Intelligent Workload Distribution Private Edition Guide

Configure IWD

5/25/2026

Contents

Learn how to configure IWD.

Related documentation:
-
-
-
-

RSS:

- For private edition

Override Helm chart values

The following table provides information on the IWD deployment settings that can be configured in the **values.yaml** file:

Parameter	Description	Default
`deploymentType`	Deployment type. Only two possible values are supported: Deployment, ReplicaSet	Deployment
`replicaCount`	Number of pods to be created	1
`image.registry`	Docker registry for iWD	pureengage-docker-staging.jfrog.io
`image.repository`	iWD Image name	nexus/iwd
`image.pullPolicy`	Image pull policy	IfNotPresent
`image.pullSecrets`	Specify docker-registry secret names as an array	[]
`affinity`	Map of node/pod affinities	{}
`nodeSelector`	Node labels for pod assignment	{}
`tolerations`	Tolerations for pod assignment	nil
`priorityClassName`	Priority class name	
`podSecurityContext`	Pod security context	{}
`securityContext`	Security context	{}
`podDisruptionBudget.enabled`	Enable or disable pod disruption budget	false
`podDisruptionBudget.minAvailable`	Set minimal number of pods available during the disruption	1
`podAnnotations`	Add annotations to pods	{}

Parameter	Description	Default
`podLabels`	Add custom labels to pods	{}
`hpa.enabled`	Enable or disable Horizontal Pod Autoscaler (HPA)	false
`hpa.minReplicas`	Minimal replicas count for HPA	1
`hpa.maxReplicas`	Maximal replicas count for HPA	10
`hpa.targetCPUPercent`	Specify target CPU utilization for HPA	60
`resources.limits.cpu`	Maximum amount of CPU K8s allocates for container	2000m
`resources.limits.memory`	Maximum amount of Memory K8s allocates for container	2000Mi
`resources.requests.cpu`	Guaranteed CPU allocation for container	300m
`resources.requests.memory`	Guaranteed Memory allocation for container	500Mi
`serviceAccount.create`	Specifies whether a service account should be created	false
`serviceAccount.annotations`	Annotations to add to service account	{}
`serviceAccount.name`	Service account name	""
`existingSecret`	Specify Secret name to read application secrets from	nil
`gauth.auth.url`	URL to Authentication service	nil
`gauth.auth.redirectUrl`	Redirect URL to Authentication service	nil
`redis.nodes`	Comma separate list of Redis nodes to connect	nil
`redis.useCluster`	Redis deployment mode	false
`redis.enableTLS`	Either to use TLS on Redis connection	false
`redis.password`	Access key for Redis authentication	nil
`nexus.url`	URL to Nexus	nil
`nexus.apikey`	Nexus API key	nil
`service.type`	Service type	ClusterIP
`monitoring.enabled`	Enable or disable pod monitor	false
`monitoring.alarms`	Create PrometheusRule k8s object with alarm definitions	false
`monitoring.dashboards`	Create ConfigMap with Grafana Dashboards	false
`networkPolicies.enabled`	Enable or disable network policies	false
`dnsConfig.options`	DNS Configuration options	{ name: ndots, value: "3" }

Configure Kubernetes

ConfigMaps

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

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.pullSecrets** to *[mycred]*.