

Control Loops for O-RU Fronthaul Recovery usecase

This page describes how to create and run the control loops for the "Hello World" O-RU Fronthaul Recovery usecase. This can be done either in docker environment using docker-compose files (available in the nonrtic repo of OSC), or in kubernetes environment using the complete ONAP installation done via OOM. Moreover, the control loop for apex policy version of the usecase can be created using Policy participant, whereas the control loop for script version of the usecase can be created using Kubernetes participant (both participants available in policy/clamp repo of ONAP).

- [Control loops in kubernetes](#)
 - [Create topic in Dmaap MR](#)
 - [Run Policy GUI](#)
 - [Commission/Instantiate control loop via GUI](#)
- - [a\) Control loop for apex policy version](#)
 - [b\) Control loop for script version](#)
- [Control loops in docker](#)
 - [a\) Control loop for apex policy version](#)
 - [b\) Control loop for script version](#)

Control loops in kubernetes

This section is related to running the control loops in a kubernetes environment. Specifically, it describes how to deploy the control loops in a full-fledge installation of ONAP assuming that the installation was done in a cluster using 'istanbul' branch of OOM.

Firstly, the common steps for creating control loops for both apex policy and script versions of the usecase are described. This is followed by the steps that are unique for setting up and testing each version individually.

Create topic in Dmaap MR

In order to create the fault notification topic in Dmaap Message Router, the first step is to find out its NodePort and NodeIP. The NodeIP is the IP address of any k8s node in the cluster where ONAP has been installed, and it can be found using the command "kubectl get nodes -o wide". The NodePort can be found using the command "kubectl -n onap get svc | grep message-router-external". Next, the topic defined for this usecase can be created using:

```
curl -k -X POST -H "Content-Type: application/json" -d '{"topicName": "unauthenticated.SEC_FAULT_OUTPUT"}' https://<NodeIP>:<NodePort-message-router>/events/unauthenticated.SEC_FAULT_OUTPUT
```

Run Policy GUI

The easiest way to create the control loops is via Policy GUI component of the clamp. The below steps describe how to start this GUI.

NOTE: At the time of writing this page (15 Dec 2021), there is a bug in the helm chart of policy/clamp in 'istanbul' branch of OOM. The bug should be fixed by the policy/clamp team. Until then, the following steps should be done to fix this problem. Run the command:

```
kubectl -n onap edit cm def-policy-clamp-be-configmap
```

(whereas "def" refers to the name of deployment and should be replaced with the name used when installing ONAP. The same should be done for all instructions given on this page that use "def" as deployment name)

and change http to https in clamp.config.controlloop.runtime.url under application.properties. Then, run this command:

```
kubectl rollout restart deployment def-policy-clamp-be
```

Next step is to find out the NodePort of policy-gui. This can be done by using the command "kubectl -n onap get svc | grep policy-gui".

Then, open a web browser and navigate to the url:

<https://<NodeIP>:<NodePort-policy-gui>/clamp/>

Use below credentials for the GUI:

username: demo@people.osaaf.org. password: demo123456!



[POLICY Framework](#)
[CLAMP Options](#)
[LOOP Instance](#)
[LOOP Operations](#)
[TOSCA Control Loop](#)
[Help](#)
[Signed in as: demo@people.osaaf.org](#)

Loop Viewer - Empty (NO loop loaded yet) - ()

No LOOP (SVG)

Loop Status:

Component Name	Component State	Description

Loop Logs

Date	Type	Component	Log

Start-up screen of the Policy GUI

Commission/Instantiate control loop via GUI

This sub-section shows how to commission and instantiate the control loops via policy-gui. The individual tosca templates for each of the apex policy and script versions are provided later in the relevant sub-sections. The screenshots shown in this sub-section are general steps that are applicable for both versions.

Go to **Tosca Control Loop** pane, and select **Upload Tosca to Commissioning** in order to upload the tosca template (provided later in the relevant sub-section).



[POLICY Framework](#)
[CLAMP Options](#)
[LOOP Instance](#)
[LOOP Operations](#)
[TOSCA Control Loop](#)
[Help](#)
[Signed in as: demo@people.osaaf.org](#)

Loop Viewer - Empty (NO loop loaded yet) - ()

No LOOP (SVG)

Loop Status:

Component Name	Component State	Description

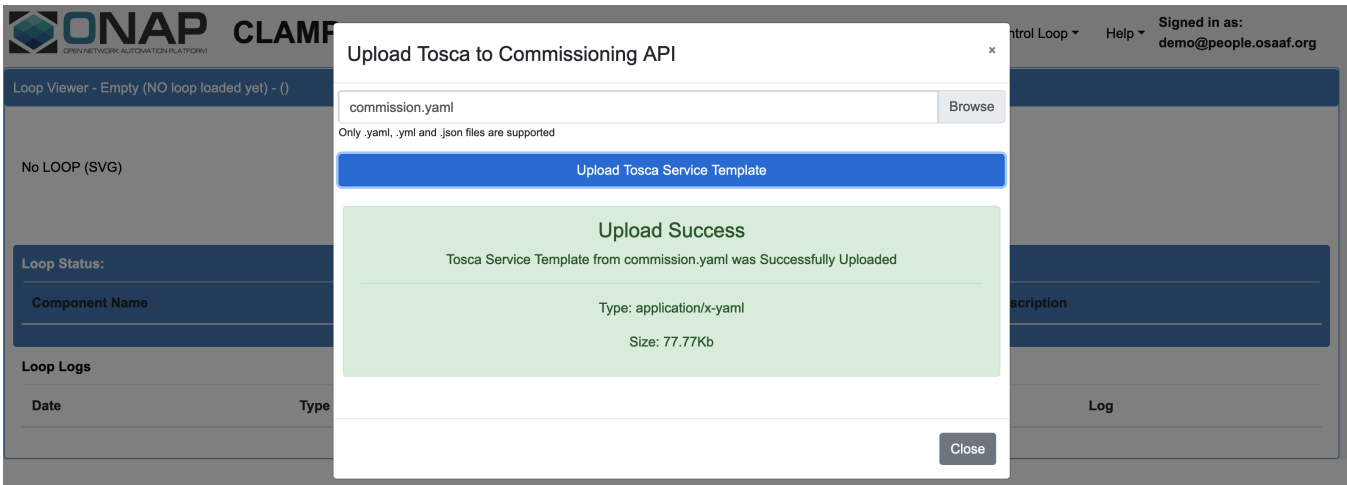
Loop Logs

Date	Type	Component	Log

Instantiation
Instantiation Management

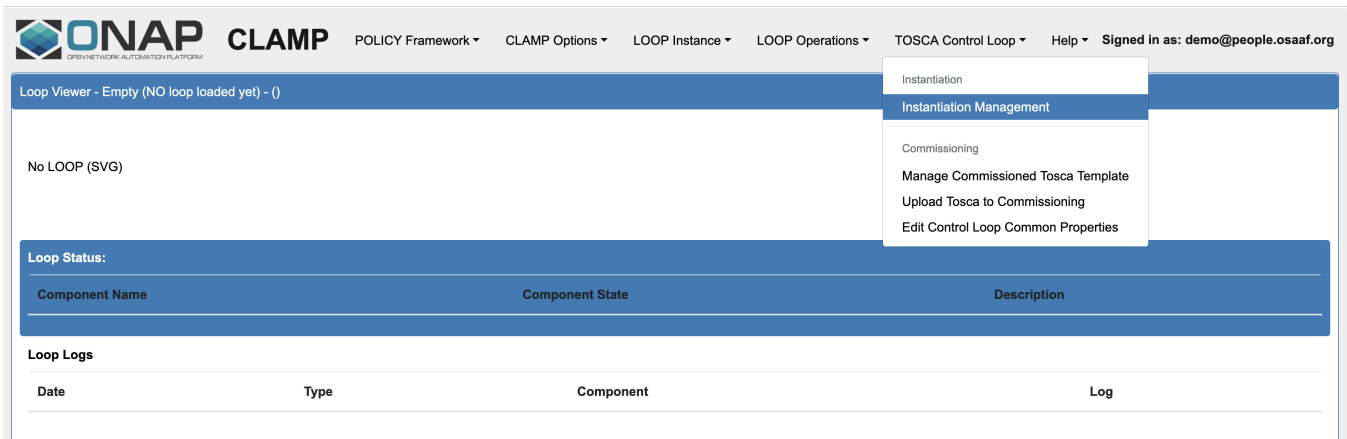
Commissioning
Manage Commissioned Tosca Template
Upload Tosca to Commissioning
Edit Control Loop Common Properties

Upload tosca template for commissioning

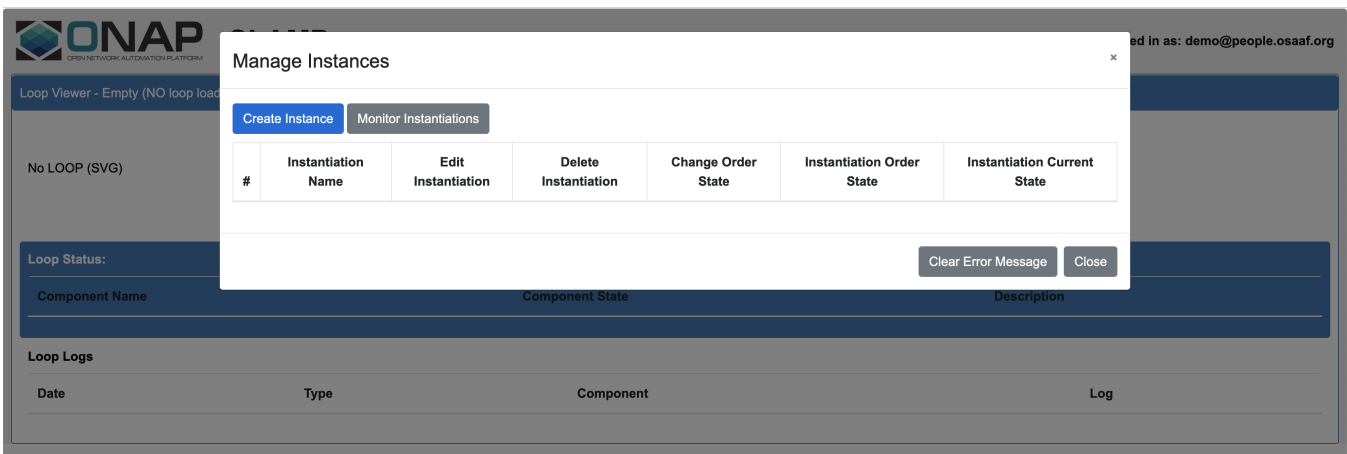


Tosca template uploaded successfully

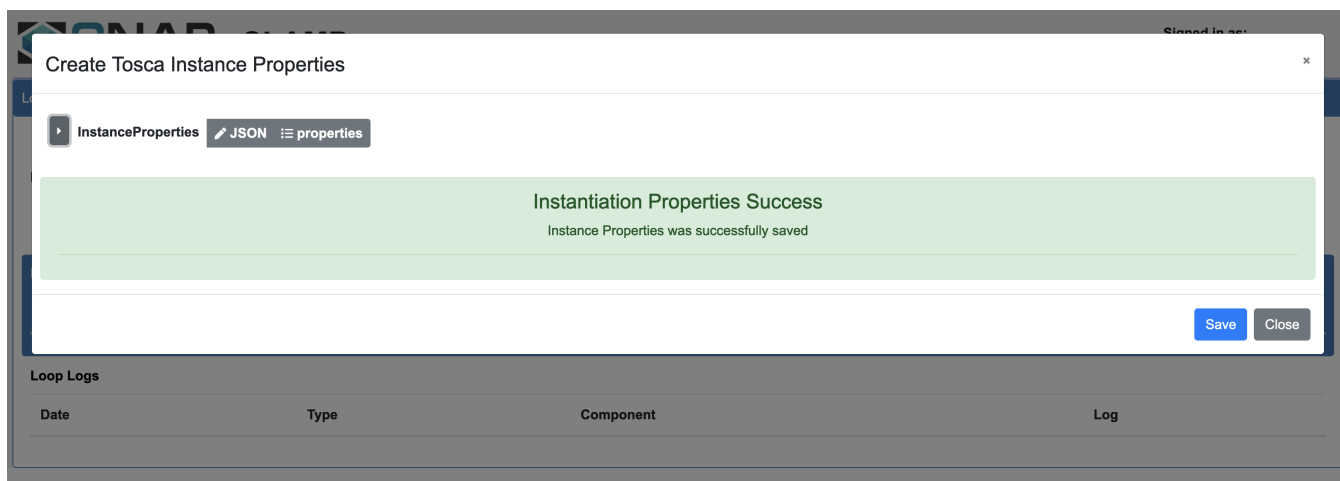
After commissioning the tosca template, the next step is to instantiate the control loop. Go to **Tosca Control Loop** pane, and select **Instantiation Management** and then press the **Create Instance** button. If no changes need to be made in the instance properties, press the **Save** button and it should show a message depicting that the instantiation operation was successful.



