Genesys Web Services and Applications Private Edition Guide

Configure GWS Ingress

6/2/2023

Contents

Learn how to configure GWS Ingress.

Related documentation:

-
-
-

RSS:

- For private edition

> **Warning**
> If you are deploying Genesys Web Services and Applications in a single namespace with other private edition services, then you do not need to deploy GWS ingress.

Override Helm chart values

You can specify parameters for the deployment by overriding Helm chart values in the **values.yaml** file. See the tables below for a full list of overridable values available for each container in GWS ingress.

For more information about how to override Helm chart values, see Overriding Helm chart values.

Parameters

Parameter	Description	Valid values	Default
podLabels	Custom labels for each pod.	A valid set of labels as "name: value"	{}
podAnnotations	Custom annotations for each pod.	A valid set of annotations as "name: value"	{}
priorityClassName	The class name Kubernetes uses to determine the priority of the pods for this container deployment relative to other pods. See the Kubernetes documentation for details.	A valid priority class name	""
	The Docker registry	A valid registry URL	""

Parameter	Description	Valid values	Default
imageGlobals.registry	from which Kubernetes pulls images.		
deploymentGlobals.deploymentTag	The deployment tag used as a suffix for the names of Kubernetes objects created by the chart. The value must be the same as the value in the GWS Helm chart.	Any lowercase alphanumeric value up to 8 characters long.	"live"
deploymentGlobals.strategy	The strategy GWS ingress uses to upgrade its containers.	RollingUpdate or Recreate	"RollingUpdate"
deploymentGlobals.securityContext.runAsNonRoot	Specifies whether the container must run as a non-root user.	true or false	true
deploymentGlobals.securityContext.runAsUser	The user ID to run the container process.	A valid user ID or null	500
deploymentGlobals.securityContext.runAsGroup	The group ID to run the container process.	A valid group ID or null	500
deploymentGlobals.securityContext.fsGroup	A supplemental group ID that applies to all containers in a pod.	A valid group ID or null	500
nodeSelector	The labels Kubernetes uses to assign pods to nodes. See the Kubernetes documentation for details.	Valid nodeSelector settings.	{}
gwsServiceProxy.deployment.replicaCount	The number of pod replicas in the deployment.	A number greater than 0	2
gwsServiceProxy.image.registry	The Docker registry from which Kubernetes pulls images. If set, this parameter overrides imageGlobals.registry.	A valid registry URL	""
gwsServiceProxy.image.repository	The name of the Docker registry repository.	A valid repository name	"gws-system-nginx"
gwsServiceProxy.image.pullPolicy	Specifies when Kubernetes pulls images from the registry on start up.	IfNotPresent or Always	"Always"
gwsServiceProxy.image.imagePullSecrets	The secret Kubernetes uses to get credentials to pull images from the registry.	A valid secret	[]

Parameter	Description	Valid values	Default
gwsServiceProxy.context.sessionCookieName	The cookie name for sticky sessions.	A valid cookie name	"GWSSESSIONID"
gwsServiceProxy.context.env.CONSUL_PORT	The port of the local Consul agent.	A valid port	8500
gwsServiceProxy.context.env.GWS_NGINX_ENABLE_MAPPING	Specifies whether mapping is enabled in Nginx.	true or false	false
gwsServiceProxy.context.env.GWS_NGINX_CONSUL_SERVICE	The name of the service to register in Consul.	A valid name	"system-nginx"
gwsServiceProxy.context.env.GWS_CONSUL_KV_PREFIX	The prefix used to locate GWS ingress data in the Consul KV datastore.	String	"gws"
gwsServiceProxy.livenessProbe.startupDelay	Specifies the time in seconds to wait before performing the first liveness probe.	Number	5
gwsServiceProxy.livenessProbe.checkInterval	Specifies the interval in seconds between liveness probes.	Number	10
gwsServiceProxy.readinessProbe.startupDelay	Specifies the time in seconds to wait before performing the first readiness probe.	Number	15
gwsServiceProxy.readinessProbe.checkInterval	Specifies the interval in seconds between readiness probes.	Number	20
gwsServiceProxy.service.ports	The HTTP ports used by service.	A valid set of ports as "name: value","port: value"	[{"name": "gws-service-proxy", "port": 80, "targetPort": 8080}, {"name": "gws-service-proxy-ext", "port": 81, "targetPort": 8081}]
entryPoints.internal.service.annotations	Custom annotations for the service.	A valid set of annotations as "name: value"	{}
entryPoints.internal.ingress.enabled	Specifies whether internal ingress is enabled. Set this value to false if you are deploying Genesys Web Services and Applications in a single namespace.	true or false	true
entryPoints.internal.ingress.ingressClassName	Defines which controller implements the Ingress resource. The value is directly propagated to the ingressClassName field of the Kubernetes	A valid IngressClass	""

Parameter	Description	Valid values	Default
	Ingress object. See Ingress and Ingress class in the Kubernetes documentation for details.		
entryPoints.internal.ingress.annotations	Custom annotations for internal ingress.	A valid set of annotations as "name: value"	{}
entryPoints.internal.ingress.paths	Paths to internal ingress, relative to the hostnames.	Valid paths	["/"]
entryPoints.internal.ingress.hosts	List of internal ingress hostnames.	Valid hostnames	["gws-int.genesys.com"]
entryPoints.internal.ingress.tls	List of TLS configurations for internal ingress. See Network requirements for an example configuration.	Valid TLS configurations	[]
entryPoints.external.ingress.enabled	Specifies whether external ingress is enabled. Set this value to false if you are deploying Genesys Web Services and Applications in a single namespace.	true or false	true
entryPoints.external.ingress.ingressClassName	Defines which controller implements the Ingress resource. The value is directly propagated to the ingressClassName field of the Kubernetes Ingress object. See Ingress and Ingress class in the Kubernetes documentation for details.	A valid IngressClass	""
entryPoints.external.ingress.annotations	Custom annotations for external ingress.	A valid set of annotations as "name: value"	{}
entryPoints.external.ingress.paths	Paths to external ingress, relative to the hostnames.	Valid paths	["/"]
entryPoints.external.ingress.hosts	List of external ingress hostnames.	Valid hostnames	["gws.genesys.com"]
entryPoints.external.ingress.tls	List of TLS configurations for external ingress. See Network requirements for an example	Valid TLS configurations	[]

Parameter	Description	Valid values	Default
	configuration.		

Configure Kubernetes

Create a Kubernetes secret for your API token from Consul. For more information about this token, see Consul's access control documentation.

```
kubectl create secret generic gws-secrets-green -n gws --from-literal='gws-consul-token='
```

Configure security

To learn more about how security is configured for private edition, be sure to read Permissions and OpenShift security settings.

The security context settings define the privilege and access control settings for pods and containers.

By default, the user and group IDs are set in the **values.yaml** file as `500:500:500`, meaning the **genesys** user.

```
deploymentGlobals:
  securityContext:
    runAsUser: 500
    runAsGroup: 500
    fsGroup: 500
    runAsNonRoot: true
```

For details about these parameters and possible values, see **deploymentGlobals.securityContext.*** in the Parameters table above.

Next steps

- Deploy GWS Ingress
- Provision Genesys Web Services and Applications