Genesys Web Services and Applications Private Edition Guide

11/20/2025

Table of Contents

Contents

Find links to all the topics in this guide.

Related documentation:

-
-

RSS:

- For private edition

Genesys Web Services and Applications is a service available with the Genesys Multicloud CX private edition offering.

Overview

Learn more about Genesys Web Services and Applications and how to get started.

- About Genesys Web Services and Applications
- Architecture
- High availability and disaster recovery

Configure and deploy

Find out how to configure and deploy Genesys Web Services and Applications.

- Before you begin
- Configure GWS Services
- Configure connections with TLS and authentication
- Deploy GWS Services
- Deploy GWS Ingress
- Provision Genesys Web Services and Applications

Upgrade, roll back, or uninstall

Find out how to upgrade, roll back, or uninstall Genesys Web Services and Applications.

- Upgrade, roll back, or uninstall

Observability

Learn how to monitor Genesys Web Services and Applications with metrics and logging.

- [[GWS/Current/GWSPEGuide/Observability|]]
- [[GWS/Current/GWSPEGuide/GWSMetrics|]]
- [[GWS/Current/GWSPEGuide/WorkspaceMetrics|]]

About Genesys Web Services and Applications

Contents

Learn about Genesys Web Services and Applications and how it works in Genesys Multicloud CX private edition.

Related documentation:

-
-
-

RSS:

- For private edition

Genesys Web Services and Applications (GWS) is a set of user interfaces and APIs that provide a web-based client interface to access Genesys services. The Genesys Web Services and Applications package contains a variety of microservices that you can implement in your contact center.

Genesys Web Services and Applications (GWS) is an application cluster composed of several microservices that run together. GWS runs on multiple containers that are categorized as below:

- Agent Setup (see Manage your Contact Center in Agent Setup): Controls your contact center and its resources:
  - The people who run and operate it – the administrators who control the technical ins and outs, the managers who run the day-to-day operations and administrative aspects of a contact center, the supervisors who oversee agents, and the agents who communicate with customers.
  - The systems and programs that make the day-to-day stuff possible – the telephony, the software, the servers, the routing and dialing strategies, and so on.
  - The features and capabilities we use to meet our business needs and requirements – Caller ID capabilities, voicemail, agent transfers and conferencing, and so on.
- Data Services: These services use multiple data sources (third-party databases) that you must maintain to store GWS data.
- Platform Services: These services are used to connect to Genesys servers such as Configuration Server, Stat Server, SIP Server, and Interaction Server.
- UI Services: These services provide user interfaces Workspace Web Edition Private Edition Guide and the underlying services needed to support them, such as the Workspace Service.
- Client Application: This can be Workspace Web Edition (WWE) Agent Workspace, a custom desktop.

A reverse proxy service is used as an ingress controller. This works as an internal application load balancer.

Supported Kubernetes platforms

Genesys Web Services and Applications is supported on the following cloud platforms:

- Azure Kubernetes Service (AKS)
- Google Kubernetes Engine (GKE)

See the Genesys Web Services and Applications Release Notes for information about when support was introduced.

Architecture

Contents

Learn about Genesys Web Services and Applications architecture

Related documentation:

-
-
-

RSS:

- For private edition

Introduction

Genesys Web Services and Applications (GWS) is an application cluster composed of several microservices that run together.