Instantiate the control loop

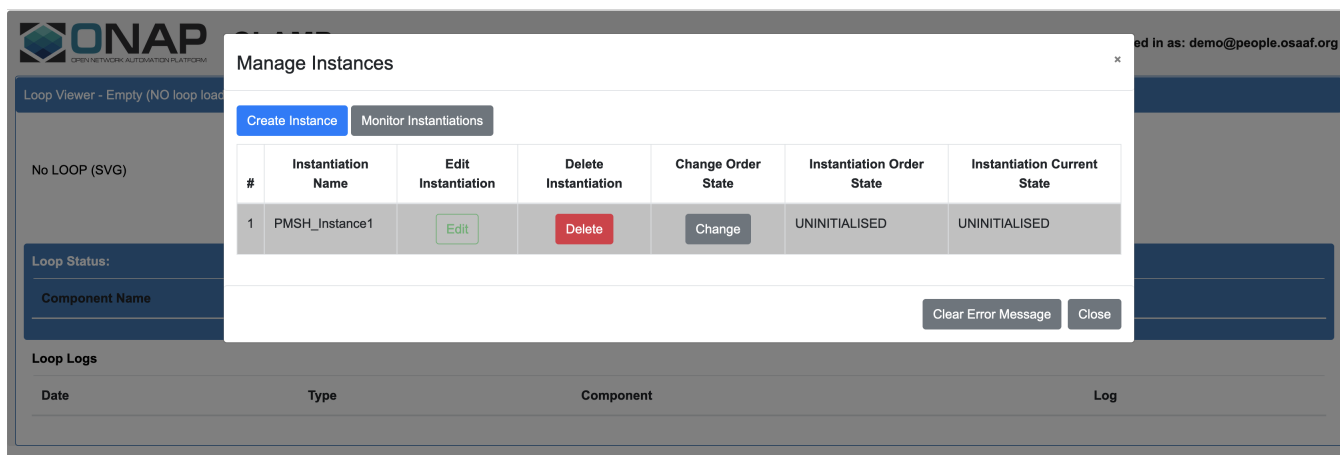


Create Instance dialog



Instantiation properties saved successfully

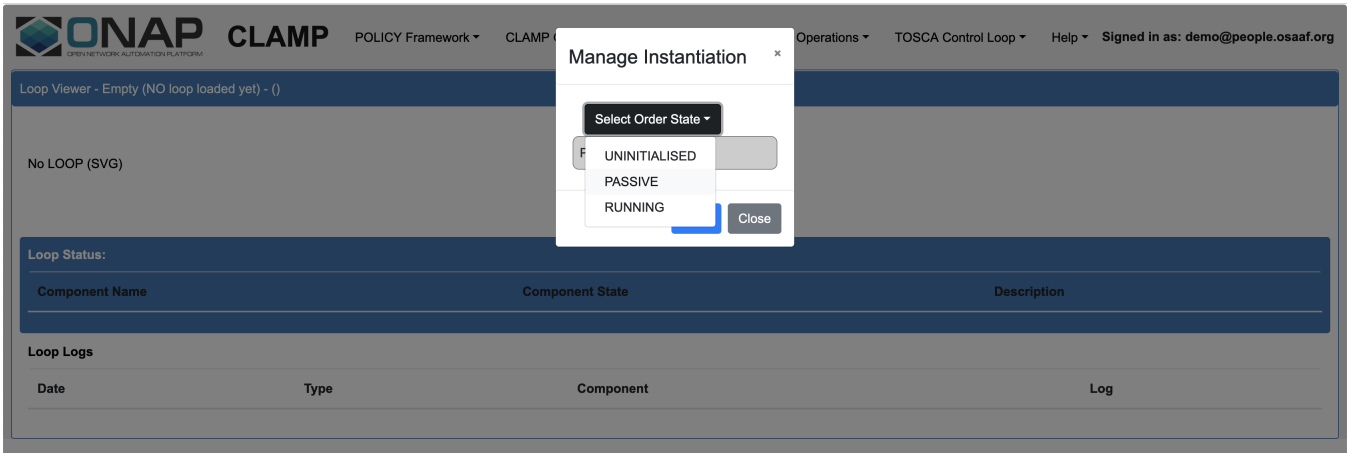
Go back again to **Instantiation Management** under **Tosca Control Loop** pane, and the newly created control loop instance in UNINITIALISED state will pop up. If nothing shows up, refresh the web browser and try again.



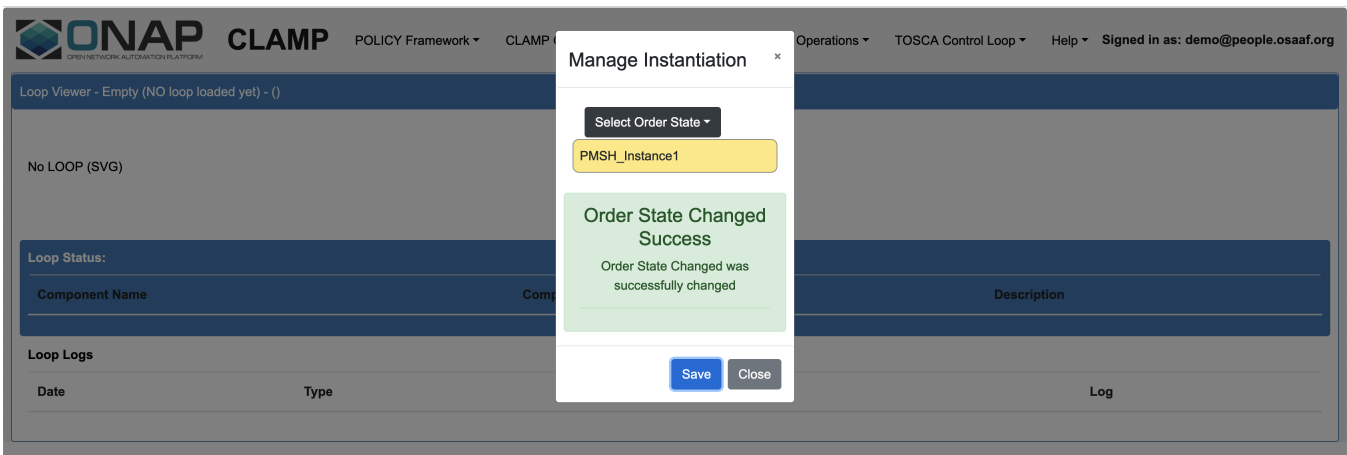
Newly created control loop instance in UNINITIALISED state

NOTE: There is a bug in the istanbul version of policy/clamp that each control loop instance is named as **PMSH_Instance1**. This should be fixed by the clamp team, however it can be ignored if the instance name is not important for the user.

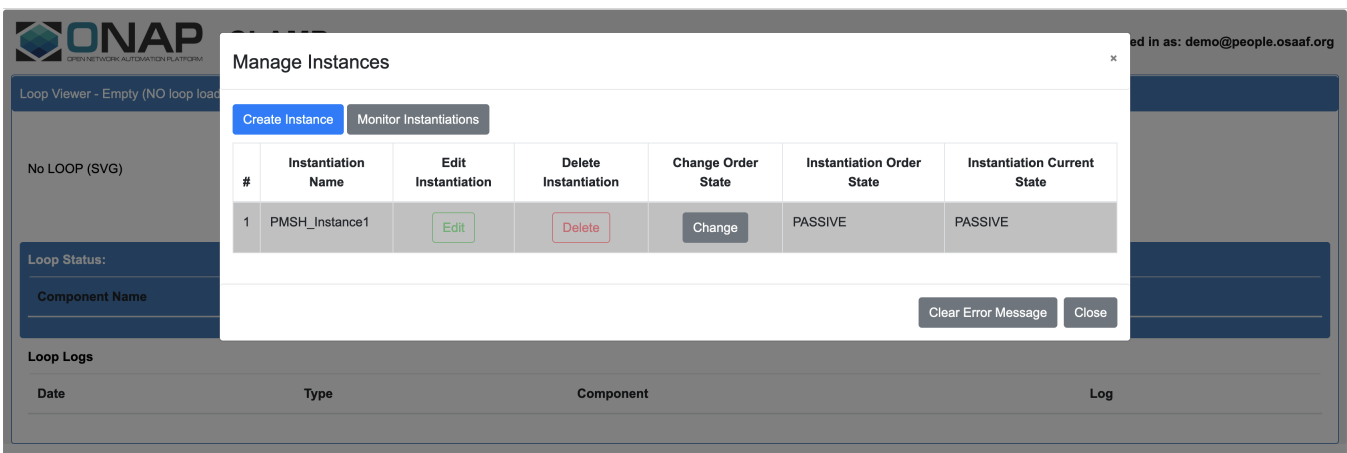
Press the **Change** button under **Change Order State**. Then, press the **Select Order State** drop-down menu, and select PASSIVE. Finally, press the **Save** button to change the control loop to PASSIVE state.



Changing the control loop to PASSIVE state



State changed successfully



Control loop changed to PASSIVE state

In a similar way, change the control loop state to RUNNING.

ONAP OPEN NETWORK AUTOMATION PLATFORM

Signed in as: demo@people.osaaf.org

Loop Viewer - Empty (NO loop load)

No LOOP (SVG)

Loop Status:

Component Name

Loop Logs

Date Type Component Log

Manage Instances

Create Instance Monitor Instantiations

#	Instantiation Name	Edit Instantiation	Delete Instantiation	Change Order State	Instantiation Order State	Instantiation Current State
1	PMSH_Instance1	Edit	Delete	Change	RUNNING	RUNNING

Clear Error Message Close

Control loop changed to RUNNING state

Once the control loop gets into the RUNNING state, the corresponding version of the usecase should be up and running.

NOTE: There is a limitation in the istanbul version of policy/clamp that only one toasca template can be commissioned at a time. So, always delete the currently commissioned template before trying a new one.

In order to delete the control loop instance, it should be first changed back to PASSIVE state and then to UNINITIALISED state. Once the instance shows PASSIVE under **Instantiation Current State**, press the **Delete** button under **Delete Instantiation**.

ONAP OPEN NETWORK AUTOMATION PLATFORM

Signed in as: demo@people.osaaf.org

Loop Viewer - Empty (NO loop load)

No LOOP (SVG)

Loop Status:

Component Name

Loop Logs

Date

Manage Instances

Create Instance Monitor Instantiations

#	Instantiation Name	Edit Instantiation	Delete Instantiation	Change Order State	Instantiation Order State	Instantiation Current State
---	--------------------	--------------------	----------------------	--------------------	---------------------------	-----------------------------

Deletion of Instantiation Success
Deletion of Instantiation was successful!!

Clear Error Message Close

Control loop instance deleted

After deleting the control loop instance, the toasca template can be decommissioned as follows.

Go to **Tosca Control Loop** pane, and select **Manage Commissioned Tosca Template**.

ONAP

CLAMP

POLICY Framework

CLAMP Options

LOOP Instance

LOOP Operations

TOSCA Control Loop

Help

Signed in as: demo@people.osaaf.org

Loop Viewer - Empty (NO loop loaded yet) - ()

No LOOP (SVG)

Loop Status:

Component Name	Component State	Description
----------------	-----------------	-------------

Loop Logs

Date	Type	Component	Log
------	------	-----------	-----

Instantiation

Instantiation Management

Commissioning

Manage Commissioned Tosca Template

Upload Tosca to Commissioning

Edit Control Loop Common Properties

Manage commissioned tosca template

Press the button **Pull Tosca Service Template** and it should show the commissioned tosca template. Once the template shows up, press the **Delete Tosca Service Template** button. This will be followed by a "Delete Successful" message.

