**22. helm get**

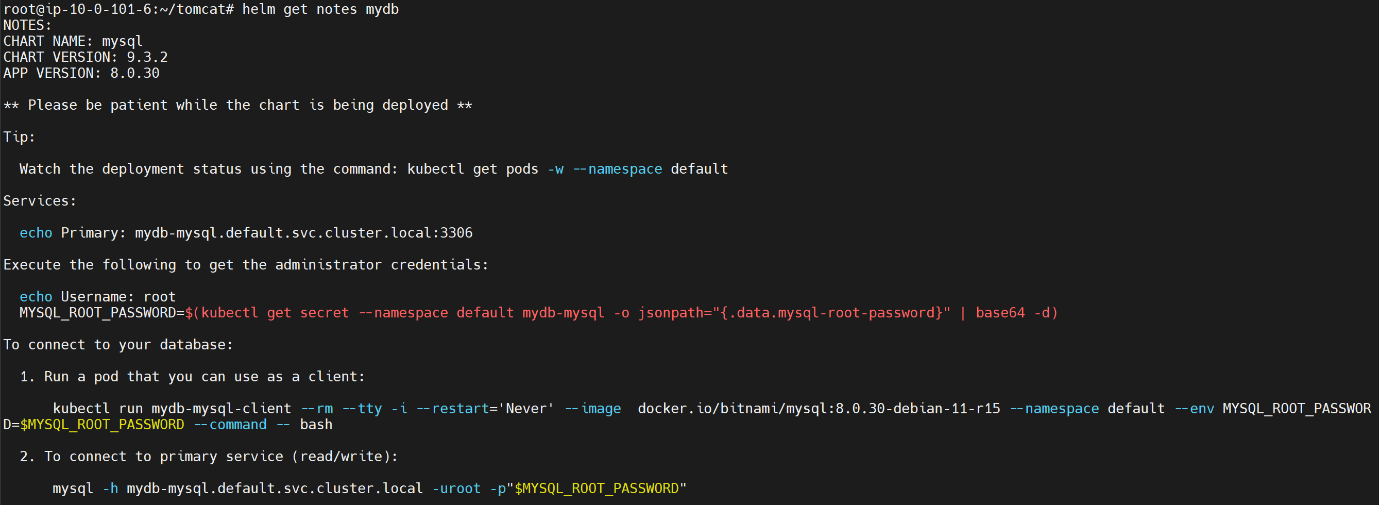
**# List the packages using helm**

--- helm list or helm ls

--- **note** - When we execute a **helm ls** command, we get the information about a particular release, which is the very minimum information. If you want to know more details, we can use the **helm get**

**# Get the release notes information of mysql**

--- helm get notes mydb



**# List the custom values you have passed through values.yml**

--- helm get values mysql

root@ip-10-0-101-6:/home/ubuntu# helm get values mysql

USER-SUPPLIED VALUES:

auth:

  rootPassword: admin123

**# List the customized values passed using values.yml in revision**

--- helm get values mysql --revision 1

root@ip-10-0-101-6:/home/ubuntu# helm get values mysql --revision 1

USER-SUPPLIED VALUES:

auth:

  rootPassword: admin123

# List the manifest for particular revision.

--- helm get manifest mysql --revision 1

**# List all the values for the chart**

--- helm get values mysql --all

COMPUTED VALUES:

architecture: standalone

auth:

  createDatabase: true

  customPasswordFiles: {}

  database: my\_database

  existingSecret: ""

  password: ""

  replicationPassword: ""

  replicationUser: replicator

  rootPassword: admin123

  usePasswordFiles: false

  username: ""

clusterDomain: cluster.local

common:

  exampleValue: common-chart

  global:

    imagePullSecrets: []

    imageRegistry: ""

    storageClass: ""

commonAnnotations: {}

commonLabels: {}

diagnosticMode:

  args:

  - infinity

  command:

  - sleep

  enabled: false

extraDeploy: []

fullnameOverride: ""

global:

  imagePullSecrets: []

  imageRegistry: ""

  storageClass: ""

image:

  debug: false

  digest: ""

  pullPolicy: IfNotPresent

  pullSecrets: []

  registry: docker.io

  repository: bitnami/mysql

  tag: 8.0.30-debian-11-r15

initdbScripts: {}

initdbScriptsConfigMap: ""

kubeVersion: ""

metrics:

  enabled: false

  extraArgs:

    primary: []

    secondary: []

  image:

    digest: ""

    pullPolicy: IfNotPresent

    pullSecrets: []

    registry: docker.io

    repository: bitnami/mysqld-exporter

    tag: 0.14.0-debian-11-r33

  livenessProbe:

    enabled: true

    failureThreshold: 3

    initialDelaySeconds: 120

    periodSeconds: 10

    successThreshold: 1

    timeoutSeconds: 1

  prometheusRule:

    additionalLabels: {}

    enabled: false

    namespace: ""

    rules: []

  readinessProbe:

    enabled: true

    failureThreshold: 3

    initialDelaySeconds: 30

    periodSeconds: 10

    successThreshold: 1

    timeoutSeconds: 1

  resources:

    limits: {}

    requests: {}

  service:

    annotations:

      prometheus.io/port: '{{ .Values.metrics.service.port }}'

      prometheus.io/scrape: "true"

    port: 9104

    type: ClusterIP

  serviceMonitor:

    annotations: {}

    enabled: false

    honorLabels: false

    interval: 30s

    jobLabel: ""

