pre>	400: Success <pre> {   "output": {     "common- header": {       "api- ver": "2.00",       "originator-id": "MSO",       "request-id": "a1171d12-2ae9-496b-a2df-99784572914d",       "sub- request-id": "a1171d12-2ae9-496b-a2df-99784572914d"     },     "payload": "{\n\" result\": \" Success\""   },   "status": {     "code": 400,     "message": "Ansible Request 0af55193-8177-41a7-b927-78646be6250b finished with Result = success, Message = FINISHED"   } } </pre>
---------------	-----------------------------------	---	---	---

\*: These LCM APIs are enhanced from R4/R5, and update the *vnf-id* field in *action-identifiers* to *pnf-name* in the request parameter.

The *playbook-name* in the payload of request is optional. If not set the *playbook-name* in the request, it will use the value in the config file */opt/onap/sdnc/data/properties/lcm-dg.properties* of SDNC.

## JIRA Status Tracking

Key	Summary	T	Created	Updated	Due	Assignee	Reporter	P	Status	Resolution
VNFRQ TS-698	PNF Software Upgrade with ansible with EM	✓	Aug 20, 2019	Apr 27, 2020		Unassigned	None	==	CLOSED	Won't Do
SO-2589	Support PNF software upgrade workflow using SDNC LCM API	🟢	Jan 08, 2020	Mar 06, 2020		Unassigned	None	==	CLOSED	Done
SO-2588	Support LCM API of SDNC to support PNF SW Upgrade	🟢	Jan 08, 2020	Mar 02, 2020		Unassigned	None	==	CLOSED	Done

SDNC-857	Provide DGs and Ansible playbooks for downloadNESw and activateNESw		Aug 20, 2019	Jan 10, 2020	Unassigned	None		<button>CLOSED</button>	Done
SDNC-856	Support LCM API for downloadNESw and activateNESw actions		Aug 20, 2019	Jan 10, 2020	Unassigned	None		<button>CLOSED</button>	Done
INT-1209	Integration document revision for PNF Sw Upgrade UC		Aug 20, 2019	Apr 08, 2020	Unassigned	None		<button>CLOSED</button>	Done
INT-1208	EMS simulator extension for PNF SW Upgrade		Aug 20, 2019	Mar 05, 2020	Unassigned	None		<button>CLOSED</button>	Done

7 issues

Head Epic :

[INT-1207](#) - Getting issue details... STATUS

## Test Status

No	Test Case	Test Status
1	UpgradePreCheck for PNF instance from SDNC to EM using LCM API	<button>COMPLETE</button>
2	DownloadNESw for PNF instance from SDNC to EM using LCM API	<button>COMPLETE</button>
3	ActivateNESw for PNF instance from SDNC to EM using LCM API	<button>COMPLETE</button>
4	UpgradePostCheck for PNF instance from SDNC to EM using LCM API	<button>COMPLETE</button>
5	Test PNF SW upgrade Workflow for PNF instance from SO to SDNC (using LCM API) to EM	<button>COMPLETE</button>

## Preparation

1. Add topics **SDNC-LCM-READ** and **SDNC-LCM-WRITE** to **DMaaP** if they don't exist.

```
curl -s -H "Content-Type: application/json" -X POST http://message-router.onap:30227/topics/create -d '{
  "topicName": "SDNC-LCM-READ",
  "partitionCount": "1",
  "replicationCount": "1"
}'

curl -s -H "Content-Type: application/json" -X POST http://message-router.onap:30227/topics/create -d '{
  "topicName": "SDNC-LCM-WRITE",
  "partitionCount": "1",
  "replicationCount": "1"
}'
```

2. Add the following line in the file in **/opt/ansible-server/Playbooks/Ansible\_inventory** in **ansible-server** container:

```
192.168.35.83 ansible_connection=ssh ansible_port=60032 ansible_user=admin ansible_ssh_private_key_file=/home/ansible/.ssh/ems.key
```

Where *ems.key* is the private key of SSH user *admin* at EMS Simulator.