ONAP

CLAMP

POLICY Framework

CLAMP Options

LOOP Instance

LOOP Operations

TOSCA Control Loop

Help

Signed in as: demo@people.osaaf.org

Loop Viewer - Empty (NO loop loaded yet) - ()

No LOOP (SVG)

Loop Status:

Component Name	Component State	Description
----------------	-----------------	-------------

Loop Logs

Date	Type	Component	Log
------	------	-----------	-----

View Tosca Template

Pull Tosca Service Template

```
{
  "policy_types": {
    "onap.policies.controlloop.operational.Common": {
      "name": "onap.policies.controlloop.operational.Common",
      "version": "1.0.0",
      "derived_from": "tosca.policies.Root",
      "metadata": {},
      "description": "Operational Policy for Control Loop execution. Originated in Frankfurt to support TOSCA Compliant Policy Type",
      "properties": {
        "abatement": {
          "name": "abatement",
          "type": "boolean",
          "type_version": "0.0.0",
          "description": "Whether an abatement event message will be expected for the control loop from DCAE.",
          "default_value": false,
          "required": true,
          "status": null,
          "constraints": null,
          "key_schema": null,
          "entry_schema": null,
          "metadata": null
        }
      },
      "operations": {
        "name": "operations",
        "type": "list",
        "type_version": "0.0.0",
        "description": "List of operations to be performed when Control Loop is triggered."
      }
    }
  }
}
```

Delete Tosca Service Template

Deleting the commissioned tosca template



OPEN NETWORK AUTOMATION PLATFORM

Loop Viewer - Empty (NO loop loaded)

No LOOP (SVG)

Loop Status:

Component Name

Loop Logs

Date

```
{  "abatement": {    "name": "abatement",    "type": "boolean",    "type_version": "0.0.0",    "description": "Whether an abatement event message will be expected for the control loop from DCAE.",    "default_value": false,    "required": true,    "status": null,    "constraints": null,    "key_schema": null,    "entry_schema": null,    "metadata": null  },  "operations": {    "name": "operations",    "type": "list",    "type_version": "0.0.0",    "description": "List of operations to be performed when Control Loop is triggered."  }},
```

Delete Successful

```
{  "errorDetails": null,  "affectedControlLoopDefinitions": [    {      "name": "ToscaServiceTemplateSimple",      "version": "1.0.0"    }  ]}
```

ed in as: demo@people.osaaf.org

Close

Tosca template deleted successfully

a) Control loop for apex policy version

This sub-section describes the steps required for bringing up the control loop with apex policy version of the usecase. The tosa template to be used for commissioning this control loop is given below. The steps for commissioning are depicted in the previous sub-section.

commission.yaml

```
tosca_definitions_version: tosca_simple_yaml_1_1_0
data_types:
  onap.datatypes.ToscaConceptIdentifier:
    derived_from: tosca.datatypes.Root
    properties:
      name:
        type: string
        required: true
      version:
        type: string
        required: true
  onap.datatype.controlloop.Target:
    derived_from: tosca.datatypes.Root
    description: Definition for a entity in A&AI to perform a control loop operation on
    properties:
      targetType:
        type: string
        description: Category for the target type
        required: true
        constraints:
          - valid_values:
              - VNF
              - VM
              - VFMODULE
              - PNF
    entityIds:
      type: map
      description: |
        Map of values that identify the resource. If none are provided, it is assumed that the
        entity that generated the ONSET event will be the target.
      required: false
      metadata:
        clamp_possible_values: ClampExecution:CSAR_RESOURCES
```



```

    entry_schema:
      type: string
onap.datatype.controlloop.Actor:
  derived_from: tosca.datatypes.Root
  description: An actor/operation/target definition
  properties:
    actor:
      type: string
      description: The actor performing the operation.
      required: true
      metadata:
        clamp_possible_values: Dictionary:DefaultActors, ClampExecution:CDS/actor
    operation:
      type: string
      description: The operation the actor is performing.
      metadata:
        clamp_possible_values: Dictionary:DefaultOperations, ClampExecution:CDS/operation
      required: true
    target:
      type: onap.datatype.controlloop.Target
      description: The resource the operation should be performed on.
      required: true
    payload:
      type: map
      description: Name/value pairs of payload information passed by Policy to the actor
      required: false
      metadata:
        clamp_possible_values: ClampExecution:CDS/payload
      entry_schema:
        type: string
onap.datatype.controlloop.Operation:
  derived_from: tosca.datatypes.Root
  description: An operation supported by an actor
  properties:
    id:
      type: string
      description: Unique identifier for the operation
      required: true
    description:
      type: string
      description: A user-friendly description of the intent for the operation
      required: false
    operation:
      type: onap.datatype.controlloop.Actor
      description: The definition of the operation to be performed.
      required: true
    timeout:
      type: integer
      description: The amount of time for the actor to perform the operation.
      required: true
    retries:
      type: integer
      description: The number of retries the actor should attempt to perform the operation.
      required: true
      default: 0
    success:
      type: string
      description: Points to the operation to invoke on success. A value of "final_success" indicates and end
to the operation.
      required: false
      default: final_success
    failure:
      type: string
      description: Points to the operation to invoke on Actor operation failure.
      required: false
      default: final_failure
    failure_timeout:
      type: string
      description: Points to the operation to invoke when the time out for the operation occurs.
      required: false
      default: final_failure_timeout

```

```

failure_retries:
  type: string
  description: Points to the operation to invoke when the current operation has exceeded its max retries.
  required: false
  default: final_failure_retries
failure_exception:
  type: string
  description: Points to the operation to invoke when the current operation causes an exception.
  required: false
  default: final_failure_exception
failure_guard:
  type: string
  description: Points to the operation to invoke when the current operation is blocked due to guard
policy_enforcement:
  required: false
  default: final_failure_guard
policy_types:
  onap.policies.controlloop.operational.Common:
    derived_from: tosca.policies.Root
    version: 1.0.0
    name: onap.policies.controlloop.operational.Common
    description: |
      Operational Policy for Control Loop execution. Originated in Frankfurt to support TOSCA Compliant
      Policy Types. This does NOT support the legacy Policy YAML policy type.
    properties:
      id:
        type: string
        description: The unique control loop id.
        required: true
      timeout:
        type: integer
        description: |
          Overall timeout for executing all the operations. This timeout should equal or exceed the total
          timeout for each operation listed.
        required: true
      abatement:
        type: boolean
        description: Whether an abatement event message will be expected for the control loop from DCAE.
        required: true
        default: false
      trigger:
        type: string
        description: Initial operation to execute upon receiving an Onset event message for the Control Loop.
        required: true
    operations:
      type: list
      description: List of operations to be performed when Control Loop is triggered.
      required: true
      entry_schema:
        type: onap.datatype.controlloop.Operation
  onap.policies.controlloop.operational.common.Apex:
    derived_from: onap.policies.controlloop.operational.Common
    type_version: 1.0.0
    version: 1.0.0
    name: onap.policies.controlloop.operational.common.Apex
    description: Operational policies for Apex PDP
    properties:
      engineServiceParameters:
        type: string
        description: The engine parameters like name, instanceCount, policy implementation, parameters etc.
        required: true
      eventInputParameters:
        type: string
        description: The event input parameters.
        required: true
      eventOutputParameters:
        type: string
        description: The event output parameters.
        required: true
    javaProperties:
      type: string

```

```

        description: Name/value pairs of properties to be set for APEX if needed.
        required: false
node_types:
  org.onap.policy.clamp.controlloop.Participant:
    version: 1.0.1
    derived_from: tosca.nodetypes.Root
    properties:
      provider:
        type: string
        required: false
  org.onap.policy.clamp.controlloop.ControlLoopElement:
    version: 1.0.1
    derived_from: tosca.nodetypes.Root
    properties:
      provider:
        type: string
        required: false
      metadata:
        common: true
      description: Specifies the organization that provides the control loop element
  participant_id:
    type: onap.datatypes.ToscaConceptIdentifier
    required: true
    metadata:
      common: true
  participantType:
    type: onap.datatypes.ToscaConceptIdentifier
    required: true
    metadata:
      common: true
      description: The identity of the participant type that hosts this type of Control Loop Element
  startPhase:
    type: integer
    required: false
    constraints:
      - greater_or_equal: 0
    metadata:
      common: true
      description: A value indicating the start phase in which this control loop element will be started, the
        first start phase is zero. Control Loop Elements are started in their start_phase order and stopped
        in reverse start phase order. Control Loop Elements with the same start phase are started and
        stopped simultaneously
  uninitializedToPassiveTimeout:
    type: integer
    required: false
    constraints:
      - greater_or_equal: 0
    default: 60
    metadata:
      common: true
      description: The maximum time in seconds to wait for a state change from uninitialized to passive
  passiveToRunningTimeout:
    type: integer
    required: false
    constraints:
      - greater_or_equal: 0
    default: 60
    metadata:
      common: true
      description: The maximum time in seconds to wait for a state change from passive to running
  runningToPassiveTimeout:
    type: integer
    required: false
    constraints:
      - greater_or_equal: 0
    default: 60
    metadata:
      common: true
      description: The maximum time in seconds to wait for a state change from running to passive
  passiveToUninitializedTimeout:
    type: integer

```

```

    required: false
    constraints:
      - greater_or_equal: 0
    default: 60
    metadata:
      common: true
    description: The maximum time in seconds to wait for a state change from passive to uninitialized
org.onap.policy.clamp.controlloop.ControlLoop:
  version: 1.0.1
  derived_from: tosca.nodetypes.Root
  properties:
    provider:
      type: string
      required: false
      metadata:
        common: true
      description: Specifies the organization that provides the control loop element
  elements:
    type: list
    required: true
    metadata:
      common: true
    entry_schema:
      type: onap.datatypes.ToscaConceptIdentifier
      description: Specifies a list of control loop element definitions that make up this control loop
definition
org.onap.policy.clamp.controlloop.PolicyControlLoopElement:
  version: 1.0.1
  derived_from: org.onap.policy.clamp.controlloop.ControlLoopElement
  properties:
    policy_type_id:
      type: onap.datatypes.ToscaConceptIdentifier
      required: true
    policy_id:
      type: onap.datatypes.ToscaConceptIdentifier
      required: false
topology_template:
  node_templates:
    org.onap.domain.linkmonitor.LinkMonitorPolicyControlLoopElement:
      version: 1.2.3
      type: org.onap.policy.clamp.controlloop.PolicyControlLoopElement
      type_version: 1.0.1
      description: Control loop element for the Link Monitor
      properties:
        provider: Ericsson
        participant_id:
          name: org.onap.PM_Policy
          version: 1.0.0
        participantType:
          name: org.onap.policy.controlloop.PolicyControlLoopParticipant
          version: 2.3.1
        policy_type_id:
          name: onap.policies.controlloop.operational.common.Apex
          version: 1.0.0
        policy_id:
          name: operational.apex.linkmonitor
          version: 1.0.0
        pdpGroup: defaultGroup
    org.onap.domain.linkmonitor.LinkMonitorControlLoopDefinition0:
      version: 1.2.3
      type: org.onap.policy.clamp.controlloop.ControlLoop
      type_version: 1.0.0
      description: Control loop for Link Monitor
      properties:
        provider: Ericsson
        elements:
          - name: org.onap.domain.linkmonitor.LinkMonitorPolicyControlLoopElement
            version: 1.2.3
org.onap.policy.controlloop.PolicyControlLoopParticipant:
  version: 2.3.1
  type: org.onap.policy.clamp.controlloop.Participant

