LF NETWORKING

LFN Developer & Testing Forum

ONAP for PNFs management

PNF upgrade automation

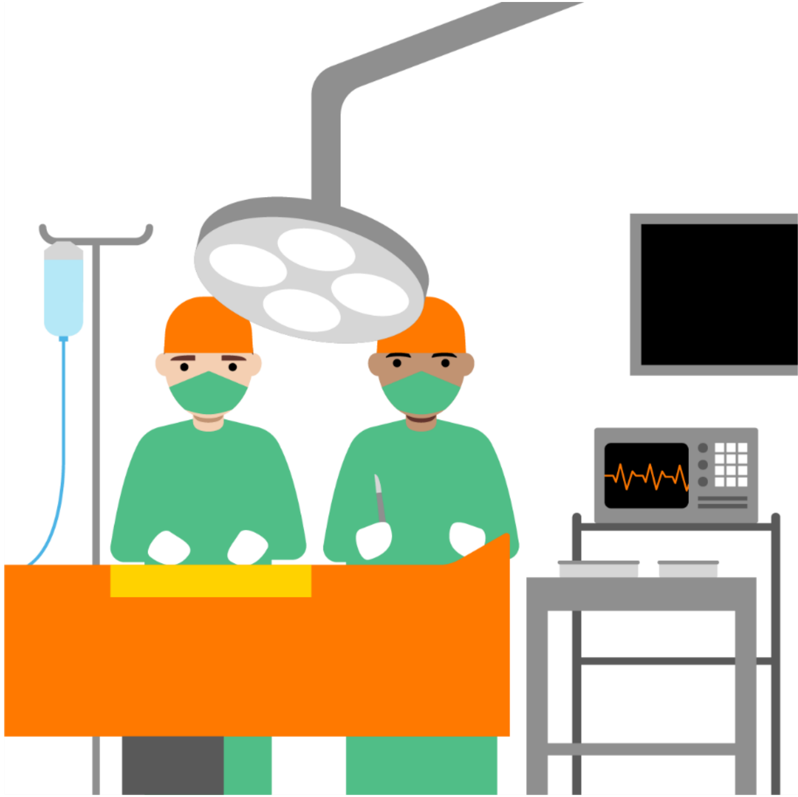
Michał Grzesik *<Michal.Grzesik@orange.com>*
Łukasz Rajewski *<Lukasz.Rajewski@orange.com>*
Orange Labs Poland

Agenda

- PNF upgrade process
- ONAP support for PNFs
- CDS for PNF upgrade operation
- Upgrade automation - Ansible & AWX
- Demo
- Summary and plans

