



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.

IWD Data Mart Private Edition Guide

Table of Contents

| | |
|---|----|
| Overview | |
| About IWD Data Mart | 6 |
| Architecture | 7 |
| High availability and disaster recovery | 9 |
| Configure and deploy | |
| Before you begin | 10 |
| Configure IWD Data Mart | 13 |
| Provision IWD Data Mart | 20 |
| Deploy IWD Data Mart | 21 |
| Upgrade, rollback, or uninstall IWD Data Mart | 25 |
| Observability | |
| Observability in IWD Data Mart | 27 |
| IWDDM metrics and alerts | 30 |

Contents

- [1 Overview](#)
- [2 Configure and deploy](#)
- [3 Observability](#)

Find links to all the topics in this guide.

Related documentation:

-

Early Adopter Program

Genesys Multicloud CX private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

IWD Data Mart (IWDDM) is a service available with the Genesys Multicloud CX private edition offering.

Overview

Learn more about IWD Data Mart, its architecture, and how to support high availability and disaster recovery.

- About IWD Data Mart
- Architecture
- High availability and disaster recovery

Configure and deploy

Find out how to configure and deploy IWD Data Mart.

- Before you begin
- Configure IWD Data Mart
- Provision IWD Data Mart
- Deploy IWD Data Mart
- Upgrade, rollback, or uninstall IWD Data Mart

Observability

Learn how to monitor IWD Data Mart with metrics and logging.

- IWDDM metrics and alerts
-

About IWD Data Mart

Learn about iWD Data Mart and how it works in Genesys Multicloud CX private edition.

Related documentation:

-

Early Adopter Program

Genesys Multicloud CX private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

IWD Data Mart offers comprehensive reporting, providing management insight into business operation. It provides key indicators of performance both through current-day statistics and on an historical basis.

IWD Data Mart retrieves reporting events from Intelligent Workload Distribution (IWD) and populates task fact tables, dimensions, and aggregate tables in IWD Data Mart DB. Data from IWD Data Mart is displayed using Genesys Customer Experience Insights (GCXI) IWD reports.

Architecture

Learn about the iWD Data Mart's architecture.

Related documentation:

-

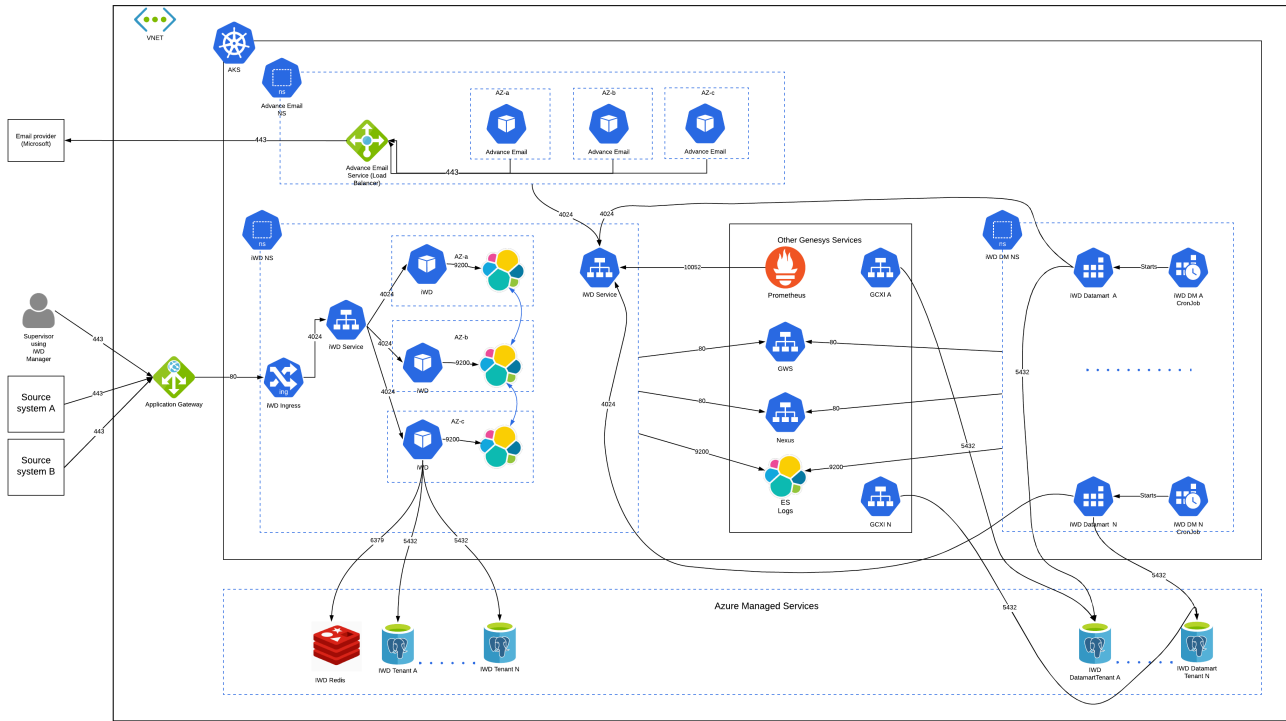
Early Adopter Program

Genesys Multicloud CX private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

IWD Data Mart is an ETL and aggregation job run via K8s CronJob every 15 minutes. The results are written to a separate DB instance for each tenant.

The architecture diagram in this topic illustrates a sample deployment of IWD Data Mart. The diagram shows Azure as the reference implementation.

For more information on the Genesys Multicloud CX private edition architecture, refer to the Architecture topic in the *Setting up Genesys Multicloud CX private edition* document.



High availability and disaster recovery

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

Related documentation:

-

Early Adopter Program

Genesys Multicloud CX private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

IWD Data Mart is a Cronjob that runs on a per-tenant basis, so High Availability (HA) is not applicable.

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

High Availability of IWD Data Mart is based on the CronJob's limited amount of working time and fact that no data is destroyed even if the IWD Data Mart job failed in progress.

Before you begin

Contents

- [1 Limitations and assumptions](#)
- [2 Download the Helm charts](#)
- [3 Third-party prerequisites](#)
- [4 Storage requirements](#)
- [5 Network requirements](#)
- [6 Browser requirements](#)
- [7 Genesys dependencies](#)
- [8 GDPR support](#)

Find out what to do before deploying IWD Data Mart.

Related documentation:

-

Early Adopter Program

Genesys Multicloud CX private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

Limitations and assumptions

The current version of IWD Data Mart:

- works as a short-living job started on schedule
- does not support scaling or HA
- requires dedicated PostgreSQL deployment per customer

IWD Data Mart is a short-living job, so Prometheus metrics cannot be pulled. Therefore, it requires a standalone Pushgateway service for monitoring.

Download the Helm charts

IWD Data Mart in Genesys Multicloud CX private edition includes the following containers:

- `iwd_dm_cloud`

The service also includes a Helm chart, which you must deploy to install the required containers for IWD Data Mart:

- `iwddm-cronjob`

See Helm Charts and Containers for IWD and IWD Data Mart for the Helm chart version you must download for your release.

To download the Helm chart, navigate to the **iwddm-cronjob** folder in the JFrog repository. For

Before you begin

information about how to download the Helm charts, see [Downloading your Genesys Multicloud CX containers](#).

Third-party prerequisites

Third-party services

| Name | Version | OpenShift | GKE | Purpose | Shared service? | Notes |
|------------|---------|-----------|-----|----------------------|-----------------|-------|
| PostgreSQL | 11.x | | | Relational database. | Optional | |

Storage requirements

All data is stored in PostgreSQL, which is external to the IWD Data Mart.

Network requirements

Not applicable

Browser requirements

Not applicable

Genesys dependencies

Intelligent Workload Distribution (IWD) with a provisioned tenant.

For the order in which the Genesys services must be deployed, refer to the [Order of services deployment](#) topic in the *Setting up Genesys Multicloud CX private edition* document.

GDPR support

Content coming soon

Configure IWD Data Mart

Contents

- [1 Override Helm chart values](#)
- [2 Configure Kubernetes](#)
 - [2.1 ConfigMaps](#)
 - [2.2 Secrets](#)
 - [2.3 Create the pull secret](#)
- [3 Configure security](#)
 - [3.1 Arbitrary UIDs in OpenShift](#)
- [4 Configure IWD REST endpoint](#)

Learn how to configure IWD Data Mart.

