Cloudify Blueprint validation under OOM

For DCAE services deployed via blueprint, below steps can be followed to trial out new blueprint/inputs

1) Using ONAP key, you can login into rancher vm on required tenant DCAE/integration-SB04/Integration-OOM-Daily/Integration-OOM-Staging-Daily

For e.g – for SB04 rancher ip
>>ssh -i ~/onap_dev ubuntu@10.12.5.47

Note: Use DCAE tenant if your components is not targeted for current release. The onap_dev is common key used across all VM in WINlab.

2) Using kubectl find the pod for bootstrap and exec into it

>>kubectl exec -it -n onap dev-dcaegen2-dcae-bootstrap-776cf86d49-mxzq6 /bin/bash

3) The blueprint for Holmes rules/engine component are under /blueprints (this is basically pulled from nexus raw during container build) and corresponding input file is under /inputs (this gets mapped during deployment from what is specified under oom repo). Here you can basically modify the image tag or any other blueprint changes to trial/deploy.

4) Deploy (or undeploy) commands.

If the component is already deployed (run "cfy deployments list" to check), then uninstall first

Uninstall

```
Syntax - cfy uninstall <deployment to be uninstalled>
>>cfy uninstall holmes_rules
```

Remove the deployment (this is required if uninstall does not remove the deployment automatically)

```
Syntax - cfy deployments delete -f <deployment name>

>>cfy deployments delete -f holmes_rules
```

Delete older version of blueprint (run "cfy blueprints list" to get current list; this is required if uninstall does not remove the deployment automatically)

```
Syntax - cfy blueprints delete <blueprint name>
>>cfy blueprints delete holmes_rules
```

Validate the new blueprint changes

```
Syntax - cfy blueprints validate <blueprint filename>

>> cfy blueprints validate k8s-holmes-rules.yaml
```

Upload the new blueprint

```
Syntax - cfy blueprints upload -b <blueprintname> <blueprint file path>
>>cfy blueprints upload -b holmes_rules /blueprints/k8s-holmes-rules.yaml
```

Create a new deployment with updated blueprint

```
Syntax - cfy deployments create -b <blueprint name> -i <input file path> <deployment name>
>>cfy deployments create -b holmes_rules -i /inputs/k8s-holmes_rules-inputs.yaml holmes_rules
```

Execute install workflow

```
Syntax - cfy executions start -d <deployment name> install
>>cfy executions start -d holmes_rules install

Note: Examples are captured around  holmes_engine; replace holmes_engine and corresponding blueprint/input
file name required.
```

5) Verify status/logs

```
>>root@onap-oom-rancher:~# kubectl get pods -n onap | grep holmes
dep-holmes-engine-mgmt-885d86d49-czh28                 0/1       Running   61        10h
dep-holmes-rule-mgmt-5ffd74f697-bb7xp                  0/1       Running   61        10h
```

The component version used in deployment can be checked using below command.

```
>> kubectl describe pod -n onap dep-holmes-engine-mgmt-885d86d49-czh28
```

Rolling upgrade of deployed components

DCAE platform for R3 supports rolling upgrade of individual component deployed via cloudify.

To upgrade a running pod, follow below syntax

cfy execution start -d <deployment name> <inputfile> execute_operation

<input file> - should be under following structure

operation: update_image

operation_kwargs:

  image: <complete path of image:tag>

node_ids:

  - <list of node_id's in blueprint to be updated>

Example: cfy executions start -d ves -p k8s-ves-image.yaml execute_operation

Where k8s-ves-image.yaml is set as below

operation: update_image

operation_kwargs:

  image: nexus3.onap.org:10001/onap/org.onap.dcaegen2.collectors.ves.vescollector:1.3.0

node_ids:

  - "ves"

Example of rolling upgrade executions (logs for Data-file collector)

cfy executions start -d datafile-collector -p k8s-datafile-collector-image.yaml execute_operation

Executing workflow execute_operation on deployment datafile-collector [timeout=900 seconds]

2018-10-10 20:42:07.234 CFY <datafile-collector> Starting 'execute_operation' workflow execution

2018-10-10 20:42:07.971 CFY <datafile-collector> [datafile-collector_8avn6c] Starting operation update_image (Operation parameters: {u'image':u'10.12.5.2:5000/onap/org.onap.dcaegen2.collectors.datafile.datafile-app-server:1.0.2'})

2018-10-10 20:42:07.971 CFY <datafile-collector> [datafile-collector_8avn6c.update_image] Sending task 'k8splugin.update_image'

2018-10-10 20:42:07.971 CFY <datafile-collector> [datafile-collector_8avn6c.update_image] Task started 'k8splugin.update_image'

2018-10-10 20:42:08.682 LOG <datafile-collector> [datafile-collector_8avn6c.update_image] INFO: Updating app image for dcae-datafile-collector from 10.12.5.2:5000/onap/org.onap.dcaegen2.collectors.datafile.datafile-app-server:1.0.0 to 10.12.5.2:5000/onap/org.onap.dcaegen2.collectors.datafile.datafile-app-server:1.0.2

2018-10-10 20:42:09.358 LOG <datafile-collector> [datafile-collector_8avn6c.update_image] INFO: Waiting up to 300 secs for dcae-datafile-collector to be updated and become ready 2018-10-10 20:42:25.703 LOG <datafile-collector> [datafile-collector_8avn6c.update_image] INFO: Update complete: dcae-datafile-collector from 10.12.5.2:5000/onap/org.onap.dcaegen2.collectors.datafile.datafile-app-server:1.0.0 to 10.12.5.2:5000/onap/org.onap.dcaegen2.collectors.datafile.datafile-app-server:1.0.2

2018-10-10 20:42:26.250 CFY <datafile-collector> [datafile-collector_8avn6c.update_image] Task succeeded 'k8splugin.update_image' 2018-10-10 20:42:26.991 CFY <datafile-collector> [datafile-collector_8avn6c] Finished operation update_image

2018-10-10 20:42:26.991 CFY <datafile-collector> 'execute_operation' workflow execution succeeded

Finished executing workflow execute_operation on deployment datafile-collector

* Run 'cfy events list -e 14cca5f1-ea89-4a0b-9ce0-bb8b25cc8cc1' to retrieve the execution's events/logs