### 3. Initial PNF 5gDU0001 entry in AAI:

```
{
  "pnf-name": "5gDU0001",
  "pnf-id": "5gDU0001",
  "ipaddress-v4-oam": "192.168.35.83",
  "sw-version": "v1",
  "in-maint": false,
  "resource-version": "1584873329418"
}
```

Where Sw Version of PNF *5gDU0001* is "v1".

### 4. Add some entries to SO catalogdb:

```
use catalogdb;

insert into pnf_resource set MODEL_UUID='f50f34be-9aa7-4a03-aa12-fd29e5f27538', MODEL_VERSION='1.0';

insert into pnf_resource_customization set MODEL_CUSTOMIZATION_UUID='c57939f8-ee46-4700-aa3a-60374507d94f',
MODEL_INSTANCE_NAME='TestPnf', PNF_RESOURCE_MODEL_UUID='f50f34be-9aa7-4a03-aa12-fd29e5f27538',
CONTROLLER_ACTOR='sdnc';

insert into pnf_resource_customization_to_service set SERVICE_MODEL_UUID='296d49aa-236a-4920-a395-5620df09e765',
RESOURCE_MODEL_CUSTOMIZATION_UUID='c57939f8-ee46-4700-aa3a-60374507d94f';

insert into workflow (ARTIFACT_UUID, ARTIFACT_NAME, NAME, VERSION, DESCRIPTION, RESOURCE_TARGET, SOURCE) values
('245a0bca-3816-45c6-9270-1de25d27fc56', 'PNFSoftwareUpgrade.bpmn', 'PNFSoftwareUpgrade', '1.0', 'PNF Software Upgrade Test', 'pnf',
'native');
```

## Detailed Description of Integration Testcases and Results

Test case ID	1
Name	UpgradePreCheck for PNF instance from SDNC to EM using LCM API
Description	UpgradePreCheck for PNF instance from SDNC to EM using LCM API
Release	Frankfurt
Pre-conditions	



Testing Steps	<div>Test Command</div> <pre>curl -s -u "admin:Kp8bJ4SXszM0WXlhak3eHlcse2gAw84vaoGGmJvUy2U" -H "Content-Type: application/json" -H "Accept: application/json" -X POST http://sdnc.onap:30202/restconf/operations/LCM:upgrade-pre-check -d @upgrade-pre-check-input.json   python -m json.tool</pre>
	<div>Contents of upgrade-pre-check-input.json</div> <pre>{   "input": {     "action": "UpgradePreCheck",     "action-identifiers": {       "pnf-name": "5gDU0001"     },     "common-header": {       "api-ver": "2.00",       "flags": {         "force": "FALSE",         "mode": "NORMAL",         "ttl": 65000       },       "originator-id": "MSO",       "request-id": "6ca0df6f-85d5-40d0-b577-58fc6963126e",       "sub-request-id": "3013c12f-46d6-4f42-86cc-6658f8e62efd",       "timestamp": "2020-03-10T08:16:29.255Z"     },     "payload": "{ \"ipaddress-v4-oam\": \"192.168.35.83\", \"playbook-name\": \"ansible_huawei_precheck\", \"oldSwVersion\": \"v1\", \"targetSwVersion\": \"v2\", \"ruleName\": \"r101\", \"additionalData\": \"{}\" }"   } }</pre>
	<div>Response of SDNC</div> <pre>{   "output": {     "payload": "{ \"result\": \"Success\" }",     "common-header": {       "api-ver": "2.00",       "originator-id": "MSO",       "sub-request-id": "3013c12f-46d6-4f42-86cc-6658f8e62efd",       "request-id": "6ca0df6f-85d5-40d0-b577-58fc6963126e"     },     "status": {       "code": 400,       "message": "Ansible Request 16df336e-2e0c-4908-bb85-b057b73e9ab1 finished with Result = success, Message = FINISHED"     }   } }</pre>
	<div>Conclusion (Pass/Fail)</div> <div>Pass</div>
Test Lab	

