Helm Generator for DCAE MS

Wiki to track the design requirements for Helm generator to support ~~DCAEGEN2-2694~~ - Helm charts generation through MOD (Part1) CLOSED

REQUIREMENTS

1. ENV SETTING SUPPORT

Component Spec

- Need spec schema update to include list of parameters (key/value for applicationEnv) -->https://git.onap.org/dcaegen2/platform/tree/mod/component-json-schemas/component-specification/dcae-cli-v2/component-spec-schema.json

```
"auxilary": {
.
.
"helm": {
         "applicationEnv": {
    "PMSH_PG_URL": "dcae-pmsh-pg-primary",
    "PMSH_PG_USERNAME": {
            "secretUid": "pgUserCredsSecretUid",
            "key": "login"
    },
    "PMSH_PG_PASSWORD": {
         "secretUid": "pgUserCredsSecretUid",
         "key": "password"
    }
    }
}
.
.
}
```

Values.yaml specification

```
applicationEnv:
  PMSH_PG_URL: dcae-pmsh-pg-primary
  PMSH_PG_USERNAME:
    secretUid: pgUserCredsSecretUid
    key: login
  PMSH_PG_PASSWORD:
    secretUid: pgUserCredsSecretUid
    key: password
```

Note: Text in blue should be mapped from component-spec. If using secret UID, its responsibility of MS developer to include them also on values.yaml

Example

```
  - uid: &pgUserCredsSecretUid pg-user-creds
    name: &pgUserCredsSecretName '{{ include "common.release" . }}-pmsh-pg-user-creds'
    type: basicAuth
    externalSecret: '{{ ternary "" (tpl (default "" .Values.postgres.config.pgUserExternalSecret) .) (hasSuffix "pmsh-pg-user-creds" .Values.postgres.config.
pgUserExternalSecret) }}'
    login: '{{ .Values.postgres.config.pgUserName }}'
    password: '{{ .Values.postgres.config.pgUserPassword }}'
    passwordPolicy: generate
```

2. CONFIG-MAP SUPPORT

Component Spec

```
  "config_map_volume": {
      "type": "object",
      "properties": {
        "config_volume": {
          "type": "object",
          "name": {
            "type": "string"
          }
        },
        "container": {
          "type": "object",
          "bind": {
            "type": "string"
          },
          "mode": {
            "type": "string"
          }
        }
      },
      "required": ["config_volume", "container"]
    },
```

Example:

```
"volumes": [{
        "config_volume": {
                "name": "dcae-external-repo-configmap-schema-map"
                },
        "container": {
                "bind": "/opt/app/VESCollector/etc/externalRepo/"
        }
},
{
        "config_volume": {
                "name": "dcae-external-repo-configmap-sa88-rel16"
                },
        "container": {
                 "bind": "/opt/app/VESCollector/etc/externalRepo/3gpp/rep/sa5/MnS/blob/SA88-Rel16/OpenAPI
/"
        }
  }
],
```

https://git.onap.org/dcaegen2/collectors/ves/tree/dpo/spec/vescollector-componentspec.json

Values.yaml specification

```
externalVolumes:
  - name: dcae-external-repo-configmap-schema-map
    type: configmap
    mountPath: /opt/app/VESCollector/etc/externalRepo/
    optional: true (default)
  - name: '{{ include "common.release" . }}-another-example'   //dcae-external-repo-configmap-sa88-rel16
    type: configmap
    mountPath: /opt/app/VESCollector/etc/externalRepo/3gpp/rep/sa5/MnS/blob/SA88-Rel16/OpenAPI
    optional: false  //If set to false, the configMap must be present in order for the microservice's pod to
start. Defaults to true.
```

3. CMVP2 Certificates support

Component Spec

```
         "tls_info": {
          "description": "Component information to use tls certificates",
          "type": "object",
          "properties": {
            "cert_directory": {
              "description": "The path in the container where the component certificates will be placed by the
init container",
              "type": "string"
            },
            "use_tls": {
              "description": "Boolean flag to determine if the application is using tls certificates",
              "type": "boolean"
            },
            "use_external_tls": {
              "description": "Boolean flag to determine if the application is using tls certificates for
external communication",
              "type": "boolean"
            }
          },
          "required": [
            "cert_directory","use_tls"
          ],
          "additionalProperties": false
        },
```

Example:

```
    "tls_info":{
        "cert_directory":"/opt/app/dcae-certificate/",
        "use_tls":true,
        "use_external_tls": true
     }
```

https://git.onap.org/dcaegen2/collectors/ves/tree/dpo/spec/vescollector-componentspec.json

Values.yaml specification

```
# CMPv2 certificate
certificates:
  - mountPath: /opt/app/dcae-certificate/external
    commonName: dcae-ves-collector  --> from spec
    dnsNames:
      - dcae-ves-collector --> from spec
    keystore:
      outputType:
        - jks
      passwordSecretRef:
        name: ves-cmpv2-keystore-password  --> TBD
        key: password
        create: true
```

requirement.yaml

```
     - name: certManagerCertificate
       version: ~8.x-0
       repository: '@local'
```

templates/certificates.yaml

```
{{ if and .Values.certDirectory .Values.global.cmpv2Enabled .Values.global.CMPv2CertManagerIntegration }}
{{ include "certManagerCertificate.certificate" . }}
{{ end }}
```

4. POLICY SIDECAR SUPPORT

Component Spec

```
    "policy_info": {
      "type": "object",
      "properties": {
        "policy":
        {
          "type": "array",
          "items":
          {
            "type": "object",
            "properties":
            {
              "node_label":
              {
                "type": "string"
              },
              "policy_id":
              {
                "type": "string"
              },
              "policy_model_id":
              {
                "type": "string"
              }
            },
            "required": ["node_label", "policy_model_id"]
          }
        }
      },
      "additionalProperties": false
    }
  }
```

Example:

```
        "policy_info":{
                "policy":[
                        {
                            "node_label":"tca_policy_00",
                            "policy_model_id":"onap.policies.monitoring.cdap.tca.hi.lo.app"
                                "policy_id":"tca_policy_id_10",
                        },
                        {
                                "node_label":"tca_policy_11",
                                "policy_id":"tca_policy_id_11",
                                "policy_model_id":"onap.policies.monitoring.cdap.tca.hi.lo.app"
                        }
                ]
        }
```

Values.yaml specification

#dcaePolicySyncImage: onap/org.onap.dcaegen2.deployments.dcae-services-policy-sync:1.0.1 From base template
policies:
  duration: 300 default

  policyRelease: onap
  policyID: |
    '["tca_policy_id_11","tca_policy_id_10"]' coming from spec file

5. POSTGRES SUPPORT

Component Spec

```
        "databases": {
          "description": "The databases the application is connecting to using the pgaas",
          "type": "object",
          "additionalProperties": {
            "type": "string",
            "enum": [
              "postgres"
            ]
          }
        },
```

- **Need secret suffix or retrieve from spec-name?**

Values.yaml specification

```
##################################################################
# Secrets Configuration.
##################################################################
secrets:
  - uid: pg-user-creds
    name: '{{ include "common.release" . }}-pmsh-pg-user-creds'
    type: basicAuth
    externalSecret: '{{ ternary "" (tpl (default "" .Values.postgres.config.pgUserExternalSecret) .) (hasSuffix
"pmsh-pg-user-creds" .Values.postgres.config.pgUserExternalSecret) }}'
    login: '{{ .Values.postgres.config.pgUserName }}'
    password: '{{ .Values.postgres.config.pgUserPassword }}'
    passwordPolicy: generate


postgres:
  nameOverride: dcae-pmsh-postgres
  service:
    name: dcae-pmsh-postgres
    name2: dcae-pmsh-pg-primary
    name3: dcae-pmsh-pg-replica
  container:
    name:
      primary: dcae-pmsh-pg-primary
      replica: dcae-pmsh-pg-replica
  persistence:
    mountSubPath: pmsh/data
    mountInitPath: pmsh
  config:
    pgUserName: pmsh
    pgDatabase: pmsh
    pgUserExternalSecret: '{{ include "common.release" . }}-pmsh-pg-user-creds'
```

Note: applicationEnv setting if required should be mapped from spec as-is (req#1). Example above contains <pmsh> part of secret name and PG name which should be mapped to component-name from spec file

