Genesys Pulse Private Edition Guide

Shared Provisioning

5/17/2026

Contents

Learn how to configure Genesys Pulse.

Related documentation:
-
-
-
-

RSS:
- For private edition

Prerequisites

Before you begin the steps on this page, complete the instructions on Before you begin.

Information you require for shared provisioning:

- Versions:
    - = 100.0.000.0015
    - = 100.0.000+0015
- K8S namespace pulse
- Project Name pulse
- Postgres credentials
    -
    -
    -
    -
    -
    -
- Docker credentials
    -
    -
- Redis credentials
    -
    -
    -
    -
- Tenant service variables
    -
    -
    -

- •
  - • GAuth/GWS service variables
    - •
    - •
    - •
    - •
    - •
    - •

- • Storage class:
  - •

- • Pulse:
  - •

    - • (optional)

Single namespace

Single namespace deployments have a software-defined networking (SDN) with multitenant mode, where namespaces are network isolated. If you plan to deploy Pulse into the single namespace, ensure that your environment meets the following requirements for inputs:

- • Back-end services deployed into the single namespace must include the string *pulse*:

- • The hostname used for Ingress must be unique, and must include the string *pulse*:

- • Internal service-to-service traffic must use the service endpoints, rather than the Ingress Controller:

Override Helm chart values

For more information about overriding Helm chart values, see the suite-level documentation: Overriding Helm chart values.

Parameter	Description	Default	Valid values
service.port	Designer service to be exposed.	8888	A valid port.

Deployment

init Helm chart

Use this chart to initialize the shared PostgreSQL database.

Get init Helm chart

Run the following commands to get the chart:

```
helm repo update
helm search repo pulsehelmrepo/init
```

Prepare override-init file (GKE)

Create a file with the following content, entering appropriate values where indicated, and save the file as **values-override-init.yaml**:

```
# Default values for init.
# This is a YAML-formatted file.
# Declare variables to be passed into your templates.

image:
  tag: ""
  pullPolicy: IfNotPresent
  registry: ""
  imagePullSecrets: [name: ""]

# tenant identification, or empty for shared deployment
tenants:
  - id:    ""
    name: ""
    key:  ""
    dcu:  ""
    # optional, available starting with 100.0.000.0015
    # enable_agent_management: true

# common configuration.
config:
  # set "true" to create config maps
  createConfigMap: true
  # set "true" to create secrets
  createSecret: true

  # Postgres config - fill when createConfigMap: true
  # Postgres config map name
  postgresConfig: "pulse-postgres-configmap"
  # Postgres hostname
  postgresHost: ""
  # Postgres port
  postgresPort: ""
  # Postgres SSL mode
  postgresEnableSSL: ""

  # Postgres secret config - fill when createSecret: true
  # Postgres User
  postgresUser: ""
  # Postgres Password
  postgresPassword: ""
  # Secret name for postgres
  postgresSecret: "pulse-postgres-secret"
  # Secret key for postgres user
  postgresSecretUser: "META_DB_ADMIN"
  # Secret key for postgres  password
  postgresSecretPassword: "META_DB_ADMINPWD"

  # Redis config - fill when createConfigMap: true
```

```
  # Redis config map name
  redisConfig: "pulse-redis-configmap"
  # Redis host
  redisHost: ""
  # Redis port
  redisPort: ""
  # Redis SSL enabled
  redisEnableSSL: "false"

  # Redis secret config - fill when createSecret: true
  # Password for Redis
  redisKey: ""
  # Secret name for Redis
  redisSecret: "pulse-redis-secret"
  # Secret key for Redis password
  redisSecretKey: "REDIS01_KEY"

  # GWS secret config - fill when createSecret: true
  # Client ID
  gwsClientId: ""
  # Client Secret
  gwsClientSecret: ""
  # Secret name
  gwsSecret: "pulse-gws-secret"
  # Secret key for Client ID
  gwsSecretClientId: "clientId"
  # Secret key for Client Secret
  gwsSecretClientSecret: "clientSecret"

  # fill database name
  dbName: ""
  # set "true" when need @host added for username
  dbUserWithHost: true
  # set "true" for CSI secrets
  mountSecrets: false

## Service account settings
serviceAccount:
  # Specifies whether a service account should be created
  create: false
  # Annotations to add to the service account
  annotations: {}
  # The name of the service account to use.
  # If not set and create is true, a name is generated using the fullname template
  name: ""

## Add annotations to all pods
##
podAnnotations: {}

## Specifies the security context for all Pods in the service
##
podSecurityContext:
    fsGroup: null
    runAsUser: null
    runAsGroup: 0
    runAsNonRoot: true

## Resource requests and limits
## ref: http://kubernetes.io/docs/user-guide/compute-resources/
##
resources:
  limits:
```

```
      memory: 256Mi
      cpu: 200m
   requests:
      memory: 128Mi
      cpu: 100m

## Priority Class
## ref: https://kubernetes.io/docs/concepts/configuration/pod-priority-preemption/
##
priorityClassName: ""

## Node labels for assignment.
## ref: https://kubernetes.io/docs/user-guide/node-selection/
##
nodeSelector: {}

## Tolerations for assignment.
## ref: https://kubernetes.io/docs/concepts/configuration/taint-and-toleration/
##
tolerations: []
```