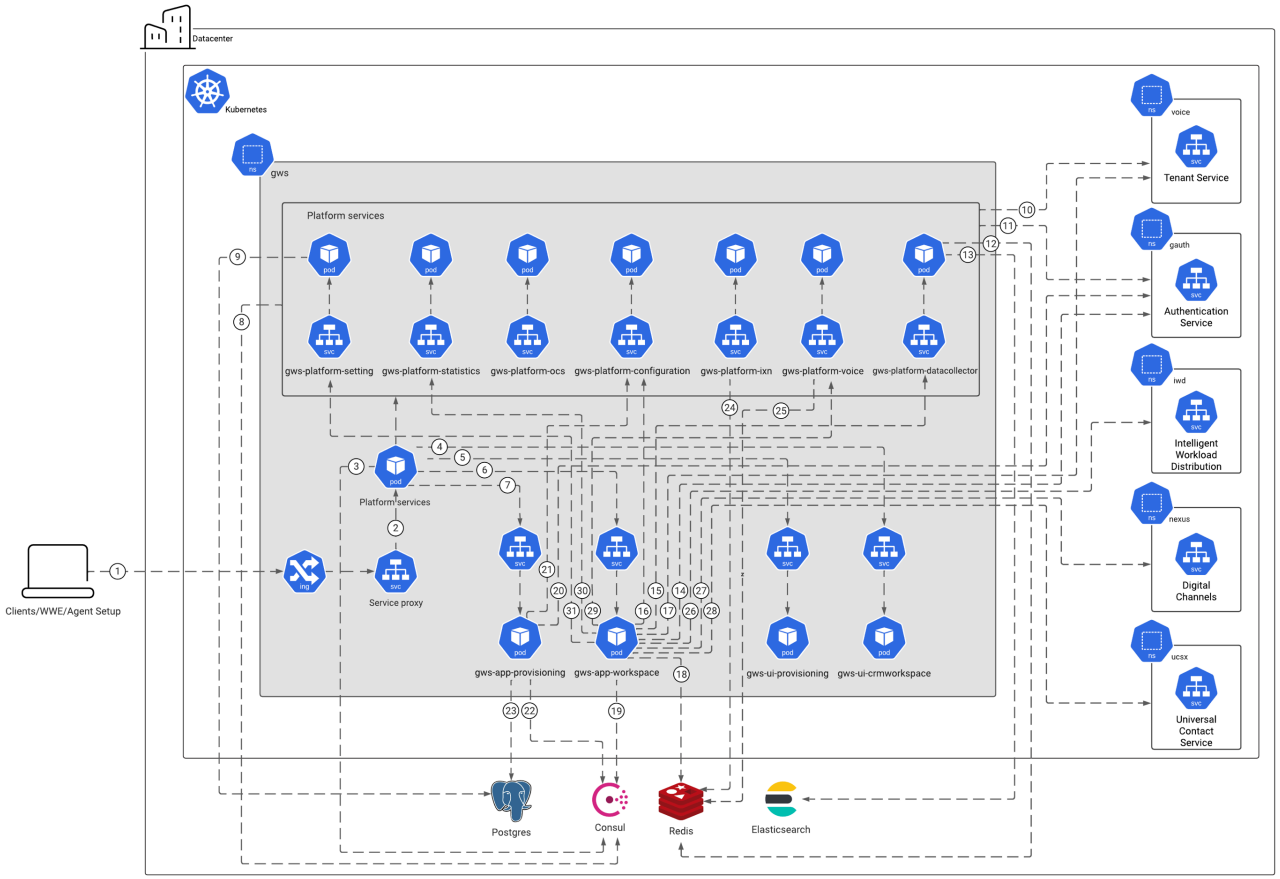
GWS runs on multiple containers as shown in the diagram below.

For information about the overall architecture of Genesys Multicloud CX private edition, see the high-level Architecture page.

See also High availability and disaster recovery for information about high availability/disaster recovery architecture.

Architecture diagram — Connections

The numbers on the connection lines refer to the connection numbers in the table that follows the diagram. The direction of the arrows indicates where the connection is initiated (the source) and where an initiated connection connects to (the destination), from the point of view of Genesys Web Services and Applications as a service in the network.



Connections table

The connection numbers refer to the numbers on the connection lines in the diagram. The **Source**, **Destination**, and **Connection Classification** columns in the table relate to the direction of the arrows in the Connections diagram above: The source is where the connection is initiated, and the destination is where an initiated connection connects to, from the point of view of Genesys Web Services and Applications as a service in the network. *Egress* means the Genesys Web Services and Applications service is the source, and *Ingress* means the Genesys Web Services and Applications service is the destination. *Intra-cluster* means the connection is between services in the cluster.

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
1	Clients/ WWE/Agent Setup	Service proxy	HTTPS	443	Ingress	REST API requests and responses, and UI static content.

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
2	Service proxy	Platform services	HTTP	80	Intra-cluster	REST API requests and responses. You can configure the port number for each platform service with the **gwsServices..service.por** parameters. See .
3	Service proxy	Consul	HTTP	8500	Intra-cluster	Service discovery information.
4	Service proxy	Gplus Adapter for Salesforce			Intra-cluster	
5	Service proxy	Agent Setup		80	Intra-cluster	UI static content. The port number is the value set for . The default is 80.
6	Service proxy	GWS Workspace Service	HTTP	80	Intra-cluster	Workspace REST API requests and responses. The port number is the value set for . The default is 80.
7	Service proxy	GWS Provisioning Service	HTTP	80	Intra-cluster	Provisioning REST API requests and responses. The port number is the value set for . The default is 80.
8	Platform services	Consul	HTTP	8500	Intra-cluster	Service discovery information.
9	GWS Setting Service	PostgreSQL	TCP	5432	Intra-cluster	Tenant information.

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
						The port number depends on your PostgreSQL configuration.
10	Platform services	Tenant Service			Egress	Tenant configuration, voice events, multimedia interactions, and statistical information. The port depends on your .
11	Platform services	Authentication Service	HTTP		Egress	Authentication and authorization information. The port depends on your .
12	GWS Data Collector Service	Redis	TCP	6379	Intra-cluster	Session data cache.
13	GWS Data Collector Service	Elasticsearch	TCP	9200	Intra-cluster	Store monitored statistics for fast access.
14	GWS Workspace Service	Authentication Service	HTTP		Egress	Authentication and authorization information. The port depends on your .
15	GWS Workspace Service	GWS Data Collector Service	HTTP	80	Intra-cluster	Contacts search requests and contacts state information. The port number is the value set for . The default is 80.

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
16	GWS Workspace Service	GWS Configuration Service	HTTP	80	Intra-cluster	Contact center configuration information. The port number is the value set for . The default is 80.
17	GWS Workspace Service	Voicemail	HTTP		Egress	Voice mailbox message count data. The port depends on your .
18	GWS Workspace Service	Redis	TCP	6379	Intra-cluster	Session data cache.
19	GWS Workspace Service	Consul	HTTP	8500	Intra-cluster	Service discovery information.
20	GWS Provisioning Service	Authentication Service	HTTP		Egress	Authentication and authorization information. The port depends on your .
21	GWS Provisioning Service	GWS Configuration Service	HTTP	80	Intra-cluster	Read and write requests for contact center configuration information. The port number is the value set for . The default is 80.
22	GWS Provisioning Service	Consul	HTTP	8500	Intra-cluster	Service discovery information.
23	GWS Provisioning Service	PostgreSQL	TCP	5432	Intra-cluster	Tenant information. The port number depends on

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
						your PostgreSQL configuration.
24	GWS Interaction Service	Redis	TCP	6379	Intra-cluster	Session data cache.
25	GWS Voice Service	Redis	TCP	6379	Intra-cluster	Session data cache.
26	GWS Workspace Service	Intelligent Workload Distribution	HTTP		Egress	IWD API requests and responses. The port depends on your .
27	GWS Workspace Service	Digital Channels	HTTP		Egress	Chat API requests and responses, and CometD polling for chat events. The port depends on your .
28	GWS Workspace Service	Universal Contact Service	HTTP		Egress	UCS API requests and responses, and CometD polling for UCS asynchronous responses. The port depends on your .
29	GWS Workspace Service	GWS Voice Service	HTTP	80	Intra-cluster	Voice API requests and responses, CometD polling for voice related events. The port number is the value set for . The default is 80.
30	GWS Workspace Service	GWS Statistics Service	HTTP	80	Intra-cluster	Statistics API requests and responses. The port

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
						number is the value set for . The default is 80.
31	GWS Workspace Service	GWS Setting Service	HTTP	80	Intra-cluster	Favorite and recent data for agents. The port number is the value set for . The default is 80.

