

DCAE mS Deployment (Standalone instantiation)

- [VESCollector](#)
 - [VM Init](#)
- [ThresholdCrossingAnalysis \(TCA/CDAP\)](#)
 - [TCA Configuration Change](#)
 - [VM Init](#)

The below steps covers manual setup of DCAE VM's and DCAE service components.

VESCollector

DCAE VES Collector can be configured on VM with ubuntu-16.04 image (m1.small should suffice if this is only service) and 20Gb cinder storage

1) Install docker

```
sudo apt-get update  
sudo apt install docker.io
```

2) Pull the latest container from onap nexus

```
sudo docker login -u docker -p docker nexus3.onap.org:10001  
sudo docker pull nexus3.onap.org:10001/onap/org.onap.dcaegen2.collectors.ves.vescollector:v1.1.0
```

3) Start the VESCollector with below command

```
sudo docker run -d --name vescollector -p 8080:8080/tcp -p 8443:8443/tcp -P -e DMAAPHOST='<dmaap IP>' nexus3.onap.org:10001/onap/org.onap.dcaegen2.collectors.ves.vescollector:v1.1.0
```

Note: Change the dmaaphost to required DMAAP ip. To change the dmaap information for a running container, stop the active container and rerun above command changing the dmaap IP.

4) Verification

1. Check logs under container /opt/app/VESCollector/logs/collector.log for errors
2. If no active feed, you can simulate an event into collector via curl

```
curl -i -X POST -d @<sampleves> --header "Content-Type: application/json" http://localhost:8080/eventListener/v5 -k
```

Note: If DMAAPHOST provided is invalid, you will see exception around publish on the collector.logs (collector queues and attempts to resend the event hence exceptions reported will be periodic).

3. Below two topic configuration are pre-set into this container. When valid DMAAP instance ip was provided and VES events are received, the collector will post to below topics.

Fault - [http://<dmaaphost>:3904/events/unauthenticated.SEC_FAULT_OUTPUT](#)

Measurement -[http://<dmaaphost>:3904/events/unauthenticated.SEC_MEASUREMENT_OUTPUT](#)

VM Init

To address windriver server in-stability, the below **init.sh** script was used to start the container on VM restart.

init.sh

```
#!/bin/sh
sudo docker ps | grep "vescollector"
if [ $? -ne 0 ]; then
    sudo docker login -u docker -p docker nexus3.onap.org:10001
    sudo docker pull nexus3.onap.org:10001/onap/org.onap.dcaeagen2.collectors.ves.vescollector:1.1
    sudo docker rm -f vescollector
    echo "Collector process not running - $(date)" >> /home/ubuntu/startuplog
    sudo docker run -d --name vescollector -p 8080:8080/tcp -p 8443:8443/tcp -P -e DMAAPHOST='10.12.25.96'
nexus3.onap.org:10001/onap/org.onap.dcaeagen2.collectors.ves.vescollector:v1.1.0
else
    echo "Collector process running - $(date)" >> /home/ubuntu/startuplog
fi
```

This script was invoked via VM init script (rc.d).

```
ln -s /home/ubuntu/init.sh /etc/init.d/init.sh

sudo update-rc.d init.sh start 2
```

ThresholdCrossingAnalysis (TCA/CDAP)

The platform deploys CDAP as cluster and instantiates TCA. For the manual setup, we will leverage the CDAP SDK docker container to deploy TCA instances. To setup TCA, choose VM with ubuntu-16.04 image, m1.medium size and 50gb cinder volumes.