Prepare override-init file (AKS)

Create a file with the following content, entering appropriate values where indicated, and save the file as **values-override-init.yaml**:

```
# Default values for init.
# This is a YAML-formatted file.
# Declare variables to be passed into your templates.

image:
   tag: ""
   pullPolicy: IfNotPresent
   registry: ""
   imagePullSecrets: [name: ""]

# tenant identification, or empty for shared deployment
tenants:
   - id:    ""
     name: ""
     key:   ""
     dcu:   ""

# common configuration.
config:
   # set "true" to create config maps
   createConfigMap: true
   # set "true" to create secrets
   createSecret: true

   # Postgres config - fill when createConfigMap: true
   # Postgres config map name
   postgresConfig: "pulse-postgres-configmap"
   # Postgres hostname
   postgresHost: ""
   # Postgres port
   postgresPort: ""
   # Postgres SSL mode
   postgresEnableSSL: ""

   # Postgres secret config - fill when createSecret: true
   # Postgres User
```

```
  postgresUser: ""
  # Postgres Password
  postgresPassword: ""
  # Secret name for postgres
  postgresSecret: "pulse-postgres-secret"
  # Secret key for postgres user
  postgresSecretUser: "META_DB_ADMIN"
  # Secret key for postgres  password
  postgresSecretPassword: "META_DB_ADMINPWD"

  # Redis config - fill when createConfigMap: true
  # Redis config map name
  redisConfig: "pulse-redis-configmap"
  # Redis host
  redisHost: ""
  # Redis port
  redisPort: ""
  # Redis SSL enabled
  redisEnableSSL: "false"

  # Redis secret config - fill when createSecret: true
  # Password for Redis
  redisKey: ""
  # Secret name for Redis
  redisSecret: "pulse-redis-secret"
  # Secret key for Redis password
  redisSecretKey: "REDIS01_KEY"

  # GWS secret config - fill when createSecret: true
  # Client ID
  gwsClientId: ""
  # Client Secret
  gwsClientSecret: ""
  # Secret name
  gwsSecret: "pulse-gws-secret"
  # Secret key for Client ID
  gwsSecretClientId: "clientId"
  # Secret key for Client Secret
  gwsSecretClientSecret: "clientSecret"

  # fill database name
  dbName: ""
  # set "true" when need @host added for username
  dbUserWithHost: true
  # set "true" for CSI secrets
  mountSecrets: false

## Service account settings
serviceAccount:
  # Specifies whether a service account should be created
  create: false
  # Annotations to add to the service account
  annotations: {}
  # The name of the service account to use.
  # If not set and create is true, a name is generated using the fullname template
  name: ""

## Add annotations to all pods
##
podAnnotations: {}

## Specifies the security context for all Pods in the service
##
```

```
podSecurityContext: {}

## Resource requests and limits
## ref: http://kubernetes.io/docs/user-guide/compute-resources/
##
resources:
  limits:
    memory: 256Mi
    cpu: 200m
  requests:
    memory: 128Mi
    cpu: 100m

## Priority Class
## ref: https://kubernetes.io/docs/concepts/configuration/pod-priority-preemption/
##
priorityClassName: ""

## Node labels for assignment.
## ref: https://kubernetes.io/docs/user-guide/node-selection/
##
nodeSelector: {}

## Tolerations for assignment.
## ref: https://kubernetes.io/docs/concepts/configuration/taint-and-toleration/
##
tolerations: []
```