```

```

    type_version: 1.0.1
    description: Participant for policy framework
    properties:
      provider: ONAP
  policies:
  - operational.apex.linkmonitor:
    type: onap.policies.controlloop.operational.common.Apex
    type_version: 1.0.0
    version: 1.0.0
    metadata:
      policy-id: operational.apex.linkmonitor
      policy-version: 1.0.0
    properties:
      engineServiceParameters:
        name: LinkMonitorApexEngine
        version: 0.0.1
        id: 101
        instanceCount: 1
        deploymentPort: 12345
      engineParameters:
        executorParameters:
          JAVASCRIPT:
            parameterClassName: org.onap.policy.apex.plugins.executor.javascript.
JavascriptExecutorParameters
            contextParameters:
              parameterClassName: org.onap.policy.apex.context.parameters.ContextParameters
              schemaParameters:
                Avro:
                  parameterClassName: org.onap.policy.apex.plugins.context.schema.avro.
AvroSchemaHelperParameters
            taskParameters:
              - key: ORU-ODU-Map
                value: |-
                  {
                    "ERICSSON-O-RU-11220": "O-DU-1122",
                    "ERICSSON-O-RU-11221": "O-DU-1122",
                    "ERICSSON-O-RU-11222": "O-DU-1122",
                    "ERICSSON-O-RU-11223": "O-DU-1122",
                    "ERICSSON-O-RU-11224": "O-DU-1123",
                    "ERICSSON-O-RU-11225": "O-DU-1123",
                    "ERICSSON-O-RU-11226": "O-DU-1123",
                    "ERICSSON-O-RU-11227": "O-DU-1124",
                    "ERICSSON-O-RU-11228": "O-DU-1125",
                    "ERICSSON-O-RU-11229": "O-DU-1125"
                  }
      policy_type_impl:
        apexPolicyModel:
          key:
            name: LinkMonitorModel
            version: 0.0.1
          keyInformation:
            key:
              name: LinkMonitorModel_KeyInfo
              version: 0.0.1
            keyInfoMap:
              entry:
                - key:
                    name: ApexMessageOutputEvent
                    version: 0.0.1
                  value:
                    key:
                      name: ApexMessageOutputEvent
                      version: 0.0.1
                      UUID: cca47d74-7754-4a61-b163-ca31f66b157b
                      description: Generated description for concept referred to by
                        key "ApexMessageOutputEvent:0.0.1"
                - key:
                    name: CreateLinkClearedOutfieldsEvent
                    version: 0.0.1
                  value:
                    key:

```

```

        name: CreateLinkClearedOutfieldsEvent
        version: 0.0.1
    UUID: a295d6a3-1b73-387e-abba-b41e9b608802
    description: Generated description for concept referred to by
        key "CreateLinkClearedOutfieldsEvent:0.0.1"
- key:
    name: CreateLinkClearedOutfieldsTask
    version: 0.0.1
    value:
        key:
            name: CreateLinkClearedOutfieldsTask
            version: 0.0.1
            UUID: fd594e88-411d-4a94-b2be-697b3a0d7adf
            description: This task creates the output fields when link failure
                is cleared.
- key:
    name: CreateLinkFailureOutfieldsEvent
    version: 0.0.1
    value:
        key:
            name: CreateLinkFailureOutfieldsEvent
            version: 0.0.1
            UUID: 02be2b5d-45b7-3c54-ae54-97f2b5c30125
            description: Generated description for concept referred to by
                key "CreateLinkFailureOutfieldsEvent:0.0.1"
- key:
    name: CreateLinkFailureOutfieldsTask
    version: 0.0.1
    value:
        key:
            name: CreateLinkFailureOutfieldsTask
            version: 0.0.1
            UUID: ac3d9842-80af-4a98-951c-bd79a431c613
            description: This task the output fields when link failure is
                detected.
- key:
    name: LinkClearedTask
    version: 0.0.1
    value:
        key:
            name: LinkClearedTask
            version: 0.0.1
            UUID: eecfde90-896c-4343-8f9c-2603ced94e2d
            description: This task sends a message to the output when link
                failure is cleared.
- key:
    name: LinkFailureInputEvent
    version: 0.0.1
    value:
        key:
            name: LinkFailureInputEvent
            version: 0.0.1
            UUID: c4500941-3f98-4080-a9cc-5b9753ed050b
            description: Generated description for concept referred to by
                key "LinkFailureInputEvent:0.0.1"
- key:
    name: LinkFailureInputSchema
    version: 0.0.1
    value:
        key:
            name: LinkFailureInputSchema
            version: 0.0.1
            UUID: 3b3974fc-3012-3b02-9f33-c9d8eefe4dc1
            description: Generated description for concept referred to by
                key "LinkFailureInputSchema:0.0.1"
- key:
    name: LinkFailureOutputEvent
    version: 0.0.1
    value:
        key:
            name: LinkFailureOutputEvent

```

```

        version: 0.0.1
        UUID: 4f04aa98-e917-4f4a-882a-c75ba5a99374
        description: Generated description for concept referred to by
            key "LinkFailureOutputEvent:0.0.1"
- key:
    name: LinkFailureOutputSchema
    version: 0.0.1
value:
    key:
        name: LinkFailureOutputSchema
        version: 0.0.1
        UUID: 2d1a7f6e-eb9a-3984-be1f-283d98111b84
        description: Generated description for concept referred to by
            key "LinkFailureOutputSchema:0.0.1"
- key:
    name: LinkFailureTask
    version: 0.0.1
value:
    key:
        name: LinkFailureTask
        version: 0.0.1
        UUID: 3351b0f4-cf06-4fa2-8823-edf67bd30223
        description: This task updates the config for O-RU when link
            failure is detected.
- key:
    name: LinkMonitorModel
    version: 0.0.1
value:
    key:
        name: LinkMonitorModel
        version: 0.0.1
        UUID: 540226fb-55ee-4f0e-a444-983a0494818e
        description: This is the Apex Policy Model for link monitoring.
- key:
    name: LinkMonitorModel_Events
    version: 0.0.1
value:
    key:
        name: LinkMonitorModel_Events
        version: 0.0.1
        UUID: 27ad3e7e-fe3b-3bd6-9081-718705c2bcea
        description: Generated description for concept referred to by
            key "LinkMonitorModel_Events:0.0.1"
- key:
    name: LinkMonitorModel_KeyInfo
    version: 0.0.1
value:
    key:
        name: LinkMonitorModel_KeyInfo
        version: 0.0.1
        UUID: ea0b5f58-eefd-358a-9660-840c640bf981
        description: Generated description for concept referred to by
            key "LinkMonitorModel_KeyInfo:0.0.1"
- key:
    name: LinkMonitorModel_Policies
    version: 0.0.1
value:
    key:
        name: LinkMonitorModel_Policies
        version: 0.0.1
        UUID: ee9e0b0f-2b7d-3ab7-9a98-c5ec05ed823d
        description: Generated description for concept referred to by
            key "LinkMonitorModel_Policies:0.0.1"
- key:
    name: LinkMonitorModel_Schemas
    version: 0.0.1
value:
    key:
        name: LinkMonitorModel_Schemas
        version: 0.0.1
        UUID: fa5f9b8f-796c-3c70-84e9-5140c958c4bb

```

```

        description: Generated description for concept referred to by
        key "LinkMonitorModel_Schemas:0.0.1"
- key:
  name: LinkMonitorModel_Tasks
  version: 0.0.1
  value:
    key:
      name: LinkMonitorModel_Tasks
      version: 0.0.1
      UUID: eec592f7-69d5-39a9-981a-e552f787ed01
      description: Generated description for concept referred to by
      key "LinkMonitorModel_Tasks:0.0.1"
- key:
  name: LinkMonitorPolicy
  version: 0.0.1
  value:
    key:
      name: LinkMonitorPolicy
      version: 0.0.1
      UUID: 6c5e410f-489a-46ff-964e-982ce6e8b6d0
      description: Generated description for concept referred to by
      key "LinkMonitorPolicy:0.0.1"
- key:
  name: MessageSchema
  version: 0.0.1
  value:
    key:
      name: MessageSchema
      version: 0.0.1
      UUID: ac4b34ac-39d6-3393-a267-8d5b84854018
      description: A schema for messages from apex
- key:
  name: NoPolicyDefinedTask
  version: 0.0.1
  value:
    key:
      name: NoPolicyDefinedTask
      version: 0.0.1
      UUID: d48b619e-d00d-4008-b884-02d76ea4350b
      description: This task sends a message to the output when an
      event is received for which no policy has been defined.
- key:
  name: OduIdSchema
  version: 0.0.1
  value:
    key:
      name: OduIdSchema
      version: 0.0.1
      UUID: 50662174-a88b-3cbd-91bd-8e91b40b2660
      description: A schema for O-DU-ID
- key:
  name: OruIdSchema
  version: 0.0.1
  value:
    key:
      name: OruIdSchema
      version: 0.0.1
      UUID: 54daf32b-015f-39cd-8530-a1175c5553e9
      description: A schema for O-RU-ID
policies:
  key:
    name: LinkMonitorModel_Policies
    version: 0.0.1
  policyMap:
    entry:
      - key:
        name: LinkMonitorPolicy
        version: 0.0.1
      value:
        policyKey:
          name: LinkMonitorPolicy

```



```
version: 0.0.1
template: Freestyle
state:
  entry:
    - key: LinkClearedState
      value:
        stateKey:
          parentKeyName: LinkMonitorPolicy
          parentKeyVersion: 0.0.1
          parentLocalName: 'NULL'
          localName: LinkClearedState
        trigger:
          name: CreateLinkClearedOutfieldsEvent
          version: 0.0.1
        stateOutputs:
          entry:
            - key: LinkClearedLogic_Output_Direct
              value:
                key:
                  parentKeyName: LinkMonitorPolicy
                  parentKeyVersion: 0.0.1
                  parentLocalName: LinkClearedState
                  localName: LinkClearedLogic_Output_Direct
                outgoingEvent:
                  name: ApexMessageOutputEvent
                  version: 0.0.1
                nextState:
                  parentKeyName: 'NULL'
                  parentKeyVersion: 0.0.0
                  parentLocalName: 'NULL'
                  localName: 'NULL'
              contextAlbumReference: []
              taskSelectionLogic:
                key: 'NULL'
                logicFlavour: UNDEFINED
                logic: ''
              stateFinalizerLogicMap:
                entry: []
              defaultTask:
                name: LinkClearedTask
                version: 0.0.1
              taskReferences:
                entry:
                  - key:
                      name: LinkClearedTask
                      version: 0.0.1
                    value:
                      key:
                        parentKeyName: LinkMonitorPolicy
                        parentKeyVersion: 0.0.1
                        parentLocalName: LinkClearedState
                        localName: LinkClearedTask
                      outputType: DIRECT
                      output:
                        parentKeyName: LinkMonitorPolicy
                        parentKeyVersion: 0.0.1
                        parentLocalName: LinkClearedState
                        localName: LinkClearedLogic_Output_Direct
            - key: LinkFailureOrClearedState
              value:
                stateKey:
                  parentKeyName: LinkMonitorPolicy
                  parentKeyVersion: 0.0.1
                  parentLocalName: 'NULL'
                  localName: LinkFailureOrClearedState
                trigger:
                  name: LinkFailureInputEvent
                  version: 0.0.1
                stateOutputs:
                  entry:
                    - key: CreateLinkClearedOutfieldsLogic_Output_Direct
```

```

value:
  key:
    parentKeyName: LinkMonitorPolicy
    parentKeyVersion: 0.0.1
    parentLocalName: LinkFailureOrClearedState
    localName: CreateLinkClearedOutfieldsLogic_Output_Direct
  outgoingEvent:
    name: CreateLinkClearedOutfieldsEvent
    version: 0.0.1
  nextState:
    parentKeyName: LinkMonitorPolicy
    parentKeyVersion: 0.0.1
    parentLocalName: 'NULL'
    localName: LinkClearedState
- key: CreateLinkFailureOutfieldsLogic_Output_Direct
value:
  key:
    parentKeyName: LinkMonitorPolicy
    parentKeyVersion: 0.0.1
    parentLocalName: LinkFailureOrClearedState
    localName: CreateLinkFailureOutfieldsLogic_Output_Direct
  outgoingEvent:
    name: CreateLinkFailureOutfieldsEvent
    version: 0.0.1
  nextState:
    parentKeyName: LinkMonitorPolicy
    parentKeyVersion: 0.0.1
    parentLocalName: 'NULL'
    localName: LinkFailureState
- key: NoPolicyDefinedLogic_Output_Direct
value:
  key:
    parentKeyName: LinkMonitorPolicy
    parentKeyVersion: 0.0.1
    parentLocalName: LinkFailureOrClearedState
    localName: NoPolicyDefinedLogic_Output_Direct
  outgoingEvent:
    name: ApexMessageOutputEvent
    version: 0.0.1
  nextState:
    parentKeyName: 'NULL'
    parentKeyVersion: 0.0.0
    parentLocalName: 'NULL'
    localName: 'NULL'
contextAlbumReference: []
taskSelectionLogic:
  key: TaskSelectionLogic
  logicFlavour: JAVASCRIPT
  logic: |-
    /*
    *