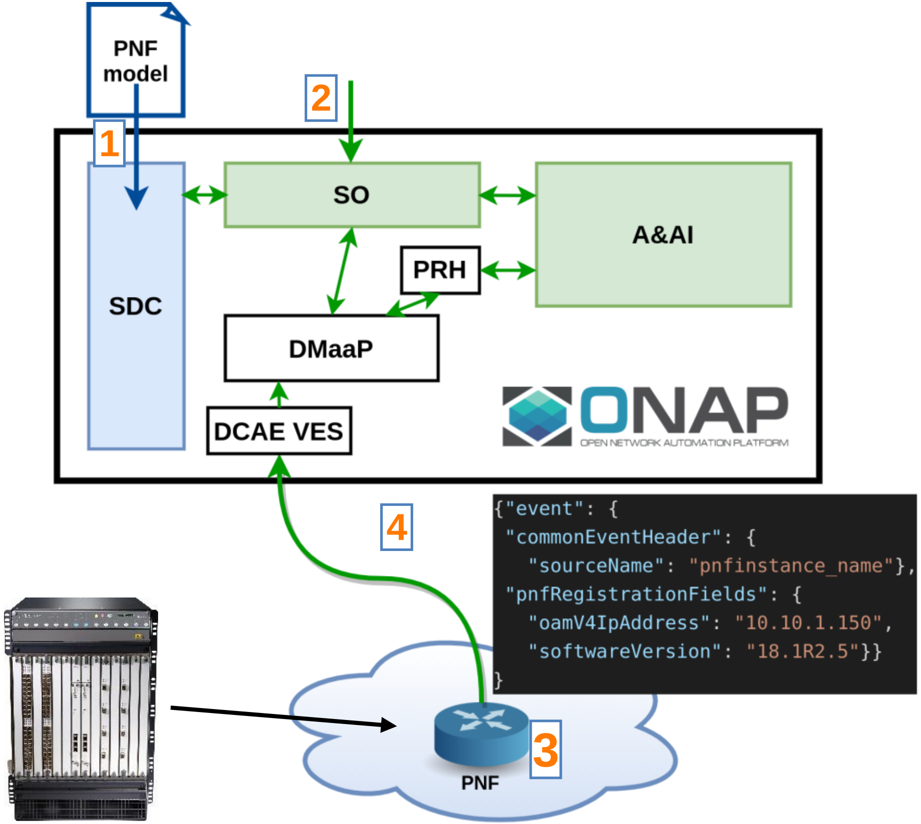
PNF Upgrade process

- Time consuming
- Limited maintenance slots
- Long procedure
  and many manual steps
- Require High Skill Expert during upgrade
- Error prone – proper actions' order required



ONAP PNF support

1. PNF Modeling
   - Onboard of TOSCA PNF Descriptor Package(s)
   - Design of the Service with PNF Included
2. Instantiation of PNF Service
   - Creation of PNF Entry in the Inventory
   - Subscription to PNF in ONAP
3. PNF Boot-strapping
4. PNF Registration
   - PNF Registration event sent to ONAP
   - PNF Information update in the Inventory



```
{"event": {
 "commonEventHeader": {
   "sourceName": "pnfinstance_name"},
 "pnfRegistrationFields": {
   "oamV4IpAddress": "10.10.1.150",
   "softwareVersion": "18.1R2.5"}}
}
```

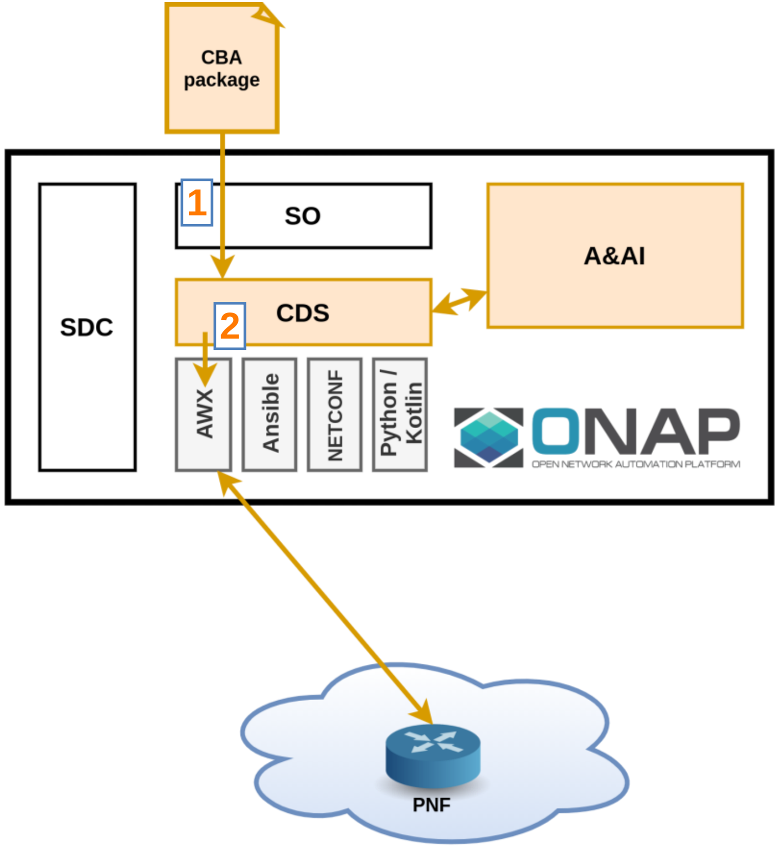
PNF Configuration

CDS (Controller Design Studio)

*key ONAP component to automate any **config** provisioning operation, such as day0, day1, or day2 configuration*