Install init Helm chart

Run:

```
helm upgrade --install pulse-init pulsehelmrepo/init --wait --wait-for-jobs --version= --
namespace=pulse -f values-override-init.yaml
```

If the installation is successful, the command finishes with exit code 0.

Validate init Helm chart

To validate Helm chart initialization, run the following command:

```
kubectl get pods -n=pulse -l "app.kubernetes.io/name=init,app.kubernetes.io/instance=pulse-
init"
NAME                    READY   STATUS      RESTARTS   AGE
pulse-init-job-5669c    0/1     Completed   0          79m
```

If the initialization was successful, the Pulse-init job has a Status of Completed.

Install pulse Helm chart

Use this chart to install the shared part.

Get pulse Helm chart

```
helm repo update
helm search repo pulsehelmrepo/pulse
```

Prepare override-pulse file (GKE)

Create a file with the following content, entering appropriate values where indicated, and save the file as **values-override-pulse.yaml**:

```
# Default values for pulse.
# This is a YAML-formatted file.
# Declare variables to be passed into your templates.

image:
  tag: ""
  pullPolicy: IfNotPresent
  registry: ""
  imagePullSecrets: [name: ""]

replicaCount: 2

# common configuration.
config:
  dbName: ""
  # set "true" when need @host added for username
  dbUserWithHost: true
  # set "true" for CSI secrets
  mountSecrets: false
  # Postgres config map name
  postgresConfig: "pulse-postgres-configmap"
  # Postgres secret name
  postgresSecret: "pulse-postgres-secret"
  # Postgres secret key for user
  postgresSecretUser: "META_DB_ADMIN"
  # Postgres secret key for password
  postgresSecretPassword: "META_DB_ADMINPWD"
  # Redis config map name
  redisConfig: "pulse-redis-configmap"
  # Redis secret name
  redisSecret: "pulse-redis-secret"
  # Redis secret key for access key
  redisSecretKey: "REDIS01_KEY"
  # GAuth secret name
  gwsSecret: "pulse-gws-secret"
  # GAuth secret key for client_id
  gwsSecretClientId: "clientId"
  # GAuth secret key for client_secret
  gwsSecretClientSecret: "clientSecret"

# monitoring settings
monitoring:
  # enable the Prometheus metrics endpoint
  enabled: false
  # port is
  # HTTP path is
  # additional annotations required for monitoring PODs
  # you can reference values of other variables as {{.Values.variable.full.name}}
  podAnnotations: {}
    # prometheus.io/scrape: "true"
    # prometheus.io/port: "{{.Values.options.managementPort}}"
    # prometheus.io/path:
"{{.Values.options.managementContext}}{{.Values.options.prometheusEndpoint}}"
  serviceMonitor:
    # enables ServiceMonitor creation
    enabled: false
    # interval at which metrics should be scraped
```

```
      scrapeInterval: 30s
      # timeout after which the scrape is ended
      scrapeTimeout:
      # namespace of the ServiceMonitor, defaults to the namespace of the service
      namespace:
      additionalLabels: {}

# common log configuration
log:
  # target directory where log will be stored, leave empty for default
  logDir: ""
  # path where volume will be mounted
  volumeMountPath: /data/log
  # log volume type: none | hostpath | pvc
  volumeType: pvc
  # log volume hostpath, used with volumeType "hostpath"
  volumeHostPath: /mnt/log
  # log PVC parameters, used with volumeType "pvc"
  pvc:
    name: pulse-logs
    accessModes:
      - ReadWriteMany
    capacity: 10Gi
    class:

# application options
options:
  authUrl: "https://"
  authUrlInt: "http://"
  gwsUrl: "https://"
  gwsUrlInt: "http://"
  # optional, required for Manage Agents since 100.0.000.0015
  # advancedServiceUrl: "https://"

## Service account settings
serviceAccount:
  # Specifies whether a service account should be created
  create: false
  # Annotations to add to the service account
  annotations: {}
  # The name of the service account to use.
  # If not set and create is true, a name is generated using the fullname template
  name: ""

## Add annotations to all pods
##
podAnnotations: {}

## Add labels to all pods
##
podLabels: {}

## Specifies the security context for all Pods in the service
##
podSecurityContext:
    fsGroup: null
    runAsUser: null
    runAsGroup: 0
    runAsNonRoot: true

## Ingress configuration
ingress:
  enabled: true
```

```
  annotations: {}
    # kubernetes.io/ingress.class: nginx
    # kubernetes.io/tls-acme: "true"
    ## recommended to increase proxy-body-size size
    # nginx.ingress.kubernetes.io/proxy-body-size: 5m
  hosts:
    - host: ""
      paths: [/]
  tls: []
  #  - secretName: chart-example-tls
  #    hosts:
  #      - chart-example.local

gateway:
  enabled: false

## Resource requests and limits
## ref: http://kubernetes.io/docs/user-guide/compute-resources/
##
resources:
  limits:
    memory: 4Gi
    cpu: 1
  requests:
    memory: 650Mi
    cpu: 100m

## HPA Settings
## Not supported in this release!
hpa:
  enabled: false

## Priority Class
## ref: https://kubernetes.io/docs/concepts/configuration/pod-priority-preemption/
##
priorityClassName: ""

## Node labels for assignment.
## ref: https://kubernetes.io/docs/user-guide/node-selection/
##
nodeSelector: {}

## Tolerations for assignment.
## ref: https://kubernetes.io/docs/concepts/configuration/taint-and-toleration/
##
tolerations: []

## Pod Disruption Budget Settings
podDisruptionBudget:
  enabled: false

## Affinity for assignment.
## Ref: https://kubernetes.io/docs/concepts/configuration/assign-pod-node/#affinity-and-anti-
affinity
##
affinity: {}

# control network policies
networkPolicies:
  enabled: false
```

Prepare override-pulse file (AKS)

