DCAE Service Component (MS) Deployment Options

Controller/Platform Components

These are DCAE components which are responsible for LCM of DCAE services. This includes below

- Cloudify
- Deployment-handler
- ConfigbindingService
- InventoryAPI
- ServiceChange Handler
- Dashboard*

All the above component will be deployed as Helm chart for Dublin release (maintained under oom/kubernetes/dcaegen2)

DCAE Services

This include all service which provides data collection and analytics services; based on usecase - different collection and analytics components can be deployed. These are deployed and managed by the DCAE Platform.

For Dublin - DCAE services will be classified below for **ONAP deployment standpoint.**

Option 1 : Static Components (instantiated through DCAE-bootstrap)

These are generic services used by ONAP integration team. These services are not binded to any specific usecase and require no dynamic policy configuration support.

- VESCollector
- HV_VES
- SNMPTrap
- Holmes Rules/Engine
- TCA
- PRH/Extension

Integration/Deployment steps

1. Component spec creation
   a. Every component to be onboarded into DCAE, should prepare a component spec (a.k.a spec) - which is meta data represented in json describing the component configuration model. Details on spec creation can be found here - RTD spec
2. Validation of spec and docker image using dcae_cli
   a. The dcae_cli tools enables MS owner to validate the component spec and data_Formats created and also "test" the container deployment of MS itself. This allows components to mimic the configuration returned from ConfigBinding Service as expected on typical cloudify based deployment by DCAE platform (Documentation on spec format and tool usage can be found under RTD dcae_cli). There is vagrant setup available which can be used to dcae_cli test/validation (vagrant vm includes Consul and Configbindingservice). Alternatively dcae_cli can be configured to interface with OOM/DCAE Platform - steps for setup and testing is documented here
   b. Once spec and data_formats are validated, add them under service component gerrit repo under following directory structure - **dpo /spec, dpo/data-formats**
3. Blueprint creation (manual for R4; new tool will be available post R4)
   a. Components developer can hand-craft the cloudify blueprints. For blueprint reference - check existing services k8s blueprint template under - DCAE Blueprint repository (refer only to blueprint with prefix - "k8s-" for OOM deployment)
4. Validation of blueprint/deployment in DCAE environment on K8S
   a. The blueprint should be tested on any DCAE deployment by executing into DCAE bootstrap pod; instruction to deploy/validate via blueprint can be found here - Cloudify Blueprint validation under OOM
5. DCAE bootstrap deployment integration

a. Add the new cloudify blueprint template on DCAE blueprint repository
b. Add any corresponding inputs under OOM dcaegen2 bootstrap chart
c. Once above are merged, update DCAE bootstrap to include Service part of OOM DCAE instantiation
    i. https://git.onap.org/dcaegen2/deployments/tree/k8s-bootstrap-container/bootstrap.sh
    ii. https://git.onap.org/oom/tree/kubernetes/dcaegen2/charts/dcae-bootstrap/values.yaml

Option 2 : On-demand deployment through CLAMP

These are MS which can be binded into Service in SDC and requiring dynamic policy configuration management support. Services involved in control loop flow will generally fall under this classification.

Note: For Dublin, none of DCAE MS will be exercising this option.

Onboarding/Deployment steps

1. Component spec creation
    a. Every component to be onboarded into DCAE, should prepare a component spec (a.k.a spec) - which is meta data represented in json describing the component configuration model. Details on spec creation can be found here - RTD spec
2. Validation of spec and docker image using dcae_cli
    a. The dcae_cli tools enables MS owner to validate the component spec and data_Formats created and also "test" the container deployment of MS itself. This allows components to mimic the configuration returned from ConfigBinding Service as expected on typical cloudify based deployment by DCAE platform (Documentation on spec format and tool usage can be found under RTD dcae_cli). There is vagrant setup available which can be used to dcae_cli test/validation (vagrant vm includes Consul and Configbindingservice). Alternatively dcae_cli can be configured to interface with OOM/DCAE Platform - steps for setup and testing is documented here
3. Blueprint creation (manual for R4; new tool will be available post R4)
    a. Components developer can hand-craft the cloudify blueprints. For blueprint reference - check existing services k8s blueprint template under - DCAE Blueprint repository (refer only to blueprint with prefix - "k8s-" for OOM deployment)
    b. Blueprint should include policy_node and taking policy_id as input (refer TCA blueprint) Note: This step is only required for MS involving policy model
4. Policy Model creation (using SDC Tosca_lab) Note: This step is only required for MS involving policy model
    a. Policy model creation require policy schema definition in spec specifying the structure of configuration to be exposed in policy. Follow Tosca Lab Tool Instructions to generate the policy models.
    Note: The policy model created from SDC Tosca_lab version tool is not compatible to new model expected by Policy Team for Dublin; hence some manual update will be required to be complaint (e.g changing nodetype naming and including policy.nodes.root node; specifics being worked with Policy team - documented Onboarding steps for DCAE MS through SDC/Policy/CLAMP (Dublin))
    b. Verify with policy team if the created policy model conforms to the specification Policy team expects
    Note: Rest of models (schema/template/translate) are not compatible to K8S deployment; Tosca_lab version in SDC will be updated in Dublin to align/support DCAE k8s model.
5. Validation of blueprint/deployment in DCAE environment on K8S
    a. The blueprint should be tested on any DCAE deployment by executing into DCAE bootstrap pod; instruction to deploy/validate via blueprint can be found here - Cloudify Blueprint validation under OOM
6. SDC/CLAMP Integration
    a. Add the new models/blueprints in service component gerrit repo under following directory structure - **dpo/spec, dpo/data-formats, dpo/tosca_models, dpo/blueprints**.
        i. Note: Both dpo/tosca_models and dpo/blueprints are temporary place holders for Dublin.
    b. Work with Integration team/usecase owner to upload the artifacts (blueprints & policy model) into SDC and trigger Service creation and distribution from SDC.
    c. Work with CLAMP team to configure the policy and trigger the MS deployment.