Related documentation:

-

Early Adopter Program

Genesys Multicloud CX private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

Override Helm chart values

The following table provides information on the IWD Data Mart deployment settings:

| Key | Type | Default | Description |
|--------------------------|--------|-----------------------------|--|
| image.imagePullSecrets | list | <code>[]</code> | imagePullSecrets must have the following format: - name: pullSecret1 - name: pullSecret2 |
| image.pullPolicy | string | <code>"IfNotPresent"</code> | Image pull policy |
| image.registry | string | <code>""</code> | Image registry |
| image.repository | string | <code>""</code> | Image repository |
| image.tag | string | <code>""</code> | Image tag |
| iwddm.cronjob.schedule | string | <code>"*/15 * * * *"</code> | IWDDM cronjob start schedule in cron format |
| iwddm.cronjob.suspend | bool | <code>false</code> | IWDDM suspend: if true, it won't start |
| iwddm.db.createConfigmap | bool | <code>false</code> | Create a config map for db access parameters |
| iwddm.db.dbname | string | <code>nil</code> | IWDDM database name |
| iwddm.db.host | string | <code>nil</code> | IWDDM database hostname |
| iwddm.db.port | int | <code>5432</code> | IWDDM database port |
| iwddm.db.secret.enabled | bool | <code>false</code> | Create IWDDM database secret |

| Key | Type | Default | Description |
|--------------------------------------|--------|--|---|
| iwddm.db.secret.password | string | `nil` | IWDDM database password to put in the secret |
| iwddm.db.secret.secretName | string | `nil` | IWDDM database secret name |
| iwddm.db.user | string | `nil` | IWDDM database user |
| iwddm.env.batchSize | int | `10000` | Events load batch size |
| iwddm.env.executionChain | string | `"full"` | |
| iwddm.env.httpProxyUrl | string | `` | IWDDM will try to connect to IWD using http proxy |
| iwddm.env.httpsProxyUrl | string | `` | IWDDM will try to connect to IWD using https proxy |
| iwddm.env.monitoring.enabled | bool | `false` | Enable push metrics to Pushgateway |
| iwddm.env.monitoring.pushgateway_url | string | `` | Since IWDDM is a short living cronjob, it uses Pushgateway for providing metrics |
| iwddm.env.restUrl | string | `` | IWD REST endpoint url |
| iwddm.env.wfm | object | `{"enabled":false}` | Push data to WFM if enabled |
| iwddm.helmTest.args | list | `["export PGPASSWORD=\${IWDDM_PASSWORD}, psql -c \"select now()\""]` | Arguments passed to the helm test command |
| iwddm.helmTest.command | list | `["/bin/bash", "-c"]` | Command for helm test |
| iwddm.helmTest.image | string | `"postgres:latest"` | Image for helm test |
| iwddm.labels | object | `{}` | Extra labels ref: https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/ |
| iwddm.nodeSelector | object | `{}` | Node labels for assignment. ref: https://kubernetes.io/docs/user-guide/node-selection/ |
| iwddm.priorityClassName | string | `` | Priority Class ref: https://kubernetes.io/docs/concepts/configuration/pod-priority-preemption/ |

| Key | Type | Default | Description |
|---------------------------------------|--------|------------|---|
| iwddm.resources.limits.cpustring | string | `"100m"`` | IWDDM Kubernetes CPU limit |
| iwddm.resources.limits.memorystring | string | `"160Mi"`` | IWDDM Kubernetes memory limit |
| iwddm.resources.requests.cpustring | string | `"50m"`` | IWDDM Kubernetes CPU request |
| iwddm.resources.requests.memorystring | string | `"128Mi"`` | IWDDM Kubernetes memory request |
| iwddm.securityContext | object | `{}` | Security Context ref: https://kubernetes.io/docs/tasks/configure-pod-container/security-context/#set-the-security-context-for-a-container Containers should run as genesys user and cannot use elevated permissions Note: These options must not be changed unless instructed by Genesys. |
| iwddm.serviceAccount.create | bool | `true` | Create service account for IWDDM |
| iwddm.serviceAccount.name | string | `nil` | The name of the ServiceAccount to use. If not set and create is true, a name is generated using the fullname template |
| iwddm.tenantId | int | `1` | Tenant short 4-digit ID |
| iwddm.volumeMounts | object | `{}` | Volumes mounted into an IWDDM container |
| iwddm.volumes | string | `nil` | Must be declared with starting - since they are parsed as a template |
| podAnnotations | object | `{}` | Add annotations to all pods |
| podLabels | object | `{}` | Add labels to all pods |

Configure Kubernetes

ConfigMaps

Not applicable as all required ConfigMaps are created via Helm Chart basing on the provided values.