1) Install docker

```
sudo apt-get update

sudo apt install docker.io
```

2) Pull CDAP SDK container

```
sudo docker pull caskdata/cdap-standalone:4.1.2
```

3) Deploy and run the CDAP container

```
sudo docker run -d --name cdap-sdk-2 -p 11011:11011 -p 11015:11015 caskdata/cdap-standalone:4.1.2
```

4) Create Namespace on CDAP application

```
curl -X PUT http://localhost:11015/v3/namespaces/cdap_tca_hi_lo
```

5) Create TCA app config file - "tca_app_config.json" under ~ubuntu as below

tca_app_config.json

```
{
  "artifact": {
    "name": "dcae-analytics-cdap-tca",
    "version": "2.0.1",
    "scope": "user"
  },
  "config": {
    "appName": "dcae-tca",
    "appDescription": "DCAE Analytics Threshold Crossing Alert Application",
    "tcaVESMessageStatusTableName": "TCAVESMessageStatusTable",
    "tcaVESMessageStatusTableTTLSeconds": 86400.0,
    "tcaAlertsAbatementTableName": "TCAAlertsAbatementTable",
    "tcaAlertsAbatementTableTTLSeconds": 1728000.0,
    "tcaVESAlertsTableName": "TCAVESAlertsTable",
    "tcaVESAlertsTableTTLSeconds": 1728000.0,
    "thresholdCalculatorFlowletInstances": 2.0,
    "tcaSubscriberOutputStreamName": "TCASubscriberOutputStream"
  }
}
```

6) Create TCA app preference file - "tca_app_preferences.json" under ~ubuntu as below

tca_app_preferences.json

```
{
  "publisherContentType" : "application/json",
  "publisherHostName" : "10.12.25.96",
  "publisherHostPort" : "3904",
  "publisherMaxBatchSize" : "1",
  "publisherMaxRecoveryQueueSize" : "100000",
  "publisherPollingInterval" : "20000",
  "publisherProtocol" : "http",
  "publisherTopicName" : "unauthenticated.DCAE_CL_OUTPUT",
  "subscriberConsumerGroup" : "OpenDCAE-cl",
  "subscriberConsumerId" : "cl",
  "subscriberContentType" : "application/json",
  "subscriberHostName" : "10.12.25.96",
  "subscriberHostPort" : "3904",
  "subscriberMessageLimit" : "-1",
  "subscriberPollingInterval" : "20000",
  "subscriberProtocol" : "http",
  "subscriberTimeoutMS" : "-1",
  "subscriberTopicName" : "unauthenticated.SEC_MEASUREMENT_OUTPUT",
  "enableAAIEnrichment" : false,
  "aaiEnrichmentHost" : "10.12.25.72",
  "aaiEnrichmentPortNumber" : 8443,
  "aaiEnrichmentProtocol" : "https",
  "aaiEnrichmentUserName" : "DCAE",
  "aaiEnrichmentUserPassword" : "DCAE",
  "aaiEnrichmentIgnoreSSLCertificateErrors" : true,
  "aaiVNFEnrichmentAPIPath" : "/aai/v11/network/generic-vnfs/generic-vnf",
  "aaiVMEEnrichmentAPIPath" : "/aai/v11/search/nodes-query",
  "tca_policy" : "{
    \"domain\": \"measurementsForVfScaling\",
    \"metricsPerEventName\": [{
      \"eventName\": \"vFirewallBroadcastPackets\",
      \"controlLoopSchemaType\": \"VNF\",
      \"policyScope\": \"DCAE\",