Create a file with the following content, entering appropriate values where indicated, and save the file as **values-override-pulse.yaml**:

```
# Default values for pulse.
# This is a YAML-formatted file.
# Declare variables to be passed into your templates.

image:
  tag: ""
  pullPolicy: IfNotPresent
  registry: ""
  imagePullSecrets: [name: ""]

replicaCount: 2

# common configuration.
config:
  dbName: ""
  # set "true" when need @host added for username
  dbUserWithHost: true
  # set "true" for CSI secrets
  mountSecrets: false
  # Postgres config map name
  postgresConfig: "pulse-postgres-configmap"
  # Postgres secret name
  postgresSecret: "pulse-postgres-secret"
  # Postgres secret key for user
  postgresSecretUser: "META_DB_ADMIN"
  # Postgres secret key for password
  postgresSecretPassword: "META_DB_ADMINPWD"
  # Redis config map name
  redisConfig: "pulse-redis-configmap"
  # Redis secret name
  redisSecret: "pulse-redis-secret"
  # Redis secret key for access key
  redisSecretKey: "REDIS01_KEY"
  # GAuth secret name
  gwsSecret: "pulse-gws-secret"
  # GAuth secret key for client_id
  gwsSecretClientId: "clientId"
  # GAuth secret key for client_secret
  gwsSecretClientSecret: "clientSecret"

# monitoring settings
monitoring:
  # enable the Prometheus metrics endpoint
  enabled: false
  # port is
  # HTTP path is
  # additional annotations required for monitoring PODs
  # you can reference values of other variables as {{.Values.variable.full.name}}
  podAnnotations: {}
    # prometheus.io/scrape: "true"
    # prometheus.io/port: "{{.Values.options.managementPort}}"
    # prometheus.io/path:
"{{.Values.options.managementContext}}{{.Values.options.prometheusEndpoint}}"
  serviceMonitor:
    # enables ServiceMonitor creation
    enabled: false
    # interval at which metrics should be scraped
```

```
        scrapeInterval: 30s
        # timeout after which the scrape is ended
        scrapeTimeout:
        # namespace of the ServiceMonitor, defaults to the namespace of the service
        namespace:
        additionalLabels: {}

# common log configuration
log:
  # target directory where log will be stored, leave empty for default
  logDir: ""
  # path where volume will be mounted
  volumeMountPath: /data/log
  # log volume type: none | hostpath | pvc
  volumeType: pvc
  # log volume hostpath, used with volumeType "hostpath"
  volumeHostPath: /mnt/log
  # log PVC parameters, used with volumeType "pvc"
  pvc:
    name: pulse-logs
    accessModes:
      - ReadWriteMany
    capacity: 10Gi
    class:

# application options
options:
  authUrl: "https://"
  authUrlInt: "http://"
  gwsUrl: "https://"
  gwsUrlInt: "http://"

## Service account settings
serviceAccount:
  # Specifies whether a service account should be created
  create: false
  # Annotations to add to the service account
  annotations: {}
  # The name of the service account to use.
  # If not set and create is true, a name is generated using the fullname template
  name: ""

## Add annotations to all pods
##
podAnnotations: {}

## Add labels to all pods
##
podLabels: {}

## Specifies the security context for all Pods in the service
##
podSecurityContext: {}

## Ingress configuration
ingress:
  enabled: true
  annotations:
    kubernetes.io/ingress.class: nginx
    kubernetes.io/tls-acme: "true"
  enabled: true
  hosts:
  - host:
```

```
    paths:
    - /
  tls:
  - hosts:
    -
    secretName: pulse-cert-tls

gateway:
  enabled: false

## Resource requests and limits
## ref: http://kubernetes.io/docs/user-guide/compute-resources/
##
resources:
  limits:
    memory: 4Gi
    cpu: 1
  requests:
    memory: 650Mi
    cpu: 100m

## HPA Settings
## Not supported in this release!
hpa:
  enabled: false

## Priority Class
## ref: https://kubernetes.io/docs/concepts/configuration/pod-priority-preemption/
##
priorityClassName: ""

## Node labels for assignment.
## ref: https://kubernetes.io/docs/user-guide/node-selection/
##
nodeSelector: {}

## Tolerations for assignment.
## ref: https://kubernetes.io/docs/concepts/configuration/taint-and-toleration/
##
tolerations: []

## Pod Disruption Budget Settings
podDisruptionBudget:
  enabled: false

## Affinity for assignment.
## Ref: https://kubernetes.io/docs/concepts/configuration/assign-pod-node/#affinity-and-anti-
affinity
##
affinity: {}

# control network policies
networkPolicies:
  enabled: false
```

Install pulse Helm chart

Run:

```
helm upgrade --install pulse pulsehelmrepo/pulse --wait --version=  --namespace=pulse -f
values-override-pulse.yaml
```

If installation is successful, the command finishes with exit code 0.

Validate pulse Helm chart

To list all running Pulse pods, run the following command:

```
kubectl get pods -n=pulse -l "app.kubernetes.io/name=pulse,app.kubernetes.io/instance=pulse"
NAME                      READY   STATUS     RESTARTS   AGE
pulse-648b9d6666-f5d84    1/1     Running    0          22m
pulse-648b9d6666-kqhs6    1/1     Running    0          68m
```

Install pulse-manage-agents helm chart

Use this chart to install the optional shared service for Manage Agents functionality. This feature is available starting with release 100.0.000.0015.