Secrets

IWD Data Mart requires several secrets to be created.

Create the pull secret

Use the following code snippet as an example of how to create pull secret:

```
kubectll create secret docker-registry mycred --docker-server=pureengage.jfrog.io --docker-username= --docker-password=
```

You can add *mycred* to Helm override values by setting **image.imagePullSecrets** to *[mycred]*.

For OpenShift, you may add the *mycred* secret in current namespace as the default pull secret:

```
oc secrets link default mycred --for=pull
```

IWDDM PostgreSQL password

It can be configured in Helm values as shown in the sample:

```
iwddm:
  db:
    secret:
      enabled: true
      secretName: iwddm-db-secret-0001
      password: somePassword1
    volumes: |-
      - name: iwddm-db-secrets
        secret:
          secretName: iwddm-db-secret-0001

    volumeMounts: |-
      - name: iwddm-db-secrets
        readOnly: true
        mountPath: "/mnt/env-file-secrets/db-file-secrets"
```

IWDDM iWD x-api-key

It must exist before your IWDDM deployment. It must be created with the following format:

```
IWDDM_API_KEY=value
```

Sample password creation:

```
kubectll create secret generic 0001-iwd-secrets \
```

Configure IWD Data Mart

```
--from-literal='IWDDM_API_KEY=123456-abcde-4568734'
```

IWD x-api-key can be configured as shown below:

```
iwddm:
  volumes: |-
    - name: iwd-secrets
      secret:
        secretName: 0001-iwd-secrets

  volumeMounts: |-
    - name: iwd-secrets
      readOnly: true
      mountPath: "/mnt/env-file-secrets/iwd-file-secrets"
```

GIM database password

Create a multi-line secret in the following way:

Create a file, **gim-secret.txt** with the following text:

```
IWDDM_GIM_DBUSER=
IWDDM_GIM_PASSWORD=
IWDDM_GIM_URL=jdbc:postgresql://:5432/
```

Create a secret from the file:

```
kubectl create secret generic gim-secrets \
--from-file=gim-secret.txt
```

Then, mount it using Helm values:

```
iwddm:
  volumes: |-
    - name: gim-secrets
      secret:
        secretName: gim-secrets

  volumeMounts: |-
    - name: gim-secrets
      readOnly: true
      mountPath: "/mnt/gim-secrets/gim-secret"
```

Configure security

IWDDM needs the username and password provided for PostgreSQL access.

Arbitrary UIDs in OpenShift

To use arbitrary UIDs in your OpenShift deployment, ensure the following settings:

Configure IWD Data Mart

```
securityContext:  
  runAsNonRoot: true  
  runAsUser: null  
  runAsGroup: 0
```

Configure IWD REST endpoint

IWD REST URL provided for IWDDM in the **iwddm.env.restUrl** parameter must be:

```
http://iwd.iwd.svc.cluster.local:4024/iwd/v3
```

The above endpoint may change depending on the configuration and where IWD is installed. Refer to IWD documentation for information on this endpoint.

Provision IWD Data Mart

- Administrator

Learn how to provision IWD Data Mart.

Related documentation:

-
-

Early Adopter Program

Genesys Multicloud CX private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

Create a Postgres database for IWDDM.

Create a user for IWDDM full access to database:

```
CREATE USER "iwddm_dm" WITH LOGIN NOSUPERUSER NOCREATEDB NOCREATEROLE NOINHERIT NOREPLICATION CONNECTION LIMIT -1 PASSWORD '';
```

IWDDM should provide access to its database to GCXI. For that, an additional user account can be created with the following script:

```
CREATE USER "iwddm_gcxi" WITH LOGIN NOSUPERUSER NOCREATEDB NOCREATEROLE NOINHERIT NOREPLICATION CONNECTION LIMIT -1 PASSWORD '';  
ALTER DEFAULT PRIVILEGES FOR USER iwddm_dm GRANT SELECT ON TABLES TO iwddm_gcxi;
```

After creating an account, the new user account can be provided to the GCXI service for database access.