Option 3 : On-Demand deployment through DCAE-Controller cli

For services not binded to specific service in SDC, CLAMP instantiation will not apply. For such cases, DCAE CLI or Dashboard can be used to deploy the component on demand. For Dublin - following service will be deployed using cli when required.

- SON Handler (New) – 5G
- DFC - 5G (W/O policy model)
- PM-Mapper (New) – 5G (W/O policy model)
- RESTConf (New) - BBS (W/O policy model)
- VES-Mapper (New) - BBS (W/O policy model)
- bbs-eventprocessor (New) - BBS (W/O policy model)
- Heartbeat (New) (W/O policy model)
- TCA-GEN2* (New) (W/O policy model)

(* Stretch goal)

Note: Policy model will not be required if the components configuration can be supported by blueprint input and consul update

1. Component spec creation
    a. Every component to be onboarded into DCAE, should prepare a component spec (a.k.a spec) - which is meta data represented in json describing the component configuration model. Details on spec creation can be found here - RTD spec
2. Validation of spec and docker image using dcae_cli

    a. The dcae_cli tools enables MS owner to validate the component spec and data_Formats created and also "test" the container deployment of MS itself. This allows components to mimic the configuration returned from ConfigBinding Service as expected on typical cloudify based deployment by DCAE platform (Documentation on spec format and tool usage can be found under RTD dcae_cli). There is vagrant setup available which can be used to dcae_cli test/validation (vagrant vm includes Consul and Configbindingservice). Alternatively dcae_cli can be configured to interface with OOM/DCAE Platform - steps for setup and testing is documented here
3. Blueprint creation (manual for R4; new tool will be available post R4)
    a. Components developer can hand-craft the cloudify blueprints. For blueprint reference - check existing services k8s blueprint template under - DCAE Blueprint repository (refer only to blueprint with prefix - "k8s-" for OOM deployment)
    b. Blueprint should include policy_node and taking policy_id as input (refer TCA policy enabled blueprint) Note: This step is only required for MS involving policy model
4. Policy Model creation (using SDC Tosca_lab)
   Note: This step is only required for MS involving policy model
    a. Policy model creation require policy schema definition in spec specifying the structure of configuration to be exposed in policy. Follow Tosca Lab Tool Instructions to generate the policy models
       Note: The policy model created from SDC Tosca_lab version tool is not compatible to new model expected by Policy Team for Dublin; hence some manual update will be required to be complaint (e.g changing nodetype naming and including policy.nodes.root node; specifics being worked with Policy team - documented Onboarding steps for DCAE MS through SDC/Policy/CLAMP (Dublin))
    b. Verify with policy team if the created policy model conforms to the specification Policy team expects
   Note: Rest of models (schema/template/translate) are not compatible to K8S deployment; Tosca_lab version in SDC will be updated in Dublin to align/support DCAE k8s model.
5. For Testing
    a. Step 1- Service owners are required to upload the policy model into Policy and create a configuration policy (policy_id created) before deployment of MS itself Note: This step is only required for MS involving policy model
    b. Step 2 - Validate the blueprint by deployment in DCAE k8s cluster. The policy_id obtained in step 1 should be specified as input and deployed using cfy command as documented here - Cloudify Blueprint validation under OOM
6. Once the spec, policy model and blueprints are validated, add it under service component gerrit repo under following directory structure - **dpo /spec, dpo/data-formats, dpo/tosca_models, dpo/blueprints**.
    a. Note: Both dpo/tosca_models and dpo/blueprints are temporary place holders for Dublin
7. Documentation
    a. As the MS will be deployed on-demand in ONAP, include the instruction for deployment of MS and validation. This info should be submitted as"readme" in gerrit and on ONAP Usecase wiki used by Integration team.

Mitigation strategy

Policy is migrating toward new design/architecture for Dublin, this requires interfacing clients to use new LC API's and onboarding MS to use new tosca policy model

Constraint with this new Policy API migration

- Toscalab tool under SDC not ready to generate policy model under new structure expected by Policy
- Policy update not supported. Any change in Policy, will require re-deployment of MS for Dublin

To reduce the risk/dependency associated, MS owner can determine if following option can be used for Dublin as work-around; this will simplify the deployment/integration steps associated with Policy Model creation.

Configuration via blueprint input and consul update

In this case, the installation configuration can be provided as input in the blueprint. Configuration such as remote xnf host – can be defined as input in the cloudify blueprint and sourced under application from the input

Example:

```
inputs:
  pg_ipAddress:
    type: string
    default: hbpostgres-write
  pg_userName:
    type: string
    default: heartbeat

node_templates:
  heartbeat:
    properties:
      application_config:
        pg_ipAddress:
          get_input: pg_ipAddress
        pg_userName:
          get_input: pg_userName
```

Once deployed, the configuration can be modified in consul for Application to source from without redeployment. This mode of deployment can be done from either CLAMP or within DCAE using Cloudify CLI (or Dashboard); no policy model associated in this mode and any action required on Policy.

Note: Any change in consul is temporal; will required to be reapplied if onap/consul is reinstalled.

Configuration update supported through Policy

This will require the original steps documented for Policy tosca_model creation (manual for Dublin) and blueprint (manual) to be followed and upload into SDC repository for distribution to CLAMP/DCAE (and Policy?)
Note: The policy model will also be required to loaded manually into Policy for Dublin. Since no automated tool exist for generating Policy model compatible for Dublin version; MS owners must work close with Policy team to validate their model created manually