```

=====LICENSE_START=====

```

    * Copyright (C) 2021 Nordix Foundation.
    *

```

=====

```

    * Licensed under the Apache License, Version 2.0 (the "License");
    * you may not use this file except in compliance with the License.
    * You may obtain a copy of the License at
    *
    *     http://www.apache.org/licenses/LICENSE-2.0
    *
    * Unless required by applicable law or agreed to in writing, software
    * distributed under the License is distributed on an "AS IS" BASIS,
    * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
    * See the License for the specific language governing permissions and
    * limitations under the License.
    *
    * SPDX-License-Identifier: Apache-2.0
    *

```

=====LICENSE_END=====

```

    */

```

```

executor.logger.info("Task Selection Execution: '"+executor.subject.id+
    "'. InputFields: '"+executor.inFields+"'");

var linkFailureInput = executor.inFields.get("LinkFailureInput");
var commonEventHeader = linkFailureInput.get("event").get("commonEventHeader");
var domain = commonEventHeader.get("domain");

taskFailure = executor.subject.getTaskKey("CreateLinkFailureOutfieldsTask");
taskCleared = executor.subject.getTaskKey("CreateLinkClearedOutfieldsTask");
taskDefault = executor.subject.getDefaultTaskKey();

if (domain == "fault") {
    var faultFields = linkFailureInput.get("event").get("faultFields");
    var alarmCondition = faultFields.get("alarmCondition");
    var eventSeverity = faultFields.get("eventSeverity");
    if (alarmCondition == "28" && eventSeverity != "NORMAL") {
        taskFailure.copyTo(executor.selectedTask);
    } else if (alarmCondition == "28" && eventSeverity == "NORMAL") {
        taskCleared.copyTo(executor.selectedTask);
    } else {
        taskDefault.copyTo(executor.selectedTask);
    }
} else {
    taskDefault.copyTo(executor.selectedTask);
}

true;
stateFinalizerLogicMap:
entry: []
defaultTask:
name: NoPolicyDefinedTask
version: 0.0.1
taskReferences:
entry:
- key:
    name: CreateLinkClearedOutfieldsTask
    version: 0.0.1
  value:
    key:
      parentKeyName: LinkMonitorPolicy
      parentKeyVersion: 0.0.1
      parentLocalName: LinkFailureOrClearedState
      localName: CreateLinkClearedOutfieldsTask
    outputType: DIRECT
    output:
      parentKeyName: LinkMonitorPolicy
      parentKeyVersion: 0.0.1
      parentLocalName: LinkFailureOrClearedState
      localName: CreateLinkClearedOutfieldsLogic_Output_Direct
- key:
    name: CreateLinkFailureOutfieldsTask
    version: 0.0.1
  value:
    key:
      parentKeyName: LinkMonitorPolicy
      parentKeyVersion: 0.0.1
      parentLocalName: LinkFailureOrClearedState
      localName: CreateLinkFailureOutfieldsTask
    outputType: DIRECT
    output:
      parentKeyName: LinkMonitorPolicy
      parentKeyVersion: 0.0.1
      parentLocalName: LinkFailureOrClearedState
      localName: CreateLinkFailureOutfieldsLogic_Output_Direct
- key:
    name: NoPolicyDefinedTask
    version: 0.0.1
  value:
    key:
      parentKeyName: LinkMonitorPolicy

```

```

        parentKeyVersion: 0.0.1
        parentLocalName: LinkFailureOrClearedState
        localName: NoPolicyDefinedTask
    outputType: DIRECT
    output:
        parentKeyName: LinkMonitorPolicy
        parentKeyVersion: 0.0.1
        parentLocalName: LinkFailureOrClearedState
        localName: NoPolicyDefinedLogic_Output_Direct
- key: LinkFailureState
  value:
    stateKey:
      parentKeyName: LinkMonitorPolicy
      parentKeyVersion: 0.0.1
      parentLocalName: 'NULL'
      localName: LinkFailureState
    trigger:
      name: CreateLinkFailureOutfieldsEvent
      version: 0.0.1
    stateOutputs:
      entry:
        - key: LinkFailureLogic_Output_Direct
          value:
            key:
              parentKeyName: LinkMonitorPolicy
              parentKeyVersion: 0.0.1
              parentLocalName: LinkFailureState
              localName: LinkFailureLogic_Output_Direct
            outgoingEvent:
              name: LinkFailureOutputEvent
              version: 0.0.1
            nextState:
              parentKeyName: 'NULL'
              parentKeyVersion: 0.0.0
              parentLocalName: 'NULL'
              localName: 'NULL'
          contextAlbumReference: []
        taskSelectionLogic:
          key: 'NULL'
          logicFlavour: UNDEFINED
          logic: ''
        stateFinalizerLogicMap:
          entry: []
        defaultTask:
          name: LinkFailureTask
          version: 0.0.1
        taskReferences:
          entry:
            - key:
                name: LinkFailureTask
                version: 0.0.1
              value:
                key:
                  parentKeyName: LinkMonitorPolicy
                  parentKeyVersion: 0.0.1
                  parentLocalName: LinkFailureState
                  localName: LinkFailureTask
                outputType: DIRECT
                output:
                  parentKeyName: LinkMonitorPolicy
                  parentKeyVersion: 0.0.1
                  parentLocalName: LinkFailureState
                  localName: LinkFailureLogic_Output_Direct
            firstState: LinkFailureOrClearedState
      tasks:
        key:
          name: LinkMonitorModel_Tasks
          version: 0.0.1
        taskMap:
          entry:
            - key:

```

```

name: CreateLinkClearedOutfieldsTask
version: 0.0.1
value:
  key:
    name: CreateLinkClearedOutfieldsTask
    version: 0.0.1
  inputFields:
    entry:
      - key: LinkFailureInput
        value:
          key: LinkFailureInput
          fieldSchemaKey:
            name: LinkFailureInputSchema
            version: 0.0.1
          optional: false
    outputFields:
      entry:
        - key: OruId
          value:
            key: OruId
            fieldSchemaKey:
              name: OruIdSchema
              version: 0.0.1
            optional: false
  taskParameters:
    entry: []
  contextAlbumReference: []
  taskLogic:
    key: TaskLogic
    logicFlavour: JAVASCRIPT
    logic: |-

```

```

/*
 * =====LICENSE_START=====
 * Copyright (C) 2021 Nordix Foundation.
 * =====
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 *
 * SPDX-License-Identifier: Apache-2.0
 * =====LICENSE_END=====
 */

```

```

        executor.logger.info("Task Execution: '"+executor.subject.id+"'. Input Fields:
        '"+executor.inFields+"'");

```

```

        var linkFailureInput = executor.inFields.get("LinkFailureInput");
        var oruId = linkFailureInput.get("event").get("commonEventHeader").get("sourceName");

        executor.outFields.put("OruId", oruId);

        executor.logger.info(executor.outFields);

```

```

        true;
- key:
  name: CreateLinkFailureOutfieldsTask
  version: 0.0.1
value:
  key:
    name: CreateLinkFailureOutfieldsTask
    version: 0.0.1
  inputFields:
    entry:

```

```

- key: LinkFailureInput
  value:
    key: LinkFailureInput
    fieldSchemaKey:
      name: LinkFailureInputSchema
      version: 0.0.1
    optional: false
outputFields:
  entry:
- key: OduId
  value:
    key: OduId
    fieldSchemaKey:
      name: OduIdSchema
      version: 0.0.1
    optional: false
- key: OruId
  value:
    key: OruId
    fieldSchemaKey:
      name: OruIdSchema
      version: 0.0.1
    optional: false
taskParameters:
  entry: []
contextAlbumReference: []
taskLogic:
  key: TaskLogic
  logicFlavour: JAVASCRIPT
  logic: |-
    /*
    * =====LICENSE_START=====
    * Copyright (C) 2021 Nordix Foundation.
    * =====
    * Licensed under the Apache License, Version 2.0 (the "License");
    * you may not use this file except in compliance with the License.
    * You may obtain a copy of the License at
    *
    *     http://www.apache.org/licenses/LICENSE-2.0
    *
    * Unless required by applicable law or agreed to in writing, software
    * distributed under the License is distributed on an "AS IS" BASIS,
    * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
    * See the License for the specific language governing permissions and
    * limitations under the License.
    *
    * SPDX-License-Identifier: Apache-2.0
    * =====LICENSE_END=====
    */

```

```

    executor.logger.info("Task Execution: '"+executor.subject.id+"'. Input Fields:
    '"+executor.inFields+"'");

```

```

    var returnValue = true;
    var linkFailureInput = executor.inFields.get("LinkFailureInput");
    var oruId = linkFailureInput.get("event").get("commonEventHeader").get("sourceName");
    var oruOduMap = JSON.parse(executor.parameters.get("ORU-ODU-Map"));

    if (oruId in oruOduMap) {
        var oduId = oruOduMap[oruId];
        executor.outFields.put("OruId", oruId);
        executor.outFields.put("OduId", oduId);
        executor.logger.info(executor.outFields);
    } else {
        executor.message = "No O-RU found in the config with this ID: " + oruId;
        returnValue = false;
    }

    returnValue;

- key:
  name: LinkClearedTask

```

```

version: 0.0.1
value:
  key:
    name: LinkClearedTask
    version: 0.0.1
  inputFields:
    entry:
      - key: OruId
        value:
          key: OruId
          fieldSchemaKey:
            name: OruIdSchema
            version: 0.0.1
            optional: false
    outputFields:
      entry:
        - key: message
          value:
            key: message
            fieldSchemaKey:
              name: MessageSchema
              version: 0.0.1
              optional: false
  taskParameters:
    entry: []
  contextAlbumReference: []
  taskLogic:
    key: TaskLogic
    logicFlavour: JAVASCRIPT
    logic: |-
      /*

```

```

      * =====LICENSE_START=====
      * Copyright (C) 2021 Nordix Foundation.
      * =====
      * Licensed under the Apache License, Version 2.0 (the "License");
      * you may not use this file except in compliance with the License.
      * You may obtain a copy of the License at
      *
      *     http://www.apache.org/licenses/LICENSE-2.0
      *
      * Unless required by applicable law or agreed to in writing, software
      * distributed under the License is distributed on an "AS IS" BASIS,
      * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
      * See the License for the specific language governing permissions and
      * limitations under the License.
      *
      * SPDX-License-Identifier: Apache-2.0
      * =====LICENSE_END=====
      */

```

```

      executor.logger.info("Task Execution: '"+executor.subject.id+"'. Input Fields:
      '"+executor.inFields+"'");

```

```

      var oruId = executor.inFields.get("OruId");

      executor.outFields.put("message", "CLEARED link failure for O-RU: " + oruId);

      executor.logger.info(executor.outFields);

```

```

      true;
- key:
  name: LinkFailureTask
  version: 0.0.1
  value:
    key:
      name: LinkFailureTask
      version: 0.0.1
    inputFields:
      entry:
        - key: OduId
          value:

```



```

        "radio_Dash_resource_Dash_management_Dash_policy_Dash_min_Dash_ratio":15,
        "user_Dash_label":"rrm-pol-1",
        "resource_Dash_type":"prb",
        "radio_Dash_resource_Dash_management_Dash_policy_Dash_dedicated_Dash_ratio":20,
        "administrative_Dash_state":"unlocked"
    });
}

linkFailureOutput.put
("o_Dash_ran_Dash_sc_Dash_du_Dash_hello_Dash_world_ColoN_radio_Dash_resource_Dash_management_Dash_policy_Dash_ra
tio", unlockMessageArray);
executor.outFields.put("LinkFailureOutput", linkFailureOutput.toString());

executor.getExecutionProperties().setProperty("OduId", oduId);
executor.getExecutionProperties().setProperty("OruId", oruId);

executor.logger.info(executor.outFields);

true;
- key:
  name: NoPolicyDefinedTask
  version: 0.0.1
value:
  key:
    name: NoPolicyDefinedTask
    version: 0.0.1
  inputFields:
    entry:
      - key: LinkFailureInput
        value:
          key: LinkFailureInput
          fieldSchemaKey:
            name: LinkFailureInputSchema
            version: 0.0.1
            optional: false
    outputFields:
      entry:
        - key: message
          value:
            key: message
            fieldSchemaKey:
              name: MessageSchema
              version: 0.0.1
              optional: false
    taskParameters:
      entry: []
    contextAlbumReference: []
  taskLogic:
    key: TaskLogic
    logicFlavour: JAVASCRIPT
    logic: |-
      /*
      * =====LICENSE_START=====
      * Copyright (C) 2021 Nordix Foundation.
      * =====
      * Licensed under the Apache License, Version 2.0 (the "License");
      * you may not use this file except in compliance with the License.
      * You may obtain a copy of the License at
      *
      *     http://www.apache.org/licenses/LICENSE-2.0
      *
      * Unless required by applicable law or agreed to in writing, software
      * distributed under the License is distributed on an "AS IS" BASIS,
      * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
      * See the License for the specific language governing permissions and
      * limitations under the License.
      *
      * SPDX-License-Identifier: Apache-2.0
      * =====LICENSE_END=====
      */