Test case ID	2
Name	DownloadNESw for PNF instance from SDNC to EM using LCM API
Description	DownloadNESw for PNF instance from SDNC to EM using LCM API
Release	Frankfurt
Pre-conditions	

Testing Steps	<div>Test Command</div> <pre>curl -s -u "admin:Kp8bJ4SXszM0WXlhak3eHlcse2gAw84vaoGGmJvUy2U" -H "Content-Type: application/json" -H "Accept: application/json" -X POST http://sdnc.onap:30202/restconf/operations/LCM:download-n-e-sw -d @download-n-e-sw-input.json   python -m json.tool</pre>
	<div>Contents of download-n-e-sw-input.json</div> <pre>{   "input": {     "action": "DownloadNESw",     "action-identifiers": {       "pnf-name": "5gDU0001"     },     "common-header": {       "api-ver": "2.00",       "flags": {         "force": "FALSE",         "mode": "NORMAL",         "ttl": 65000       },       "originator-id": "MSO",       "request-id": "6ca0df6f-85d5-40d0-b577-58fc6963126e",       "sub-request-id": "3013c12f-46d6-4f42-86cc-6658f8e62efd",       "timestamp": "2020-03-10T08:16:29.255Z"     },     "payload": "{\n\"ipaddress-v4-oam\": \"192.168.35.83\\\", \"playbook-name\": \"ansible_huawei_downloadnesw\\\", \"swToBeDownloaded\": [{\n\"swLocation\": \"http://192.168.35.96:10080/ran_du_pkg1-v2.zip\\\", \"swFileSize\": 353, \"swFileCompression\": \"ZIP\\\", \"swFileFormat\": \"binary\\\"}]]\"",   },   "swToBeDownloaded": [{\n\"swLocation\": \"http://192.168.35.96:10080/ran_du_pkg1-v2.zip\\\", \"swFileSize\": 353, \"swFileCompression\": \"ZIP\\\", \"swFileFormat\": \"binary\\\"}]]"   } }</pre>
	<div>Response of SDNC</div> <pre>{   "output": {     "payload": "{\n\"result\": \"Success\\\"}",     "common-header": {       "api-ver": "2.00",       "originator-id": "MSO",       "sub-request-id": "3013c12f-46d6-4f42-86cc-6658f8e62efd",       "request-id": "6ca0df6f-85d5-40d0-b577-58fc6963126e"     },     "status": {       "code": 400,       "message": "Ansible Request bfcaa172-0f5b-46db-a973-6adfe79b07e9 finished with Result = success, Message = FINISHED"     }   } }</pre>
Conclusion (Pass/Fail)	Pass
Test Lab	

Test case ID	3
Name	ActivateNESw for PNF instance from SDNC to EM using LCM API
Description	ActivateNESw for PNF instance from SDNC to EM using LCM API
Release	Frankfurt
Pre-conditions	

Testing Steps	<div>Test Command</div> <pre>curl -s -u "admin:Kp8bJ4SXszM0WXlhak3eHlcse2gAw84vaoGGmJvUy2U" -H "Content-Type: application/json" -H "Accept: application/json" -X POST http://sdnc.onap:30202/restconf/operations/LCM:activate-n-e-sw -d @activate-n-e-sw-input.json   python -m json.tool</pre>
	<div>Contents of activate-n-e-sw-input.json</div> <pre>{   "input": {     "action": "ActivateNESw",     "action-identifiers": {       "pnf-name": "5gDU0001"     },     "common-header": {       "api-ver": "2.00",       "flags": {         "force": "FALSE",         "mode": "NORMAL",         "ttl": 65000       },       "originator-id": "MSO",       "request-id": "6ca0df6f-85d5-40d0-b577-58fc6963126e",       "sub-request-id": "3013c12f-46d6-4f42-86cc-6658f8e62efd",       "timestamp": "2020-03-10T08:16:29.255Z"     },     "payload": "{\n\"ipaddress-v4-oam\": \"192.168.35.83\", \"playbook-name\": \"ansible_huawei_activatenesw\", \"swVersionToBeActivated\": \"v2\"}"   } }</pre>
	<div>Response of SDNC</div> <pre>{   "output": {     "payload": "{\n\"result\": \"Success\"}",     "common-header": {       "api-ver": "2.00",       "originator-id": "MSO",       "sub-request-id": "3013c12f-46d6-4f42-86cc-6658f8e62efd",       "request-id": "6ca0df6f-85d5-40d0-b577-58fc6963126e"     },     "status": {       "code": 400,       "message": "Ansible Request 26592778-8ff5-405e-9aa3-f180a0b5e05a finished with Result = success, Message = FINISHED"     }   } }</pre>
Conclusion (Pass/Fail)	Pass
Test Lab	

Test case ID	4
Name	UpgradePostCheck for PNF instance from SDNC to EM using LCM API
Description	UpgradePostCheck for PNF instance from SDNC to EM using LCM API
Release	Frankfurt
Pre-conditions	

Testing Steps	<div>Test Command</div> <pre>curl -s -u "admin:Kp8bJ4SXszM0WXlhak3eHlcse2gAw84vaoGGmJvUy2U" -H "Content-Type: application/json" -H "Accept: application/json" -X POST http://sdnc.onap:30202/restconf/operations/LCM:upgrade-post-check -d @upgrade-post-check-input.json   python -m json.tool</pre>
	<div>Contents of upgrade-post-check-input.json</div> <pre>{   "input": {     "action": "UpgradePostCheck",     "action-identifiers": {       "pnf-name": "5gDU0001"     },     "common-header": {       "api-ver": "2.00",       "flags": {         "force": "FALSE",         "mode": "NORMAL",         "ttl": 65000       },       "originator-id": "MSO",       "request-id": "6ca0df6f-85d5-40d0-b577-58fc6963126e",       "sub-request-id": "3013c12f-46d6-4f42-86cc-6658f8e62efd",       "timestamp": "2020-03-10T08:16:29.255Z"     },     "payload": "{\n\"ipaddress-v4-oam\": \"192.168.35.83\", \"playbook-name\": \"ansible_huawei_postcheck\", \"oldSwVersion\": \"v1\", \"targetSwVersion\": \"v2\", \"ruleName\": \"r102\", \"additionalData\": \"{}\"}"   } }</pre>
	<div>Response of SDNC</div> <pre>{   "output": {     "payload": "{\n\"result\": \"Success\"}",     "common-header": {       "api-ver": "2.00",       "originator-id": "MSO",       "sub-request-id": "3013c12f-46d6-4f42-86cc-6658f8e62efd",       "request-id": "6ca0df6f-85d5-40d0-b577-58fc6963126e"     },     "status": {       "code": 400,       "message": "Ansible Request 4e3ef116-5e0f-4681-8b3b-b55a1a1b83cb finished with Result = success, Message = FINISHED"     }   } }</pre>
Conclusion (Pass/Fail)	Pass
Test Lab	

Test case ID	5
Name	Test PNF SW upgrade Workflow for PNF instance from SO to SDNC (using LCM API) to EM
Description	Test PNF SW upgrade Workflow for PNF instance from SO to SDNC (using LCM API) to EM
Release	Frankfurt
Pre-conditions	
Testing	NOTE: Befor testing this case, reset the EMS Simulator to initial status.

