



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Genesys Pulse Private Edition Guide

Table of Contents

Overview	
About Genesys Pulse	6
Architecture	8
High availability and disaster recovery	15
Configure and deploy	
Before you begin	16
Shared Provisioning	22
Tenant Provisioning	42
Deploy Pulse Web Service	72
Upgrade, roll back, or uninstall	
Upgrade, roll back, or uninstall Genesys Pulse	77
Observability	
Observability in Genesys Pulse	83
Pulse metrics and alerts	87
Tenant Permissions Service metrics and alerts	90
Tenant Load Distribution Server (LDS) metrics and alerts	93
Tenant Data Collection Unit (DCU) metrics and alerts	97
Logging	103

Contents

- [1 Overview](#)
- [2 Configure and deploy](#)
- [3 Upgrade, roll back, or uninstall](#)
- [4 Observability](#)

Find links to all the topics in this guide.

Related documentation:

-
-
-

RSS:

- [For private edition](#)

Genesys Pulse is a service available with the Genesys Multicloud CX private edition offering.

Overview

Learn more about Genesys Pulse, its architecture, and how to support high availability and disaster recovery.

- [About Genesys Pulse](#)
- [Architecture](#)
- [High availability and disaster recovery](#)

Configure and deploy

Find out how to configure and deploy Genesys Pulse.

- [Before you begin](#)
- [Shared Provisioning](#)
- [Tenant Provisioning](#)
- [Deploy Pulse Web Service](#)

Upgrade, roll back, or uninstall

Find out how to upgrade, roll back, or uninstall Genesys Pulse.

-
- Upgrade, roll back, or uninstall Genesys Pulse
-

Observability

Learn how to monitor Genesys Pulse with metrics and logging.

- Observability in Genesys Pulse
 - Pulse metrics and alerts
 - Tenant Permissions Service metrics and alerts
 - Tenant Load Distribution Server (LDS) metrics and alerts
 - Tenant Data Collection Unit (DCU) metrics and alerts
 - Logging
-

About Genesys Pulse

Contents

- [1 Supported Kubernetes platforms](#)

Learn about Genesys Pulse and how it works in Genesys Multicloud CX private edition.

Related documentation:

-
-
-
-

RSS:

- [For private edition](#)

Genesys Pulse is a widget-based performance dashboard solution for monitoring contact center resources in real time.

Supported Kubernetes platforms

Genesys Engagement Service (GES) is supported on the following Kubernetes platforms:

- Google Kubernetes Engine (GKE)
- Azure Kubernetes Service (AKS)

For more information about when support was introduced, see the Real-Time Reporting (Genesys Pulse) Release Notes.

Architecture

Contents

- [1 Introduction](#)
- [2 Architecture diagram — Connections](#)
- [3 Connections table](#)

Learn about Genesys Pulse architecture

Related documentation:

-
-
-
-

RSS:

- [For private edition](#)

Introduction

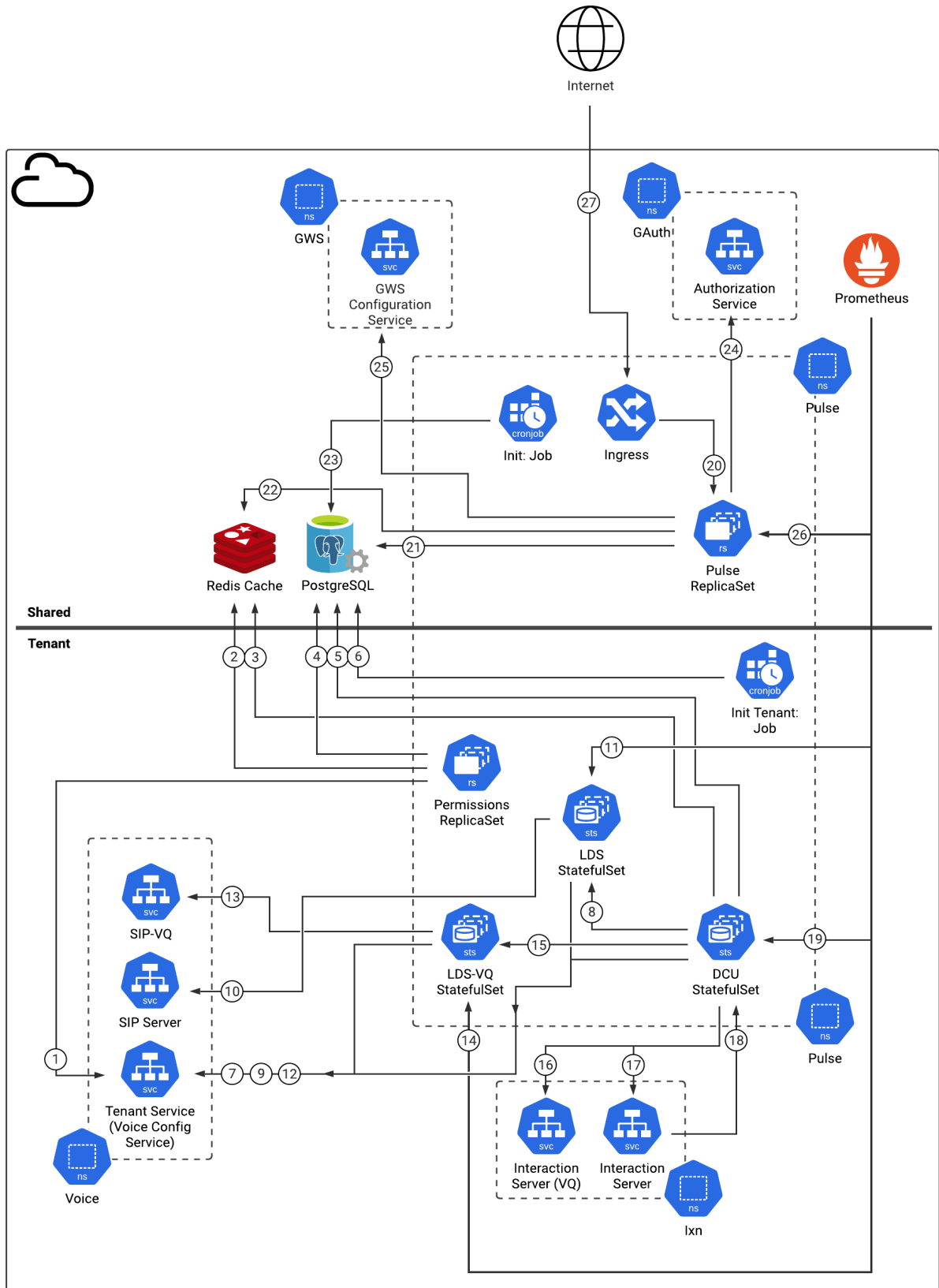
The Connections table, which follows the architecture diagram, provides information about the objects and connections shown in the diagram.

For information about the overall architecture of Genesys Multicloud CX private edition, see the high-level Architecture page.

See also High availability and disaster recovery for information about high availability/disaster recovery architecture.

Architecture diagram — Connections

The numbers on the connection lines refer to the connection numbers in the table that follows the diagram. The direction of the arrows indicates where the connection is initiated (the source) and where an initiated connection connects to (the destination), from the point of view of Genesys Pulse as a service in the network.



Connections table

The connection numbers refer to the numbers on the connection lines in the diagram. The **Source**, **Destination**, and **Connection Classification** columns in the table relate to the direction of the arrows in the Connections diagram above: The source is where the connection is initiated, and the destination is where an initiated connection connects to, from the point of view of Genesys Pulse as a service in the network. *Egress* means the Genesys Pulse service is the source, and *Ingress* means the Genesys Pulse service is the destination. *Intra-cluster* means the connection is between services in the cluster.

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
1	Permissions ReplicaSet	Tenant Service (Voice Config Service)	TCP	8888	Ingress	Permissions Service retrieves configuration data (such as Agents, Queues), in order to check user access permissions.
2	Permissions ReplicaSet	Redis	TCP	6380	Intra-cluster	Permissions Service stores information about user permissions to access configuration objects.
3	DCU StatefulSet	Redis	TCP	6380	Intra-cluster	Pulse Data Collection Unit (DCU) stores current and historical statistic values.
4	Permissions ReplicaSet	PostgreSQL	TCP	5432	Intra-cluster	Permissions Service reads Pulse metadata to check access permissions.
5	DCU StatefulSet	PostgreSQL	TCP	5432	Intra-cluster	Pulse DCU reads Pulse metadata required for

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
						collecting statistic values.
6	DCU StatefulSet	PostgreSQL	TCP	5432	Intra-cluster	Init Tenant job uses this connection to initialize OOB Widget Templates.
7	DCU StatefulSet	Tenant Service (Voice Config Service)	TCP	8888	Intra-cluster	Pulse DCU retrieves configuration data.
8	DCU StatefulSet	LDS StatefulSet	TCP	8000	Intra-cluster	Pulse DCU subscribes for events from Voice SIP Service.
9	DCU StatefulSet	Tenant Service (Voice Config Service)	TCP	8888	Intra-cluster	Pulse Load Distribution Server (LDS) retrieves configuration data.
10	LDS StatefulSet	SIP Server	TCP	8000	Intra-cluster	Pulse LDS broadcasts events from Voice SIP Service to connected clients.
11	Prometheus	LDS StatefulSet	HTTP	9091	Intra-cluster	Metrics for monitoring and alerting with Prometheus.
12	LDS-VQ StatefulSet	Tenant Service (Voice Config Service)	TCP	8888	Intra-cluster	Pulse LDS VQ retrieves configuration data.
13	LDS-VQ StatefulSet	SIP-VQ	TCP	8000	Intra-cluster	Pulse LDS broadcasts events from Voice SIP VQ Service to connected clients.

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
14	Prometheus	LDS-VQ StatefulSet	HTTP	9091	Intra-cluster	Metrics for monitoring and alerting with Prometheus.
15	DCU StatefulSet	LDS-VQ StatefulSet	TCP	8000	Intra-cluster	Pulse DCU subscribes for events from Voice SIP VQ Service.
16	DCU StatefulSet	Interaction Server (VQ)	TCP	7122	Intra-cluster	Pulse DCU subscribes for events from the Interaction VQ Server.
17	DCU StatefulSet	Interaction Server	TCP	7120	Intra-cluster	Pulse DCU subscribes for events from the Interaction Server.
18	Interaction Server	DCU StatefulSet	TCP	2060	Intra-cluster	Interaction Server connection to Pulse DCU.
19	Prometheus	DCU StatefulSet	HTTP	9091	Intra-cluster	Metrics for monitoring and alerting with Prometheus.
20	Ingress controller	Pulse ReplicaSet	HTTP	8080	Intra-cluster	Application Gateway connection to Pulse.
21	Pulse ReplicaSet	PostgreSQL	TCP	5432	Intra-cluster	Pulse reads and writes metadata (Dashboards, Widgets, and Widget Templates).
22	Pulse ReplicaSet	Redis	TCP	6380	Intra-cluster	Pulse reads current and historical statistic values, and

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
						reads information about user permissions to access configuration objects.
23	Init Job	PostgreSQL	TCP	5432	Intra-cluster	Init job uses this connection to initialize or update the database schema.
24	Pulse ReplicaSet	Authorization Service	HTTP	80	Intra-cluster	Pulse queries the Genesys Authentication Service to validate user identity and obtain privilege information for the authenticated user.
25	Pulse ReplicaSet	GWS Configuration Service	HTTP	80	Intra-cluster	Pulse queries the GWS Configuration Service to read configuration data (such as Agents, Queues).
26	Prometheus	Pulse ReplicaSet	HTTP	8090	Intra-cluster	Metrics for monitoring and alerting with Prometheus.
27	Public Internet	Ingress	HTTPS	443	Intra-cluster	Inbound web traffic.

High availability and disaster recovery

Find out how this service provides disaster recovery in the event the service goes down.