```

```

        executor.logger.info("Task Execution: '"+executor.subject.id+"'. Input Fields:
        '"+executor.inFields+"'");

        executor.outFields.put("message", "No policy defined for this event");

        executor.logger.info(executor.outFields);

        true;
events:
  key:
    name: LinkMonitorModel_Events
    version: 0.0.1
  eventMap:
    entry:
      - key:
          name: ApexMessageOutputEvent
          version: 0.0.1
        value:
          key:
            name: ApexMessageOutputEvent
            version: 0.0.1
          nameSpace: org.onap.policy.apex.auth.clieditor
          source: APEX
          target: APEX
          parameter:
            entry:
              - key: message
                value:
                  key: message
                  fieldSchemaKey:
                    name: MessageSchema
                    version: 0.0.1
                    optional: false
            - key:
                name: CreateLinkClearedOutfieldsEvent
                version: 0.0.1
              value:
                key:
                  name: CreateLinkClearedOutfieldsEvent
                  version: 0.0.1
                nameSpace: org.onap.policy.apex.auth.clieditor
                source: APEX
                target: APEX
                parameter:
                  entry:
                    - key: OruId
                      value:
                        key: OruId
                        fieldSchemaKey:
                          name: OruIdSchema
                          version: 0.0.1
                          optional: false
            - key:
                name: CreateLinkFailureOutfieldsEvent
                version: 0.0.1
              value:
                key:
                  name: CreateLinkFailureOutfieldsEvent
                  version: 0.0.1
                nameSpace: org.onap.policy.apex.auth.clieditor
                source: APEX
                target: APEX
                parameter:
                  entry:
                    - key: OduId
                      value:
                        key: OduId
                        fieldSchemaKey:
                          name: OduIdSchema
                          version: 0.0.1
                          optional: false

```

```

      - key: OruId
        value:
          key: OruId
          fieldSchemaKey:
            name: OruIdSchema
            version: 0.0.1
            optional: false
- key:
  name: LinkFailureInputEvent
  version: 0.0.1
  value:
    key:
      name: LinkFailureInputEvent
      version: 0.0.1
    nameSpace: org.onap.policy.apex.auth.clieditor
    source: DMAAP
    target: APEX
    parameter:
      entry:
        - key: LinkFailureInput
          value:
            key: LinkFailureInput
            fieldSchemaKey:
              name: LinkFailureInputSchema
              version: 0.0.1
              optional: false
- key:
  name: LinkFailureOutputEvent
  version: 0.0.1
  value:
    key:
      name: LinkFailureOutputEvent
      version: 0.0.1
    nameSpace: org.onap.policy.apex.auth.clieditor
    source: APEX
    target: OAM
    parameter:
      entry:
        - key: LinkFailureOutput
          value:
            key: LinkFailureOutput
            fieldSchemaKey:
              name: LinkFailureOutputSchema
              version: 0.0.1
              optional: false
schemas:
  key:
    name: LinkMonitorModel_Schemas
    version: 0.0.1
  schemas:
    entry:
      - key:
        name: LinkFailureInputSchema
        version: 0.0.1
      value:
        key:
          name: LinkFailureInputSchema
          version: 0.0.1
        schemaFlavour: Avro
        schemaDefinition: |-
          {
            "type": "record",
            "name": "Link_Failure_Input",
            "fields": [
              {
                "name": "event",
                "type": {
                  "type": "record",
                  "name": "Event_Type",
                  "fields": [
                    {

```

```
"name": "commonEventHeader",
"type": {
  "type": "record",
  "name": "Common_Event_Header_Type",
  "fields": [
    {
      "name": "domain",
      "type": "string"
    },
    {
      "name": "eventId",
      "type": "string"
    },
    {
      "name": "eventName",
      "type": "string"
    },
    {
      "name": "eventType",
      "type": "string"
    },
    {
      "name": "sequence",
      "type": "int"
    },
    {
      "name": "priority",
      "type": "string"
    },
    {
      "name": "reportingEntityId",
      "type": "string"
    },
    {
      "name": "reportingEntityName",
      "type": "string"
    },
    {
      "name": "sourceId",
      "type": "string"
    },
    {
      "name": "sourceName",
      "type": "string"
    },
    {
      "name": "startEpochMicrosec",
      "type": "string"
    },
    {
      "name": "lastEpochMicrosec",
      "type": "string"
    },
    {
      "name": "nfNamingCode",
      "type": "string"
    },
    {
      "name": "nfVendorName",
      "type": "string"
    },
    {
      "name": "timeZoneOffset",
      "type": "string"
    },
    {
      "name": "version",
      "type": "string"
    },
    {
      "name": "vesEventListenerVersion",
```

```

    "type": "string"
  }
}
],
{
  "name": "faultFields",
  "type": {
    "type": "record",
    "name": "Fault_Fields_Type",
    "fields": [
      {
        "name": "faultFieldsVersion",
        "type": "string"
      },
      {
        "name": "alarmCondition",
        "type": "string"
      },
      {
        "name": "alarmInterfaceA",
        "type": "string"
      },
      {
        "name": "eventSourceType",
        "type": "string"
      },
      {
        "name": "specificProblem",
        "type": "string"
      },
      {
        "name": "eventSeverity",
        "type": "string"
      },
      {
        "name": "vfStatus",
        "type": "string"
      },
      {
        "name": "alarmAdditionalInformation",
        "type": {
          "type": "record",
          "name": "Alarm_Additional_Information_Type",
          "fields": [
            {
              "name": "eventTime",
              "type": "string"
            },
            {
              "name": "equipType",
              "type": "string"
            },
            {
              "name": "vendor",
              "type": "string"
            },
            {
              "name": "model",
              "type": "string"
            }
          ]
        }
      }
    ]
  }
},
]
}
}
}
}
}
```

```

    }
    - key:
      name: LinkFailureOutputSchema
      version: 0.0.1
    value:
      key:
        name: LinkFailureOutputSchema
        version: 0.0.1
        schemaFlavour: Avro
        schemaDefinition: "{\n  \"name\": \"Link_Failure_Output\",\n  \"type\": \"record\",\n  \"fields\": [\n    {\n      \"name\": \"o_DasH_ran_DasH_sc_DasH_du_DasH_hello_DasH_world_ColoN_radio_DasH_resource_DasH_management_DasH_policy_DasH_ratio\",\n      \"type\": {\n        \"type\": \"array\",\n        \"items\": {\n          \"name\": \"o_DasH_ran_DasH_sc_DasH_du_DasH_hello_DasH_world_ColoN_radio_DasH_resource_DasH_management_DasH_policy_DasH_ratio_record\",\n          \"type\": \"record\",\n          \"fields\": [\n            {\n              \"name\": \"id\",\n              \"type\": \"string\",\n              \"fields\": {\n                \"name\": \"radio_DasH_resource_DasH_management_DasH_policy_DasH_max_DasH_ratio\",\n                \"type\": \"int\"\n              },\n              {\n                \"name\": \"radio_DasH_resource_DasH_management_DasH_policy_DasH_members\",\n                \"type\": {\n                  \"type\": \"array\",\n                  \"items\": {\n                    \"name\": \"radio_DasH_resource_DasH_management_DasH_policy_DasH_members_record\",\n                    \"type\": \"record\",\n                    \"fields\": [\n                      {\n                        \"name\": \"mobile_DasH_country_DasH_code\",\n                        \"type\": \"string\",\n                        \"fields\": {\n                          \"name\": \"mobile_DasH_network_DasH_code\",\n                          \"type\": \"string\",\n                          \"fields\": {\n                            {\n                              \"name\": \"slice_DasH_differentiator\",\n                              \"type\": \"int\"\n                            },\n                            {\n                              \"name\": \"slice_DasH_service_DasH_type\",\n                              \"type\": \"int\"\n                            }\n                          },\n                          {\n                            \"name\": \"radio_DasH_resource_DasH_management_DasH_policy_DasH_min_DasH_ratio\",\n                            \"type\": \"int\"\n                          },\n                          {\n                            \"name\": \"user_DasH_label\",\n                            \"type\": \"string\"\n                          },\n                          {\n                            \"name\": \"resource_DasH_type\",\n                            \"type\": \"string\"\n                          },\n                          {\n                            \"name\": \"radio_DasH_resource_DasH_management_DasH_policy_DasH_dedicated_DasH_ratio\",\n                            \"type\": \"int\"\n                          },\n                          {\n                            \"name\": \"administrative_DasH_state\",\n                            \"type\": \"string\"\n                          }\n                        }\n                      },\n                      {\n                        \"name\": \"administrative_DasH_state\",\n                        \"type\": \"string\"\n                      }\n                    }\n                  }\n                }\n              }\n            }\n          }\n        }\n      }\n    }\n  ]\n}"
      - key:
        name: MessageSchema
        version: 0.0.1
      value:
        key:
          name: MessageSchema
          version: 0.0.1
          schemaFlavour: Java
          schemaDefinition: java.lang.String
        - key:
          name: OduIdSchema
          version: 0.0.1
        value:
          key:
            name: OduIdSchema
            version: 0.0.1
            schemaFlavour: Java
            schemaDefinition: java.lang.String
          - key:
            name: OruIdSchema
            version: 0.0.1
          value:
            key:
              name: OruIdSchema
              version: 0.0.1
              schemaFlavour: Java
              schemaDefinition: java.lang.String
eventOutputParameters:
  RestProducer:
    carrierTechnologyParameters:
      carrierTechnology: RESTCLIENT
      parameterClassName: org.onap.policy.apex.plugins.event.carrier.restclient.
RestClientCarrierTechnologyParameters
  parameters:
    url: http://sdnr-simulator.nonrtric:9990/rests/data/network-topology:network-topology

```

```

/topology=topology-netconf/node={OduId}/yang-ext:mount/o-ran-sc-du-hello-world:network-function/distributed-
unit-functions={OduId}/radio-resource-management-policy-ratio=rrm-pol-1
  httpMethod: PUT
  httpHeaders:
    - - Authorization
    - Basic YWRtaW46S3A4Yko0U1hzek0wV1hsaGFrm2VibGNzZTJnQXc4NHZhb0dHbUp2VXkyVQ==
  eventProtocolParameters:
    eventProtocol: JSON
    parameters:
      pojoField: LinkFailureOutput
  eventNameFilter: LinkFailureOutputEvent
StdOutProducer:
  carrierTechnologyParameters:
    carrierTechnology: FILE
    parameters:
      standardIo: true
  eventProtocolParameters:
    eventProtocol: JSON
    parameters:
      pojoField: message
  eventNameFilter: ApexMessageOutputEvent
eventInputParameters:
  DMaaPConsumer:
    carrierTechnologyParameters:
      carrierTechnology: RESTCLIENT
      parameterClassName: org.onap.policy.apex.plugins.event.carrier.restclient.
RestClientCarrierTechnologyParameters
  parameters:
    url: http://message-router:3904/events/unauthenticated.SEC_FAULT_OUTPUT/users/link-monitor-
nonrtric?timeout=15000&limit=100
  eventProtocolParameters:
    eventProtocol: JSON
    parameters:
      versionAlias: version
      pojoField: LinkFailureInput
  eventName: LinkFailureInputEvent

```

NOTE: The default hostname/port for sdnr-simulator and message-router are specified in lines 1547 and 1573 respectively of the above file. They should be replaced with actual values if using different hostname/port.