1. **Design part**
   - Definitions of actions for service

     CBA package (TOSCA format)
2. **Run part**
   - Execute defined actions



CDS Blueprint

```
"workflows": {
   "upgrade-pnf": {
      "steps": {
         "retrieve-oam-ip-address": {
            "target": "retrieve-oam-ip-address",
            "activities": [
               { "call_operation": "" }
            ],
            "on_success": [
               "execute-upgrade"
            ]
         },
         "execute-upgrade": {
            "target": "execute-remote-awx-ansible",
            "activities": [
               { "call_operation": "" }
            ],
            "on_success": [
               "update-pnf-sw-version"
            ]
         },
         "update-pnf-sw-version": {
            "target": "update-pnf-sw-version",
            "activities": [
               { "call_operation": "" }
            ]
         }
      },
```

```
"inputs": {
   "endpoint-selector": {
      "required": false,
      "default": {
         "type": "token-auth",
         "url": "http://10.10.4.XXX",
         "token": "Bearer DhXKdA57CbpDxcgXXXXXXXXXXXXXX"
      },
      "type": "json",
      "description": "awx authentication data"
   },
   "workflow-job-template-id": {
      "required„ : false,
      "default": "41",
      "type": "string",
      "description": "awx workflow template id"
   },
   "pnf-instance-name": {
      "required": true,
      "type": "string",
      "description": "the name of pnf instance"
   },
   "os_filename": {
      "required": true,
      "type": "string",
      "description": "os software filename"
   }
},
```

CDS Blueprint

```
"workflows": {
   "upgrade-pnf": {
      "steps": {
         "retrieve-oam-ip-address": {
            "target": "retrieve-oam-ip-address",
            "activities": [
               { "call_operation": "" }
            ],
            "on_success": [
               "execute-upgrade"
            ]
         },
         "execute-upgrade": {
            "target": "execute-remote-awx-ansible",
            "activities": [
               { "call_operation": "" }
            ],
            "on_success": [
               "update-pnf-sw-version"
            ]
         },
         "update-pnf-sw-version": {
            "target": "update-pnf-sw-version",
            "activities": [
               { "call_operation": "" }
            ]
         }
      },
```

```
"node_templates": {
   "retrieve-oam-ip-address": {
      "type": "component-resource-resolution",
      "interfaces": {
         "ResourceResolutionComponent": {
            "operations": {
               "process": {
                  "inputs": {
                     "artifact-prefix-names": [
                        "pnf-oam-ip"
                     ]
                  }
               }
            }
         }
      },
      "artifacts": {
         "pnf-oam-ip-template": {
            "type": "artifact-template-velocity",
            "file": "Templates/pnf-oam-ip-template.vtl"
         },
         "pnf-oam-ip-mapping": {
            "type": "artifact-mapping-resource",
            "file": "Templates/pnf-oam-ip-mapping.json"
         }
      }
   }
```

CDS Blueprint

```
"workflows": {
   "upgrade-pnf": {
      "steps": {
         "retrieve-oam-ip-address": {
            "target": "retrieve-oam-ip-address",
            "activities": [
               { "call_operation": "" }
            ],
            "on_success": [
               "execute-upgrade"
            ]
         },
         "execute-upgrade": {
            "target": "execute-remote-awx-ansible",
            "activities": [
               { "call_operation": "" }
            ],
            "on_success": [
               "update-pnf-sw-version"
            ]
         },
         "update-pnf-sw-version": {
            "target": "update-pnf-sw-version",
            "activities": [
               { "call_operation": "" }
            ]
         }
      },
```