Get pulse helm chart

Run the following commands to get the chart:

```
helm repo update
helm search repo /pulse-manage-agents
```

Prepare override file

Create a file with the following content, entering appropriate values where indicated, and save the file as **values-override-pulse-manage-agents.yaml**:

```
# Default values for pulse-manage-agents.
# This is a YAML-formatted file.
# Declare variables to be passed into your templates.
  image:
  tag: ""
  pullPolicy: IfNotPresent
  registry: ""
  imagePullSecrets: [name: ""]

replicaCount: 2

# monitoring settings
monitoring:
  # enable the Prometheus metrics endpoint
  enabled: false
  # port number of the Prometheus metrics endpoint
  port: 9103
  # additional annotations required for monitoring PODs
  # you can reference values of other variables as {{.Values.variable.full.name}}
  podAnnotations: {}
  # additional annotations required for monitoring Service
  # you can reference values of other variables as {{.Values.variable.full.name}}
  serviceAnnotations: {}
    # prometheus.io/scrape: "true"
    # prometheus.io/port: "{{.Values.monitoring.port}}"
    # prometheus.io/path: "/metrics"
```

```
    serviceMonitor:
      # enables ServiceMonitor creation
      enabled: false
      # interval at which metrics should be scraped
      scrapeInterval: 30s
      # timeout after which the scrape is ended
      scrapeTimeout:
      # namespace of the ServiceMonitor, defaults to the namespace of the service
      namespace:
      additionalLabels: {}
    dashboards:
      # enables Grafana dashboards
      enabled: true
      # namespace of the ConfigMap with Grafana dashboards,
      # defaults to the namespace of the POD
      namespace:
      additionalLabels: {}
    alerts:
      # enables alert rules
      enabled: false
      # namespace of the alert rules, defaults to the namespace of the POD
      namespace:
      additionalLabels: {}

# common log configuration
log:
  # target directory where log will be stored, leave empty for default
  logDir: ""
  # path where volume will be mounted
  volumeMountPath: /data/log
  # log volume type: none | hostpath | pvc
  volumeType: pvc
  # log volume hostpath, used with volumeType "hostpath"
  volumeHostPath: /mnt/log
  # log PVC parameters, used with volumeType "pvc"
  pvc:
    name: pulse-manage-agents-logs
    accessModes:
      - ReadWriteMany
    capacity: 10Gi
    class:

# application options
env:
  CLOUD: MULTICLOUD
  amGwsUrlInt:
  amGwsPort: 80
  amGauthUrlInt:

# CORS functionality checks that Origin header in OPTIONS command is one of these.
whitelistedOrigins:
  - https://

## Service account settings
serviceAccount:
  # Specifies whether a service account should be created
  create: false
  # Annotations to add to the service account
  annotations: {}
  # The name of the service account to use.
  # If not set and create is true, a name is generated using the fullname template
  name: ""
```

```
## Add annotations to all pods
##
podAnnotations: {}

## Add labels to all pods
##
podLabels: {}

## Specifies the security context for all Pods in the service
##
podSecurityContext:
  runAsNonRoot: true
  runAsUser: 500
  runAsGroup: 500
  fsGroup: 0

## Ingress configuration
ingress:
  enabled: true
  annotations: {}
    # kubernetes.io/ingress.class: nginx
    # kubernetes.io/tls-acme: "true"
    ## recommended to increase proxy-body-size size
    # nginx.ingress.kubernetes.io/proxy-body-size: 5m
  hosts:
    - host: ""
      paths: [/]
  tls: []
  #  - secretName: chart-example-tls
  #    hosts:
  #      - chart-example.local

gateway:
  enabled: false

## Resource requests and limits
## ref: http://kubernetes.io/docs/user-guide/compute-resources/
## resources:
  limits:
    memory: 256Mi
    cpu: 200m
  requests:
    memory: 155Mi
    cpu: 10m

## HPA Settings
## Not supported in this release!
hpa:
  enabled: false

## Priority Class
## ref: https://kubernetes.io/docs/concepts/configuration/pod-priority-preemption/
##
priorityClassName: ""

## Node labels for assignment.
## ref: https://kubernetes.io/docs/user-guide/node-selection/
##
nodeSelector: {}

## Tolerations for assignment.
## ref: https://kubernetes.io/docs/concepts/configuration/taint-and-toleration/
##
```

```
tolerations: []

## Pod Disruption Budget Settings
podDisruptionBudget:
  enabled: false

## Affinity for assignment.
## Ref: https://kubernetes.io/docs/concepts/configuration/assign-pod-node/#affinity-and-anti-
affinity
##
affinity: {}

# control network policies
networkPolicies:
  enabled: false
```