    labels: {}

    metricRelabelings: []

    namespace: ""

    relabelings: []

    scrapeTimeout: ""

    selector: {}

nameOverride: ""

namespaceOverride: ""

networkPolicy:

  allowExternal: true

  enabled: false

  explicitNamespacesSelector: {}

primary:

  affinity: {}

  args: []

  command: []

  configuration: |-

    [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

  containerSecurityContext:

    enabled: true

    runAsNonRoot: true

    runAsUser: 1001

  customLivenessProbe: {}

  customReadinessProbe: {}

  customStartupProbe: {}

  existingConfigmap: ""

  extraEnvVars: []

  extraEnvVarsCM: ""

  extraEnvVarsSecret: ""

  extraFlags: ""

  extraVolumeMounts: []

  extraVolumes: []

  hostAliases: []

  initContainers: []

  lifecycleHooks: {}

  livenessProbe:

    enabled: true

    failureThreshold: 3

    initialDelaySeconds: 5

    periodSeconds: 10

    successThreshold: 1

    timeoutSeconds: 1

  name: primary

  nodeAffinityPreset:

    key: ""

    type: ""

    values: []

  nodeSelector: {}

  pdb:

    create: false

    maxUnavailable: ""

    minAvailable: 1

  persistence:

    accessModes:

    - ReadWriteOnce

    annotations: {}

    enabled: true

    existingClaim: ""

    selector: {}

    size: 8Gi

    storageClass: ""

  podAffinityPreset: ""

  podAnnotations: {}

  podAntiAffinityPreset: soft

  podLabels: {}

  podManagementPolicy: ""

  podSecurityContext:

    enabled: true

    fsGroup: 1001

  priorityClassName: ""

  readinessProbe:

    enabled: true

    failureThreshold: 3

    initialDelaySeconds: 5

    periodSeconds: 10

    successThreshold: 1

    timeoutSeconds: 1

  resources:

    limits: {}

    requests: {}

  schedulerName: ""

  service:

    annotations: {}

    clusterIP: ""

    externalTrafficPolicy: Cluster

    extraPorts: []

    loadBalancerIP: ""

    loadBalancerSourceRanges: []

    nodePorts:

      mysql: ""

    ports:

      mysql: 3306

    sessionAffinity: None

    sessionAffinityConfig: {}

    type: ClusterIP

  sidecars: []

  startupProbe:

    enabled: true

    failureThreshold: 10

    initialDelaySeconds: 15

    periodSeconds: 10

    successThreshold: 1

    timeoutSeconds: 1

  terminationGracePeriodSeconds: ""

  tolerations: []

  topologySpreadConstraints: []

  updateStrategy:

    type: RollingUpdate

rbac:

  create: false

  rules: []

secondary:

  affinity: {}

  args: []

  command: []

  configuration: |-

    [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

  containerSecurityContext:

    enabled: true

    runAsNonRoot: true

    runAsUser: 1001

  customLivenessProbe: {}

  customReadinessProbe: {}

  customStartupProbe: {}

  existingConfigmap: ""

  extraEnvVars: []

  extraEnvVarsCM: ""

  extraEnvVarsSecret: ""

  extraFlags: ""

  extraVolumeMounts: []

  extraVolumes: []

  hostAliases: []

  initContainers: []

  lifecycleHooks: {}

  livenessProbe:

    enabled: true

    failureThreshold: 3

    initialDelaySeconds: 5

    periodSeconds: 10

    successThreshold: 1

    timeoutSeconds: 1

  name: secondary

  nodeAffinityPreset:

    key: ""

    type: ""

    values: []

  nodeSelector: {}

  pdb:

    create: false

    maxUnavailable: ""

    minAvailable: 1

  persistence:

    accessModes:

    - ReadWriteOnce

    annotations: {}

    enabled: true

    selector: {}

    size: 8Gi

    storageClass: ""

  podAffinityPreset: ""

  podAnnotations: {}

  podAntiAffinityPreset: soft

  podLabels: {}

  podManagementPolicy: ""

  podSecurityContext:

    enabled: true

    fsGroup: 1001

  priorityClassName: ""

  readinessProbe:

    enabled: true

    failureThreshold: 3

    initialDelaySeconds: 5

    periodSeconds: 10

    successThreshold: 1

    timeoutSeconds: 1

  replicaCount: 1

  resources:

    limits: {}

    requests: {}

  schedulerName: ""

  service:

    annotations: {}

    clusterIP: ""

    externalTrafficPolicy: Cluster

    extraPorts: []

    loadBalancerIP: ""

    loadBalancerSourceRanges: []

    nodePorts:

      mysql: ""

    ports:

      mysql: 3306

    sessionAffinity: None

    sessionAffinityConfig: {}

    type: ClusterIP

  sidecars: []

  startupProbe:

    enabled: true

    failureThreshold: 15

    initialDelaySeconds: 15

    periodSeconds: 10

    successThreshold: 1

    timeoutSeconds: 1

  terminationGracePeriodSeconds: ""

  tolerations: []

  topologySpreadConstraints: []

  updateStrategy:

    type: RollingUpdate

serviceAccount:

  annotations: {}

  automountServiceAccountToken: true

  create: true

  name: ""

volumePermissions:

  enabled: false

  image:

    digest: ""

    pullPolicy: IfNotPresent

    pullSecrets: []

    registry: docker.io

    repository: bitnami/bitnami-shell

    tag: 11-debian-11-r32

  resources: {}

--- **important** – we can pass customized values for above.