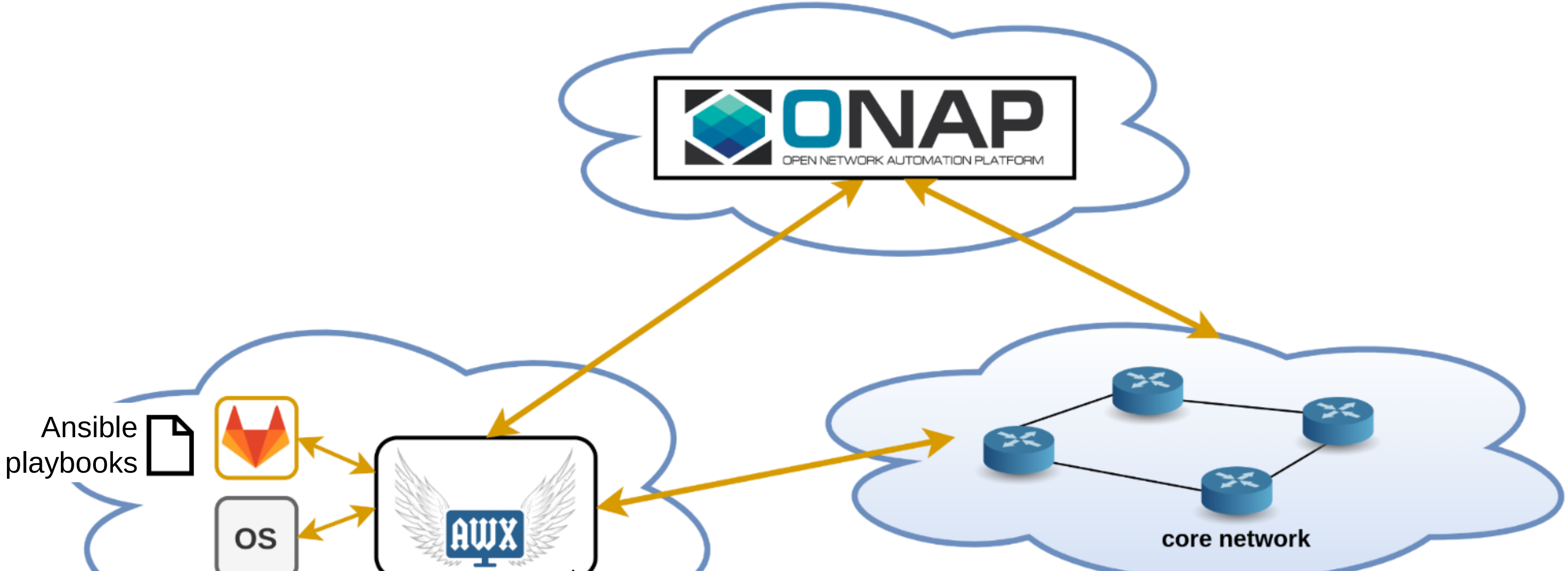
```
"node_templates": {
   "execute-remote-awx-ansible": {
      "type": "component-remote-ansible-executor",
      "interfaces": {
         "ComponentRemoteAnsibleExecutor": {
            "operations": {
               "process": {
                  "inputs": {
                     "endpoint-selector": {
                        "get_input": "endpoint-selector"
                     },
                     "workflow-job-template-id": {
                        "get_input": "workflow-job-template-id"
                     },
                     "extra-vars": "*extra-vars"
                  }
               }
            }
         }
      }
   }
```

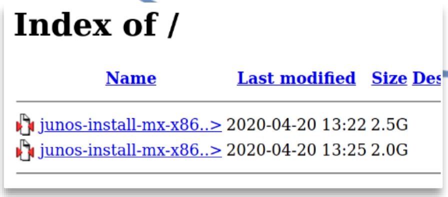
CDS Blueprint

```
"workflows": {
   "upgrade-pnf": {
      "steps": {
         "retrieve-oam-ip-address": {
            "target": "retrieve-oam-ip-address",
            "activities": [
               { "call_operation": "" }
            ],
            "on_success": [
               "execute-upgrade"
            ]
         },
         "execute-upgrade": {
            "target": "execute-remote-awx-ansible",
            "activities": [
               { "call_operation": "" }
            ],
            "on_success": [
               "update-pnf-sw-version"
            ]
         },
         "update-pnf-sw-version": {
            "target": "update-pnf-sw-version",
            "activities": [
               { "call_operation": "" }
            ]
         }
      },
```

```
"node_templates": {
   "update-pnf-sw-version": {
      "type": "component-remote-python-executor",
      "interfaces": {
         "ComponentRemotePythonExecutor": {
            "operations": {
               "process": {
                  "implementation": {
                     "primary": "component-script"
                  },
                  "inputs": {
                     "command": "python updatePnf.py",
                     "packages": [
                        {
                           "type": "pip",
                           "package": [
                              "requests"
                           ]
                        }
                     ],
                     "argument-properties":
                         "*update-pnf-sw-version"
                  }
               }
            }
         }
      }
```