      \"policyName\": \"DCAE.Config_tca-hi-lo\",
      \"policyVersion\": \"v0.0.1\",
      \"thresholds\": [{
        \"closedLoopControlName\": \"ControlLoop-vFirewall-d0a1dfc6-94f5-4fd4-a5b5-4630b438850a\",
        \"version\": \"1.0.2\",
        \"fieldPath\": \"$.event.measurementsForVfScalingFields.vNicUsageArray[*].receivedTotalPacketsDelta\",
        \"thresholdValue\": 300,
        \"direction\": \"LESS_OR_EQUAL\",
        \"severity\": \"MAJOR\",
        \"closedLoopEventStatus\": \"ONSET\"
      }], {
        \"closedLoopControlName\": \"ControlLoop-vFirewall-d0a1dfc6-94f5-4fd4-a5b5-4630b438850a\",
        \"version\": \"1.0.2\",
        \"fieldPath\": \"$.event.measurementsForVfScalingFields.vNicUsageArray[*].receivedTotalPacketsDelta\",
        \"thresholdValue\": 700,
        \"direction\": \"GREATER_OR_EQUAL\",
        \"severity\": \"CRITICAL\",
        \"closedLoopEventStatus\": \"ONSET\"
      }
    ]
  }, {
    \"eventName\": \"vLoadBalancer\",
    \"controlLoopSchemaType\": \"VM\",
    \"policyScope\": \"DCAE\",
    \"policyName\": \"DCAE.Config_tca-hi-lo\",
    \"policyVersion\": \"v0.0.1\",
    \"thresholds\": [{
      \"closedLoopControlName\": \"ControlLoop-vDNS-6f37f56d-a87d-4b85-b6a9-cc953cf779b3\",
```

```

        \ "version\": \"1.0.2\",
        \ "fieldPath\": \"$.event.measurementsForVfScalingFields.vNicUsageArray[*].
receivedTotalPacketsDelta\",
        \ "thresholdValue\": 300,
        \ "direction\": \"GREATER_OR_EQUAL\",
        \ "severity\": \"CRITICAL\",
        \ "closedLoopEventStatus\": \"ONSET\"
    }], {
        \ "eventName\": \"Measurement_vGMUX\",
        \ "controlLoopSchemaType\": \"VNF\",
        \ "policyScope\": \"DCAE\",
        \ "policyName\": \"DCAE.Config_tca-hi-lo\",
        \ "policyVersion\": \"v0.0.1\",
        \ "thresholds\": [{
            \ "closedLoopControlName\": \"ControlLoop-vCPE-48f0c2c3-a172-4192-9ae3-052274181b6e\",
            \ "version\": \"1.0.2\",
            \ "fieldPath\": \"$.event.measurementsForVfScalingFields.additionalMeasurements[*].
arrayOfFields[0].value\",
            \ "thresholdValue\": 0,
            \ "direction\": \"EQUAL\",
            \ "severity\": \"MAJOR\",
            \ "closedLoopEventStatus\": \"ABATED\"
        }], {
            \ "closedLoopControlName\": \"ControlLoop-vCPE-48f0c2c3-a172-4192-9ae3-052274181b6e\",
            \ "version\": \"1.0.2\",
            \ "fieldPath\": \"$.event.measurementsForVfScalingFields.additionalMeasurements[*].
arrayOfFields[0].value\",
            \ "thresholdValue\": 0,
            \ "direction\": \"GREATER\",
            \ "severity\": \"CRITICAL\",
            \ "closedLoopEventStatus\": \"ONSET\"
        }
    }
}
}

