Genesys Pulse Private Edition Guide

Logging

5/25/2026

Contents

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	statserver	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:
```

```
        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**:

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