Related documentation:

-
-
-
-

RSS:

- [For private edition](#)

Service	High Availability	Disaster Recovery	Where can you host this service?
Genesys Pulse	N = 2 (active-active)	Pilot light	Primary unit only

See High Availability information for all services: [High availability and disaster recovery](#)

Genesys Pulse pods run in an Active-Active High Availability mode. Configure each pod with at least two replicas, and at least one replica in each Availability Zone (AZ).

Before you begin

Contents

- [1 Limitations and assumptions](#)
- [2 Download the Helm charts](#)
 - [2.1 Genesys Pulse Containers](#)
 - [2.2 Genesys Pulse Helm Charts](#)
- [3 Third-party prerequisites](#)
- [4 Storage requirements](#)
 - [4.1 Logs Volume](#)
 - [4.2 Genesys Pulse Collector Health Volume](#)
 - [4.3 Stat Server Backup Volume](#)
- [5 Network requirements](#)
- [6 Genesys dependencies](#)
- [7 GDPR support](#)

Find out what to do before deploying Genesys Pulse.

Related documentation:

-
-
-

RSS:

- [For private edition](#)

Limitations and assumptions

There are no known limitations.

Download the Helm charts

For more information about how to download the Helm charts in Jfrog Edge, see the suite-level documentation: [Downloading your Genesys Multicloud CX containers](#)

To learn what Helm chart version you must download for your release, see [Helm charts and containers for Genesys Pulse](#)

Genesys Pulse Containers

Container	Description	Docker Path
collector	Genesys Pulse Collector	/pulse/collector:
cs_proxy	Configuration Server Proxy	/pulse/cs_proxy:
init	Init container, used for DB initialization	/pulse/init:
lds	Load Distribution Server (LDS)	/pulse/lds:
monitor_dcu_push_agent	Provides monitoring data from Stat Server and Genesys Pulse Collector	/pulse/monitor_dcu_push_agent:
monitor_lds_push_agent	Provides monitoring data from LDS	/pulse/monitor_lds_push_agent:

Before you begin

Container	Description	Docker Path
pulse	Genesys Pulse Backend	/pulse/pulse:
ss	Stat Server	/pulse/ss:
userpermissions	User Permissions service	/pulse/userpermissions:

Genesys Pulse Helm Charts

Helm Chart	Containers	Shared	Helm Path
Init	init	yes	/init-.tgz
Pulse	pulse	yes	/pulse-.tgz
LDS	cs_proxy, lds, monitor_lds_push_agent		/lds-.tgz
DCU	cs_proxy, ss, collector, monitor_dcu_push_agent		/dcu-.tgz
Permissions	cs_proxy, userpermissions		/permissions-.tgz
Init Tenant	init		/init-tenant-.tgz
Monitor	-	yes	/monitor-.tgz

Third-party prerequisites

Appropriate CLI must be installed.

For more information about setting up your Genesys Multicloud CX private edition platform, see [Software requirements](#).

Third-party services

Name	Version	Purpose	Notes
PostgreSQL	11.x	Relational database.	
Redis	6.x	Used for caching. Only distributions of Redis that support Redis cluster mode are supported, however, some services may not support cluster mode.	Note: Pulse does not currently support Redis Cluster; use Redis in stand-alone mode.
A container image registry and Helm chart repository		Used for downloading Genesys containers and Helm charts into the customer's repository to support a CI/CD pipeline. You can use any Docker OCI compliant registry.	

Before you begin

Name	Version	Purpose	Notes
Load balancer		VPC ingress. For NGINX Ingress Controller, a single regional Google external network LB with a static IP and wildcard DNS entry will pass HTTPS traffic to NGINX Ingress Controller which will terminate SSL traffic and will be setup as part of the platform setup.	

Storage requirements

Logs Volume

Persistent Volume	Size	Type	IOPS	POD	Containers	Critical	Backup needed
pulse-dcu-logs	10Gi	RW	high	DCU	csproxy, collector, statserver	Y	Y
pulse-lds-logs	10Gi	RW	high	lds	csproxy, lds	Y	Y
pulse-permissions-logs	10Gi	RW	high	permissions	csproxy, permissions	Y	Y
pulse-logs	10Gi	RW	high	pulse	pulse	Y	Y

The logs volume stores log files:

- To use the persistent volume, set the log.volumeType to the pvc.
- To use the local storage, set the log.volumeType to the hostpath.

Genesys Pulse Collector Health Volume

Local Volume	POD	Containers
collector-health	dcu	collector, monitor-sidecar

Genesys Pulse Collector health volume provides non-persistent storage for store Genesys Pulse Collector health state files for monitoring.

Before you begin

Stat Server Backup Volume

Persistent Volume	Size	Type	IOPS	POD	Containers	Critical	Backup needed
statserver-backup	1Gi	RWO	medium	dcu	statserver	N	N

Stat Server backup volume provides disk space for Stat Server's state backup. The Stat Server backup volume stores the server state between restarts of the container.

Network requirements

No special requirements.

Genesys dependencies

Ensure that the following services are deployed and running before you deploy Genesys Pulse:

- Genesys Authentication:
 - The client secret is provisioned with the following values (see Provision Genesys Authentication):
 - Client ID: "pulse_client"
 - Grant Types: "authorization_code"
 - RedirectURIs: "https://"
 - CORS is allowed for https:// (see Provision Genesys Authentication)
- Genesys Web Services and Applications
- Agent Setup
- Tenant Service:
 - The Tenant UUID (v4) is provisioned, example: "9350e2fc-a1dd-4c65-8d40-1f75a2e080dd"
 - The Tenant service is available as host. For example, in GKE, it is: "tenant-.voice" port: 8888
- Voice Microservice:
 - The Voice service is available as host. For example, in GKE, it is: "tenant-.voice" port: 8000

Important

All services listed must be accessible from within the cluster where Genesys Pulse will

Before you begin

be deployed.

For more information, see [Order of services deployment](#).

GDPR support

Genesys Pulse supports the General Data Protection Regulation (GDPR). See [Genesys Pulse Support for GDPR](#) for details.

Shared Provisioning

Contents

- **1 Prerequisites**
 - **1.1 Single namespace**
- **2 Override Helm chart values**
- **3 Deployment**
 - **3.1 init Helm chart**
 - **3.2 Install pulse Helm chart**
- **4 Install pulse-manage-agents helm chart**
 - **4.1 Get pulse helm chart**
 - **4.2 Prepare override file**
 - **4.3 Install pulse-manage-agents helm chart**
 - **4.4 Validate pulse-manage-agents helm chart**
- **5 Validation**
 - **5.1 Check logs for error**
 - **5.2 Health validation**
- **6 Troubleshooting**
 - **6.1 Check init Helm manifests**
 - **6.2 Check Pulse Helm manifests**
- **7 Configure security**
 - **7.1 Arbitrary UIDs**

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
 -
 -
 -

- GAAuth/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:

```
kubectrl 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"
  # GAAuth secret name
  gwsSecret: "pulse-gws-secret"
  # GAAuth secret key for client_id
  gwsSecretClientId: "clientId"
  # GAAuth 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"
  # GAAuth secret name
  gwsSecret: "pulse-gws-secret"
  # GAAuth secret key for client_id
  gwsSecretClientId: "clientId"
  # GAAuth 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.

Tenant Provisioning

Contents

- **1 Prerequisites**
 - **1.1 Single namespace**
- **2 Tenant provisioning**
 - **2.1 Install init tenant chart**
 - **2.2 Install dcu helm chart**
 - **2.3 Install lds helm chart**
 - **2.4 Install permissions helm chart**
- **3 Troubleshooting**

- Administrator

Learn how to provision Genesys Pulse.

Related documentation:

-
-
-
-

RSS:

- [For private edition](#)

Prerequisites

Before performing the steps described on this page, complete the Before you begin instructions, and ensure that you have the following information:

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

- GAAuth/GWS service variables:
 -
 -
 -
 -
 -
- Storage class:
 -
 -
- Pulse:
 -

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:
 -
 -

Tenant provisioning

Install init tenant chart

Get the **init-tenant** helm chart:

```
helm repo update
helm search repo /init-tenant
```

Prepare the **override file**:

- Update the `values-override-init-tenant.yaml` file (AKS):

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

# * Images
# Replace for your values: registry and secret
```

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

configurator:
  enabled: true
  # set service domain used to access voice service
  # example for GKE VPC case: voice.svc.gke1-uswest1.gcpe002.gencpe.com
  voiceDomain: "voice.svc."
  # set service domain used to access ixn service
  # example for GKE VPC case: ixn.svc.gke1-uswest1.gcpe002.gencpe.com
  ixnDomain: "ixn.svc."
  # set service domain used to access pulse service
  # example for GKE VPC case: pulse.svc.gke1-uswest1.gcpe002.gencpe.com
  pulseDomain: "pulse.svc."
  # set configuration server password, used when create secrets
  cfgUser: "default"
  # set configuration server user, used when create secrets
  cfgPassword: "password"
  # common log configuration
  cfgHost: "tenant-9350e2fc-a1dd-4c65-8d40-1f75a2e080dd.voice.svc."

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: none
  # log volume hostpath, used with volumeType "hostpath"
  volumeHostPath: /mnt/log
  # log PVC parameters, used with volumeType "pvc"
  pvc:
    name: pulse-init-tenant-logs
    accessModes:
      - ReadWriteMany
    capacity: 10Gi
    class:

# * Tenant info
# Replace for your values
tenant:
  # Tenant UUID
  id:
  # Tenant SID (like 0001)
  sid:

# 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"

## 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: []

# * Templates
templates:
  - Agent_Group_Status.gpb
  - Agent_KPIs.gpb
  - Agent_Login.gpb
  - Alert_Widget.gpb
  - Callback_Activity.gpb
  - Campaign_Activity.gpb
  - Campaign_Callback_Status.gpb
  - Campaign_Group_Activity.gpb
  - Campaign_Group_Status.gpb
  - Chat_Agent_Activity.gpb
  - Chat_Queue_Activity.gpb
  - Chat_Service_Level_Performance.gpb
  - Chat_Waiting_Statistics.gpb
  - Email_Agent_Activity.gpb
  - Email_Queue_Activity.gpb
```

- Facebook_Media_Activity.gpb
- IFRAME.gpb
- IWD_Agent_Activity.gpb
- IWD_Queue_Activity.gpb
- Queue_KPIs.gpb
- Queue_Overflow_Reason.gpb
- Static_Text.gpb
- Twitter_Media_Activity.gpb
- eServices_Agent_Activity.gpb
- eServices_Queue_KPIs.gpb

- Update the values-override-init-tenant.yaml file (GKE):

Important

Enable configurator only for configurations in GKE with VPC scoped DNS.

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

* Images
Replace for your values: registry and secret
image:
 tag: ""
 pullPolicy: IfNotPresent
 registry: ""
 imagePullSecrets: [name: ""]

configurator:
 enabled: true
 # set service domain used to access voice service
 # example for GKE VPC case: voice.svc.gke1-uswest1.gcpe002.gcpe.com
 voiceDomain: "voice.svc."
 # set service domain used to access ixn service
 # example for GKE VPC case: ixn.svc.gke1-uswest1.gcpe002.gcpe.com
 ixnDomain: "ixn.svc."
 # set service domain used to access pulse service
 # example for GKE VPC case: pulse.svc.gke1-uswest1.gcpe002.gcpe.com
 pulseDomain: "pulse.svc."
 # set configuration server password, used when create secrets
 cfgUser: "default"
 # set configuration server user, used when create secrets
 cfgPassword: "password"
 # common log configuration
 cfgHost: "tenant-.voice.svc."

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: none
 # log volume hostpath, used with volumeType "hostpath"
 volumeHostPath: /mnt/log
 # log PVC parameters, used with volumeType "pvc"
 pvc:
 name: pulse-init-tenant-logs
```

```
 accessModes:
 - ReadWriteMany
 capacity: 10Gi
 class: nfs-client

* Tenant info
Replace for your values
tenant:
 # Tenant UUID
 id:
 # Tenant SID (like 0001)
 sid:

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"

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: []

* Templates
templates:
 - Agent_Group_Status.gpb
 - Agent_KPIs.gpb
 - Agent_Login.gpb
 - Alert_Widget.gpb
 - Callback_Activity.gpb
 - Campaign_Activity.gpb
 - Campaign_Callback_Status.gpb
 - Campaign_Group_Activity.gpb
 - Campaign_Group_Status.gpb
 - Chat_Agent_Activity.gpb
 - Chat_Queue_Activity.gpb
 - Chat_Service_Level_Performance.gpb
 - Chat_Waiting_Statistics.gpb
 - Email_Agent_Activity.gpb
 - Email_Queue_Activity.gpb
 - Facebook_Media_Activity.gpb
 - IFRAME.gpb
 - IWD_Agent_Activity.gpb
 - IWD_Queue_Activity.gpb
 - Queue_KPIs.gpb
 - Queue_Overflow_Reason.gpb
 - Static_Text.gpb
 - Twitter_Media_Activity.gpb
 - eServices_Agent_Activity.gpb
 - eServices_Queue_KPIs.gpb

```

**Install the init-tenant helm chart:**

To install the init-tenant helm chart, run the following command:

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

If installation is successful, the exit code 0 appears.