After commissioning the above toscatemplate, control loop can be instantiated using the steps described in previous sub-section. Once the control loop is in RUNNING state, the below steps can be done to test the correct working of the apex policy.

- First of all, deploy the sdnr-simulator in the cluster (if not using the real SDNR in ONAP). The sdnr simulator can be found in the nonrtric repo of OSC.

```

git clone "https://gerrit.o-ran-sc.org/r/nonrtric"
git checkout -b e-release --track origin/e-release

cd nonrtric/test/usecases/oruclosedlooprecovery/scriptversion/helm/sdnr-simulator/
helm package .

helm install sdnr-simulator sdnr-simulator-0.1.0.tgz --set image.repository=registry.nordix.org/onap/sdnr-
simulator --set image.tag=1.0.0 --set messagerouter.host="http://message-router.onap" --set messagerouter.port="
3904" --namespace nonrtric --create-namespace --wait

```

- In order to make sure that the apex policy has been deployed successfully, the REST APIs for policy-pap and policy-api components can be used. However, these components do not expose the NodePorts. Hence, a NodePort needs to be opened for accessing each of these APIs.

```

kubectl expose deployment def-policy-pap --type=NodePort --name=policy-pap-public

kubectl expose deployment def-policy-api --type=NodePort --name=policy-api-public

```

- Find the NodePort numbers allocated in the cluster for these two components.

```
kubectl -n onap get svc | grep policy-pap-public

kubectl -n onap get svc | grep policy-api-public
```

- Making this REST call to the policy-api component should return the deployed policy.

```
curl -k -u 'policyadmin:zb!XztG34' -X GET "https://<NodeIP>:<NodePort-policy-api>/policy/api/v1/policytypes
/onap.policies.controlloop.operational.common.Apex/versions/1.0.0/policies/operational.apex.linkmonitor/versions
/1.0.0"
```

- The status of deployed policy can be checked by making a REST call to policy-pap component.

```
curl -k -u 'policyadmin:zb!XztG34' -X GET "https://<NodeIP>:<NodePort-policy-pap>/policy/pap/v1/policies/status"
```

The above command should show a state of "SUCCESS" for the LinkMonitor policy.

- Finally, to test that the apex policy is actually working, an example LinkFailureEvent can be sent to the Dmaap MR.

```
cd nonrtrc/test/usecases/oruclosedlooprecovery/apexpolicyversion/LinkMonitor

curl -k -X POST -H accept:application/json -H Content-Type:application/json "https://<NodeIP>:<NodePort-message-
router>/events/unauthenticated.SEC_FAULT_OUTPUT/" -d @./events/LinkFailureEvent.json
```

The logs of the sdnr-simulator should show that a PUT request has been successfully received.

"PUT /rests/data/network-topology:network-topology/topology=topology-netconf/node=HCL-O-DU-1123/yang-ext:mount/o-ran-sc-du-hello-world:network-function/du-to-ru-connection=ERICSSON-O-RU-11225 HTTP/1.1" 200

b) Control loop for script version

This sub-section describes the steps required for bringing up the control loop with script version of the usecase. The tosca template to be used for commissioning this control loop is given below. The steps for commissioning are depicted in the sub-section [Commission/Instantiate control loop via GUI](#).

commission.yaml

```
tosca_definitions_version: tosca_simple_yaml_1_1_0
data_types:
  onap.datatypes.ToscaConceptIdentifier:
    derived_from: tosca.datatypes.Root
    properties:
      name:
        type: string
        required: true
      version:
        type: string
        required: true
node_types:
  org.onap.policy.clamp.controlloop.Participant:
    version: 1.0.1
    derived_from: tosca.nodetypes.Root
    properties:
      provider:
        type: string
        required: false
  org.onap.policy.clamp.controlloop.ControlLoop:
    version: 1.0.1
    derived_from: tosca.nodetypes.Root
    properties:
      provider:
        type: string
        required: false
    elements:
```



```

    type: list
    required: true
    entry_schema:
      type: onap.datatypes.ToscaConceptIdentifier
org.onap.policy.clamp.controlloop.ControlLoopElement:
  version: 1.0.1
  derived_from: tosca.nodetypes.Root
  properties:
    provider:
      type: string
      required: false
    participant_id:
      type: onap.datatypes.ToscaConceptIdentifier
      required: true
org.onap.policy.clamp.controlloop.K8SMicroserviceControlLoopElement:
  version: 1.0.1
  derived_from: org.onap.policy.clamp.controlloop.ControlLoopElement
  properties:
    chart:
      type: string
      required: true
    configs:
      type: list
      required: false
    requirements:
      type: string
      required: false
    templates:
      type: list
      required: false
      entry_schema:
    values:
      type: string
      required: true
topology_template:
  node_templates:
    org.onap.domain.linkmonitor.LinkMonitorControlLoopDefinition1:
      version: 1.2.3
      type: org.onap.policy.clamp.controlloop.ControlLoop
      type_version: 1.0.1
      description: Control loop for Link Monitor
      properties:
        provider: Ericsson
        elements:
          - name: org.onap.domain.linkmonitor.OruAppK8SMicroserviceControlLoopElement
            version: 1.2.3
          - name: org.onap.domain.linkmonitor.MessageGeneratorK8SMicroserviceControlLoopElement
            version: 1.2.3
          - name: org.onap.domain.linkmonitor.SdnrSimulatorK8SMicroserviceControlLoopElement
            version: 1.2.3
    org.onap.k8s.controlloop.K8SControlLoopParticipant:
      version: 2.3.4
      type: org.onap.policy.clamp.controlloop.Participant
      type_version: 1.0.1
      description: Participant for k8s
      properties:
        provider: ONAP
org.onap.domain.linkmonitor.OruAppK8SMicroserviceControlLoopElement:
  version: 1.2.3
  type: org.onap.policy.clamp.controlloop.K8SMicroserviceControlLoopElement
  type_version: 1.0.1
  description: Control loop element for oru-app
  properties:
    provider: ONAP
    participant_id:
      name: K8sParticipant0
      version: 1.0.0
    participantType:
      name: org.onap.k8s.controlloop.K8SControlLoopParticipant
      version: 2.3.4
    chart:

```

```

    chartId:
      name: oru-app
      version: 0.1.0
    releaseName: oru-app
    repository:
      repoName: chartmuseum
    namespace: nonrtric
    overrideParams:
      image.repository: nexus3.o-ran-sc.org:10002/o-ran-sc/nonrtric-o-ru-closed-loop-recovery
      image.tag: 1.0.1
      messagerouter.host: http://message-router.onap
      messagerouter.port: 3904
      sdnr.host: http://sdnr-simulator
      sdnr.port: 9990
org.onap.domain.linkmonitor.MessageGeneratorK8SMicroserviceControlLoopElement:
  version: 1.2.3
  type: org.onap.policy.clamp.controlloop.K8SMicroserviceControlLoopElement
  type_version: 1.0.1
  description: Control loop element for message-generator
  properties:
    provider: ONAP
    participant_id:
      name: K8sParticipant0
      version: 1.0.0
    participantType:
      name: org.onap.k8s.controlloop.K8SControlLoopParticipant
      version: 2.3.4
  chart:
    chartId:
      name: message-generator
      version: 0.1.0
    releaseName: message-generator
    repository:
      repoName: chartmuseum
    namespace: nonrtric
    overrideParams:
      image.repository: registry.nordix.org/onap/message-generator
      image.tag: 1.0.0
      messagerouter.host: http://message-router.onap
      messagerouter.port: 3904
org.onap.domain.linkmonitor.SdnrSimulatorK8SMicroserviceControlLoopElement:
  version: 1.2.3
  type: org.onap.policy.clamp.controlloop.K8SMicroserviceControlLoopElement
  type_version: 1.0.1
  description: Control loop element for sdnr-simulator
  properties:
    provider: ONAP
    participant_id:
      name: K8sParticipant0
      version: 1.0.0
    participantType:
      name: org.onap.k8s.controlloop.K8SControlLoopParticipant
      version: 2.3.4
  chart:
    chartId:
      name: sdnr-simulator
      version: 0.1.0
    releaseName: sdnr-simulator
    repository:
      repoName: chartmuseum
    namespace: nonrtric
    overrideParams:
      image.repository: registry.nordix.org/onap/sdnr-simulator
      image.tag: 1.0.0
      messagerouter.host: http://message-router.onap
      messagerouter.port: 3904

```

This control loop will bring up three micro-services in the nonrtrc namespace: oru-app (running the actual logic of the usecase), message-generator (sending the LinkFailure messages at random intervals), and sdnr-simulator (for receiving the REST calls made by oru-app). Make sure that the sdnr-simulator is not already running in the nonrtrc namespace, otherwise the control loop instantiation might fail.

NOTE: The default hostname/port for sdnr and message-router are specified in **overrideParams** of the above file. They should be replaced with actual values if using different hostname/port.f

Before commissioning this toasca template, some preparations need to be done in the kubernetes-participant component of the clamp.

- First step is to copy the kube config file of the cluster inside the kubernetes-participant. Find the pod-name of this component using:

```
kubectl -n onap get pod | grep k8s-ppnt
```

Copy the config file using this command:

```
kubectl cp ~/.kube/config onap/<POD-NAME-k8s-ppnt>:/home/policy/.kube/config
```

In order to make sure that the kubernetes-participant is properly configured, get into the pod using "kubectl -n onap exec -it <POD-NAME-k8s-ppnt> sh" and run the following command:

```
kubectl get ns
```

This should show all the namespaces in the cluster where ONAP is deployed.

- Next step is to copy the helm charts of all three components into the kubernetes-participant. The helm charts are located in the nonrtrc repo of OSC.

```
cd nonrtrc/test/usecases/oruclosedlooprecovery/scriptversion/helm/sdnr-simulator/
helm package .
kubectl cp ./sdnr-simulator-0.1.0.tgz onap/<POD-NAME-k8s-ppnt>:/home/policy/local-charts/sdnr-simulator-0.1.0.tgz

cd nonrtrc/test/usecases/oruclosedlooprecovery/scriptversion/helm/message-generator/
helm package .
kubectl cp ./message-generator-0.1.0.tgz onap/<POD-NAME-k8s-ppnt>:/home/policy/local-charts/message-generator-0.1.0.tgz

cd nonrtrc/test/usecases/oruclosedlooprecovery/scriptversion/helm/oru-app/
helm package .
kubectl cp ./oru-app-0.1.0.tgz onap/<POD-NAME-k8s-ppnt>:/home/policy/local-charts/oru-app-0.1.0.tgz
```

- Finally, install chartmuseum into the kubernetes-participant and push the above helm charts into it. Get into the pod using "kubectl -n onap exec -it <POD-NAME-k8s-ppnt> sh" and run the following commands:

```
mkdir -p ~/helm3-storage
curl -LO https://s3.amazonaws.com/chartmuseum/release/latest/bin/linux/amd64/chartmuseum
chmod +x ./chartmuseum
./chartmuseum --storage local --storage-local-rootdir /home/policy/helm3-storage -port 8080 &

curl --data-binary "@local-charts/sdnr-simulator-0.1.0.tgz" http://localhost:8080/api/charts
curl --data-binary "@local-charts/message-generator-0.1.0.tgz" http://localhost:8080/api/charts
curl --data-binary "@local-charts/oru-app-0.1.0.tgz" http://localhost:8080/api/charts

helm repo add chartmuseum http://localhost:8080
helm repo update
```

Once the kubernetes-participant is set up, the toasca template can be commissioned. After that, the control loop can be instantiated using the steps described in the sub-section [Commission/Instantiate control loop via GUI](#). Once the control loop is in RUNNING state, check that all three micro-services have been created in the nonrtrc namespace.

```
kubect1 -n nonrtric get pod
```

In order to test the correct working of the usecase, check logs in each of the three components. There should be messages flowing in this order:

message-generator oru-app sdnr-simulator

Control loops in docker

This section is related to running the control loops in a docker environment. Separate docker-compose files are available in the nonrtric repo of OSC for bringing up the apex policy as well as the script versions of the usecase.

a) Control loop for apex policy version

This sub-section describes the steps for running the control loop for apex policy version of the usecase using docker.

- The first step is to clone the nonrtric repo and start the Dmaap message-router. Then, two topics are created in the message-router: **POLICY-CLRUNTIME-PARTICIPANT** (to be used by controlloop-runtime component of policy/clamp) and **unauthenticated.SEC_FAULT_OUTPUT** (for handling fault notification events).

```
git clone "https://gerrit.o-ran-sc.org/r/nonrtric"
git checkout e-release --track origin/e-release

cd nonrtric/test/auto-test
./startMR.sh remote docker --env-file ../common/test_env-oran-e-release.sh

docker rename message-router onap-dmaap

curl -X POST -H "Content-Type: application/json" -d '{"topicName": "POLICY-CLRUNTIME-PARTICIPANT"}'
http://localhost:3904/events/POLICY-CLRUNTIME-PARTICIPANT

curl -X POST -H "Content-Type: application/json" -d '{"topicName": "unauthenticated.SEC_FAULT_OUTPUT"}'
http://localhost:3904/events/unauthenticated.SEC_FAULT_OUTPUT
```

- After creating the topics in the message-router, start the ONAP Policy Framework using the docker-compose file available in nonrtric repo.

```
cd nonrtric/docker-compose/docker-compose-policy-framework
docker-compose up -d
```

- The next step is to start the controlloop-runtime and policy-participant components of the clamp.

```
cd nonrtric/test/usecases/oruclosedlooprecovery/apexpolicyversion/LinkMonitor/docker-compose-controlloop
docker-compose up -d
```

Check the logs of policy-participant using the command "docker logs -f policy-participant" and wait until these messages start appearing in the logs:

"com.att.nsa.apiClient.http.HttpClient : --> HTTP/1.1 200 OK"

- Once all the components get up and running, the control loop can be commissioned and instantiated. This can be done by making a REST call to the controlloop-runtime component of the clamp. The toasca template for commissioning and the instantiation payload are provided in this directory of the nonrtric repo:

```
cd nonrtric/test/usecases/oruclosedlooprecovery/apexpolicyversion/LinkMonitor/controlloop-rest-payloads
```

Commission the toasca template using this REST call:

```
curl -X POST -k -u 'healthcheck:zb!XztG34' -H Content-Type:application/yaml https://localhost:6969/onap/controlloop/v2/commission/ --data-binary @commission.yaml
```

It should give the following response:

```
{
  "errorDetails": null,
  "affectedControlLoopDefinitions": [
    {
      "name": "org.onap.domain.linkmonitor.LinkMonitorPolicyControlLoopElement",
      "version": "1.2.3"
    },
    {
      "name": "org.onap.domain.linkmonitor.LinkMonitorControlLoopDefinition0",
      "version": "1.2.3"
    },
    {
      "name": "org.onap.policy.controlloop.PolicyControlLoopParticipant",
      "version": "2.3.1"
    }
  ]
}
```

Make the following REST call to instantiate the control loop:

```
curl -X POST -k -u 'healthcheck:zb!XztG34' -H Content-Type:application/json https://localhost:6969/onap/controlloop/v2/instantiation/ --data-binary @instantiation.json
```

It should give the following response:

```
{
  "errorDetails": null,
  "affectedControlLoops": [
    {
      "name": "LinkMonitorInstance0",
      "version": "1.0.1"
    }
  ]
}
```

Change the control loop from default UNINITIALISED state to PASSIVE using the following REST call:

```
curl -X PUT -k -u 'healthcheck:zb!XztG34' -H Content-Type:application/json https://localhost:6969/onap/controlloop/v2/instantiation/command/ --data-binary @instantiation-command.json
```

It should give the same response as above.

Next step is to change the control loop from PASSIVE to RUNNING state. Edit the "instantiation-command.json" file and replace PASSIVE with RUNNING. Making the above REST call once again will change the control loop to RUNNING state.

- Once the control loop is in RUNNING state, check whether the apex policy has been deployed successfully in the policy framework. Making the below REST call to policy-api component should return the deployed policy.

```
curl -u 'healthcheck:zb!XztG34' -X GET "http://localhost:6869/policy/api/v1/policytypes/onap.policies.controlloop.operational.common.Apex/versions/1.0.0/policies/operational.apex.linkmonitor/versions/1.0.0"
```

Make the below REST call to policy-pap component and make sure that it returns a state of "SUCCESS" for the deployed policy.

```
curl -u 'healthcheck:zb!XztG34' -X GET "http://localhost:6868/policy/pap/v1/policies/status"
```

- Start the sdnr-simulator in a docker container that will receive the REST call made by apex policy when a link failure event is received.

```
docker run --rm --name sdnr-sim --network nonrtic-docker-net -e MR-HOST="http://onap-dmaap" -e MR-PORT="3904" registry.nordix.org/onap/sdnr-simulator:1.0.0
```

- Send the example link failure event.

```
cd nonrtic/test/usecases/oruclosedlooprecovery/apexpolicyversion/LinkMonitor

curl -X POST -H accept:application/json -H Content-Type:application/json "http://localhost:3904/events/unauthenticated.SEC_FAULT_OUTPUT/" -d @./events/LinkFailureEvent.json
```

The logs of sdnr-simulator should show that the following REST call is received:

```
"PUT /rests/data/network-topology:network-topology/topology=topology-netconf/node=HCL-O-DU-1123/yang-ext:mount/o-ran-sc-du-hello-world:network-function/du-to-ru-connection=ERICSSON-O-RU-11225 HTTP/1.1" 200 -
```

- In order to stop the docker containers and free up resources on the host machine, use the following commands:

```
cd nonrtric/docker-compose/docker-compose-policy-framework
docker-compose down

cd nonrtric/test/usecases/oruclosedlooprecovery/apexpolicyversion/LinkMonitor/docker-compose-controlloop
docker-compose down

docker stop sdnr-sim
docker rm sdnr-sim

docker volume rm docker-compose-policy-framework_db-vol
```

b) Control loop for script version

This sub-section describes the steps for running the control loop for script version of the usecase using docker. This version of the control loop will bring up four micro-services in the nonrtric namespace: oru-app (running the actual logic of the usecase), message-generator (sending the LinkFailure messages at random intervals), sdnr-simulator (for receiving the REST calls made by oru-app), and dmaap-mr (a message-router stub where the LinkFailure messages will be sent).

NOTE: The below instructions refer to bringing up the micro-services in a minikube cluster on the host machine, and it is assumed that the minikube is already up and running. The instructions should be modified accordingly when using a different environment.

- The first step is to clone the nonrtric repo and start the Dmaap message-router. Then, a topic named **POLICY-CLRUNTIME-PARTICIPANT** is created in the message-router (to be used by controlloop-runtime component of policy/clamp).

```
git clone "https://gerrit.o-ran-sc.org/r/nonrtric"
git checkout e-release --track origin/e-release

cd nonrtric/test/auto-test
./startMR.sh remote docker --env-file ../common/test_env-oran-e-release.sh

docker rename message-router onap-dmaap

curl -X POST -H "Content-Type: application/json" -d '{"topicName": "POLICY-CLRUNTIME-PARTICIPANT"}'
http://localhost:3904/events/POLICY-CLRUNTIME-PARTICIPANT
```

- Build a docker image for each of the four micro-services and make it available for use inside the minikube. Open a new terminal window (keep it separate and do not run any other commands except the ones given below) and run the following commands:

```
eval $(minikube docker-env)

cd nonrtric/test/usecases/oruclosedlooprecovery/scriptversion/app
docker build -t oru-app .

cd nonrtric/test/usecases/oruclosedlooprecovery/scriptversion/simulators
docker build -f Dockerfile-sdnr-sim -t sdnr-simulator .
docker build -f Dockerfile-message-generator -t message-generator:v2 .

cd nonrtric/test/mrstub/
docker build -t mrstub .
```

Make sure that all four docker images have been successfully created by running the "docker images" command.

- Next step is to prepare the kube config file of minikube for mounting it inside the k8s-participant component of policy/clamp. First of all, copy the kube config file inside the config directory used by docker-compose file that runs k8s-participant.

```
cd nonrtric/test/usecases/oruclosedlooprecovery/scriptversion/docker-compose-controlloop

cp ~/.kube/config ./config/kube-config
```

Open the copied kube-config file (located at nonrtric/test/usecases/oruclosedlooprecovery/scriptversion/docker-compose-controlloop/config/kube-config) and make the following changes:

1. replace everything under "cluster" with these two lines:
server: <https://host.docker.internal:443>
2. replace `<PORT>` with the port in original kube-config file before editing (i.e., before doing the above step)
3. replace last two lines in the file with:

```
client-certificate: /home/policy/.minikube/profiles/minikube/client.crt
```

```
client-key: /home/policy/.minikube/profiles/minikube/client.key
```

- Open the docker-compose file (located at `nontritic/test/usecases/oruclosedlooprecovery/scriptversion/docker-compose-controlloop/docker-compose.yml`) and replace the last line under volumes of `k8s-participant` with these two lines:

```
- ./config/kube-config:/home/policy/.kube/config:ro
- ~/.minikube/profiles/minikube:/home/policy/.minikube/profiles/minikube
```

- Start all the components using this docker-compose file:

```
docker-compose up -d
```

Check the logs of `k8s-participant` using the command `"docker logs -f k8s-participant"` and wait until these messages start appearing in the logs:

```
"com.att.nsa.apiClient.http.HttpClient : --> HTTP/1.1 200 OK"
```

- Once all the components get up and running, the control loop can be commissioned and instantiated. This can be done by making a REST call to the `controlloop-runtime` component of the `clamp`. The `tosca` template for commissioning and the instantiation payload are provided in this directory of the `nontritic` repo:

```
cd nontritic/test/usecases/oruclosedlooprecovery/scriptversion/controlloop-rest-payloads
```

Commission the `tosca` template using this REST call:

```
curl -X POST -k -u 'healthcheck:zb!XztG34' -H Content-Type:application/yaml https://localhost:6969/onap/controlloop/v2/commission/ --data-binary @commission.yaml
```

It should give the following response:

```
{"errorDetails":null,"affectedControlLoopDefinitions":[{"name":"org.onap.domain.linkmonitor.LinkMonitorControlLoopDefinition1","version":"1.2.3"}, {"name":"org.onap.k8s.controlloop.K8SControlLoopParticipant","version":"2.3.4"}, {"name":"org.onap.domain.linkmonitor.OruAppK8SMicroserviceControlLoopElement","version":"1.2.3"}, {"name":"org.onap.domain.linkmonitor.MessageGeneratorK8SMicroserviceControlLoopElement","version":"1.2.3"}, {"name":"org.onap.domain.linkmonitor.SdnrSimulatorK8SMicroserviceControlLoopElement","version":"1.2.3"}, {"name":"org.onap.domain.linkmonitor.DmaapMrK8SMicroserviceControlLoopElement","version":"1.2.3"}]}
```

Make the following REST call to instantiate the control loop:

```
curl -X POST -k -u 'healthcheck:zb!XztG34' -H Content-Type:application/json https://localhost:6969/onap/controlloop/v2/instantiation/ --data-binary @instantiation.json
```

It should give the following response:

```
{"errorDetails":null,"affectedControlLoops":[{"name":"LinkMonitorInstance1","version":"1.0.1"}]}
```

Change the control loop from default `UNINITIALISED` state to `PASSIVE` using the following REST call:

```
curl -X PUT -k -u 'healthcheck:zb!XztG34' -H Content-Type:application/json https://localhost:6969/onap/controlloop/v2/instantiation/command/ --data-binary @instantiation-command.json
```

It should give the same response as above.

Next step is to change the control loop from PASSIVE to RUNNING state. Edit the "instantiation-command.json" file and replace PASSIVE with RUNNING. Making the above REST call once again will change the control loop to RUNNING state.

- Once the control loop is in RUNNING state, check that all four micro-services have been created in the nonrtric namespace.

```
kubectl -n nonrtric get pod
```

In order to test the correct working of the usecase, check logs in each of the four components. There should be messages flowing in this order:

message-generator dmaap-mr oru-app sdnr-simulator

- In order to stop the docker containers and free up resources on the host machine, use the following commands:

```
cd nonrtric/test/usecases/oruclosedlooprecovery/scriptversion/docker-compose-controlloop
docker-compose down

docker volume rm docker-compose-controlloop_db-vol
```