Control Loop Sub Committee - Beijing Integration Testing Plan

clamp-onap-bejing-cl-take2.mp4

Testing Assumptions

A single VES Collector will be pre-deployed for all flows

DCAE project will 'hand-craft' a blueprint for TCA deployment using Kubernetes

SDC (DCAE-Design Studio) will not generate the blueprint programmatically

CLAMP will deploy an instance of TCA microservice for each use case (vCPE, vFirewall, etc.)

CLAMP will create the initial configuration policy for TCA and make any subsequent updates to it through API to Policy

Initial TCA configuration policy and subsequent updates will be sent by policy to DCAE policy handler

TCA microservice will process updates to its configuration policy

TCA microservice will not be capable of processing multiple configuration policies

Preparation for Testing

DCAE

Deploy VES Blueprint to DCAE which will deploy the VES collector

Generate TCA Blueprint

Confirm validity of previously-generated (and uploaded to Policy) TCA Policy model

Policy

Make sure that policy model for TCA is already uploaded to Policy GUI - this should be done as part of initial Policy container deployment

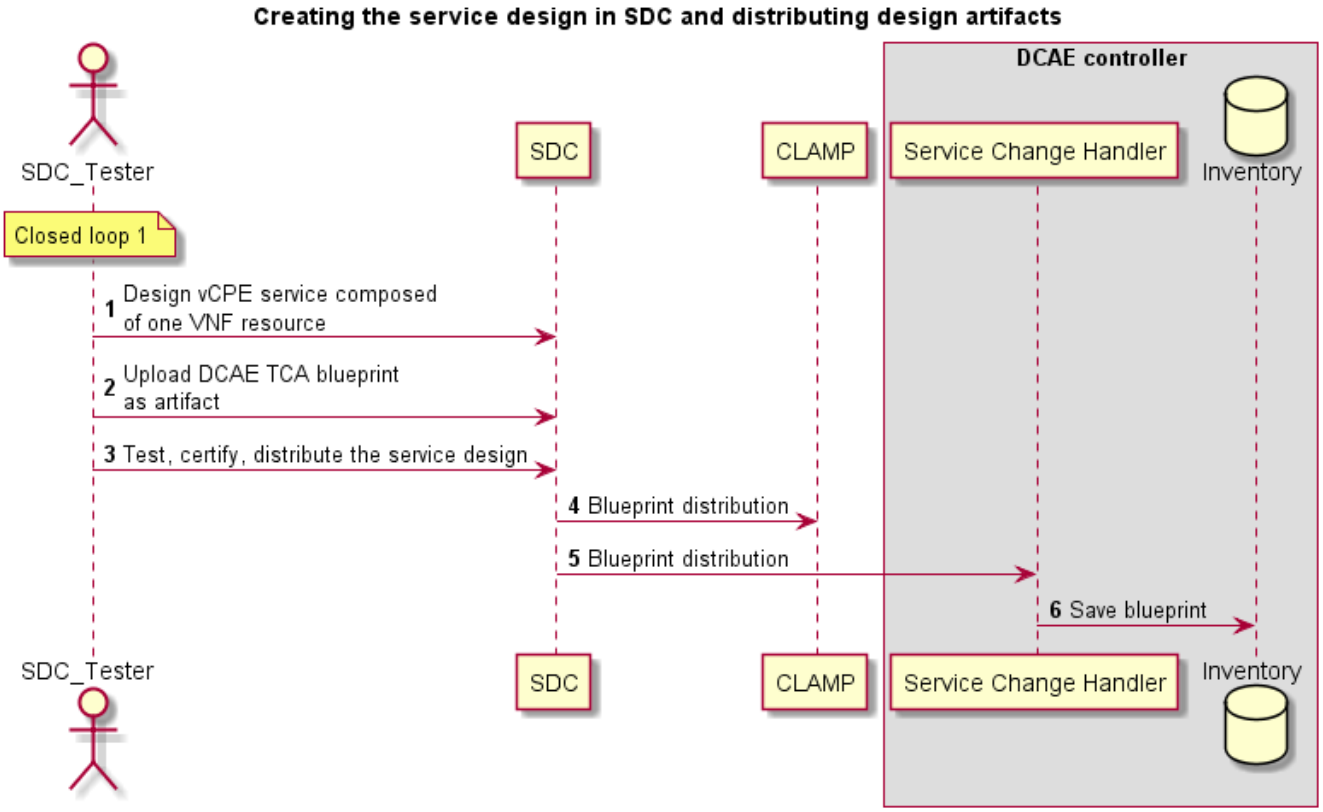
CLAMP

Store policy model for TCA in local repository

Testing Flow

This sequence of flows below will be repated for all services being tested (vCPE, vFirewall, vDNS)

Flow 1: Design and Distribute First Control Loop



UML Code for Flow 1

```
@startuml
title Creating the service design in SDC and distributing design artifacts
actor SDC_Tester
participant SDC
participant CLAMP
box "DCAE controller"
participant "Service Change Handler" as SCH
database Inventory
end box
autonumber
note over SDC_Tester: Closed loop 1
SDC_Tester -> SDC : Design vCPE service composed\nof one VNF resource
SDC_Tester -> SDC : Upload DCAE TCA blueprint\nas artifact
SDC_Tester -> SDC : Test, certify, distribute the service design
SDC -> CLAMP : Blueprint distribution
SDC -> SCH : Blueprint distribution
SCH -> Inventory : Save blueprint
@enduml
```

Testing Directions

Log into SDC as designer (cs0008)

Create a service

In Composition, create a resource instance in the service

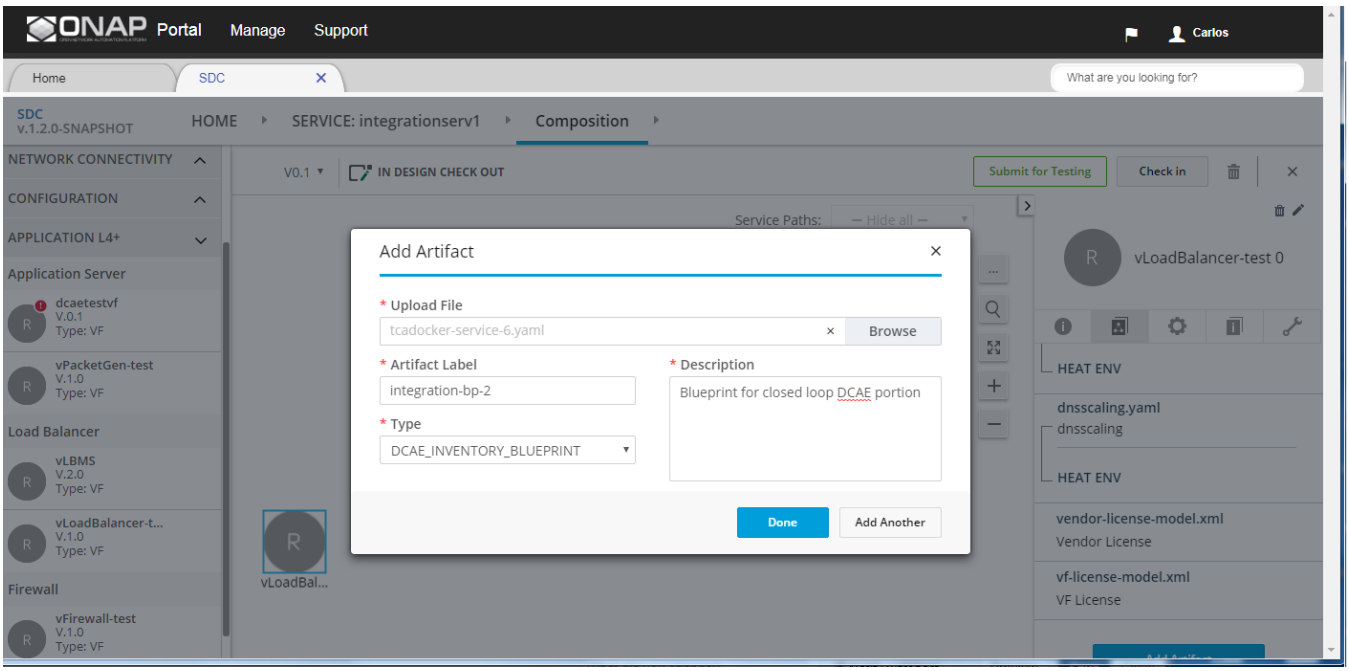
On the composition canvas, click on the resource instance

On the panel on the right, click on the second tab (Deployment Artifacts)

Click "Add Artifact"

Assign values to the artifacts as in the below screenshot, and upload the provided blueprint.

Blueprint: tcadocker-service_v3_demo.yaml



Submit the Service for Testing

Log in as a Tester user (jm0007)

Test the Service and Approve It

Log in as a Governance User (gv0001)

Approve Service for Distribution

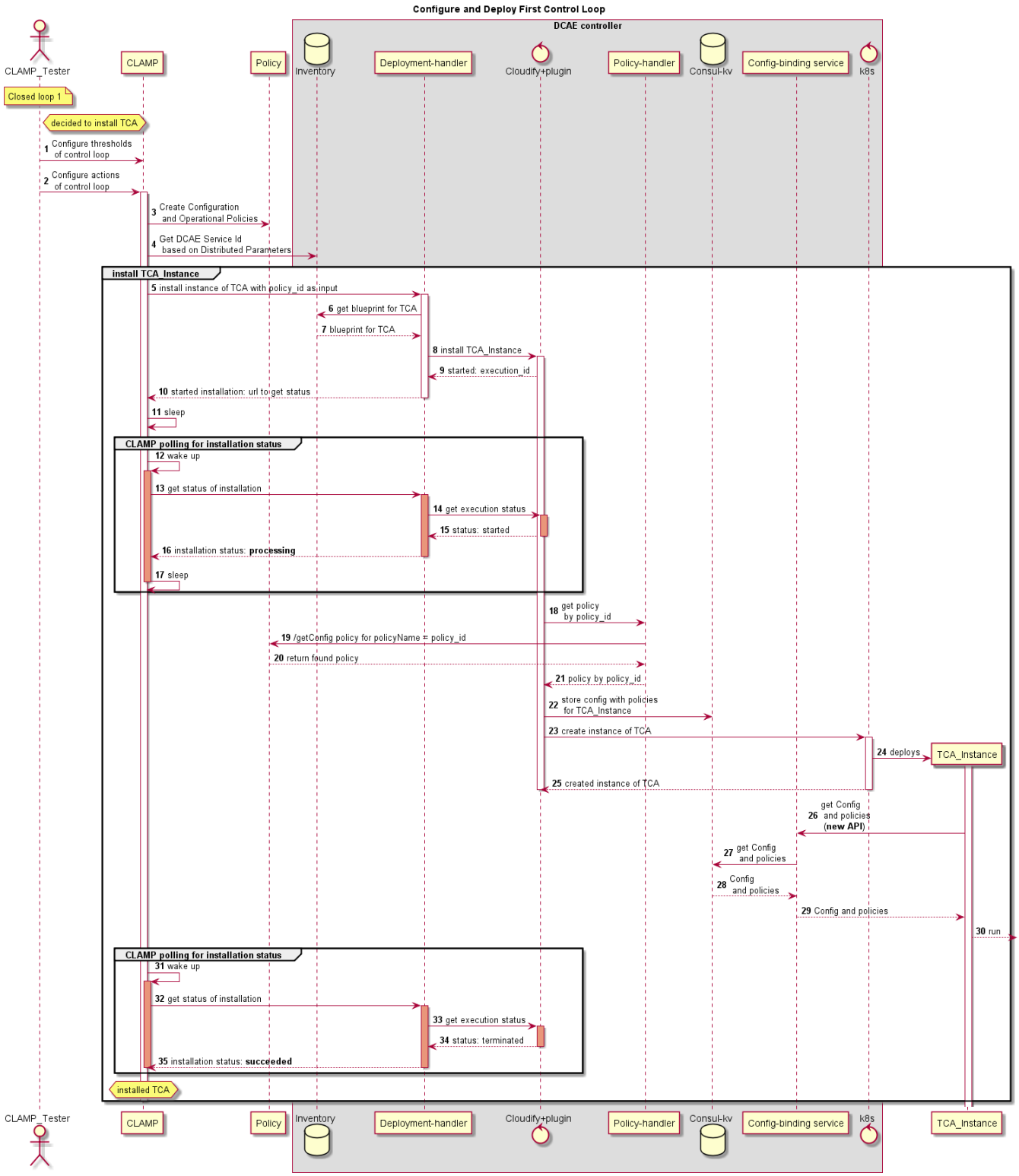
Log in as Operations User (op0001)

Distribute Service

Monitor Service to see that both DCAE and Clamp clients have successfully deployed the artifact

Step Range	Description	Status	Notes
1-3	Upload artifact and distribute	Tested	
4	Process distribution in CLAMP	Tested	
5-6	Process distribution in DCAE	Tested	

Flow 2: Configure and Deploy First Control Loop



Step Range	Description	Status	Notes
1-3	CLAMP Creates Policies	Tested with workaround	Has been tested, but outstanding bug on Policy, which requires manual workaround on Policy GUI ~~POLICY-775~~ - CLAMP Create Configuration Policy fails CLOSED - fixed based on POLICY-777

4-11	CLAMP Starts deployment in DCAE	Tested	
12-17	CLAMP starts getting deployment status	Tested	
18-22	DCAE gets and stores TCA policy		
23-25	DCAE deploys TCA	Tested	
26-30	TCA gets Policy		
31-35	CLAMP gets final status	Tested	Fix required for successful status to be passed back - ~~DCAEGEN2-482~~ - Multi-port component deployment does not become healthy CLOSED - this is done

UML Code for Flow 2

```
@startuml
title Configure and Deploy First Control Loop
actor CLAMP_Tester
participant CLAMP
participant Policy
box "DCAE controller"
database Inventory
participant "Deployment-handler" as DH
control "Cloudify+plugin" as Cloudify
participant "Policy-handler" as PH
database "Consul-kv" as consul
participant "Config-binding service" as CBS
control k8s
end box
participant TCA_Instance
autonumber
note over CLAMP_Tester: Closed loop 1
hnote right CLAMP_Tester: decided to install TCA
CLAMP_Tester -> CLAMP : Configure thresholds\n of control loop
CLAMP_Tester -> CLAMP : Configure actions\n of control loop
activate CLAMP
CLAMP -> Policy : Create Configuration\n and Operational Policies
CLAMP -> Inventory : Get DCAE Service Id\n based on Distributed Parameters
group install TCA_Instance
CLAMP -> DH : install instance of TCA with policy_id as input
activate DH
DH -> Inventory : get blueprint for TCA
Inventory --> DH : blueprint for TCA
DH -> Cloudify : install TCA_Instance
activate Cloudify
Cloudify --> DH : started: execution_id
DH --> CLAMP : started installation: url to get status
deactivate DH
CLAMP -> CLAMP : sleep
group CLAMP polling for installation status
    CLAMP -> CLAMP : wake up
    activate CLAMP #DarkSalmon
    CLAMP -> DH : get status of installation
    activate DH #DarkSalmon
    DH -> Cloudify : get execution status
    activate Cloudify #DarkSalmon
    Cloudify --> DH : status: started
    deactivate Cloudify
    DH --> CLAMP : installation status: **processing**
    deactivate DH
    CLAMP -> CLAMP : sleep
    deactivate CLAMP
end group
Cloudify -> PH : get policy\n by policy_id
```

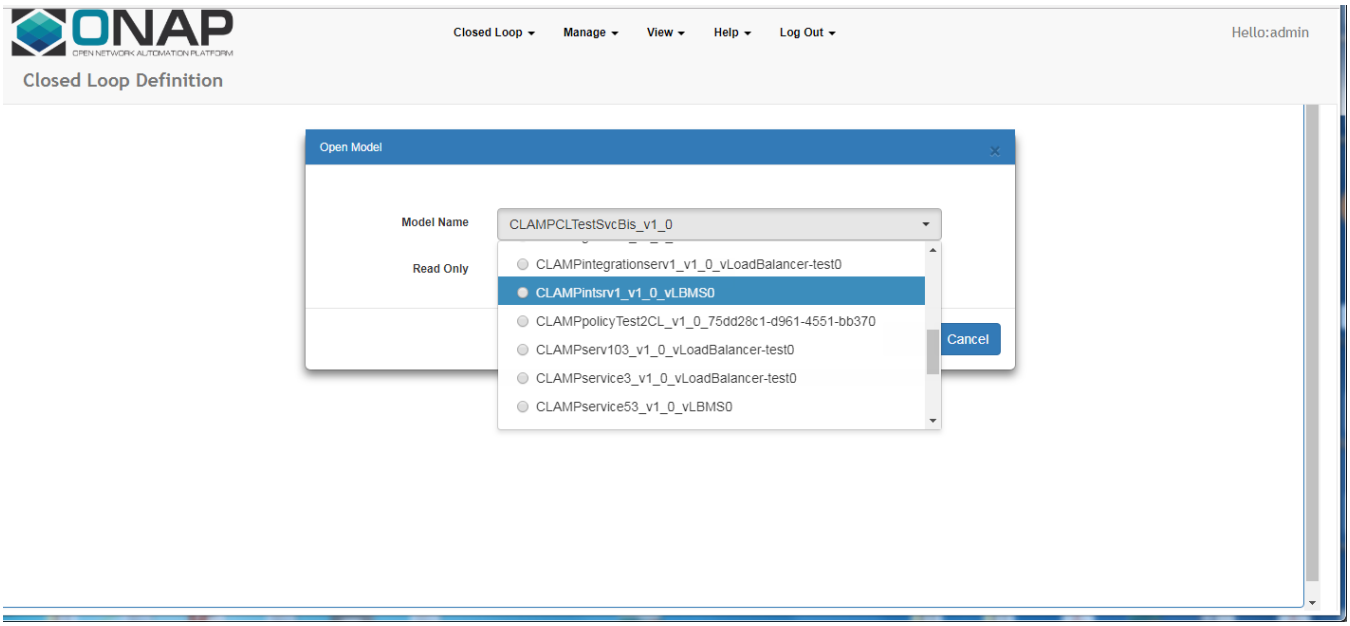
```
PH -> Policy : /getConfig policy for policyName = policy_id
Policy --> PH : return found policy
PH --> Cloudify : policy by policy_id
Cloudify -> consul: store config with policies\n for TCA_Instance
Cloudify -> k8s: create instance of TCA
activate k8s
create TCA_Instance
k8s -> TCA_Instance: deploys
activate TCA_Instance
k8s --> Cloudify: created instance of TCA
deactivate k8s
deactivate Cloudify
TCA_Instance -> CBS: get Config\n and policies\n (**new API**)
CBS -> consul: get Config\n and policies
consul --> CBS: Config\n and policies
CBS --> TCA_Instance: Config and policies
TCA_Instance -->]: run
group CLAMP polling for installation status
    CLAMP -> CLAMP : wake up
    activate CLAMP #DarkSalmon
    CLAMP -> DH : get status of installation
    activate DH #DarkSalmon
    DH -> Cloudify : get execution status
    activate Cloudify #DarkSalmon
    Cloudify --> DH : status: terminated
    deactivate Cloudify
    DH --> CLAMP : installation status: **succeeded**
    deactivate DH
    deactivate CLAMP
end group
hnote over CLAMP: installed TCA
deactivate CLAMP
end group
@enduml
```

Testing Directions

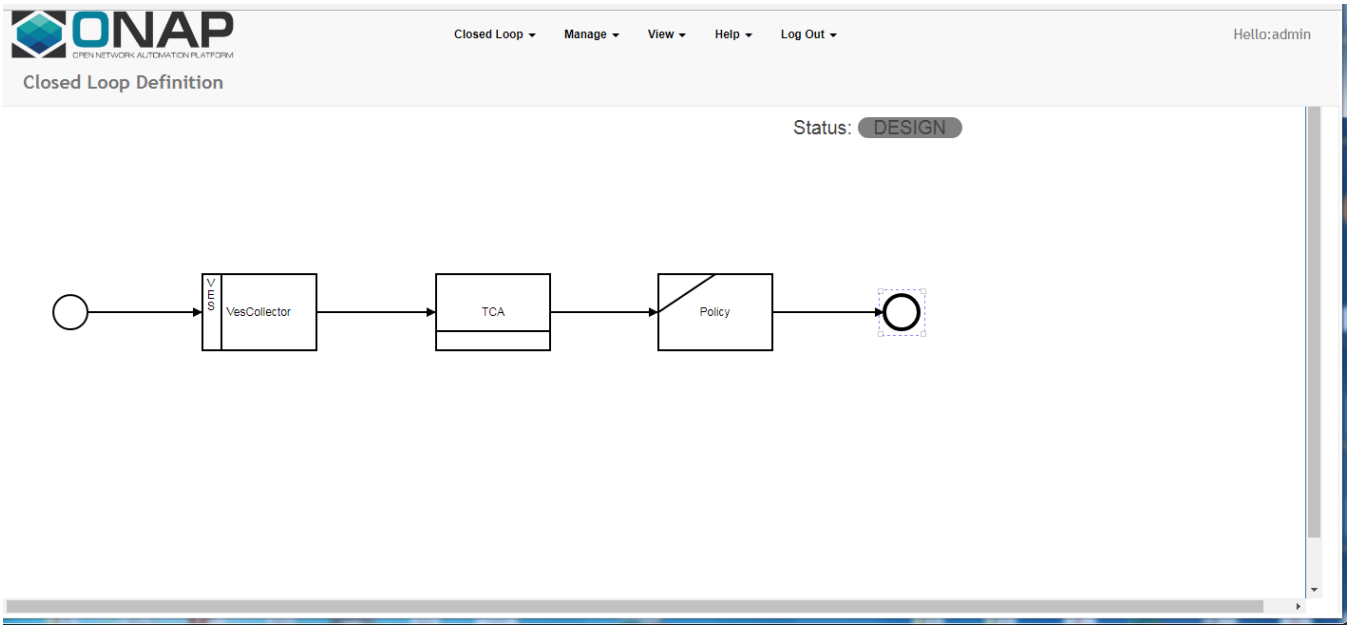
Log into CLAMP using credentials admin/password

Select Open CL from Closed Loop Menu

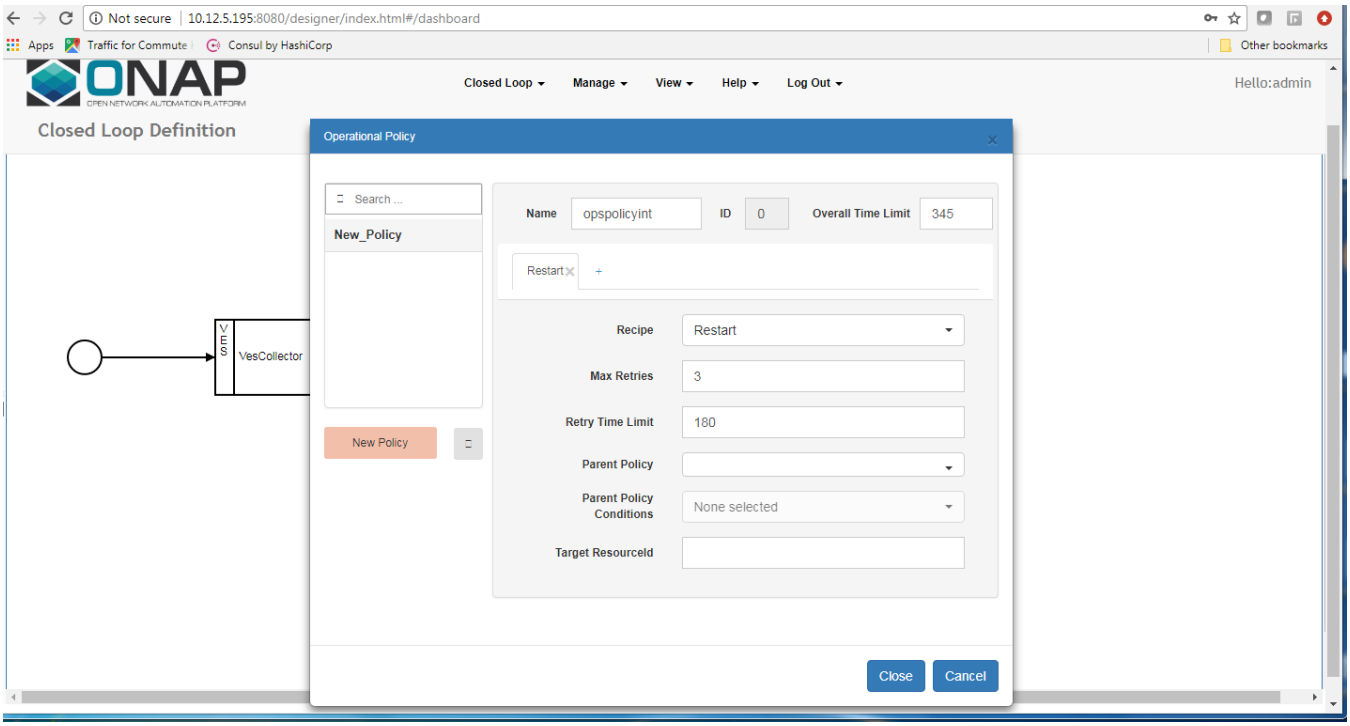
You will find the Closed Loop model that has been distributed from SDC. Its name has the form: CLAMP + <Service name> + <version> + <resource name>. For example, below there is a closed loop model for serivce 'intsrv1', version 1.0 and resource vLBMS0.



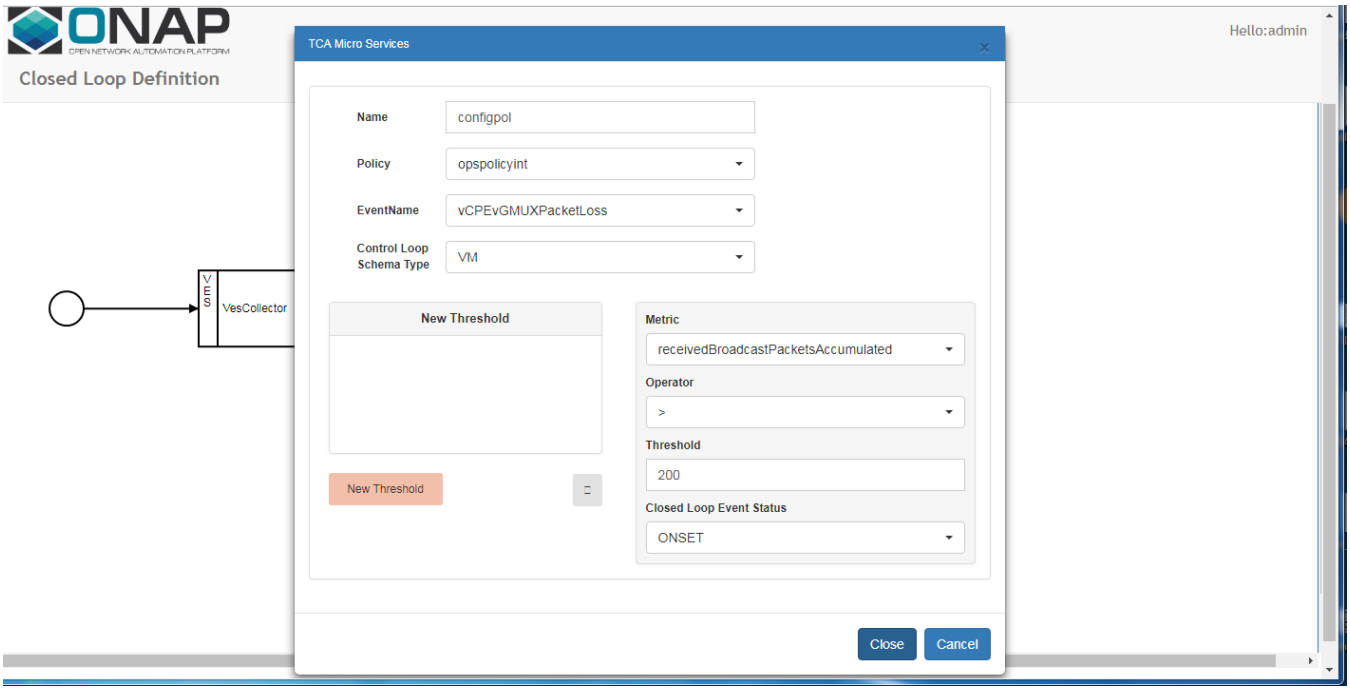
This will bring up a view of the control loop model. This allows you to create the TCA configuration policy and Operational Policy.



Fill in the details of the Operational Policy by clicking on Policy



Fill in the details of the Configuration Policy by clicking on TCA



Choose Save CL from Closed Loop Menu

Choose Submit from Manage Menu

The Status will change to "Distributed"

Choose Deploy from Manage Menu

The deploy window provides a JSON of inputs to provide to the deploy call:

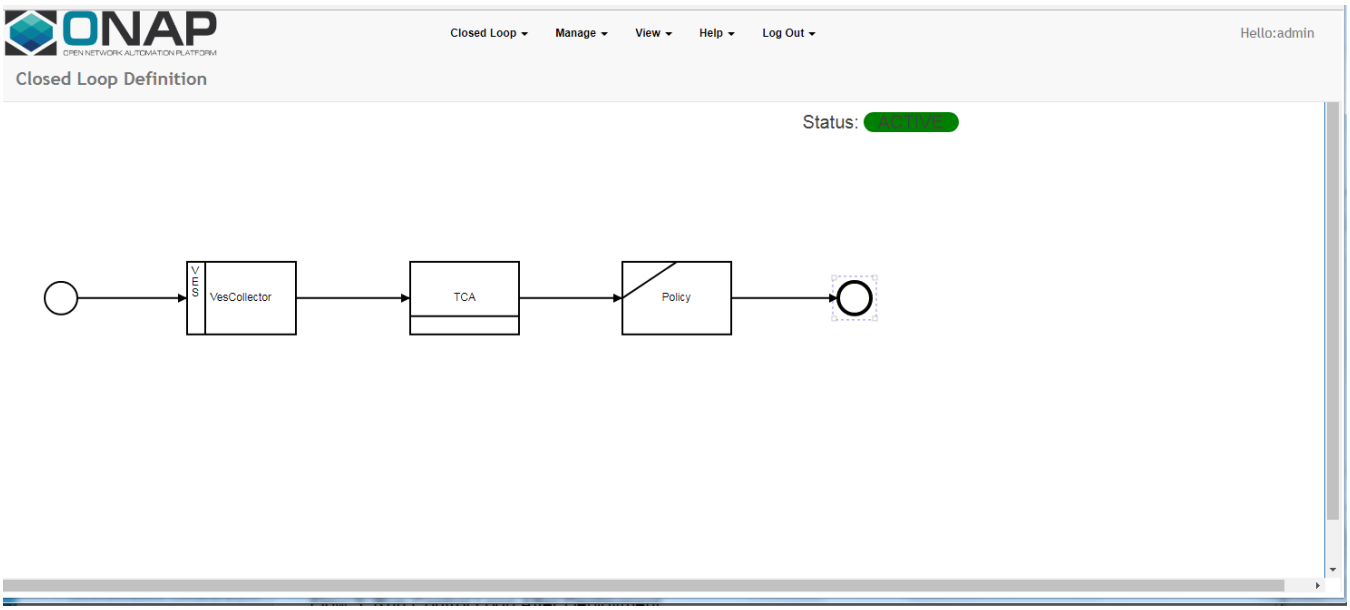
- The policyId field should be kept as-is
- The rest of the JSON object should be replaced by the following. external_port and scn_name need to be unique across existing deployments of TCA microservice

inputs

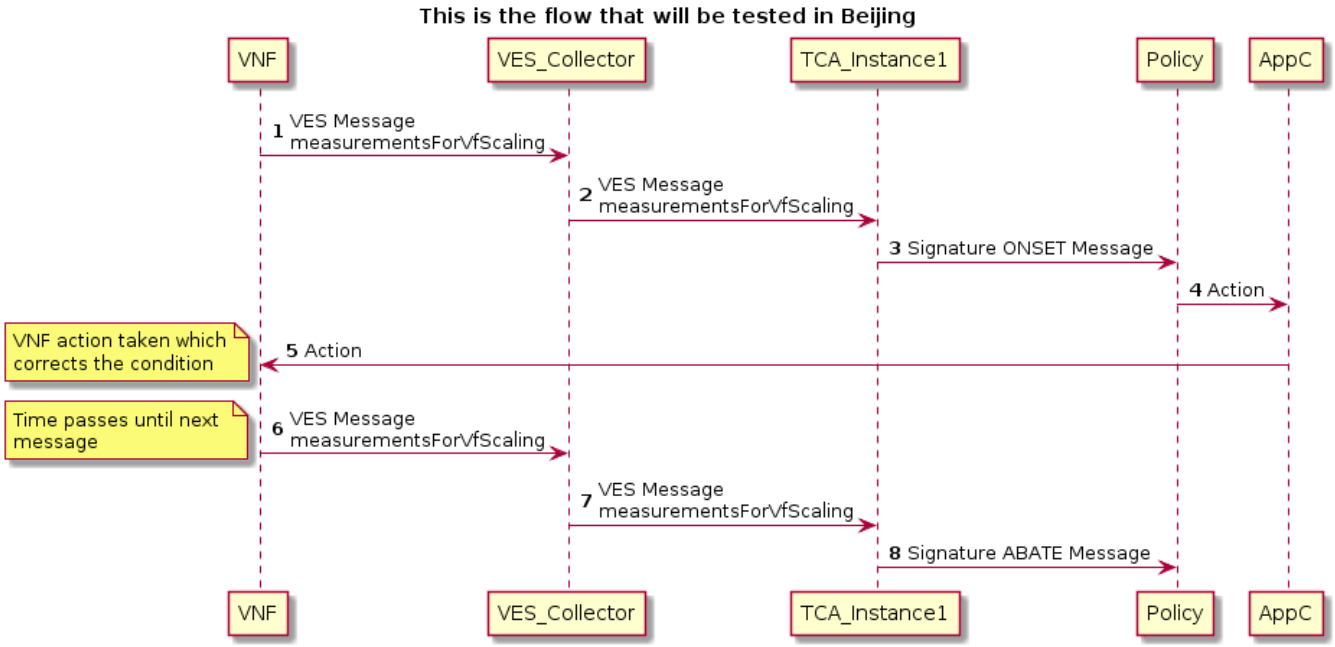
```
,"aaiEnrichmentHost":"aai","aaiEnrichmentPort":"30233","enableAAIEnrichment":"true","enableRedisCaching":"
false","dmaap_host":"10.12.5.127","dmaap_port":"3904","redisHosts":"na","consul_host":"10.12.5.130","
consul_port":"8500","cbs_host":"config-binding-service","cbs_port":"10000","tag_version":"nexus3.onap.org:10001
/onap/org.onap.dcaegen2.deployments.tca-cdap-container:1.0.0","dh_override":"dockerhost","dh_location_id":"
zone1","scn_name":"dcaegen2-analytics_tca_clampinstance_936","external_port":"32138"}
```

This may take on the order of tens of seconds, as it waits for DCAE to report final success.

Once the Deployment is successful in DCAE, Status will change to "Active"



Flow 3: Run Control Loop After Deployment



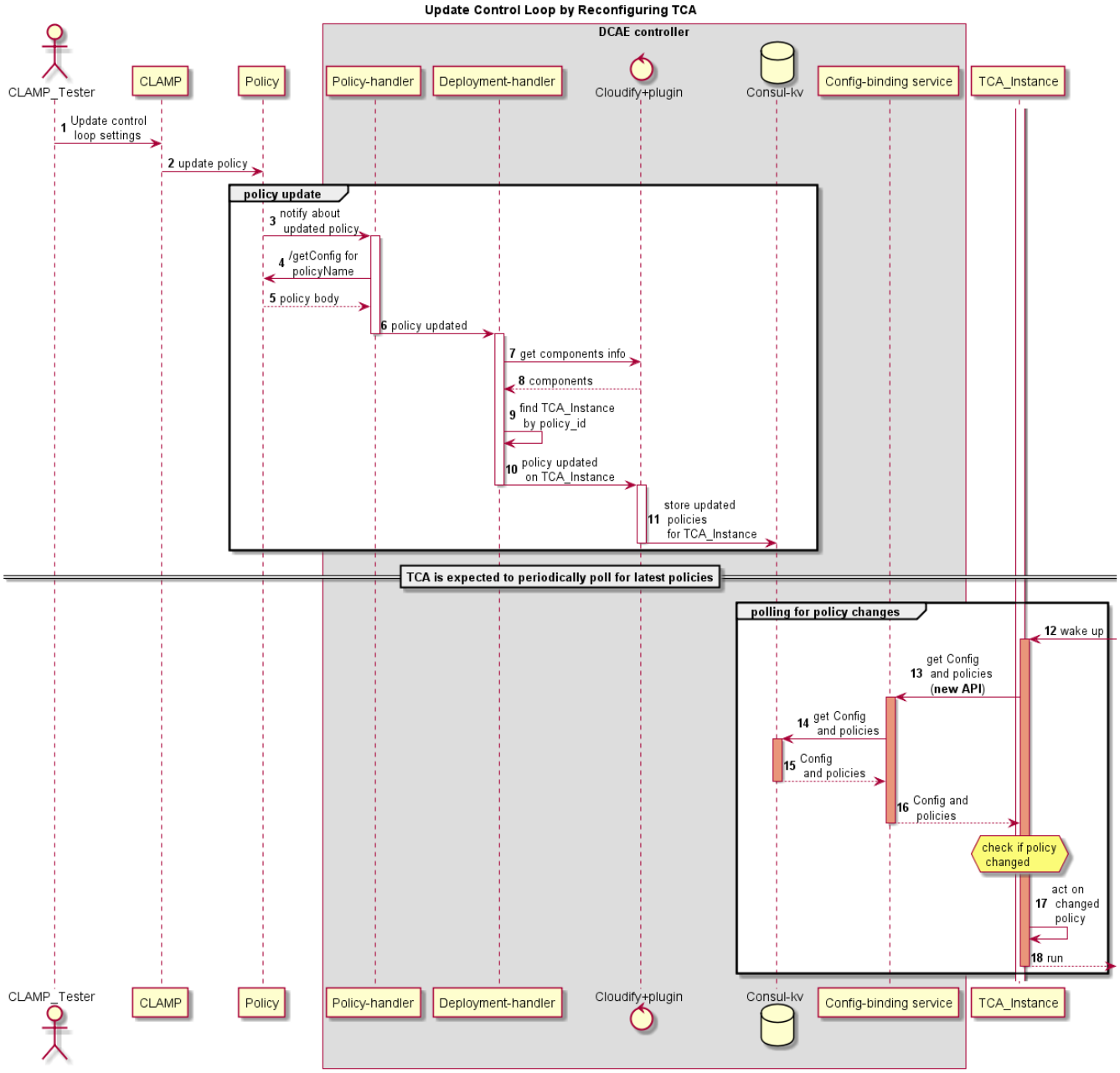
UML Code for Flow 3

```
@startuml
title This is the flow that will be tested in Beijing
participant VNF
participant VES_Collector
participant TCA_Instance1
participant Policy
participant AppC
autonumber
VNF -> VES_Collector : VES Message\nmeasurementsForVfScaling
VES_Collector -> TCA_Instance1 : VES Message\nmeasurementsForVfScaling
TCA_Instance1 -> Policy : Signature ONSET Message
Policy -> AppC : Action
AppC -> VNF : Action
note left
VNF action taken which
corrects the condition
end note
VNF -> VES_Collector : VES Message\nmeasurementsForVfScaling
note left
Time passes until next
message
end note
VES_Collector -> TCA_Instance1 : VES Message\nmeasurementsForVfScaling
TCA_Instance1 -> Policy : Signature ABATE Message
@enduml
```

Step Range	Description	Status	Notes
1-5	Onset after deploying CL		App-C bug on receiving RESET request - ~~APPC-900~~ - APPC returns error for vCPE restart message from Policy CLOSED
6-8	Abate after deploying CL		

Flow 4: Update Control Loop by Reconfiguring TCA

Reconfigure


Update Control Loop by Reconfiguring TCA
DCAE controller
CLAMP_Tester
CLAMP
Policy
Policy-handler
Deployment-handler
Cloudify+plugin
Consul-kv
Config-binding service
TCA_Instance
1 Update control loop settings
2 update policy
policy update
3 notify about updated policy
4 /getConfig for policyName
5 policy body
6 policy updated
7 get components info
8 components
9 find TCA_Instance by policy_id
10 policy updated on TCA_Instance
11 store updated policies for TCA_Instance
TCA is expected to periodically poll for latest policies
polling for policy changes
12 wake up
13 get Config and policies (new API)
14 get Config and policies
15 Config and policies
16 Config and policies
check if policy changed
17 act on changed policy
18 run

UML Code for Flow 4

```
@startuml
title Update Control Loop by Reconfiguring TCA
actor CLAMP_Tester
participant CLAMP
participant Policy
box "DCAE controller"
participant "Policy-handler" as PH
participant "Deployment-handler" as DH
control "Cloudify+plugin" as Cloudify
database "Consul-kv" as consul
participant "Config-binding service" as CBS
end box
participant TCA_Instance
autonumber
CLAMP_Tester -> CLAMP : Update control\n loop settings
CLAMP -> Policy: update policy
group policy update
    Policy -> PH : notify about\n updated policy
    activate PH
    PH -> Policy : /getConfig for\n policyName
    Policy --> PH : policy body
    PH -> DH : policy updated
    deactivate PH
    activate DH
    DH -> Cloudify : get components info
    Cloudify --> DH : components
    DH -> DH : find TCA_Instance\n by policy_id
    DH -> Cloudify : policy updated\n on TCA_Instance
    deactivate DH
    activate Cloudify
    Cloudify -> consul: store updated\n policies\n for TCA_Instance
    deactivate Cloudify
end
==TCA is expected to periodically poll for latest policies==
group polling for policy changes
    activate TCA_Instance
    TCA_Instance <-]: wake up
    activate TCA_Instance #DarkSalmon
    TCA_Instance -> CBS: get Config\n and policies\n (**new API**)
    activate CBS #DarkSalmon
    CBS -> consul: get Config\n and policies
    activate consul #DarkSalmon
    consul --> CBS: Config\n and policies
    deactivate consul
    CBS --> TCA_Instance: Config and\n policies
    deactivate CBS
    hnote over TCA_Instance: check if policy\n changed
    TCA_Instance -> TCA_Instance: act on\n changed\n policy
    TCA_Instance -->]: run
    deactivate TCA_Instance
end
@enduml
```

Run Control Loop Again

After the control loop is reconfigured, we test that the changes have taken effect. This is done by running the control loop again. For example, if the threshold value was increased, we would initiate an event with the old threshold value, and see that the signature is no longer produced.