**Validate the init-tenant helm chart:**

To validate the init-tenant helm chart, run the following command:

```
kubectl get pods -n="pulse" -l "app.kubernetes.io/name=init-tenant,app.kubernetes.io/instance=pulse-init-tenant"
```

If the deployment was successful, the pulse-init-tenant job is listed as Completed/. For example:

| NAME                            | READY | STATUS    | RESTARTS | AGE   |
|---------------------------------|-------|-----------|----------|-------|
| pulse-init-tenant-100-job-qszgl | 0/1   | Completed | 0        | 2d20h |

## Install dcu helm chart

### Get the dcu helm chart:

```
helm repo update
helm search repo /dcu
```

### Prepare the override file:

- Update the values-override-dcu.yaml file (AKS):

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

replicaCount: ""

* Tenant info
tenant identification, or empty for shared deployment
tenant:
 # Tenant UUID
 id: ""
 # Tenant SID (like 0001)
 sid: ""

* 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-dcu-logs
 accessModes:
 - ReadWriteMany
 capacity: 10Gi
 class:

* Config info
Set your values.
config:
 dbName: ""
 # set "true" when need @host added for username
 dbUserWithHost: true
 mountSecrets: false
 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"
 redisConfig: "pulse-redis-configmap"
 # Redis secret name
 redisSecret: "pulse-redis-secret"
 # Redis secret key for access key
 redisSecretKey: "REDIS01_KEY"
```

```
* Image
container image common settings
image:
 tag: ""
 pullPolicy: IfNotPresent
 registry: ""
 imagePullSecrets: [name: ""]

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: {}

Add labels to all pods
##
podLabels: {}

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: {}
```

```

* Monitoring settings
monitoring:
 # enable the Prometheus metrics endpoint
 enabled: false
 # enable golden signals metrics (not supported for PE)
 goldenSignals:
 enabled: false
 # port number of the Prometheus metrics endpoint
 port: 9091
 # HTTP path to scrape for metrics
 path: /metrics
 # 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.monitoring.port}}"
 # prometheus.io/path: "/metrics"
 podMonitor:
 # enables PodMonitor creation for the POD
 enabled: true
 # interval at which metrics should be scraped
 scrapeInterval: 30s
 # timeout after which the scrape is ended
 scrapeTimeout:
 # namespace of the PodMonitor, defaults to the namespace of the POD
 namespace:
 additionalLabels: {}
 alerts:
 # enables alert rules
 enabled: true
 # alert condition duration
 duration: 5m
 # namespace of the alert rules, defaults to the namespace of the POD
 namespace:
 additionalLabels: {}

#####

* Configuration for the Collector container
collector:
 # resource limits for container
 resources:
 # minimum resource requirements to start container
 requests:
 # minimal amount of memory required to start a container
 memory: "300Mi"
 # minimal CPU to reserve
 cpu: "200m"
 # resource limits for containers
 limits:
 # maximum amount of memory a container can use before being evicted
 # by the OOM Killer
 memory: "4Gi"
 # maximum amount of CPU resources that can be used and should be tuned
 # to reflect
 # what the application can effectively use before needing to be
 # horizontally scaled out
 cpu: "8000m"
 # securityContext: {}

* Configuration for the StatServer container

```

```

statsserver:
 # resource limits for container
 resources:
 # minimum resource requirements to start container
 requests:
 # minimal amount of memory required to start a container
 memory: "300Mi"
 # minimal CPU to reserve
 cpu: "100m"
 # resource limits for containers
 limits:
 # maximum amount of memory a container can use before being evicted
 # by the OOM Killer
 memory: "4Gi"
 # maximum amount of CPU resources that can be used and should be tuned
to reflect
 # what the application can effectively use before needing to be
horizontally scaled out
 cpu: "4000m"
 # securityContext: {}

* Configuration for the monitor sidecar container
monitorSidecar:
 # resource limits for container
 resources:
 # disabled: true
 # minimum resource requirements to start container
 requests:
 # minimal amount of memory required to start a container
 memory: "30Mi"
 # minimal CPU to reserve
 cpu: "2m"
 # resource limits for containers
 limits:
 # maximum amount of memory a container can use before being evicted
 # by the OOM Killer
 memory: "70Mi"
 # maximum amount of CPU resources that can be used and should be tuned
to reflect
 # what the application can effectively use before needing to be
horizontally scaled out
 cpu: "10m"
 # securityContext: {}

#####

* Configuration for the Configuration Server Proxy container
csproxy:
 # define domain for the configuration host
 params:
 cfgHost: "tenant-.voice."
 # resource limits for container
 resources:
 # minimum resource requirements to start container
 requests:
 # minimal amount of memory required to start a container
 memory: "200Mi"
 # minimal CPU to reserve
 cpu: "50m"
 # resource limits for containers
 limits:
 # maximum amount of memory a container can use before being evicted
 # by the OOM Killer

```

```

 memory: "2Gi"
 # maximum amount of CPU resources that can be used and should be tuned
to reflect
 # what the application can effectively use before needing to be
horizontally scaled out
 cpu: "1000m"
 # securityContext: {}

volumeClaims contains persistent volume claims for services
All available storage classes can be found here:
https://github.com/genesysengage/tfm-azure-core-aks/blob/master/k8s-module/
storage.tf
volumeClaims:
 # statserverBackup is storage for statserver backup data
 statserverBackup:
 name: statserver-backup
 accessModes:
 - ReadWriteOnce
 # capacity is storage capacity
 capacity: "1Gi"
 # class is storage class. Must be set explicitly.
 class:

```

- Update the values-override-dcu.yaml file (GKE):

```

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

replicaCount: ""

* Tenant info
tenant identification, or empty for shared deployment
tenant:
 # Tenant UUID
 id: ""
 # Tenant SID (like 0001)
 sid: ""

* 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-dcu-logs
 accessModes:
 - ReadWriteMany
 capacity: 10Gi
 class:

* Config info
Set your values.
config:
 dbName: ""
 # set "true" when need @host added for username
 dbUserWithHost: true

```

```
mountSecrets: false
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"
redisConfig: "pulse-redis-configmap"
Redis secret name
redisSecret: "pulse-redis-secret"
Redis secret key for access key
redisSecretKey: "REDIS01_KEY"

* Image
container image common settings
image:
 tag: ""
 pullPolicy: IfNotPresent
 registry: ""
 imagePullSecrets: [name: ""]

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
 name: ""

Add annotations to all pods
##
podAnnotations: {}

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

Add labels to all pods
##
podLabels: {}

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: {}

* Monitoring settings
monitoring:
 # enable the Prometheus metrics endpoint
 enabled: false
 # enable golden signals metrics (not supported for PE)
 goldenSignals:
 enabled: false
 # port number of the Prometheus metrics endpoint
 port: 9091
 # HTTP path to scrape for metrics
 path: /metrics
 # 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.monitoring.port}}"
 # prometheus.io/path: "/metrics"
 podMonitor:
 # enables PodMonitor creation for the POD
 enabled: true
 # interval at which metrics should be scraped
 scrapeInterval: 30s
 # timeout after which the scrape is ended
 scrapeTimeout:
 # namespace of the PodMonitor, defaults to the namespace of the POD
 namespace:
 additionalLabels: {}
 alerts:
 # enables alert rules
 enabled: true
 # alert condition duration
 duration: 5m
 # namespace of the alert rules, defaults to the namespace of the POD
 namespace:
 additionalLabels: {}

#####

* Configuration for the Collector container
collector:
 # resource limits for container
 resources:
 # minimum resource requirements to start container

```



```

requests:
 # minimal amount of memory required to start a container
 memory: "300Mi"
 # minimal CPU to reserve
 cpu: "200m"
resource limits for containers
limits:
 # maximum amount of memory a container can use before being evicted
 # by the OOM Killer
 memory: "4Gi"
 # maximum amount of CPU resources that can be used and should be tuned
to reflect
 # what the application can effectively use before needing to be
horizontally scaled out
 cpu: "8000m"
securityContext:
runAsUser: 500
runAsGroup: 500

* Configuration for the StatServer container
statsserver:
resource limits for container
resources:
minimum resource requirements to start container
requests:
 # minimal amount of memory required to start a container
 memory: "300Mi"
 # minimal CPU to reserve
 cpu: "100m"
resource limits for containers
limits:
 # maximum amount of memory a container can use before being evicted
 # by the OOM Killer
 memory: "4Gi"
 # maximum amount of CPU resources that can be used and should be tuned
to reflect
 # what the application can effectively use before needing to be
horizontally scaled out
 cpu: "4000m"
securityContext:
runAsUser: 500
runAsGroup: 500

* Configuration for the monitor sidecar container
monitorSidecar:
resource limits for container
resources:
disabled: true
minimum resource requirements to start container
requests:
 # minimal amount of memory required to start a container
 memory: "30Mi"
 # minimal CPU to reserve
 cpu: "2m"
resource limits for containers
limits:
 # maximum amount of memory a container can use before being evicted
 # by the OOM Killer
 memory: "70Mi"
 # maximum amount of CPU resources that can be used and should be tuned
to reflect
 # what the application can effectively use before needing to be
horizontally scaled out

```

```

 cpu: "10m"
 # securityContext:
 # runAsUser: 500
 # runAsGroup: 500

#####

* Configuration for the Configuration Server Proxy container
csproxy:
 # define domain for the configuration host
 params:
 cfgHost: "tenant-.voice."
 # resource limits for container
 resources:
 # minimum resource requirements to start container
 requests:
 # minimal amount of memory required to start a container
 memory: "200Mi"
 # minimal CPU to reserve
 cpu: "50m"
 # resource limits for containers
 limits:
 # maximum amount of memory a container can use before being evicted
 # by the OOM Killer
 memory: "2Gi"
 # maximum amount of CPU resources that can be used and should be tuned
 # to reflect
 # what the application can effectively use before needing to be
 # horizontally scaled out
 cpu: "1000m"
 # securityContext:
 # runAsUser: 500
 # runAsGroup: 500

volumeClaims contains persistent volume claims for services
All available storage classes can be found here:
https://github.com/genesysengage/tfm-azure-core-aks/blob/master/k8s-module/
storage.tf
volumeClaims:
 # statserverBackup is storage for statserver backup data
 statserverBackup:
 name: statserver-backup
 accessModes:
 - ReadWriteOnce
 # capacity is storage capacity
 capacity: "1Gi"
 # class is storage class. Must be set explicitly.
 class:

```

**Install the dcu helm chart**

To install the dcu helm chart, run the following command:

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

**Validate the dcu helm chart**

To validate the dcu helm chart, run the following command:

```
kubectl get pods -n=pulse -l "app.kubernetes.io/name=dcu,app.kubernetes.io/
instance=pulse-dcu-"
```

Check the output to ensure that all pulse-dcu pods are running, for example:

| NAME | READY | STATUS | RESTARTS | AGE |
|------|-------|--------|----------|-----|
|------|-------|--------|----------|-----|

```
pulse-dcu-100-0 3/3 Running 0 5m23s
pulse-dcu-100-1 3/3 Running 0 4m47s
```

## Install lds helm chart

### Get the lds helm chart:

```
helm repo update
helm search repo /lds
```

### Prepare the override file:

- Update values in the values-override-lds.yaml file (AKS):

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

replicaCount: 2

* Tenant info
tenant identification, or empty for shared deployment
tenant:
 # Tenant UUID
 id: ""
 # Tenant SID (like 0001)
 sid: ""

* 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-lds-logs
 accessModes:
 - ReadWriteMany
 capacity: 10Gi
 class:

* Container image common settings
image:
 tag: ""
 pullPolicy: IfNotPresent
 registry: ""
 imagePullSecrets: [name: ""]

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: {}

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

 ## 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: {}

 # * Monitoring settings
 monitoring:
 # enable the Prometheus metrics endpoint
 enabled: false
 # enable golden signals metrics (not supported for PE)
 goldenSignals:
 enabled: false
 # port number of the Prometheus metrics endpoint
 port: 9091
 # HTTP path to scrape for metrics
 path: /metrics
 # 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.monitoring.port}}"
```

```
prometheus.io/path: "/metrics"
podMonitor:
 # enables PodMonitor creation for the POD
 enabled: true
 # interval at which metrics should be scraped
 scrapeInterval: 30s
 # timeout after which the scrape is ended
 scrapeTimeout:
 # namespace of the PodMonitor, defaults to the namespace of the POD
 namespace:
 additionalLabels: {}
alerts:
 # enables alert rules
 enabled: true
 # alert condition duration
 duration: 5m
 # namespace of the alert rules, defaults to the namespace of the POD
 namespace:
 additionalLabels: {}

* Configuration for the LDS container
lds:
 # resource limits for container
 resources:
 # minimum resource requirements to start container
 requests:
 # minimal amount of memory required to start a container
 memory: "50Mi"
 # minimal CPU to reserve
 cpu: "50m"
 # resource limits for containers
 limits:
 # maximum amount of memory a container can use before being evicted
 # by the OOM Killer
 memory: "4Gi"
 # maximum amount of CPU resources that can be used and should be tuned
 # to reflect
 # what the application can effectively use before needing to be
 # horizontally scaled out
 cpu: "4000m"
 # securityContext: {}

* Configuration for the monitor sidecar container
monitorSidecar:
 # resource limits for container
 resources:
 # minimum resource requirements to start container
 requests:
 # minimal amount of memory required to start a container
 memory: "30Mi"
 # minimal CPU to reserve
 cpu: "2m"
 # resource limits for containers
 limits:
 # maximum amount of memory a container can use before being evicted
 # by the OOM Killer
 memory: "70Mi"
 # maximum amount of CPU resources that can be used and should be tuned
 # to reflect
 # what the application can effectively use before needing to be
 # horizontally scaled out
 cpu: "10m"
 # securityContext: {}
```

```
* Configuration for the Configuration Server Proxy container
csproxy:
 # define domain for the configuration host
 params:
 cfgHost: "tenant-.voice."
 resources:
 # minimum resource requirements to start container
 requests:
 # minimal amount of memory required to start a container
 memory: "200Mi"
 # minimal CPU to reserve
 cpu: "50m"
 # resource limits for containers
 limits:
 # maximum amount of memory a container can use before being evicted
 # by the OOM Killer
 memory: "2Gi"
 # maximum amount of CPU resources that can be used and should be tuned
 # to reflect
 # what the application can effectively use before needing to be
 # horizontally scaled out
 cpu: "1000m"
 # securityContext: {}
```

- Update values in the values-override-lds.yaml file (GKE):

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

replicaCount: 2

* Tenant info
tenant identification, or empty for shared deployment
tenant:
 # Tenant UUID
 id: ""
 # Tenant SID (like 0001)
 sid: ""

* 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-lds-logs
 accessModes:
 - ReadWriteMany
 capacity: 10Gi
 class:

* Container image common settings
image:
 tag: ""
 pullPolicy: IfNotPresent
```

```
registry: ""
imagePullSecrets: [name: ""]

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:
 runAsNonRoot: true
 runAsUser: 500
 runAsGroup: 500
 fsGroup: 0

Add labels to all pods
##
podLabels: {}

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: {}

* Monitoring settings
```

```
monitoring:
 # enable the Prometheus metrics endpoint
 enabled: false
 # enable golden signals metrics (not supported for PE)
 goldenSignals:
 enabled: false
 # port number of the Prometheus metrics endpoint
 port: 9091
 # HTTP path to scrape for metrics
 path: /metrics
 # 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.monitoring.port}}"
 # prometheus.io/path: "/metrics"
 podMonitor:
 # enables PodMonitor creation for the POD
 enabled: true
 # interval at which metrics should be scraped
 scrapeInterval: 30s
 # timeout after which the scrape is ended
 scrapeTimeout:
 # namespace of the PodMonitor, defaults to the namespace of the POD
 namespace:
 additionalLabels: {}
 alerts:
 # enables alert rules
 enabled: true
 # alert condition duration
 duration: 5m
 # namespace of the alert rules, defaults to the namespace of the POD
 namespace:
 additionalLabels: {}

* Configuration for the LDS container
lds:
 # resource limits for container
 resources:
 # minimum resource requirements to start container
 requests:
 # minimal amount of memory required to start a container
 memory: "50Mi"
 # minimal CPU to reserve
 cpu: "50m"
 # resource limits for containers
 limits:
 # maximum amount of memory a container can use before being evicted
 # by the OOM Killer
 memory: "4Gi"
 # maximum amount of CPU resources that can be used and should be tuned
 # to reflect
 # what the application can effectively use before needing to be
 # horizontally scaled out
 cpu: "4000m"
 # securityContext:
 # runAsUser: 500
 # runAsGroup: 500

* Configuration for the monitor sidecar container
monitorSidecar:
 # resource limits for container
```



```
resources:
 # minimum resource requirements to start container
 requests:
 # minimal amount of memory required to start a container
 memory: "30Mi"
 # minimal CPU to reserve
 cpu: "2m"
 # resource limits for containers
 limits:
 # maximum amount of memory a container can use before being evicted
 # by the OOM Killer
 memory: "70Mi"
 # maximum amount of CPU resources that can be used and should be tuned
to reflect
 # what the application can effectively use before needing to be
horizontally scaled out
 cpu: "10m"
 # securityContext:
 # runAsUser: 500
 # runAsGroup: 500

* Configuration for the Configuration Server Proxy container
csproxy:
 # define domain for the configuration host
 params:
 cfgHost: "tenant-.voice."
 resources:
 # minimum resource requirements to start container
 requests:
 # minimal amount of memory required to start a container
 memory: "200Mi"
 # minimal CPU to reserve
 cpu: "50m"
 # resource limits for containers
 limits:
 # maximum amount of memory a container can use before being evicted
 # by the OOM Killer
 memory: "2Gi"
 # maximum amount of CPU resources that can be used and should be tuned
to reflect
 # what the application can effectively use before needing to be
horizontally scaled out
 cpu: "1000m"
 # securityContext:
 # runAsUser: 500
 # runAsGroup: 500
```

### Update values in the values-override-lds-vq.yaml file:

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

```
lds:
 params:
 cfgApp: "pulse-lds-vq-$(K8S_POD_INDEX % 2)"

log:
 pvc:
 name: pulse-lds-vq-logs
```

### Install the lds helm chart:

To install the lds helm chart, run the following command:

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

If the installation is successful, the exit code 0 appears.

### Validate the lds helm chart:

To validate the lds helm chart, run the following command:

```
kubectl get pods -n=pulse -l "app.kubernetes.io/name=lds,app.kubernetes.io/
instance=pulse-lds-"
```

Verify that the command reports all pulse-lds-vq pods as Running, for example:

| NAME            | READY | STATUS  | RESTARTS | AGE   |
|-----------------|-------|---------|----------|-------|
| pulse-lds-100-0 | 3/3   | Running | 0        | 2d20h |
| pulse-lds-100-1 | 3/3   | Running | 0        | 2d20h |

## Install permissions helm chart

### Get the permissions helm chart

```
helm repo update
helm search repo /permissions
```

### Prepare the override file:

- Update values in the values-override-permissions.yaml file (AKS):

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

* Image configuration
image:
 tag: ""
 pullPolicy: IfNotPresent
 registry: ""
 imagePullSecrets: [name: ""]

* Tenant info
tenant identification, or empty for shared deployment
tenant:
 # Tenant UUID
 id: ""
 # Tenant SID (like 0001)
 sid: ""

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"

* Configuration for the Configuration Server Proxy container
csproxy:
 # define domain for the configuration host
 params:
 cfgHost: "tenant-voice."
 # resource limits for container
 resources:
 # minimum resource requirements to start container
 requests:
 # minimal amount of memory required to start a container
 memory: "200Mi"
 # minimal CPU to reserve
 cpu: "50m"
 # resource limits for containers
 limits:
 # maximum amount of memory a container can use before being evicted
 # by the OOM Killer
 memory: "2Gi"
 # maximum amount of CPU resources that can be used and should be tuned
 # to reflect
 # what the application can effectively use before needing to be
 # horizontally scaled out
 cpu: "1000m"
 # securityContext: {}

* 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-permissions-logs
 accessModes:
 - ReadWriteMany
 capacity: 10Gi
 class:

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: "1Gi"
 cpu: "500m"

```

```
requests:
 memory: "400Mi"
 cpu: "50m"

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: {}
```

- Update values in the values-override-permissions.yaml file (GKE):

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

* Image configuration
image:
 tag: ""
 pullPolicy: IfNotPresent
 registry: ""
 imagePullSecrets: [name: ""]

* Tenant info
tenant identification, or empty for shared deployment
tenant:
 # Tenant UUID
 id: ""
 # Tenant SID (like 0001)
 sid: ""

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"

* Configuration for the Configuration Server Proxy container
csproxy:
define domain for the configuration host
params:
 cfgHost: "tenant-.voice."
resource limits for container
resources:
minimum resource requirements to start container
requests:
minimal amount of memory required to start a container
memory: "200Mi"
minimal CPU to reserve
cpu: "50m"
resource limits for containers
limits:
maximum amount of memory a container can use before being evicted
by the OOM Killer
memory: "2Gi"
maximum amount of CPU resources that can be used and should be tuned
to reflect
what the application can effectively use before needing to be
horizontally scaled out
cpu: "1000m"
securityContext:
runAsUser: 500
runAsGroup: 500

* 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-permissions-logs
 accessModes:
 - ReadWriteMany
 capacity: 10Gi
 class:

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: "1Gi"
 cpu: "500m"
 requests:
 memory: "400Mi"
 cpu: "50m"

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: {}
```

**Install the permissions helm chart:** To install the permissions helm chart, run the following command:

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

If installation is successful, the exit code 0 appears.

**Validate the permissions helm chart:**

To validate the permissions helm chart, run the following command:

```
kubectl get pods -n=pulse -l "app.kubernetes.io/name=permissions,app.kubernetes.io/instance=pulse-permissions-"
```

Verify that the command report all pulse-permissions pods as Running, for example:

| NAME                                  | READY | STATUS  | RESTARTS | AGE   |
|---------------------------------------|-------|---------|----------|-------|
| pulse-permissions-100-c5ff8bb7d-jl7d7 | 2/2   | Running | 2        | 2d20h |

## Troubleshooting

### **Check init-tenant helm chart manifests:**

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

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

### **Check dcu helm chart manifests:**

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

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

### **Check lds helm chart manifests:**

To output the lds chart manifest into the **helm-template** directory, run the following command:

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

### **Check permissions Helm chart manifests:**

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

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

# Deploy Pulse Web Service

## Contents

- [1 Assumptions](#)
- [2 Prerequisites](#)
- [3 Validation steps](#)



Learn how to deploy Pulse Web Service (Pulse) into a private edition environment.

### Related documentation:

- 
- 
- 
- 

### RSS:

- [For private edition](#)

## Assumptions

- The instructions on this page assume you are deploying the service in a service-specific namespace, named in accordance with the requirements on [Creating namespaces](#). If you are using a single namespace for all private edition services, replace the namespace element in the commands on this page with the name of your single namespace or project.
- Similarly, the configuration and environment setup instructions assume you need to create namespace-specific (in other words, service-specific) secrets. If you are using a single namespace for all private edition services, you might not need to create separate secrets for each service, depending on your credentials management requirements. However, if you do create service-specific secrets in a single namespace, be sure to avoid naming conflicts.

## Prerequisites

Before you begin, ensure that you have the following information:

- Genesys Pulse access URL.
- Agent Setup access URL, administrator credentials.
- WWE access URL, agent credentials.

## Validation steps

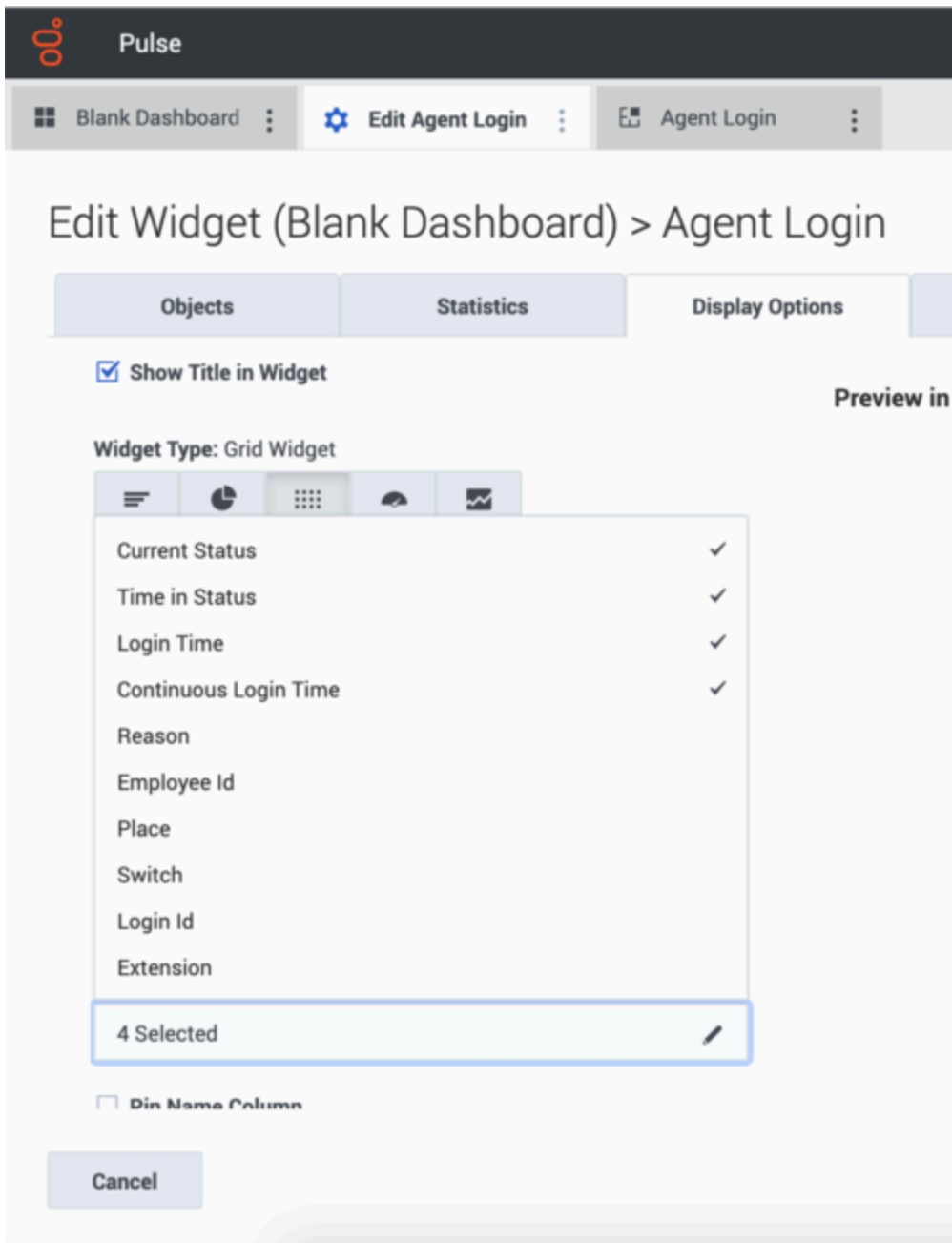
1. Create a user.
  1. Use an administrator account to log in to Agent Setup.
  2. Navigate to the **Users** tab.

3. Click **New User**.
4. Populate required fields, for example:
  - **First Name:** pulsetest
  - **Last Name:** pulsetest
  - **Username:** pulsetest
  - **Password:** Password@1
  - **Password Confirm:** Password@1
  - Clear **Reset Password**.
5. Expand the **User** list on the left side, and switch to **Access Group**.
6. Click **Add**.
7. **Supervisors** to create a Pulse User.

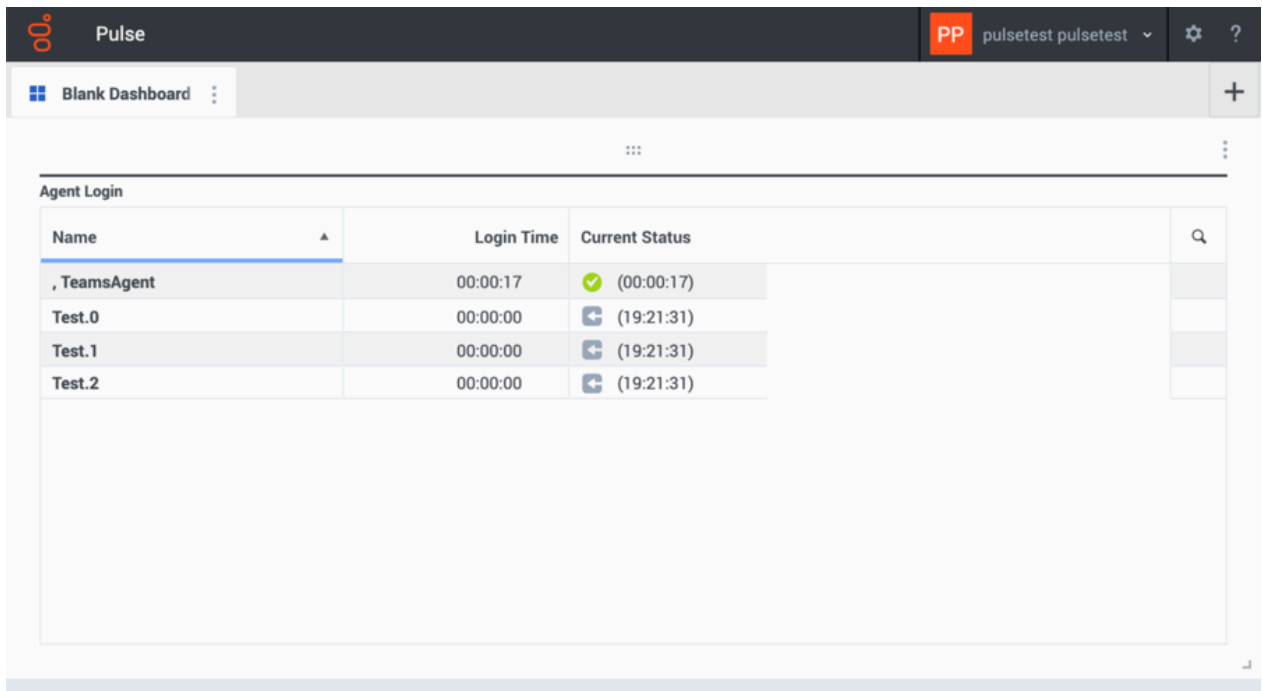
### Tip

A Pulse Administrator role is also available, which gives the new user the ability to access Dashboard / Wallboard Management, Widget Template Management, and Widget Management pages. To create an Pulse Administrator user, select **Administrator**.

8. Click **Save**.
2. Log in to Genesys Pulse as the newly created user, and create a new Widget.
    1. Pulse displays an option to create **Dashboard** or **Wallboard** on the first login; choose the **Dashboard** option.
    2. To create a new Widget, click **+ Add a Widget**.
    3. Select the **Agent Login** template.
    4. Select an agent that has access to WWE.
    5. Switch to the **Display Options** tab, and scroll down to the **Statistics** control.



6. From the available **Statistics**, select **Current Status**.
7. Click **Save**.
8. A spinner appears on the Widget to indicate that processing is underway. Wait until some zero data appears (the selected agent appears as Logged Out).
3. Log in to WWE with the agent account you selected in the Widget earlier, and change status to **Available**.
4. Return to Genesys Pulse. Verify the current agent status in the Widget:



The screenshot shows the Pulse web interface. At the top, there is a dark header with the Pulse logo and the text "Pulse". On the right side of the header, there is a user profile section with the initials "PP" and the name "pulsetest pulsetest". Below the header, there is a navigation bar with a "Blank Dashboard" button and a plus sign. The main content area is titled "Agent Login" and contains a table with the following data:

| Name         | Login Time | Current Status |
|--------------|------------|----------------|
| , TeamsAgent | 00:00:17   | ✔ (00:00:17)   |
| Test.0       | 00:00:00   | ⏪ (19:21:31)   |
| Test.1       | 00:00:00   | ⏪ (19:21:31)   |
| Test.2       | 00:00:00   | ⏪ (19:21:31)   |

# Upgrade, roll back, or uninstall Genesys Pulse

## Contents

- [1 Supported upgrade strategies](#)
- [2 Timing](#)
  - [2.1 Scheduling considerations](#)
- [3 Monitoring](#)
- [4 Preparatory steps](#)
- [5 Rolling Update](#)
  - [5.1 Rolling Update: Upgrade](#)
  - [5.2 Rolling Update: Verify the upgrade](#)
  - [5.3 Rolling Update: Rollback](#)
  - [5.4 Rolling Update: Verify the rollback](#)
- [6 Uninstall](#)

Learn how to upgrade, roll back, or uninstall Genesys Pulse.

### Related documentation:

- 
- 
- 

### RSS:

- [For private edition](#)

### Important

The instructions on this page assume you have deployed the services in service-specific namespaces. If you are using a single namespace for all private edition services, replace the namespace element in the commands on this page with the name of your single namespace or project.

## Supported upgrade strategies

Genesys Pulse supports the following upgrade strategies:

| Service       | Upgrade Strategy | Notes |
|---------------|------------------|-------|
| Genesys Pulse | Rolling Update   |       |

For a conceptual overview of the upgrade strategies, refer to Upgrade strategies in the Setting up Genesys Multicloud CX Private Edition guide.

## Timing

A regular upgrade schedule is necessary to fit within the Genesys policy of supporting N-2 releases, but a particular release might warrant an earlier upgrade (for example, because of a critical security fix).

If the service you are upgrading requires a later version of any third-party services, upgrade the third-party service(s) before you upgrade the private edition service. For the latest supported versions of third-party services, see the Software requirements page in the suite-level guide.

### Scheduling considerations

Genesys recommends that you upgrade the services methodically and sequentially: Complete the upgrade for one service and verify that it upgraded successfully before proceeding to upgrade the next service. If necessary, roll back the upgrade and verify successful rollback.

### Monitoring

Monitor the upgrade process using standard Kubernetes and Helm metrics, as well as service-specific metrics that can identify failure or successful completion of the upgrade (see Observability in Genesys Pulse).

Genesys recommends that you create custom alerts for key indicators of failure — for example, an alert that a pod is in pending state for longer than a timeout suitable for your environment. Consider including an alert for the absence of metrics, which is a situation that can occur if the Docker image is not available. Note that Genesys does not provide support for custom alerts that you create in your environment.

### Preparatory steps

Ensure that your processes have been set up to enable easy rollback in case an upgrade leads to compatibility or other issues.

Each time you upgrade a service:

1. Review the release note to identify changes.
2. Ensure that the new package is available for you to deploy in your environment.
3. Ensure that your existing **-values.yaml** file is available and update it if required to implement changes.

Use the following commands to get the updated Helm charts in your repository:

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

where is the name of the Helm chart you need. To update shared and tenant provisioning, run the command for each of the following charts:

- init
- pulse
- init-tenant
- dcu
- lds
- permissions

For example, for the **pulse** Helm chart:

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

## Rolling Update

### Rolling Update: Upgrade

Execute the following command to upgrade :

```
helm upgrade --install -f -values.yaml -n
```

**Tip:** If your review of Helm chart changes (see Preparatory Step 3) identifies that the only update you need to make to your existing **-values.yaml** file is to update the image version, you can pass the image tag as an argument by using the **--set** flag in the command:

```
helm upgrade --install -f -values.yaml --set .image.tag=
```

You can find additional information about the `helm upgrade --install` commands on the Shared Provisioning and Tenant Provisioning pages.

The following sections describe recommended arguments to use when you run the Helm upgrade command:

#### Shared provisioning

When you run the Helm upgrade command for Shared provisioning:

- For the **init** Helm chart, set the **wait** and **wait-for-jobs** flags, so that Helm waits until all jobs have completed before reporting success:

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

This command also updates the schema, if an update is required.

- For the **pulse** Helm chart, set the **wait** flag, so that Helm waits until all pods and other Kubernetes resources in the replica set are in Ready state before reporting success:

```
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.

#### Tenant provisioning

When you run the Helm upgrade command for Shared provisioning:

- For the **init-tenant** Helm chart, set the **wait** and **wait-for-jobs** flags, so that Helm waits until all jobs have completed before reporting success:



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

- For the **dcu**, **lds**, **lds-vq**, and **permissions** Helm charts, set the **wait** flag, so that Helm waits until all pods and other StatefulSet Kubernetes resources are in Ready state before reporting success:

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

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

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

You can optionally omit the argument `--version=""` to upgrade to the latest available version.

### Rolling Update: Verify the upgrade

Follow usual Kubernetes best practices to verify that the new service version is deployed. See the information about initial deployment for additional functional validation that the service has upgraded successfully.

### Rolling Update: Rollback

Execute the following command to roll back the upgrade to the previous version:

```
helm rollback
```

or, to roll back to an even earlier version:

```
helm rollback
```

Alternatively, you can re-install the previous package:

1. Revert the image version in the `.image.tag` parameter in the **-values.yaml** file. If applicable, also revert any configuration changes you implemented for the new release.
2. Execute the following command to roll back the upgrade:

```
helm upgrade --install -f -values.yaml
```

**Tip:** You can also directly pass the image tag as an argument by using the `--set` flag in the command:

```
helm upgrade --install -f -values.yaml --set .image.tag=
```

### Rolling Update: Verify the rollback

Verify the rollback in the same way that you verified the upgrade (see Rolling Update: Verify the upgrade).

## Uninstall

### Warning

Uninstalling a service removes all Kubernetes resources associated with that service. Genesys recommends that you contact Genesys Customer Care before uninstalling any private edition services, particularly in a production environment, to ensure that you understand the implications and to prevent unintended consequences arising from, say, unrecognized dependencies or purged data.

Execute the following command to uninstall :

```
helm uninstall -n
```

For example, to uninstall:

- **init:**  

```
helm uninstall pulse-init --namespace=pulse
```
- **pulse:**  

```
helm uninstall pulse --namespace=pulse
```
- **init-tenant:**  

```
helm uninstall pulse-init-tenant- --namespace=pulse
```
- **dcu:**  

```
helm uninstall pulse-dcu- --namespace=pulse
```
- **lds:**  

```
helm uninstall pulse-lds- --namespace=pulse
helm uninstall pulse-lds-vq- --namespace=pulse
```
- **permissions:**  

```
helm uninstall pulse-permissions- --namespace=pulse
```

### Important

If you want to remove all custom Pulse dashboard configurations (shared or personal) and all widget templates, you must manually delete the Pulse database schema, which also deletes all data in the schema.

# Observability in Genesys Pulse

## Contents

- **1 Monitoring**
  - 1.1 Enable monitoring
  - 1.2 Configure metrics
- **2 Alerting**
  - 2.1 Configure alerts
- **3 Logging**

Learn about the logs, metrics, and alerts you should monitor for Genesys Pulse.

**Related documentation:**

- 
- 
- 
- 

**RSS:**

- [For private edition](#)

## Monitoring

Private edition services expose metrics that can be scraped by Prometheus, to support monitoring operations and alerting.

- As described on [Monitoring overview and approach](#), you can use a tool like Grafana to create dashboards that query the Prometheus metrics to visualize operational status.
- As described on [Customizing Alertmanager configuration](#), you can configure Alertmanager to send notifications to notification providers such as PagerDuty, to notify you when an alert is triggered because a metric has exceeded a defined threshold.

The services expose a number of Genesys-defined and third-party metrics. The metrics that are defined in third-party software used by private edition services are available for you to use as long as the third-party provider still supports them. For descriptions of available Genesys Pulse metrics, see:

- [Tenant Data Collection Unit \(DCU\) metrics](#)
- [Tenant Load Distribution Server \(LDS\) metrics](#)
- [Pulse Web Service metrics](#)
- [Tenant Permissions Service metrics](#)

See also [System metrics](#).

You can find an example of Grafana dashboards and metrics inside the Monitor helm chart.

## Enable monitoring

Monitoring settings can be found in the [Override Helm chart values](#) section:

- For more information about configuring monitoring for the Pulse web service, see **Monitoring settings** in **values-override-pulse.yaml** on the Shared Provisioning page.
- For more information about configuring monitoring for the Data Collection Units and Load Distribution Servers, see **Monitoring settings** in **values-override-dcu.yaml** and **values-override-lds.yaml** on the Tenant Provisioning page.

| Service                               | CRD or annotations? | Port | Endpoint/Selector                                                                         | Metrics update interval |
|---------------------------------------|---------------------|------|-------------------------------------------------------------------------------------------|-------------------------|
| Tenant Data Collection Unit (DCU)     | PodMonitor          | 9091 | See selector details on the Tenant Data Collection Unit (DCU) metrics and alerts page     | 30 seconds              |
| Tenant Load Distribution Server (LDS) | PodMonitor          | 9091 | See selector details on the Tenant Load Distribution Server (LDS) metrics and alerts page | 30 seconds              |
| Pulse Web Service                     | ServiceMonitor      | 8090 | See selector details on the Pulse Web Service metrics and alerts page                     | 30 seconds              |
| Tenant Permissions Service            |                     |      |                                                                                           |                         |

## Configure metrics

The Pulse services use PodMonitor/ServiceMonitor custom resource definitions (CRDs), which are defined in the Helm charts by default. No further configuration is required to define or make available these metrics. You cannot define your own custom metrics.

This document describes critical metrics that Pulse services make available. You can also query Prometheus directly or using a dashboard to see all the metrics available from the Pulse services.

## Alerting

Private edition services define a number of alerts based on Prometheus metrics thresholds.

### Important

You can use general third-party functionality to create rules to trigger alerts based on metrics values you specify. Genesys does not provide support for custom alerts that you create in your environment.

For descriptions of available Genesys Pulse alerts, see:

- Tenant Data Collection Unit (DCU) alerts
- Tenant Load Distribution Server (LDS) alerts
- Pulse Web Service alerts
- Tenant Permissions Service alerts

Pulse alerts are defined as **PrometheusRule** objects in `/*.yaml` files in the **monitor** Helm chart. Pulse does not support customizing the alerts or defining additional PrometheusRule objects to create alerts based on the service-provided metrics.

### Configure alerts

Private edition services define a number of alerts by default (for Genesys Pulse, see the pages linked to above). No further configuration is required.

The alerts are defined as **PrometheusRule** objects in a **prometheus-rule.yaml** file in the Helm charts. As described above, Genesys Pulse does not support customizing the alerts or defining additional **PrometheusRule** objects to create alerts based on the service-provided metrics.

No further configuration is required for Pulse alerts.

## Logging

Logging

# Pulse metrics and alerts

## Contents

- [1 Metrics](#)
- [2 Alerts](#)

Find the metrics Pulse exposes and the alerts defined for Pulse.

**Related documentation:**

- 

| Service | CRD or annotations? | Port | Endpoint/Selector                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Metrics update interval |
|---------|---------------------|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|
| Pulse   | ServiceMonitor      | 8090 | <pre> selector:   matchLabels:     app.kubernetes.io/ name: {{ include "common.util.chart.name" . }}   app.kubernetes.io/ instance: {{ include "common.util.chart.fullname" . }}   service: {{ .Release.Namespace }}   servicename: {{ include "common.util.chart.name" . }}   tenant: "shared" </pre> <p>Endpoints to query:</p> <ul style="list-style-type: none"> <li>• For a list of metrics: <b>/actuator/metrics/</b></li> <li>• For metric output: <b>/actuator/metrics/pulse.health.all / actuator/metrics/pulse.health.connections</b></li> </ul> | 30 seconds              |

See details about:

- Pulse metrics
- Pulse alerts

## Metrics

The **pulse\_\*\_Boolean** metrics are readable only from Prometheus directly. You cannot read them using the cURL command-line tool.



| Metric and description                                                                                                    | Metric details                                                                           | Indicator of |
|---------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|--------------|
| <b>pulse_health_all_Boolean</b><br>Overall Pulse application status.                                                      | <b>Unit:</b><br><b>Type:</b> Gauge<br><b>Label:</b><br><b>Sample value:</b> 0.5          | Error        |
| <b>pulse_health_connections_Boolean</b><br>Status of the connections to the external services (Auth, GWS, Redis, and DB). | <b>Unit:</b><br><b>Type:</b> Gauge<br><b>Label:</b> connection<br><b>Sample value:</b> 0 | Error        |

## Alerts

Alerts are based on Pulse, Java, and Kubernetes cluster metrics.

The following alerts are defined for Pulse.

| Alert                             | Severity | Description                                                         | Based on                                                                   | Threshold      |
|-----------------------------------|----------|---------------------------------------------------------------------|----------------------------------------------------------------------------|----------------|
| pulse_service_down                | Critical | All Pulse instances are down.                                       | up                                                                         | for 15 minutes |
| pulse_critical_pulse_health       | Critical | Detected critical number of healthy Pulse containers.               | pulse_health_all_Boolean                                                   | 50%            |
| pulse_critical_running_instances  | Critical | Triggered when Pulse instances are down.                            | kube_deployment_status_replicas_available, kube_deployment_status_replicas | 75%            |
| pulse_too_frequent_restarts       | Critical | Detected too frequent restarts of Pulse Pod container.              | kube_pod_container_status_restarts_total                                   | 2 for 1 hour   |
| pulse_critical_cpu                | Critical | Detected critical CPU usage by Pulse Pod.                           | container_cpu_usage_seconds_total, kube_pod_container_resource_limits      | 90%            |
| pulse_critical_memory             | Critical | Detected critical memory usage by Pulse Pod.                        | container_memory_working_set_bytes, kube_pod_container_resource_limits     | 90%            |
| pulse_critical_hikari_connections | Critical | Detected critical Hikari connections pool usage by Pulse container. | hikaricp_connections_active, hikaricp_connections                          | 90%            |
| pulse_critical_5xx                | Critical | Detected critical 5xx errors per second for Pulse container.        | http_server_requests_5xx_count                                             | 15%            |

# Tenant Permissions Service metrics and alerts

## Contents

- [1 Metrics](#)
- [2 Alerts](#)

Find the metrics Tenant Permissions Service exposes and the alerts defined for Tenant Permissions Service.

