**20. helm template**

--- **note** - you will learn how to use a helm template Command, before we go ahead and use it, let us see why we need it.

**Helm --dry-run**

* The helm team has introduced the **helm --dry-run** option so that we can debug and see if the templates that are getting generated are okay for an installation or upgrade.
* several teams or projects have started using these templates for a different purpose altogether. They were copying these templates that were generated by the **--dry-run** and they were using directly with kubectl Command.
* They were taking these templates using these with kubectl commands of their own and submitting these to the kubernetes so that it will create the required resources but they will have some issues when they do it, because the **--dry-run** command will include some non yaml syntax or elements into these templates that get generated.
* The dry run can be used for both installation and upgradation. upgrade output will be completely different from installation

**# Installation**

--- helm install mysql bitnami/mysql --values /root/mysql/values.yml --dry-run

**# Upgrade**

--- helm upgrade mysql bitnami/mysql --values /root/mysql/values.yml --dry-run

**Helm template**

--- **note** – to overcome the problems we face in the --dry-run command, helm team introduced helm template.

# Generate template for mysql.

--- helm template mysql bitnami/mysql --values /root/mysql/values.yml

---

# Source: mysql/templates/serviceaccount.yaml

apiVersion: v1

kind: ServiceAccount

metadata:

  name: mysql

  namespace: "default"

  labels:

    app.kubernetes.io/name: mysql

    helm.sh/chart: mysql-9.3.2

    app.kubernetes.io/instance: mysql

    app.kubernetes.io/managed-by: Helm

  annotations:

automountServiceAccountToken: true

secrets:

  - name: mysql

---

# Source: mysql/templates/secrets.yaml

apiVersion: v1

kind: Secret

metadata:

  name: mysql

  namespace: "default"

  labels:

    app.kubernetes.io/name: mysql

    helm.sh/chart: mysql-9.3.2

    app.kubernetes.io/instance: mysql

    app.kubernetes.io/managed-by: Helm

type: Opaque

data:

  mysql-root-password: "YWRtaW4xMjM="

  mysql-password: "eHFZOE82M0E5cQ=="

---

# Source: mysql/templates/primary/configmap.yaml

apiVersion: v1

kind: ConfigMap

metadata:

  name: mysql

  namespace: "default"

  labels:

    app.kubernetes.io/name: mysql

    helm.sh/chart: mysql-9.3.2

    app.kubernetes.io/instance: mysql

    app.kubernetes.io/managed-by: Helm

    app.kubernetes.io/component: primary

data:

  my.cnf: |-

    [mysqld]

    default\_authentication\_plugin=mysql\_native\_password

    skip-name-resolve

    explicit\_defaults\_for\_timestamp

    basedir=/opt/bitnami/mysql

    plugin\_dir=/opt/bitnami/mysql/lib/plugin

    port=3306

    socket=/opt/bitnami/mysql/tmp/mysql.sock

    datadir=/bitnami/mysql/data

    tmpdir=/opt/bitnami/mysql/tmp

    max\_allowed\_packet=16M

    bind-address=0.0.0.0

    pid-file=/opt/bitnami/mysql/tmp/mysqld.pid

    log-error=/opt/bitnami/mysql/logs/mysqld.log

    character-set-server=UTF8

    collation-server=utf8\_general\_ci

    slow\_query\_log=0

    slow\_query\_log\_file=/opt/bitnami/mysql/logs/mysqld.log

    long\_query\_time=10.0

    [client]

    port=3306

    socket=/opt/bitnami/mysql/tmp/mysql.sock

    default-character-set=UTF8

    plugin\_dir=/opt/bitnami/mysql/lib/plugin

    [manager]

    port=3306

    socket=/opt/bitnami/mysql/tmp/mysql.sock

    pid-file=/opt/bitnami/mysql/tmp/mysqld.pid

---

# Source: mysql/templates/primary/svc-headless.yaml

apiVersion: v1

kind: Service

metadata:

  name: mysql-headless

  namespace: "default"

  labels:

    app.kubernetes.io/name: mysql

    helm.sh/chart: mysql-9.3.2

    app.kubernetes.io/instance: mysql

    app.kubernetes.io/managed-by: Helm

    app.kubernetes.io/component: primary

  annotations:

spec:

  type: ClusterIP

  clusterIP: None

  publishNotReadyAddresses: true

  ports:

    - name: mysql

      port: 3306

      targetPort: mysql

  selector:

    app.kubernetes.io/name: mysql

    app.kubernetes.io/instance: mysql

    app.kubernetes.io/component: primary

---

# Source: mysql/templates/primary/svc.yaml

apiVersion: v1

kind: Service

metadata:

  name: mysql

  namespace: "default"

  labels:

    app.kubernetes.io/name: mysql

    helm.sh/chart: mysql-9.3.2

    app.kubernetes.io/instance: mysql

    app.kubernetes.io/managed-by: Helm

    app.kubernetes.io/component: primary

  annotations:

spec:

  type: ClusterIP

  sessionAffinity: None

  ports:

    - name: mysql

      port: 3306

      protocol: TCP

      targetPort: mysql

      nodePort: null

  selector:

    app.kubernetes.io/name: mysql

    app.kubernetes.io/instance: mysql

    app.kubernetes.io/component: primary

---

# Source: mysql/templates/primary/statefulset.yaml

apiVersion: apps/v1

kind: StatefulSet

metadata:

  name: mysql

  namespace: "default"

  labels:

    app.kubernetes.io/name: mysql

    helm.sh/chart: mysql-9.3.2

    app.kubernetes.io/instance: mysql

    app.kubernetes.io/managed-by: Helm

    app.kubernetes.io/component: primary

spec:

  replicas: 1

  podManagementPolicy: ""

  selector:

    matchLabels:

      app.kubernetes.io/name: mysql

      app.kubernetes.io/instance: mysql

      app.kubernetes.io/component: primary

  serviceName: mysql

  updateStrategy:

    type: RollingUpdate

  template:

    metadata:

      annotations:

        checksum/configuration: 5c52e411b3a9519246a44d9021f3fe600758cd9fa87562f356592bafbdff7c31

      labels:

        app.kubernetes.io/name: mysql

        helm.sh/chart: mysql-9.3.2

        app.kubernetes.io/instance: mysql

        app.kubernetes.io/managed-by: Helm

        app.kubernetes.io/component: primary

    spec:

      serviceAccountName: mysql

      affinity:

        podAffinity:

        podAntiAffinity:

          preferredDuringSchedulingIgnoredDuringExecution:

            - podAffinityTerm:

                labelSelector:

                  matchLabels:

                    app.kubernetes.io/name: mysql

                    app.kubernetes.io/instance: mysql

                namespaces:

                  - "default"

                topologyKey: kubernetes.io/hostname

              weight: 1

        nodeAffinity:

      securityContext:

        fsGroup: 1001

      initContainers:

      containers:

        - name: mysql

          image: docker.io/bitnami/mysql:8.0.30-debian-11-r15

          imagePullPolicy: "IfNotPresent"

          securityContext:

            runAsNonRoot: true

            runAsUser: 1001

          env:

            - name: BITNAMI\_DEBUG

              value: "false"

            - name: MYSQL\_ROOT\_PASSWORD

              valueFrom:

                secretKeyRef:

                  name: mysql

                  key: mysql-root-password

            - name: MYSQL\_DATABASE

              value: "my\_database"

          envFrom:

          ports:

            - name: mysql

              containerPort: 3306

          livenessProbe:

            failureThreshold: 3

            initialDelaySeconds: 5

            periodSeconds: 10

            successThreshold: 1

            timeoutSeconds: 1

            exec:

              command:

                - /bin/bash

                - -ec

                - |

                  password\_aux="${MYSQL\_ROOT\_PASSWORD:-}"

                  if [[ -f "${MYSQL\_ROOT\_PASSWORD\_FILE:-}" ]]; then

                      password\_aux=$(cat "$MYSQL\_ROOT\_PASSWORD\_FILE")

                  fi

                  mysqladmin status -uroot -p"${password\_aux}"

          readinessProbe:

            failureThreshold: 3

            initialDelaySeconds: 5

            periodSeconds: 10

            successThreshold: 1

            timeoutSeconds: 1

            exec:

              command:

                - /bin/bash

                - -ec

                - |

                  password\_aux="${MYSQL\_ROOT\_PASSWORD:-}"

                  if [[ -f "${MYSQL\_ROOT\_PASSWORD\_FILE:-}" ]]; then

                      password\_aux=$(cat "$MYSQL\_ROOT\_PASSWORD\_FILE")

                  fi

                  mysqladmin status -uroot -p"${password\_aux}"

          startupProbe:

            failureThreshold: 10

            initialDelaySeconds: 15

            periodSeconds: 10

            successThreshold: 1

            timeoutSeconds: 1

            exec:

              command:

                - /bin/bash

                - -ec

                - |

                  password\_aux="${MYSQL\_ROOT\_PASSWORD:-}"

                  if [[ -f "${MYSQL\_ROOT\_PASSWORD\_FILE:-}" ]]; then

                      password\_aux=$(cat "$MYSQL\_ROOT\_PASSWORD\_FILE")

                  fi

                  mysqladmin status -uroot -p"${password\_aux}"

          resources:

            limits: {}

            requests: {}

          volumeMounts:

            - name: data

              mountPath: /bitnami/mysql

            - name: config

              mountPath: /opt/bitnami/mysql/conf/my.cnf

              subPath: my.cnf

      volumes:

        - name: config

          configMap:

            name: mysql

  volumeClaimTemplates:

    - metadata:

        name: data

        labels:

          app.kubernetes.io/name: mysql

          app.kubernetes.io/instance: mysql

          app.kubernetes.io/component: primary

        annotations:

      spec:

        accessModes:

          - "ReadWriteOnce"

        resources:

          requests:

            storage: "8Gi"