IWD Data Mart Private Edition Guide

Deploy IWD Data Mart

5/13/2026

Contents

Learn how to deploy IWD Data Mart (IWDDM) into a private edition environment.

Related documentation:

-
-
-
-

RSS:

- For private edition

> **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.

Kubernetes

Prepare

1. Create a new project using the following command:

   ```
   kubectl create namespace 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

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

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

2. Set up essential IWDDM Helm values:
    - image.registry
    - image.imagePullSecrets (if needed)
    - image.repository
    - image.tag
    - image.repository
    - iwddm.tenantId
    - iwddm.db.*
    - iwddm.db.secret.*
    - iwddm.volumes
    - iwddm.volumeMounts
    - iwddm.env.gim.enabled: true (given that GIM DB secret is provided)
      Use the 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: ""
  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"
```

3. Install IWD Data Mart using the following command:

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

Google Kubernetes Engine (GKE)

Prepare

1. Log in to the GKE cluster.

```
gcloud container clusters get-credentials
```

2. Create a new project:
    1. Create a *create-iwddm-namespace.json* :

```
{
  "apiVersion": "v1",
  "kind": "Namespace",
  "metadata": {
    "name": "iwddm",
    "labels": {
      "name": "iwddm"
    }
  }
}
```

    2. Create a namespace using the above JSON:

```
kubectl apply -f create-iwddm-namespace.json
```

    3. Confirm the namespace creation:

```
kubectl describe namespace iwddm
```

3. Create a pull secret for accessing the JFrog registry. See Configure Kubernetes.
4. Download the IWD helm chart from the JFrog repository. See Download the Helm charts.
5. 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

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

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

2. Set up essential IWDDM Helm values:
    - image.registry
    - image.imagePullSecrets (if needed)
    - image.repository
    - image.tag
    - image.repository
    - iwddm.tenantId
    - iwddm.db.*
    - iwddm.db.secret.*
    - iwddm.volumes
    - iwddm.volumeMounts
    - iwddm.env.gim.enabled: true (given that GIM DB secret is provided)
      Use the 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: ""
  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"
```

3. Install IWD Data Mart using the following command:

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

Azure Kubernetes Service (AKS)

Prepare

1. Log in to the AKS cluster.

```
az aks get-credentials --resource-group  --name  --admin
```

2. Create a new project:
    1. Create a *create-iwddm-namespace.json* :

```
{
  "apiVersion": "v1",
  "kind": "Namespace",
  "metadata": {
    "name": "iwddm",
    "labels": {
      "name": "iwddm"
    }
  }
}
```

    2. Create a namespace using the above JSON:

```
kubectl apply -f create-iwddm-namespace.json
```

    3. Confirm the namespace creation:

```
kubectl describe namespace iwddm
```

3. Create a pull secret for accessing the JFrog registry. See Configure Kubernetes.
4. Download the IWD helm chart from the JFrog repository. See Download the Helm charts.
5. 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

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

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

2. Set up essential IWDDM Helm values:
    - image.registry
    - image.imagePullSecrets (if needed)
    - image.repository
    - image.tag
    - image.repository
    - iwddm.tenantId
    - iwddm.db.*
    - iwddm.db.secret.*
    - iwddm.volumes
    - iwddm.volumeMounts
    - iwddm.env.gim.enabled: true (given that GIM DB secret is provided)
      Use the sample override file:

```
image:
  registry: "pureengageuse1-docker-multicloud.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.${IWD_NAMESPACE}.svc.${DNS_SCOPE}:4024/iwd/v3"
    monitoring:
      enabled: false
      pushgateway_url: ""
  gim:
    enabled: true
  wfm:
    enabled: false
    rest_url:   "http://wfm-t101-backend.${WFM_NS}.svc.${DNS_SCOPE}:7010/wfm/api/v3"
    health_url: "http://wfm-
t101-backend.${WFM_NS}.svc.${DNS_SCOPE}:7010/?Handler=DISCO"
  gauth:
    enabled: false
    rest_url: http://gauth-auth.${GAUTH_NAMESPACE}.svc.${DNS_SCOPE}:80
    client_id: iwddm_client
```

```
    client_secret:
      valueFrom:
        secretKeyRef:
          name: shared-gauth-iwddm-client-secret
          key: gauth-iwddm-client-secret
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"
```

3. Install 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: iwdem
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.