```

Note: Dmaap configuration are specified on this file on publisherHostName and subscriberHostName. To be changed as required**

6 a) To enable AAI Enrichment in TCA for CL event, following updates should be done to TCA app preference file before proceeding

```

"enableAAIEnrichment" : true
"aaiEnrichmentHost" : <should be set to aail_ip_addr>
"aaiEnrichmentIgnoreSSLCertificateErrors" : true

```

7) Copy below script to CDAP server (this gets latest image from nexus and deploys TCA application) and execute it

reload-tca.sh

```
#!/bin/sh
TCA_JAR=dcae-analytics-cdap-tca-2.0.0.jar
rm -f /home/ubuntu/$TCA_JAR
cd /home/ubuntu/
wget https://nexus3.onap.org/service/local/repositories/releases/content/org/onap/dcaeegen2/analytics/tca/dcae-
analytics-cdap-tca/2.0.0/$TCA_JAR
if [ $? -eq 0 ]; then
    if [ -f /home/ubuntu/$TCA_JAR ]; then
        echo "Restarting TCA CDAP application using $TCA_JAR artifact"
    else
        echo "ERROR: $TCA_JAR missing"
        exit 1
    fi
else
    echo "ERROR: $TCA_JAR not found in nexus"
    exit 1
fi
# stop programs
curl -X POST http://localhost:11015/v3/namespaces/cdap_tca_hi_lo/apps/dcae-tca/workers/TCADMaapMRPublisherWorker
/stop
curl -X POST http://localhost:11015/v3/namespaces/cdap_tca_hi_lo/apps/dcae-tca/workers
/TCADMaapMRSubscriberWorker/stop
curl -X POST http://localhost:11015/v3/namespaces/cdap_tca_hi_lo/apps/dcae-tca/flows/TCAVESCollectorFlow/stop
# delete application
curl -X DELETE http://localhost:11015/v3/namespaces/cdap_tca_hi_lo/apps/dcae-tca
# delete artifact
curl -X DELETE http://localhost:11015/v3/namespaces/cdap_tca_hi_lo/artifacts/dcae-analytics-cdap-tca/versions/2.
0.1
# load artifact
curl -X POST --data-binary @/home/ubuntu/$TCA_JAR http://localhost:11015/v3/namespaces/cdap_tca_hi_lo/artifacts
/dcae-analytics-cdap-tca
# create app
curl -X PUT -d @/home/ubuntu/tca_app_config.json http://localhost:11015/v3/namespaces/cdap_tca_hi_lo/apps/dcae-
tca
# load preferences
curl -X PUT -d @/home/ubuntu/tca_app_preferences.json http://localhost:11015/v3/namespaces/cdap_tca_hi_lo/apps
/dcae-tca/preferences
# start programs
curl -X POST http://localhost:11015/v3/namespaces/cdap_tca_hi_lo/apps/dcae-tca/workers/TCADMaapMRPublisherWorker
/start
curl -X POST http://localhost:11015/v3/namespaces/cdap_tca_hi_lo/apps/dcae-tca/workers
/TCADMaapMRSubscriberWorker/start
curl -X POST http://localhost:11015/v3/namespaces/cdap_tca_hi_lo/apps/dcae-tca/flows/TCAVESCollectorFlow/start
echo
# get status of programs
curl http://localhost:11015/v3/namespaces/cdap_tca_hi_lo/apps/dcae-tca/workers/TCADMaapMRPublisherWorker/status
curl http://localhost:11015/v3/namespaces/cdap_tca_hi_lo/apps/dcae-tca/workers/TCADMaapMRSubscriberWorker/status
curl http://localhost:11015/v3/namespaces/cdap_tca_hi_lo/apps/dcae-tca/flows/TCAVESCollectorFlow/status
echo
```

Note: Attached the script also for reference : [reload-tca.sh](#)

8) Verify TCA application and logs via CDAP GUI processes

http://<hostname/ip>:11011/oldcdap/ns/cdap_tca_hi_lo/apps/dcae-tca/overview/programs

The overall flow can be checked here

http://<hostname/ip>:11011/oldcdap/ns/cdap_tca_hi_lo/apps/dcae-tca/programs/flows/TCAVESCollectorFlow/runs

TCA Configuration Change

Typical configuration changes include changing DMAAP host and/or Policy configuration. If necessary, modify the file on step #6 and run the script noted as step #7 to redeploy TCA with updated configuration.

VM Init

To address windriver server in-stability, the below **init.sh** script was used to restart the container on VM restart. This script was invoked via VM init script (rc.d).

init.sh

```
#!/bin/sh
#docker run -d --name cdap-sdk -p 11011:11011 -p 11015:11015 caskdata/cdap-standalone:4.1.2
sudo docker restart cdap-sdk-2
sleep 30
# start program
curl -X POST http://localhost:11015/v3/namespaces/cdap_tca_hi_lo/apps/dcae-tca/workers/TCADMaapMRPublisherWorker/start
curl -X POST http://localhost:11015/v3/namespaces/cdap_tca_hi_lo/apps/dcae-tca/workers/TCADMaapMRSubscriberWorker/start
curl -X POST http://localhost:11015/v3/namespaces/cdap_tca_hi_lo/apps/dcae-tca/flows/TCAVESCollectorFlow/start
```

This script was invoked via VM init script (rc.d).

```
ln -s /home/ubuntu/init.sh /etc/init.d/init.sh
sudo update-rc.d init.sh start 2
```