



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

[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
- 4 Configure IWD REST endpoint
- 5 Configure connection to WFM

Learn how to configure IWD Data Mart.

Related documentation:

-
-
-

RSS:

- [For private edition](#)

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
iwddm.db.secret.password	string	<code>nil</code>	IWDDM database password to put in the secret

Key	Type	Default	Description
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/
iwddm.resources.limits.cpu	string	`"100m"`	IWDDM Kubernetes CPU limit
iwddm.resources.limits.memory	string	`"160Mi"`	IWDDM Kubernetes

Key	Type	Default	Description
			memory limit
iwddm.resources.requests.cpu	string	`"50m"`	IWDDM Kubernetes CPU request
iwddm.resources.requests.memory	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:

```
kubectl 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]*.

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:

```
kubectl create secret generic 0001-iwd-secrets \
--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:

```
kubectll 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.

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.

Configure connection to WFM

IWD DM can connect and push metrics to Workforce Management (WFM).

The following Helm values must be configured:

Helm key	Sample value
iwddm.env.wfm.enabled	true
iwddm.env.wfm.health_url	http://wfm-t-backend.wfm.svc.cluster.local.:7010/wfm/api/v3
iwddm.env.wfm.rest_url	http://wfm-t-backend.wfm.svc.cluster.local.:7010/?Handler=DISCO
iwddm.env.wfm.gauth.enabled	true
iwddm.env.wfm.gauth.rest_url	http://gauth-auth-active.gauth.svc.cluster.local.:80
iwddm.env.wfm.gauth.client_id	
iwddm.env.wfm.gauth.client_secret	valueFrom: secretKeyRef: key: name: