Intelligent Workload Distribution Private Edition Guide

Deploy Intelligent Workload Distribution

4/15/2026

Contents

Learn how to deploy Intelligent Workload Distribution (IWD) 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.

Kubernetes

Prepare

1. Create a new project using the following command:

   ```
   kubectl create namespace iwd
   ```

2. Create a pull secret for accessing the JFrog registry. See Create the pull secret.
3. Download the IWD helm chart from the JFrog repository. See Download the Helm charts.
4. Create a gauth client.
   IWD requires *clientId* and *clientSecret* registered in Authentication Service. These must be provided during Helm Chart deployment. Create new client credentials if they are not already created . Refer to the GWS documentation for more information.

Deploy

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

   ```
   $ helm show values iwd-.tgz > values.yaml
   ```

   For information on parameters and values in the **values.yaml** file, see Override Helm chart values.
   Sample override file:

   ```
   replicaCount: 1

   image:
     registry: pureengage-docker-staging.jfrog.io
     repository: nexus/iwd
     pullSecrets: []

   gauth:
     auth:
       url: http://gauth-auth.gauth
       redirectUrl: https://gauth.${domain}

   redis:
     nodes: redis://infra-redis-redis-cluster.infra.svc.cluster.local:6379
     useCluster: true
     enableTLS: false
     password:

   gws:
     url: http://gauth-auth.gauth
     clientId:
     clientSecret:
     apiKey:

   ingress:
     enabled: true
     hosts:
       - host: iwd.${domain}
         paths:
           - path: '/iwd/'
             port: 4024
     tls:
       - hosts:
           - iwd.${domain}
         secretName: letsencrypt

   nexus:
     url: http://nexus.nexus
     apikey:

   elasticsearch:
     host: elastic-es-http.infra.svc.cluster.local
     port: 9200
   ```

2. Install IWD using the following command (replace with applicable values):

   ```
   helm install iwd ./iwd-.tgz -f override_values.yaml
   --set gws.clientId=
   --set gws.clientSecret=
   --set redis.password=
   --set nexus.apikey=
   --set gws.apiKey='None'
   --namespace=iwd
   ```

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-iwd-namespace.json* :

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

   2. Create a namespace using the above JSON:

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

   3. Confirm the namespace creation:

      ```
      kubectl describe namespace iwd
      ```

3. Create a pull secret for accessing the JFrog registry.

   ```
   kubectl create secret docker-registry jfrog-stage-credentials \
    --docker-server=pureengage-docker-staging.jfrog.io \
    --docker-username= \
    --docker-password= \
    --docker-email=
   ```

4. Download the IWD helm chart from the JFrog repository. See Download the Helm charts.

5. Create a gauth client.
   IWD requires *clientId* and *clientSecret* registered in Authentication Service. These must be provided during Helm Chart deployment. Create new client credentials if they are not already created . Refer to the GWS documentation for more information.

Deploy

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

   ```
   $ helm show values iwd-.tgz > values.yaml
   ```

   For information on parameters and values in the **values.yaml** file, see Override Helm chart values.

Sample override file:

```
replicaCount: 1

image:
  registry: pureengage-docker-staging.jfrog.io
  repository: nexus/iwd
  pullSecrets:
    - name: "pullsecret"

gauth:
  auth:
    url: http://gauth-auth.gws
    redirectUrl: https://gws.nlb02-useast1.gcpe002.gencpe.com

redis:
  nodes: redis://infra-redis-redis-cluster.infra.svc.gke2-useast1.gcpe002.gencpe.com:6379
  useCluster: true
  enableTLS: false
  #password: xxx #in secrets

gws:
  url: http://gauth-auth.gws
  #clientId: xxx #in secrets
  #clientSecret: xxx #in secrets
  #apiKey: xxx #in secrets

ingress:
  enabled: true
  hosts:
    - host: iwd.nlb02-useast1.gcpe002.gencpe.com
      paths:
        - path: '/iwd/'
          port: 4024
  annotations:
    cert-manager.io/issuer-name: ca-cluster-issuer
    kubernetes.io/ingress.class: nginx
  tls:
    - hosts:
      - iwd.nlb02-useast1.gcpe002.gencpe.com
      secretName: iwd-ingress-cert

nexus:
  url: http://nexus.nexus
  #apikey: xxx #in secrets

elasticsearch:
  host: elastic-elasticsearch-master.infra.svc.gke2-useast1.gcpe002.gencpe.com
  port: 9200

monitoring:
  # Deploy ServiceMonitor
  enabled: true
  # Create PrometheusRule k8s object with alarm definitions
  alarms: true
  # Create ConfigMap with Grafana Dashboards
  dashboards: true
```