Requirement.yaml

requirement yaml content

```
  - name: postgres
    version: ~8.x-0
    repository: '@local'
    condition: postgres.enabled
```

6. DMAAP – Secure Topic/Feed (WIP)

Component Spec

TBD

Values.yaml specification

```
##################################################################
# Secrets Configuration.
##################################################################
secrets:
  - uid: &aafCredsUID aafcreds
    type: basicAuth
    login: '{{ .Values.aafCreds.identity }}'
    password: '{{ .Values.aafCreds.password }}'
    passwordPolicy: required


# AAF Credentials
aafCreds:
  identity: dcae@dcae.onap.org
  password: demo123456!

credentials:
- name: AAF_USER
  uid: *aafCredsUID
  key: login
- name: AAF_PASSWORD
  uid: *aafCredsUID
  key: password
```

Note: applicationConfig should use same names as defined under credentials

Example:

```
enable_tls: true
aaf_identity: ${AAF_USER}
aaf_password: ${AAF_PASSWORD}
streams_publishes:
  ves-3gpp-fault-supervision:
  type: kafka
  aaf_credentials:
    username: ${AAF_USER}
    password: ${AAF_PASSWORD}
  kafka_info:
    bootstrap_servers: message-router-kafka:9092
    topic_name: SEC_3GPP_FAULTSUPERVISION_OUTPUT
```

7. SERVICE MAPPING

Component Spec

```
"auxilary": {
.
.
"helm": {
      "services": [
        {
          "type": "NodePort",
          "name": "dcae-ves-collector",
          "ports": [
             {
                "name": "http",
                "port": 8443,
                "plain_port": 8080,
                "port_protocol": "http",
                "nodePort": 17,
                "useNodePortExt": true
             }
          ]
       }
      ]
    }
  }
.
.
}
```

- Schema change required need to determine if nodeport vs clusterip
  - Require type/name/ports
    - type - Nodeport or ClusterIPO
    - ports - list of objects mapped from spec as-is
    - constraints for ports can be added later

https://git.onap.org/dcaegen2/platform/tree/mod/component-json-schemas/component-specification/dcae-cli-v2/component-spec-schema.json

Values.yaml specification

```
service:
  type: ClusterIP
  name: dcae-tcagen2
  ports:
    - port: 9091
      name: http
```

OR

```
global:
  nodePortPrefix: 302
  nodePortPrefixExt: 304

        # service configuration
service:
  type: NodePort
  name: dcae-ves-collector
  ports:
    - name: http
      port: 8443
      plain_port: 8080
      port_protocol: http
      nodePort: 17
      useNodePortExt: true
```

OR

Based on https://gerrit.onap.org/r/c/oom/+/121390

```
service:
  type: NodePort
  name: dcae-ves-collector
  has_internal_only_ports: true
  ports:
    - name: http
      port: 8443
      plain_port: 8080
      port_protocol: http
      nodePort: 17
      useNodePortExt: true
    - name: metrics
      port: 4444
      internal_only: true
```

REVISED V3 SPEC

Component	V3 Schema	V2 Schema	With CMPV2	With Postgres	With Policy
VESCollector	vescollector-componentspec-v3-helm	vescollector-componentspec	vescollector-componentspec-cmpv2-v3-helm	vescollector-componentspec-postgres-v3-helm	
TCAgen2	tcagen2_spec-v3-helm	tcagen2_spec			tcagen2_spec-policy-v3-helm
PRH	prh-componentspec-v3-helm (pending test)	prh-componentspec			
hv_vescollector	hv-ves-collector-componentspec-v3-helm (pending test)	hv-ves-collector.componentspec			
PM-Mapper	pmmapper-component-spec-v3-helm (need to update publisher and subscriber and pending test)	pmmapper-component-spec			
DataFileCollector (DFC)	datafile-component-spec-v3-helm (need to update publisher and subscriber and pending test)	datafile-component-spec			

REFERENCE

Discussed ppt slides Helm_deployment.pptx

MOD-HelmGenerator-Requirements_v0.2.docx

MOD-HelmGenerator-Usecase_v0.2.docx