**Related documentation:**

- 
- 
- 

**RSS:**

- [For private edition](#)

| Service                    | CRD or annotations?                                                | Port | Endpoint/Selector | Metrics update interval |
|----------------------------|--------------------------------------------------------------------|------|-------------------|-------------------------|
| Tenant Permissions Service | The Pulse Permissions service does not make any metrics available. |      |                   |                         |

See details about:

- [Tenant Permissions Service metrics](#)
- [Tenant Permissions Service alerts](#)

## Metrics

The Pulse Permissions service does not expose metrics.

| Metric and description | Metric details | Indicator of |
|------------------------|----------------|--------------|
|------------------------|----------------|--------------|

## Alerts

Alerts are based on Kubernetes cluster metrics.

The following alerts are defined for Tenant Permissions Service.

| Alert                                        | Severity | Description                      | Based on                                                                   | Threshold |
|----------------------------------------------|----------|----------------------------------|----------------------------------------------------------------------------|-----------|
| pulse_permissions_critical_running_instances | Critical | Triggered when Pulse Permissions | kube_deployment_status_replicas_available, kube_deployment_status_replicas | 75%       |

## Tenant Permissions Service metrics and alerts

---

| Alert                               | Severity | Description                                                  | Based on                                                               | Threshold    |
|-------------------------------------|----------|--------------------------------------------------------------|------------------------------------------------------------------------|--------------|
|                                     |          | instances are down.                                          |                                                                        |              |
| pulse_permissions_too_few_instances | Critical | Detected too frequent restarts of Permissions Pod container. | kube_pod_container_status_restarts_total                               | 2 for 1 hour |
| pulse_permissions_critical_cpu      | Critical | Detected critical CPU usage by Pulse Permissions Pod.        | container_cpu_usage_seconds_total, kube_pod_container_resource_limits  | 90%          |
| pulse_permissions_critical_memory   | Critical | Detected critical memory usage by Pulse Permissions Pod.     | container_memory_working_set_bytes, kube_pod_container_resource_limits | 90%          |

# Tenant Load Distribution Server (LDS) metrics and alerts

## Contents

- [1 Metrics](#)
- [2 Alerts](#)

Find the metrics Tenant Load Distribution Server (LDS) exposes and the alerts defined for Tenant Load Distribution Server (LDS).

**Related documentation:**

- 

| Service                               | CRD or annotations? | Port | Endpoint/Selector                                                                                                                                                                                                                                                                                                                                                   | Metrics update interval |
|---------------------------------------|---------------------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|
| Tenant Load Distribution Server (LDS) | PodMonitor          | 9091 | <pre> selector:   matchLabels:     app.kubernetes.io/ name: {{include "common.util.chart.name" . }}   app.kubernetes.io/ instance: {{include "common.util.chart.fullname" . }}   service: {{.Release.Namespace }}   servicename: {{include "common.util.chart.name" . }}   tenant: {{.Values.tenant.sid }} Endpoints to query: /metrics/                     </pre> | 30 seconds              |

See details about:

- Tenant Load Distribution Server (LDS) metrics
- Tenant Load Distribution Server (LDS) alerts

## Metrics

| Metric and description                                                                                                                 | Metric details                                                                                                | Indicator of |
|----------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|--------------|
| <p><b>pulse_monitor_check_duration_seconds</b></p> <p>The duration in seconds of the last health check performed by Monitor Agent.</p> | <p><b>Unit:</b> seconds</p> <p><b>Type:</b> Gauge</p> <p><b>Label:</b> tenant</p> <p><b>Sample value:</b></p> | Error        |
| <p><b>pulse_lds_uptime_seconds</b></p>                                                                                                 | <p><b>Unit:</b> seconds</p>                                                                                   | Error        |

## Tenant Load Distribution Server (LDS) metrics and alerts

| Metric and description                                                                                                       | Metric details                                                                                       | Indicator of |
|------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|--------------|
| The LDS container uptime in seconds.                                                                                         | <b>Type:</b> Gauge<br><b>Label:</b> tenant<br><b>Sample value:</b>                                   |              |
| <b>pulse_lds_senders_number</b><br>The number of upstream servers to which the LDS is connected.                             | <b>Unit:</b><br><b>Type:</b> Gauge<br><b>Label:</b> tenant<br><b>Sample value:</b> 2                 | Error        |
| <b>pulse_lds_receivers_number</b><br>The number of clients connected to the LDS.                                             | <b>Unit:</b><br><b>Type:</b> Gauge<br><b>Label:</b> tenant<br><b>Sample value:</b> 2                 | Error        |
| <b>pulse_lds_sender_connected_seconds</b><br>Duration in seconds of connection to the upstream server.                       | <b>Unit:</b> seconds<br><b>Type:</b> Gauge<br><b>Label:</b> tenant, sender<br><b>Sample value:</b>   | Error        |
| <b>pulse_lds_sender_disconnected_seconds</b><br>Duration in seconds of disconnection from the upstream server.               | <b>Unit:</b> seconds<br><b>Type:</b> Gauge<br><b>Label:</b> tenant, sender<br><b>Sample value:</b>   | Error        |
| <b>pulse_lds_sender_registered_dns_number</b><br>The number of DNSs registered on the upstream server.                       | <b>Unit:</b><br><b>Type:</b> Gauge<br><b>Label:</b> tenant, sender<br><b>Sample value:</b> 1000      | Saturation   |
| <b>pulse_lds_sender_registration_errors_number</b><br>The number of failed registrations of DNSs on the upstream server.     | <b>Unit:</b><br><b>Type:</b> Gauge<br><b>Label:</b> tenant, sender<br><b>Sample value:</b> 0         | Error        |
| <b>pulse_lds_receiver_connected_seconds</b><br>Duration in seconds of client connection to the LDS.                          | <b>Unit:</b> seconds<br><b>Type:</b> Gauge<br><b>Label:</b> tenant, receiver<br><b>Sample value:</b> | Error        |
| <b>pulse_lds_receiver_registered_dns_number</b><br>The number of DNSs registered by the client.                              | <b>Unit:</b><br><b>Type:</b> Gauge<br><b>Label:</b> tenant, receiver<br><b>Sample value:</b> 1000    | Saturation   |
| <b>pulse_lds_receiver_registration_errors_number</b><br>The number of failed registrations of DNSs received from the client. | <b>Unit:</b><br><b>Type:</b> Gauge<br><b>Label:</b> tenant, receiver<br><b>Sample value:</b> 0       | Error        |

## Alerts

Alerts are based on LDS and Kubernetes cluster metrics.

The following alerts are defined for Tenant Load Distribution Server (LDS).

| Alert                                  | Severity | Description                                          | Based on                                                                 | Threshold      |
|----------------------------------------|----------|------------------------------------------------------|--------------------------------------------------------------------------|----------------|
| pulse_lds_monitor_data_unavailable     | Critical | Pulse LDS Monitor Agents do not provide data.        | pulse_monitor_check_duration_seconds, kube_statefulset_replicas          | for 15 minutes |
| pulse_lds_critical_non_ready_instances | Critical | Triggered when Pulse LDS instances are down.         | kube_statefulset_status_replicas_ready, kube_statefulset_status_replicas | for 15 minutes |
| pulse_lds_too_frequent_restarts        | Critical | Detected too frequent restarts of LDS Pod container. | kube_pod_container_status_restarts_total                                 | 2 for 1 hour   |
| pulse_lds_critical_cpu_usage           | Critical | Detected critical CPU usage by Pulse LDS Pod.        | container_cpu_usage_seconds_total, kube_pod_container_resource_limits    | 90%            |
| pulse_lds_critical_memory_usage        | Critical | Detected critical memory usage by Pulse LDS Pod.     | container_memory_working_set_bytes, kube_pod_container_resource_limits   | 90%            |
| pulse_lds_no_connecting_senders        | Critical | Pulse LDS is not connected to upstream servers.      | pulse_lds_senders_number                                                 | for 15 minutes |
| pulse_lds_no_registered_senders        | Critical | No DNs are registered on Pulse LDS.                  | pulse_lds_sender_registered_number                                       | for 30 minutes |



# Tenant Data Collection Unit (DCU) metrics and alerts

## Contents

- [1 Metrics](#)
- [2 Alerts](#)

Find the metrics Tenant Data Collection Unit (DCU) exposes and the alerts defined for Tenant Data Collection Unit (DCU).

**Related documentation:**

- 

| Service                           | CRD or annotations? | Port | Endpoint/Selector                                                                                                                                                                                                                                                                                                                                             | Metrics update interval |
|-----------------------------------|---------------------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|
| Tenant Data Collection Unit (DCU) | PodMonitor          | 9091 | <pre> selector:   matchLabels:     app.kubernetes.io/ name: {{include "common.util.chart.name" . }}   app.kubernetes.io/ instance: {{include "common.util.chart.fullname" . }} service: {{.Release.Namespace }} servicename: {{include "common.util.chart.name" . }} tenant: {{.Values.tenant.sid }} Endpoints to query: /metrics/                     </pre> | 30 seconds              |

See details about:

- Tenant Data Collection Unit (DCU) metrics
- Tenant Data Collection Unit (DCU) alerts

## Metrics

| Metric and description                                                                                                                 | Metric details                                                                                                | Indicator of |
|----------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|--------------|
| <p><b>pulse_monitor_check_duration_seconds</b></p> <p>The duration in seconds of the last health check performed by Monitor Agent.</p> | <p><b>Unit:</b> seconds</p> <p><b>Type:</b> Gauge</p> <p><b>Label:</b> tenant</p> <p><b>Sample value:</b></p> | Error        |
| <p><b>pulse_collector_uptime_seconds</b></p>                                                                                           | <p><b>Unit:</b> seconds</p>                                                                                   | Error        |

| Metric and description                                         | Metric details                                                                      | Indicator of |
|----------------------------------------------------------------|-------------------------------------------------------------------------------------|--------------|
| The Collector container uptime in seconds.                     | <b>Type:</b> Gauge<br><b>Label:</b> tenant<br><b>Sample value:</b>                  |              |
| <b>pulse_collector_snapshot_writing_status</b>                 | <b>Unit:</b>                                                                        |              |
| The status of writing Collector snapshots to the Redis.        | <b>Type:</b> Gauge<br><b>Label:</b> tenant<br><b>Sample value:</b> 1                | Error        |
| <b>pulse_collector_active_layouts_count</b>                    | <b>Unit:</b>                                                                        |              |
| The number of active layouts.                                  | <b>Type:</b> Gauge<br><b>Label:</b> tenant<br><b>Sample value:</b> 100              | Saturation   |
| <b>pulse_collector_connection_status</b>                       | <b>Unit:</b>                                                                        |              |
| The status of the Collector connection to the upstream server. | <b>Type:</b> Gauge<br><b>Label:</b> tenant, connection<br><b>Sample value:</b> 1    | Error        |
| <b>pulse_collector_connection_connected_seconds</b>            | <b>Unit:</b> seconds                                                                |              |
| Duration in seconds of connection to the upstream server.      | <b>Type:</b> Gauge<br><b>Label:</b> tenant, connection<br><b>Sample value:</b>      | Error        |
| <b>pulse_collector_connection_disconnected_seconds</b>         | <b>Unit:</b> seconds                                                                |              |
| Duration in seconds of disconnection from the upstream server. | <b>Type:</b> Gauge<br><b>Label:</b> tenant, connection<br><b>Sample value:</b>      | Error        |
| <b>pulse_collector_statistics_total_count</b>                  | <b>Unit:</b>                                                                        |              |
| The total number of Collector statistics.                      | <b>Type:</b> Gauge<br><b>Label:</b> tenant, connection<br><b>Sample value:</b> 1000 | Saturation   |
| <b>pulse_collector_statistics_opened_count</b>                 | <b>Unit:</b>                                                                        |              |
| The number of successfully open Collector statistics.          | <b>Type:</b> Gauge<br><b>Label:</b> tenant, connection<br><b>Sample value:</b> 1000 | Saturation   |
| <b>pulse_collector_statistics_failed_count</b>                 | <b>Unit:</b>                                                                        |              |
| The number of Collector statistics that failed to open.        | <b>Type:</b> Gauge<br><b>Label:</b> tenant, connection<br><b>Sample value:</b> 0    | Error        |
| <b>pulse_statserver_uptime_seconds</b>                         | <b>Unit:</b> seconds                                                                |              |
| The Stat Server container uptime in seconds.                   | <b>Type:</b> Gauge<br><b>Label:</b> tenant<br><b>Sample value:</b>                  | Error        |
| <b>pulse_statserver_clients_number</b>                         | <b>Unit:</b>                                                                        |              |
|                                                                |                                                                                     | Error        |