Install pulse-manage-agents helm chart

Run the following command to install pulse-manage-agents:

```
helm upgrade --install pulse pulsehelmrepo/pulse-manage-agents --wait --version=   --
namespace=pulse -f values-override-pulse-manage-agents.yaml
```

If the installation is successful, the command finishes with exit code 0.

Validate pulse-manage-agents helm chart

To validate Helm chart initialization, run the following command:

```
oc get pods -n=pulse -l "app.kubernetes.io/name=pulse-manage-agents,app.kubernetes.io/
instance=pulse-manage-agents"
```

The output should show that all pulse-manage-agents pods are Running:

```
NAME                                   READY   STATUS    RESTARTS   AGE
pulse-manage-agents-b999c7758-hscfz    1/1     Running   0          5d18h
pulse-manage-agents-b999c7758-s67zt    1/1     Running   0          5d18h
```

Validation

Use the following procedures to validate the deployment.

Check logs for error

1. To check the log files, run the following command:

   ```
   kubectl get pods
    os logs
   ```

   Where: is the pod identifier.

Health validation

1. To download the health validation metrics, run the following command:
   GET /actuator/metrics/pulse.health.all

2. Open two Command Prompt windows, and run the following commands:

   1. **Console 1**:

      ```
      kubectl get pods --namespace pulse -l "app.kubernetes.io/
      name=pulse,app.kubernetes.io/instance=pulse" -o
      jsonpath="{.items[0].metadata.name}"
      kubectl --namespace pulse port-forward  8090:8090
      ```

   2. **Console 2**:

      ```
      curl -X GET http://127.0.0.1:8090/actuator/metrics/pulse.health.all       -H
      'Content-Type: application/json'
      ```

      If Pulse is running correctly and can connect to Redis and PostgreSQL, the following results appear:

   - http response is 200

   - json response has `measurements.statistic.value` of 1.0, for example:

     ```
     {
         "name": "pulse.health.all",
         "description": "Provides overall application status",
         "baseUnit": "Boolean",
         "measurements": [
           {
             "statistic": "VALUE",
             "value": 1
           }
         ],
         "availableTags": [
           {
             "tag": "deployment.code",
             "values": [
               "pulse"
             ]
           },
           {
             "tag": "application.name",
             "values": [
               "pulse"
             ]
           }
         ]
     }
     ```

Troubleshooting

If you encounter problems during deployment, examine the init Helm and Pulse Helm manifests.

Check init Helm manifests

To output init Helm manifest files into the **helm-template** directory, run the following command:

```
helm template --version= --namespace=pulse --debug --output-dir helm-template init
pulsehelmrepo/init -f values-override-init.yaml.yaml
```

Where: is the Helm chart version.

Check Pulse Helm manifests

To output Pulse Helm manifest files into the **helm-template** directory, run the following command:

```
helm template --version= --namespace=pulse --debug --output-dir helm-template pulse
pulsehelmrepo/pulse -f values-override-pulse.yaml
```

Where: is the Helm chart version.