Deploy IWD Data Mart

Contents

- [1 Prepare your environment](#)
- [2 Deploy](#)
- [3 Validate the deployment](#)

Learn how to deploy IWD Data Mart.

Related documentation:

-

Early Adopter Program

Genesys Multicloud CX private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

Prepare your environment

Important

- Make sure to review Before you begin for the full list of prerequisites required to deploy IWD Data Mart.
- The sample code snippets and install commands in this document use an example version of IWD Data Mart. Ensure that you replace the example version with the version that is applicable for your deployment.

1. Create a new project using the following command:

```
kubectl create namespace iwddm
```

For OpenShift:

```
oc get clusterversion  
oc new-project iwddm
```

2. Create a pull secret for accessing the JFrog registry. See Configure Kubernetes.
3. Download the IWD helm chart from the JFrog repository. See Download the Helm charts.
4. IWD Data Mart requires the Digital Channels API key. The key must be provisioned and shared via Digital Channels or IWD. See IWD x-api-key.

Deploy

Extract parameters from chart to see multiple (default) values used to fine tune the installation.

```
$ helm show values /iwddm- > values.yaml
```

Set up essential IWDDM Helm values:

1. image.registry
2. image.imagePullSecrets (if needed)
3. image.repository
4. image.tag
5. image.repository
6. iwddm.tenantId
7. iwddm.db.*
8. iwddm.db.secret.*
9. iwddm.volumes
10. iwddm.volumeMounts
11. iwddm.env.gim.enabled: true (given that GIM DB secret is provided)

Sample override file:

```
image:
  registry: "pureengage-docker-staging.jfrog.io"
  repository: "iwddm/iwd_dm_cloud"
  tag: ""
  pullPolicy: IfNotPresent
  imagePullSecrets:
    - name: pullsecret
iwddm:
  tenantId: #sample 100
  db:
    createConfigmap: true
    host:
    port: 5432
    dbname:
    user:
    secret:
      enabled: true
      secretName:
      password:
  cronjob:
    schedule: "*/3 * * * *"
    suspend: false
  securityContext: {}
  env:
    executionChain: "full"
    restUrl: "http://iwd.iwd.svc.cluster.local:4024/iwd/v3"
  monitoring:
    enabled: false
    pushgateway_url: ""
```

Deploy IWD Data Mart

```
volumes: |-
  - name: iwddm-db-secrets
    secret:
      secretName:
  - name: iwd-secrets
    secret:
      secretName:
volumeMounts:
  iwddm-db-secrets:
    readOnly: true
    mountPath: "/mnt/env-secrets/db-secrets"
  iwd-secrets:
    readOnly: true
    mountPath: "/mnt/env-secrets/iwd-secrets"
```

Deploy IWD Data Mart using the following command:

```
helm upgrade --install iwddm-{short_tenant_id} /iwddm-cronjob --version={version} -f
./values.private.yml
```

Validate the deployment

Watch the helm output at the end of installation. Pods must be in a Running state and they must pass all READY checks.

See the following sample output:

```
Release "iwddm" has been upgraded. Happy Helming!
NAME: iwddm
LAST DEPLOYED: Tue Jul 18 10:18:07 2021
NAMESPACE: iwddm
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
Please be patient while iwddm 100.0.0741322 is being deployed
```

Note that IWDDM is a short-living job. So, pods will be created or deleted based on schedule.

Upgrade, rollback, or uninstall IWD Data Mart

Contents

- [1 Upgrade IWD Data Mart](#)
- [2 Rollback IWD Data Mart](#)
- [3 Uninstall IWD Data Mart](#)

Learn how to upgrade, rollback or uninstall IWD Data Mart.

Related documentation:

-

Early Adopter Program

Genesys Multicloud CX private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

Upgrade IWD Data Mart

Update Docker image versions in the values file or get a new Helm chart, then install it with the following command:

```
helm upgrade --install -f
```

Rollback IWD Data Mart

Set the previous Docker image versions in the values file or get the previous Helm chart, then install it with the following command:

```
helm upgrade --install -f
```

Uninstall IWD Data Mart

Uninstall IWDDM using:

```
helm uninstall
```

Manually delete all external secrets, if needed.

Observability in IWD Data Mart

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 IWD Data Mart.

Related documentation:

-
-

Early Adopter Program

Genesys Multicloud CX private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

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 IWD Data Mart metrics, see:

- IWD DataMart metrics

See also System metrics.

Enable monitoring

| Service | CRD or annotations? | Port | Endpoint/Selector | Metrics update interval |
|--------------|---------------------|------|-------------------|-------------------------|
| IWD DataMart | n/a | n/a | n/a | n/a |

Configure metrics

IWD Data Mart works as a short living cronjob, so it uses Pushgateway for providing metrics. To enable pushing metrics, find the following Helm values as an example:

```
iwddm:
  env:
    monitoring:
      enabled: true
      pushgateway_url: "http://prometheus-pushgateway.monitoring.svc.cluster.local:9091"
```

Alerting

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

Important

While you can use general third-party functionality to create rules to trigger alerts based on metrics values you specify, private edition does not enable you to create custom alerts, and Genesys does not provide support for custom alerting.

For descriptions of available IWD Data Mart alerts, see:

- IWD DataMart alerts

Configure alerts

Private edition services define a number of alerts by default (for IWD Data Mart, 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, IWD Data Mart does not support customizing the alerts or defining additional **PrometheusRule** objects to create alerts based on the service-provided metrics.

Logging

IWDDM container logs to *stdout* in structured JSON format.

IWDDM metrics and alerts

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

Related documentation:

-

Early Adopter Program

Genesys Multicloud CX private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

Contents

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

| Service | CRD or annotations? | Port | Endpoint/Selector | Metrics update interval |
|--|---------------------|------|-------------------|-------------------------|
| IWDDM | n/a | n/a | n/a | n/a |
| Note: As a serverless component, IWDDM is run via a Kubernetes CronJob. By default, the job runs every 15 minutes and pushes information into the Prometheus Pushgateway. | | | | |

See details about:

- IWDDM metrics
- IWDDM alerts

Metrics

Labeling is used to distinguish the characteristics of the metrics that is being measured. Along with the metrics the following labels can be used:

| Label | Description |
|-----------------|--|
| tenant | Name of the tenant taken from IWDDM_CCID environment variable |
| job | Job name. Possible value is: <ul style="list-style-type: none"> • iwddm_metrics |
| execution_chain | Indicates execution chain. Possible values are: <ul style="list-style-type: none"> • intraday • historical • initialize • full |
| table | Allows to organize metrics by table names |
| dimension | Allows to organize metrics by dimension names |

| Metric and description | Metric details | Indicator of |
|---|--|--------------|
| iwddm_job_active Indicates that IWDDM is active Values: <ul style="list-style-type: none"> • 1 - job is started • 0 - job is stopped or failed | Unit: Type: gauge Label: tenant, job, execution_chain Sample value: iwddm_job_active{execution_chain="TEST", job="iwddm_metrics", tenant="TEST"} 1 | Error |
| iwddm_job_last_start | Unit: milliseconds | Error |

| Metric and description | Metric details | Indicator of |
|---|--|--------------|
| <p>Indicates when the IWDDM job started</p> <p>Value:</p> <p>Unix timestamp when job started</p> | <p>Type: gauge Label: tenant, job, execution_chain Sample value: iwddm_job_last_start{execution_chain="TEST", job="iwddm_metrics", tenant="TEST"} 1618322383</p> | |
| <p>iwddm_job_last_success</p> <p>Indicates when the IWDDM job succeeded</p> <p>Value:</p> <p>Unix timestamp when job succeeded</p> | <p>Unit: milliseconds</p> <p>Type: gauge Label: tenant, job, execution_chain Sample value: iwddm_job_last_success{execution_chain="TEST", job="iwddm_metrics", tenant="TEST"} 1618322383</p> | Error |
| <p>iwddm_job_last_fail</p> <p>Indicates when the IWDDM job failed</p> <p>Value:</p> <p>Unix timestamp when job failed</p> | <p>Unit: milliseconds</p> <p>Type: gauge Label: tenant, job, execution_chain Sample value: iwddm_job_last_fail{execution_chain="TEST", job="iwddm_metrics", tenant="TEST"} 1618322383</p> | Error |

Alerts