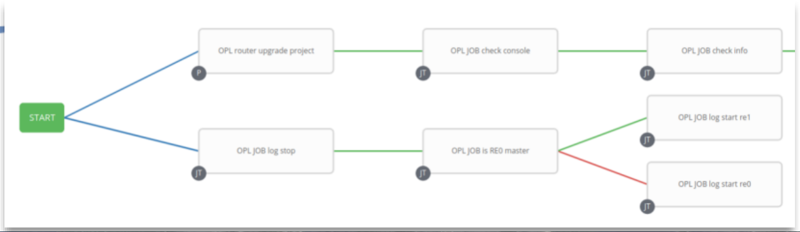
PNF Upgrade - architecture





Index of /

Name	Last modified	Size	Des
junos-install-mx-x86..>	2020-04-20 13:22	2.5G	
junos-install-mx-x86..>	2020-04-20 13:25	2.0G	



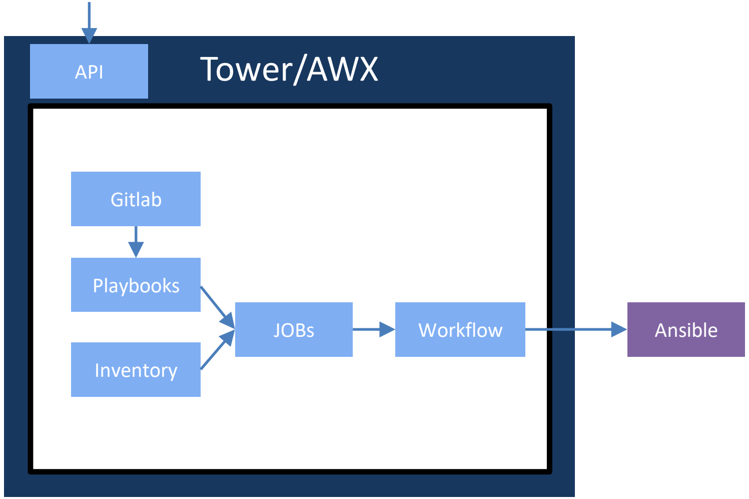
Ansible & AWX

Ansible playbooks:

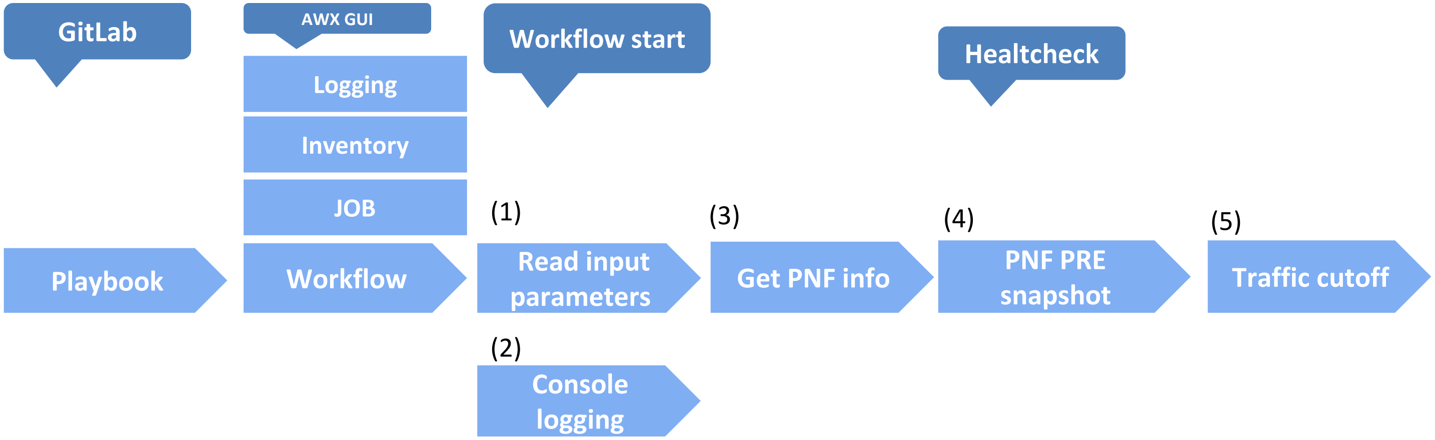
- for config operations
- for config snapshots

Ansible AWX:

- manage Ansible playbooks
- create and visualize AWX workflows
- create decision/approval points
- sourcing Ansible YAML from GitLab



Block #1 (preparation)



(1) Input parameters*: **OS filename***, ***pnf oam ip address, + additional (default)***.