Testing Directions

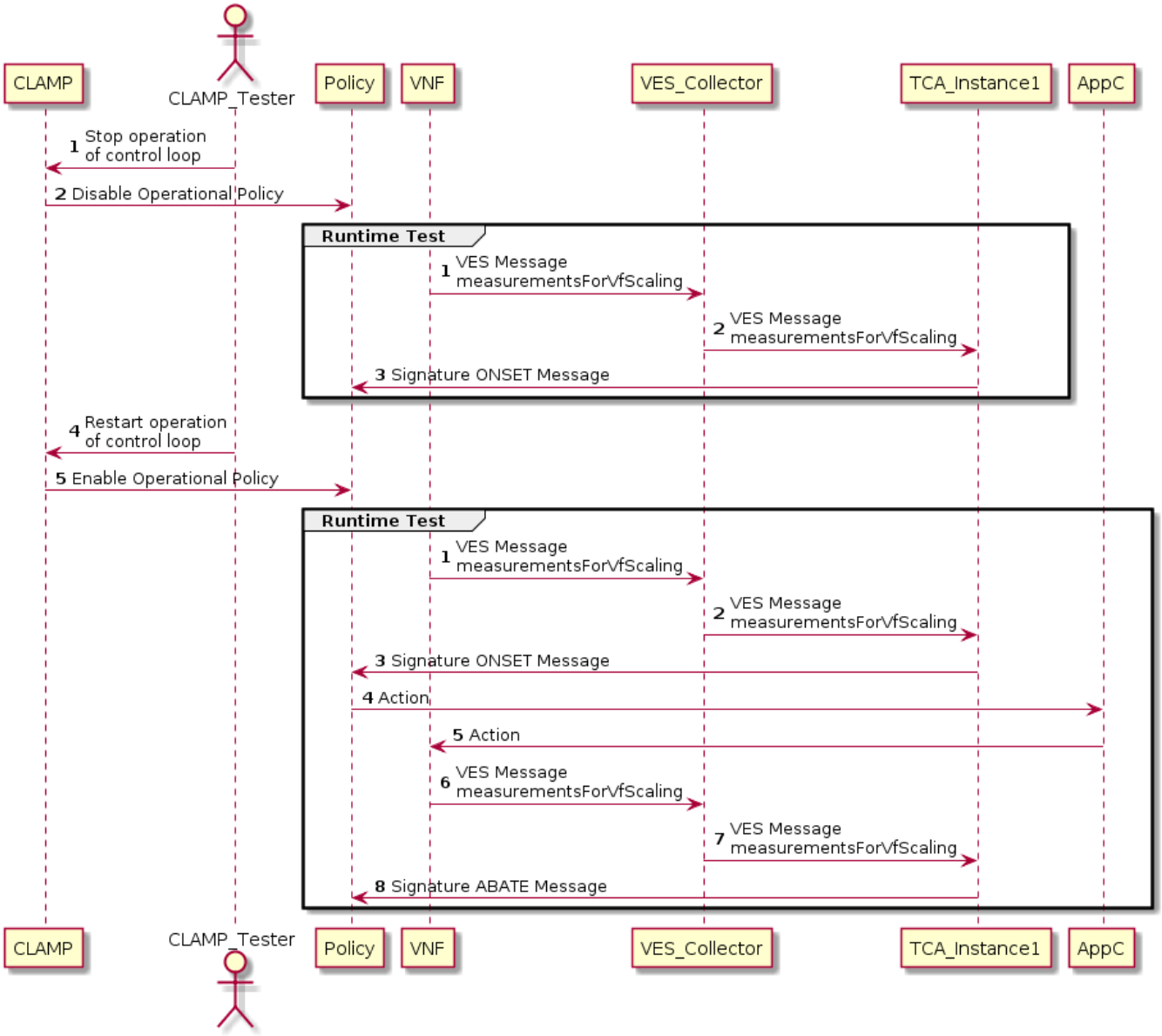
Click on either the Policy box or the TCA box in order to bring up the details of the given policy

Update the values on the policy

Choose Update from the Manage CL Menu

Step Range	Description	Status	Notes
1-2	CLAMP updates policy		~~POLICY-779~~ - Policy Update hangs CLOSED Policy update API call does not return; root cause thought to be in ~~POLICY-777~~ - PAP: frequent failing of provisioning transactions because of DB locking table errors CLOSED - these have been fixed
3-5	Updated policy sent to DCAE		
6-11	Updated policy stored in DCAE		
12-17	TCA retrieves new policy		

Flow 5: Stop and Restart Control Loop


CLAMP
CLAMP_Tester
Policy
VNF
VES_Collector
TCA_Instance1
AppC
1 Stop operation of control loop
2 Disable Operational Policy
Runtime Test
1 VES Message measurementsForVfScaling
2 VES Message measurementsForVfScaling
3 Signature ONSET Message
4 Restart operation of control loop
5 Enable Operational Policy
Runtime Test
1 VES Message measurementsForVfScaling
2 VES Message measurementsForVfScaling
3 Signature ONSET Message
4 Action
5 Action
6 VES Message measurementsForVfScaling
7 VES Message measurementsForVfScaling
8 Signature ABATE Message
CLAMP
CLAMP_Tester
Policy
VNF
VES_Collector
TCA_Instance1
AppC

UML Code for Flow 5

```
@startuml
participant CLAMP
actor CLAMP_Tester
participant Policy
autonumber
CLAMP_Tester -> CLAMP : Stop operation\nof control loop
CLAMP -> Policy : Disable Operational Policy
participant VNF
participant VES_Collector
participant TCA_Instance1
participant AppC
autonumber
group Runtime Test
VNF -> VES_Collector : VES Message\nmeasurementsForVfScaling
VES_Collector -> TCA_Instance1 : VES Message\nmeasurementsForVfScaling
TCA_Instance1 -> Policy : Signature ONSET Message
end
CLAMP_Tester -> CLAMP : Restart operation\nof control loop
CLAMP -> Policy : Enable Operational Policy
participant VNF
participant VES_Collector
participant TCA_Instance1
participant AppC
autonumber
group Runtime Test
VNF -> VES_Collector : VES Message\nmeasurementsForVfScaling
VES_Collector -> TCA_Instance1 : VES Message\nmeasurementsForVfScaling
TCA_Instance1 -> Policy : Signature ONSET Message
Policy -> AppC : Action
AppC -> VNF : Action
VNF -> VES_Collector : VES Message\nmeasurementsForVfScaling
VES_Collector -> TCA_Instance1 : VES Message\nmeasurementsForVfScaling
TCA_Instance1 -> Policy : Signature ABATE Message
end
@enduml
```

Testing Directions

Choose Stop from the Manage Menu

The Status will change to "Stopped"

To verify, check Policy GUI to confirm that the BRMS policy under the control loop's scope has been removed

Choose Restart from the Manage Menu

The Status will change back to "Active"

To verify, check Policy GUI to confirm that the BRMS policy under the control loop's scope has been returned

Step Range	Description	Status	Notes
1-2	Operational Policy (action) is disabled		
	Disabling of action is tested		
	Operational Policy (action) is enabled		
	Onset is tested after enabling action		
	Abatement is tested after enabling action		

Flow 6: Undeploy Control Loop

Choose Undeploy from the Manage Menu

The Status will change to "Distributed"

To verify, check DCAE to make sure that the TCA instance has been removed

Dashboard