**Test Command**

```
curl -s -u "InfraPortalClient:password1$" -H "Content-Type: application/json" -H "Accept: application/json" -H "X-ONAP-RequestID: 2cd4a080-88c0-4de3-9255-1560468f5df9" -H "X-ONAP-PartnerName: SO-REST" -H "X-RequestorID: so-rest" -X POST http://so.onap:30277/onap/so/infra/instanceManagement/v1/serviceInstances/25c651cb-4ddb-4755-9aea-42e36ea34442/pnfs/5gDU0001/workflows/245a0bca-3816-45c6-9270-1de25d27fc56 -d @PNFSoftwareUpgrade-sdnc.json | python -m json.tool
```

**Contents of PNFSoftwareUpgrade-sdnc.json**

```
{
  "requestDetails": {
    "modelInfo": {
      "modelUuid": "296d49aa-236a-4920-a395-5620df09e765"
    },
    "requestInfo": {
      "instanceName": "TestPnfSwUp01"
    },
    "requestParameters": {
      "userParams": [
        {
          "name": "pnfName",
          "value": "5gDU0001"
        },
        {
          "name": "targetSoftwareVersion",
          "value": "v2"
        }
      ]
    },
    "payload": "{\"ipaddressV4Oam\": \"192.168.35.83\", \"oldSwVersion\": \"v1\", \"preCheckRuleName\": \"r101\", \"preCheckAdditionalData\": \"{}\", \"preCheckPlaybook\": \"ansible_huawei_precheck\", \"swToBeDownloaded\": [{\"swLocation\": \"http://192.168.35.96:10080/ran_du_pkg1-v2.zip\", \"swFileSize\": 353, \"swFileCompression\": \"ZIP\", \"swFileFormat\": \"binary\"}], \"downloadNESwPlaybook\": \"ansible_huawei_downloadnesw\", \"activateNESwPlaybook\": \"ansible_huawei_activatenesw\", \"postCheckRuleName\": \"r102\", \"postCheckAdditionalData\": \"{}\", \"postCheckPlaybook\": \"ansible_huawei_postcheck\"}"
  }
}
```

**Response of SO**

```
{
  "requestReferences": {
    "requestId": "2cd4a080-88c0-4de3-9255-1560468f5df9",
    "instanceId": "5gDU0001",
    "requestSelfLink": "http://so.onap:30277/orchestrationRequests/v1/2cd4a080-88c0-4de3-9255-1560468f5df9"
  }
}
```

Retrieve the status of the workflow:

**Test Command**

```
curl -s -u "InfraPortalClient:password1$" -H "Accept: application/json" -X GET http://so.onap:30277/onap/so/infra/orchestrationRequests/v7/2cd4a080-88c0-4de3-9255-1560468f5df9 | python -m json.tool
```

### Status of the workflow

```
{
  "request": {
    "requestId": "2cd4a080-88c0-4de3-9255-1560468f5df9",
    "startTime": "Sun, 22 Mar 2020 12:42:05 GMT",
    "requestScope": "pnf",
    "requestType": "forCustomWorkflow",
    "requestDetails": {
      "modelInfo": {
        "modelId": "296d49aa-236a-4920-a395-5620df09e765",
        "modelUuid": "296d49aa-236a-4920-a395-5620df09e765"
      },
      "requestInfo": {
        "source": null,
        "instanceName": "TestPnfSwUp01",
        "suppressRollback": false
      },
      "requestParameters": {
        "userParams": [
          {
            "name": "pnfName",
            "value": "5gDU0001"
          },
          {
            "name": "targetSoftwareVersion",
            "value": "v2"
          }
        ]
      },
      "payload": "{\n  \"ipaddressV4Oam\": \"192.168.35.83\", \"oldSwVersion\": \"v1\", \"preCheckRuleName\": \"r101\", \"preCheckAdditionalData\": \"\", \"preCheckPlaybook\": \"ansible_huawei_precheck\", \"swToBeDownloaded\": [{\n    \"swLocation\": \"http://192.168.35.96:10080/ran_du_pkg1-v2.zip\", \"swFileSize\": 353, \"swFileCompression\": \"ZIP\", \"swFileFormat\": \"binary\"}], \"downloadNESwPlaybook\": \"ansible_huawei_downloadnesw\", \"activateNESwPlaybook\": \"ansible_huawei_activatenesw\", \"postCheckRuleName\": \"r102\", \"postCheckAdditionalData\": \"\", \"postCheckPlaybook\": \"ansible_huawei_postcheck\"}",
      "instanceReferences": {
        "serviceInstanceId": "25c651cb-4ddb-4755-9aea-42e36ea34442"
      },
      "requestStatus": {
        "requestState": "IN_PROGRESS",
        "timestamp": "Sun, 22 Mar 2020 12:42:05 GMT"
      }
    }
  }
}
```

While the **requestStatus** is "IN\_PROGRESS", then wait a minute to retrieve the status of workflow again and again, until the **requestStatus** is "COMPLETED".

#### Status of the workflow

```
{
  "request": {
    "requestId": "2cd4a080-88c0-4de3-9255-1560468f5df9",
    "startTime": "Sun, 22 Mar 2020 12:42:05 GMT",
    "requestScope": "pnf",
    "requestType": "forCustomWorkflow",
    "requestDetails": {
      "modelInfo": {
        "modelId": "296d49aa-236a-4920-a395-5620df09e765",
        "modelUuid": "296d49aa-236a-4920-a395-5620df09e765"
      },
      "requestInfo": {
        "source": null,
        "instanceName": "TestPnfSwUp01",
        "suppressRollback": false
      },
      "requestParameters": {
        "userParams": [
          {
            "name": "pnfName",
            "value": "5gDU0001"
          },
          {
            "name": "targetSoftwareVersion",
            "value": "v2"
          }
        ]
      },
      "payload": "{\n  \"ipaddressV4Oam\": \"192.168.35.83\", \"oldSwVersion\": \"v1\", \"preCheckRuleName\": \"r101\", \"preCheckAdditionalData\": \"\", \"preCheckPlaybook\": \"ansible_huawei_precheck\", \"swToBeDownloaded\": [{\n    \"swLocation\": \"http://192.168.35.96:10080/ran_du_pkg1-v2.zip\", \"swFileSize\": 353, \"swFileCompression\": \"ZIP\", \"swFileFormat\": \"binary\"}], \"downloadNESwPlaybook\": \"ansible_huawei_downloadnesw\", \"activateNESwPlaybook\": \"ansible_huawei_activatenesw\", \"postCheckRuleName\": \"r102\", \"postCheckAdditionalData\": \"\", \"postCheckPlaybook\": \"ansible_huawei_postcheck\"}",
      "instanceReferences": {
        "serviceInstanceId": "25c651cb-4ddb-4755-9aea-42e36ea34442"
      },
      "requestStatus": {
        "requestState": "COMPLETED",
        "statusMessage": "STATUS: PNF has been upgraded successfully.",
        "percentProgress": 100,
        "timestamp": "Sun, 22 Mar 2020 12:43:06 GMT"
      }
    }
  }
}
```

While the **requestStatus** is "COMPLETED", then retrieve the Sw Version of PNF 5gDU0001 in AAI:

#### Command

```
curl -s -k -u "AAI:AAI" -H "X-FromAppId: aairest" -H "X-TransactionId: 1001" -H "Content-Type: application/json" -H "Accept: application/json" -X GET https://aai.onap:30233/aai/v19/network/pnfs/pnf/5gDU0001 | python -m json.tool
```

#### Response of retrieving AAI

```
{
  "pnf-name": "5gDU0001",
  "pnf-id": "5gDU0001",
  "ipaddress-v4-oam": "192.168.35.83",
  "sw-version": "v2",
  "in-maint": false,
  "resource-version": "1584880971237"
}
```

The Sw Version of PNF *5gDU0001* in AAI is upgraded to "v2".

Conclusion  
(Pass  
/Fail)

Pass

Test  
Lab