| Metric and description                                                                            | Metric details                                                                        | Indicator of |
|---------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|--------------|
| The number of clients connected to the Stat Server.                                               | <b>Type:</b> Gauge<br><b>Label:</b> tenant<br><b>Sample value:</b> 1                  |              |
| <b>pulse_statserver_messages_received_total_count</b>                                             | <b>Unit:</b>                                                                          |              |
| The total number of messages received by the Stat Server.                                         | <b>Type:</b> Gauge<br><b>Label:</b> tenant<br><b>Sample value:</b> 10000              | Traffic      |
| <b>pulse_statserver_messages_sent_total_count</b>                                                 | <b>Unit:</b>                                                                          |              |
| The total number of messages sent by the Stat Server.                                             | <b>Type:</b> Gauge<br><b>Label:</b> tenant<br><b>Sample value:</b> 10000              | Traffic      |
| <b>pulse_statserver_server_connected_number</b>                                                   | <b>Unit:</b>                                                                          |              |
| The number of Stat Server connections to upstream servers.                                        | <b>Type:</b> Gauge<br><b>Label:</b> tenant, type<br><b>Sample value:</b> 1            | Error        |
| <b>pulse_statserver_server_messages_received_total_count</b>                                      | <b>Unit:</b>                                                                          |              |
| The total number of messages received by the Stat Server from the upstream server.                | <b>Type:</b> Gauge<br><b>Label:</b> tenant, server, type<br><b>Sample value:</b> 1    | Traffic      |
| <b>pulse_statserver_server_connected_seconds</b>                                                  | <b>Unit:</b> seconds                                                                  |              |
| Duration in seconds of the Stat Server connection to the upstream server.                         | <b>Type:</b> Gauge<br><b>Label:</b> tenant, server, type<br><b>Sample value:</b>      | Error        |
| <b>pulse_statserver_server_disconnects_count</b>                                                  | <b>Unit:</b> seconds                                                                  |              |
| Duration in seconds of the Stat Server disconnection from the upstream server.                    | <b>Type:</b> Gauge<br><b>Label:</b> tenant, server, type<br><b>Sample value:</b>      | Error        |
| <b>pulse_statserver_dn_registered</b>                                                             | <b>Unit:</b>                                                                          |              |
| The number of successful registration attempts during current session with the upstream T-Server. | <b>Type:</b> Gauge<br><b>Label:</b> tenant, server, type<br><b>Sample value:</b> 1000 | Saturation   |
| <b>pulse_statserver_dn_failed</b>                                                                 | <b>Unit:</b>                                                                          |              |
| The number of DNs for which registration failed after predefined number of attempts.              | <b>Type:</b> Gauge<br><b>Label:</b> tenant, server, type<br><b>Sample value:</b> 0    | Error        |
| <b>pulse_statserver_server_latency_avg</b>                                                        | <b>Unit:</b>                                                                          |              |
| The average Stat Server server latency in seconds.                                                | <b>Type:</b> Gauge<br><b>Label:</b> tenant, server, type<br><b>Sample value:</b> 0.25 | Latency      |
| <b>pulse_statserver_server_latency</b>                                                            | <b>Unit:</b>                                                                          |              |
|                                                                                                   |                                                                                       | Latency      |

| Metric and description                                                                                                                             | Metric details                                                                                        | Indicator of |
|----------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|--------------|
| The minimum Stat Server server latency in seconds.                                                                                                 | <b>Type:</b> Gauge<br><b>Label:</b> tenant, server, type<br><b>Sample value:</b> 0.25                 |              |
| <b>pulse_statserver_server_latency_min</b><br>The maximum Stat Server server latency in seconds.                                                   | <b>Unit:</b><br><b>Type:</b> Gauge<br><b>Label:</b> tenant, server, type<br><b>Sample value:</b> 0.25 | Latency      |
| <b>pulse_statserver_server_events_received_total_count</b><br>The total number of T-Events received by the Stat Server from the upstream T-Server. | <b>Type:</b> Gauge<br><b>Label:</b> tenant, server, type<br><b>Sample value:</b> 10000                | Traffic      |

## Alerts

Alerts are based on Collector, Stat Server, and Kubernetes cluster metrics.

The following alerts are defined for Tenant Data Collection Unit (DCU).

| Alert                                         | Severity | Description                                          | Based on                                                                  | Threshold      |
|-----------------------------------------------|----------|------------------------------------------------------|---------------------------------------------------------------------------|----------------|
| pulse_dcu_monitor_data_unavailable            | Critical | Pulse DCU Monitor Agents do not provide data.        | pulse_monitor_check_duration_seconds, kube_statefulset_replicas           | for 15 minutes |
| pulse_dcu_critical_no_running_instances       | Critical | Triggered when Pulse DCU instances are down.         | kube_statefulset_status_replicas_ready, kube_statefulset_status_replicas  | for 15 minutes |
| pulse_dcu_too_frequent_restarts               | Critical | Detected too frequent restarts of DCU Pod container. | kube_pod_container_status_restarts_total                                  | 2 for 1 hour   |
| pulse_dcu_critical_cpu_usage                  | Critical | Detected critical CPU usage by Pulse DCU Pod.        | container_cpu_usage_seconds_total, kube_pod_container_resource_limits     | 90%            |
| pulse_dcu_critical_memory_usage               | Critical | Detected critical memory usage by Pulse DCU Pod.     | container_memory_working_set_bytes, kube_pod_container_resource_limits    | 90%            |
| pulse_dcu_critical_disk_usage                 | Critical | Detected critical disk usage by Pulse DCU Pod.       | kubelet_volume_stats_available_bytes, kubelet_volume_stats_capacity_bytes | 90%            |
| pulse_dcu_critical_collector_snapshot_writing | Critical | Pulse DCU Collector does not write snapshots.        | pulse_collector_snapshot_writing_status                                   | for 15 minutes |

Tenant Data Collection Unit (DCU) metrics and alerts

| Alert                                                    | Severity | Description                                                          | Based on                                                   | Threshold      |
|----------------------------------------------------------|----------|----------------------------------------------------------------------|------------------------------------------------------------|----------------|
| pulse_dcu_critical_collector_not_connected_configservers | Critical | Pulse DCU Collector is not connected to ConfigServer.                | pulse_collector_connection_status                          | for 15 minutes |
| pulse_dcu_critical_collector_not_connected_dbserver      | Critical | Pulse DCU Collector is not connected to DbServer.                    | pulse_collector_connection_status                          | for 15 minutes |
| pulse_dcu_critical_collector_not_connected_statserver    | Critical | Pulse DCU Collector is not connected to Stat Server.                 | pulse_collector_connection_status                          | for 15 minutes |
| pulse_dcu_critical_ss_failed_dn_registrations            | Critical | Detected critical DN registration failures on Pulse DCU Stat Server. | pulse_statserver_dn_failed, pulse_statserver_dn_registered | 0.5%           |
| pulse_dcu_critical_ss_server_not_connected_configserver  | Critical | Pulse DCU Stat Server is not connected to ConfigServer.              | pulse_statserver_server_connected_seconds                  | for 15 minutes |
| pulse_dcu_critical_ss_server_not_connected_tserver       | Critical | Pulse DCU Stat Server is not connected to T-Servers.                 | pulse_statserver_server_connected_number                   | 2              |
| pulse_dcu_critical_ss_server_not_connected_ixnserver     | Critical | Pulse DCU Stat Server is not connected to IxnServers.                | pulse_statserver_server_connected_seconds                  | 2              |

# Logging

## Contents

- [1 Log output](#)
  - [1.1 Output all logs to stdout](#)
  - [1.2 Common log output settings](#)
- [2 Log level](#)

Learn how to store logs for Genesys Pulse.

**Related documentation:**

- 
- 
- 
- 

**RSS:**

- [For private edition](#)

## Log output

For verbose log levels of legacy components, set the output to **HostPath or PVC** as described in the following table:

| Pod         | Container   | Stdout   | HostPath or PVC |
|-------------|-------------|----------|-----------------|
| pulse       | pulse       | info     | HTTP access log |
| dcu         | csproxy     | standard | all             |
| dcu         | collector   | standard | all             |
| dcu         | statsserver | standard | all             |
| lds         | csproxy     | standard | all             |
| lds         | lds         | standard | all             |
| permissions | csproxy     | standard | all             |
| permissions | permissions | debug    | -               |

## Output all logs to stdout

To output all logs to stdout only, use simple config:

1. Set the following values in the .yaml files for the Pulse, DCU, LDS, and Permissions charts:

```
common log configuration
log:
 # log volume type: none | hostpath | pvc
 volumeType: none
```

2. Apply tenant configuration using GWS API::

1. Run the configurator container:

```
docker run \
```



```
-e CCID= \
-e GAUTHURL= \
-e GWSURL= \
-e CLIENT_ID= \
-e CLIENT_SECRET= \
-e CONFIG_PATH=/usr/src/scripts/cfgupdate/data \
-v :/usr/src/scripts/cfgupdate/data \
--rm /pulse/configurator: sh ./cfgupdate/run.sh
```

### Where:

- is the command to run (plan or apply).
  - / is the credentials pair, with Client Credentials grant type.
  - is the absolute path to the folder containing configuration files (**pulse-csp.json**, **pulse-dcu-statserver.json**, **pulse-lds.json**, and **pulse-lds-vq.json**).
2. Set the following values in **pulse-csp.json**, **pulse-dcu-statserver.json**, **pulse-lds.json**, and **pulse-lds-vq.json**:

```
{
 "log": {
 "all": "stdout"
 }
}
```

## Common log output settings

Instead of outputting all logs to stdout, you can control common log output settings individually, by setting the following options:

```
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: hostpath
 # log volume hostpath, used with volumeType "hostpath"
 volumeHostPath: /mnt/log
 # log PVC parameters, used with volumeType "pvc"
 pvc:
 name:
 accessModes:
 - ReadWriteMany
 capacity: 10Gi
 class: files-standard-zrs-retain
```

Where **name**: is set as follows:

- For Pulse chart:

```
pvc:
 name: pulse-logs
 ...
```

- For DCU chart:

```
pvc:
```

## Logging

---

```
 name: pulse-dcu-logs
 ...
```

- For LDS chart:

```
 pvc:
 name: pulse-lds-logs
 ...
```

- For Permissions Chart:

```
 pvc:
 name: pulse-permissions-logs

```

## Log level

Set Pulse container (Pulse pod) log level in **values-override-pulse.yaml**:

```
options:
 # application log level
 logLevel: INFO
 # append access log
 accessLogAppend: true
 # enable access log
 accessLogEnable: true
 # enable extended NCSA format
 accessLogExtended: false
 # enable logging of the request cookies
 accessLogCookies: false
 # enable logging of request processing time
 accessLogLatency: false
 # enable logging of the request hostname
 accessLogServer: false
 # number of days before rotated log files are deleted
 accessLogRetentionPeriod: 31
```

Set the Collector container (DCU pod) log level in **values-override-dcu.yaml**:

```
collector:
 appOptions:
 # [log]
 collector__log__buffering: "true"
 collector__log__time_format: iso8601
 collector__log__verbose: all
 collector__log__segment: "1 hr"
 collector__log__expire: "3 day"
 collector__log__standard: stdout
 # [log-db]
 collector__log_db__buffering: "true"
 collector__log_db__time_format: iso8601
 collector__log_db__verbose: standard
 collector__log_db__segment: "1 hr"
 collector__log_db__expire: "3 day"
 collector__log_db__standard: stdout
```

Set the Permissions container (Permissions pod) log level in **values-override-permissions.yaml**:

## Logging

---

```
app:
 options:
 # Level of logging. Can be NONE, ERROR, WARN, INFO, DEBUG
 logLevel: DEBUG
```