2. Install IWD using the following command (replace with applicable values):

```
helm install iwd ./iwd-.tgz -f override_values.yaml
--set gws.clientId=
--set gws.clientSecret=
```

```
--set redis.password=
--set nexus.apikey=
--set gws.apiKey='None'
--namespace=iwd
```

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-iwd-namespace.json* :

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

   2. Create a namespace using the above JSON:

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

   3. Confirm the namespace creation:

      ```
      kubectl describe namespace iwd
      ```

3. Create a pull secret for accessing the JFrog registry.

   ```
   kubectl create secret docker-registry pullsecret \
   --docker-server=pureengageuse1-docker-multicloud.jfrog.io \
   --docker-username= \
   --docker-password= \
   --docker-email=
   ```

4. Download the IWD helm chart from the JFrog repository. See Download the Helm charts.
5. Create a gauth client.
   IWD requires *clientId* and *clientSecret* registered in Authentication Service. These must be provided during Helm Chart deployment. Create new client credentials if they are not already created . Refer to the GWS documentation for more information.

Deploy

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

   ```
   $ helm show values iwd-.tgz > values.yaml
   ```

   For information on parameters and values in the **values.yaml** file, see Override Helm chart values.
   Sample override file:

   ```
   replicaCount: 1

   image:
     registry: pureengageuse1-docker-multicloud.jfrog.io
     repository: nexus/iwd
     pullSecrets:
       - name: "pullsecret"

   gauth:
     auth:
       url: http://gauth-auth.${GAUTH_NAMESPACE}
       redirectUrl: https://gauth.${DOMAIN}

   redis:
     nodes: redis://${REDIS_ADDR}:${REDIS_PORT}
     useCluster: true
     enableTLS: false
     #password: xxx #in secrets

   gws:
     url: http://gauth-auth.${GAUTH_NAMESPACE}
     #clientId: xxx #in secrets
     #clientSecret: xxx #in secrets
     #apiKey: xxx #in secrets

   ingress:
     enabled: true
     hosts:
       - host: iwd.${domain}
         paths:
           - path: '/iwd/'
             port: 4024
     annotations:
       cert-manager.io/issuer-name: ca-cluster-issuer
       kubernetes.io/ingress.class: nginx
     tls:
       - hosts:
         - iwd.${domain}
         secretName: iwd-ingress-cert

   nexus:
     url: http://nexus.${NEXUS_NAMESPACE}
     #apikey: xxx #in secrets

   elasticsearch:
     host: ${ES_ADDR}
     port: 9200

   monitoring:
     # Deploy ServiceMonitor
     enabled: true
     # Create PrometheusRule k8s object with alarm definitions
     alarms: true
   ```

```
  # Create ConfigMap with Grafana Dashboards
  dashboards: true
```

2. Install IWD using the following command (replace with applicable values):

```
helm install iwd ./iwd-.tgz -f override_values.yaml
 --set gws.clientId=
 --set gws.clientSecret=
 --set redis.password=
 --set nexus.apikey=
 --set gws.apiKey='None'
 --namespace=iwd
```

Validate the deployment

Watch the helm output at the end of installation. It provides the status and additional information about where to log in to the IWD UI.

See the following sample output:

```
Release "iwd" has been upgraded. Happy Helming!
NAME: iwd
LAST DEPLOYED: Tue Jul 13 10:18:07 2021
NAMESPACE: iwd
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
Please be patient while iwd 100.0.0741322 is being deployed
```