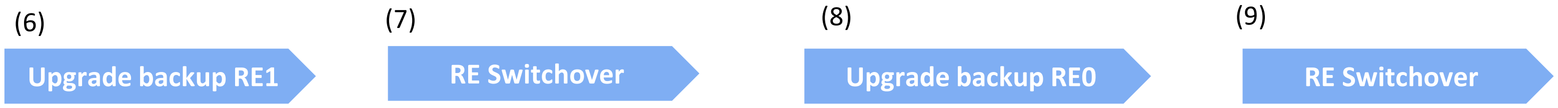
(2) In parallel console logging is started.

(3) Check PNF resources and liveness. Decide if PNF is available to upgrade.

(4) Taking snapshot of particular part of device (routes, state of interfaces, etc.)

(5) PNF must be isolated before upgrade.

Block #2 (upgrade)



(6) Upgrade backup RE1

(7) RE Switchover

(8) Upgrade backup RE0

(9) RE Switchover

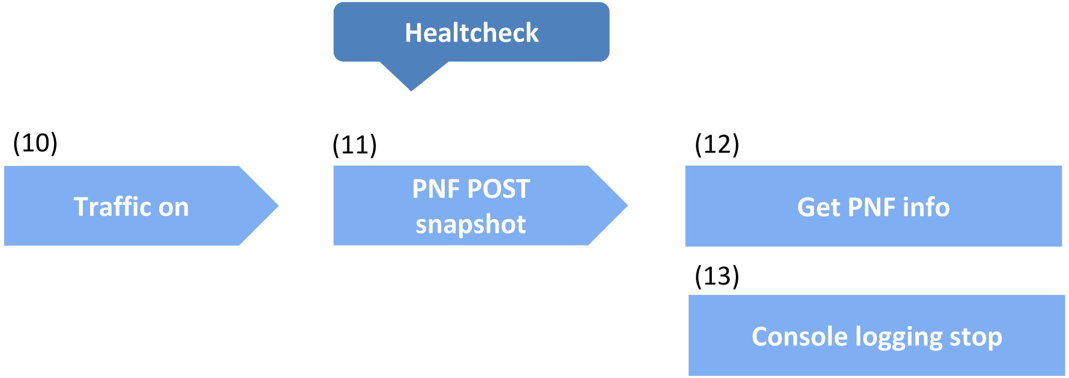
(6) Upgrade backup Router Engine (RE1)

(7) Switchover Router Engine role. Backup RE1 becomes master engine

(8) Upgrade backup Router Engine (RE0)

(9) Switchover Router Engine role. Backup RE0 becomes master engine

Block #3 (post upgrade)



(10) Traffic on node must be back. Interfaces goes back to pre upgrade state.

(11) Post upgrade snapshoot is performer and compared with pre-upgrade healtcheck.

(12) Check PNF resources and liveness. Check version and platform after upgrade

(13) Console logging is stopped.

PNF Upgrade

DEMO

PNF Management with ONAP

- Automated solution for real operational/large scale deployments
- Perform e2e PNF (re)configuration in one request via ONAP
- Management operations’ design and implementation are external to ONAP
- ONAP support PNF to achieve the full benefits of unified management of network services and automation capabilities
- Automation of data gathering and system integration for PNF (re)configuration
- Easy inventory, control, versioning, integration of diverse management/scripting logic

Plans

- Explore PNF support for ONAP Guilin +
- Dedicated PNF reconfiguration with SO using *onapsdk* python package
- More complex use-case (PNF, CNF, VNF)
- Study of different workflow composition options

LF NETWORKING

LFN Developer & Testing Forum