High availability and disaster recovery

Find out how this service provides disaster recovery in the event the service goes down.

Related documentation:

-
-
-

RSS:

- For private edition

Service	High Availability	Disaster Recovery	Where can you host this service?
Genesys Web Services and Applications	N = N (N+1)	Active-spare	Primary or secondary unit

See High Availability information for all services: High availability and disaster recovery

Before you begin

Contents

Find out what to do before deploying Genesys Web Services and Applications.

Related documentation:

-
-
-

RSS:

- For private edition

Download the Helm charts

Genesys Web Services and Applications (GWS) in Genesys Multicloud CX private edition is made up of multiple containers and Helm charts. The pages in this "Configure and deploy" chapter walk you through how to deploy the following Helm charts:

- GWS services (gws-services) - all the GWS components.
- GWS ingress (gws-ingress) - provides internal and external access to GWS services. Internal ingress is used for cross-component communication inside the GWS deployment. It also can be used by other clients located inside the same Kubernetes cluster. External ingress provides access to GWS services to clients located outside the Kubernetes cluster. 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.

GWS also includes a Helm chart for Nginx (wwe-nginx) for Workspace Web Edition - see the Workspace Web Edition Private Edition Guide for details about how to deploy this chart.

See Helm charts and containers for Genesys Web Services and Applications for the Helm chart versions you must download for your release.

For information about downloading Helm charts from JFrog Edge, see Downloading your Genesys Multicloud CX containers.

Third-party prerequisites

Install the prerequisite dependencies listed in the **Third-party services** table before you deploy Genesys Web Services and Applications. See Software requirements for a full list of prerequisites and third-party services required by all Genesys Multicloud CX private edition services.

Third-party services

Name	Version	Purpose	Notes

Elasticsearch	7.x	Used for text searching and indexing. Deployed per service that needs Elasticsearch during runtime.	**GWS requires Elasticsearch 7.17+**. You must configure additional options for Elasticsearch to support the Data Collector Service.
			
			`action.auto_create_index: false`
			
			`thread_pool.write.queue_size: -1`
			
			You can set up Elasticsearch as a shared or dedicated service.

Redis	6.x	Used for caching. Only distributions of Redis that support Redis cluster mode are supported, however, some services may not support cluster mode.	The Redis server must run in cluster mode. You can set up Redis as a shared or dedicated service.
PostgreSQL	11.x	Relational database.	**GWS supports PostgreSQL 12.x.** You can set up PostgreSQL as a shared or dedicated service.
Consul	1.13.x	Service discovery, service mesh, and key/value store.	**GWS supports Consul 1.8**. Consul can be installed either inside or outside the Kubernetes cluster. GWS pods require a Consul agent that is running at the Kubernetes node and GWS only communicates with this Consul agent. You must configure a connection to the Consul server in the local Consul agent.
Ingress controller		HTTPS ingress controller.	
HTTPS certificates - Let's Encrypt		Use with cert-manager to provide free rotating TLS certificates for NGINX Ingress Controller. **Note:** Let's Encrypt is a suite-wide	

Name	Version	Purpose	Notes
		requirement if you choose an Ingress Controller that needs it.	
HTTPS certificates - cert-manager		Use with Let's Encrypt to provide free rotating TLS certificates for NGINX Ingress Controller.	
Load balancer		VPC ingress. For NGINX Ingress Controller, a single regional Google external network LB with a static IP and wildcard DNS entry will pass HTTPS traffic to NGINX Ingress Controller which will terminate SSL traffic and will be setup as part of the platform setup.	
A container image registry and Helm chart repository		Used for downloading Genesys containers and Helm charts into the customer's repository to support a CI/CD pipeline. You can use any Docker OCI compliant registry.	
Command Line Interface		The command line interface tools to log in and work with the Kubernetes clusters.	

Storage requirements

GWS uses PostgreSQL to store tenant information, Redis to cache session data, and Elasticsearch to store monitored statistics for fast access. If you set up any of these services as dedicated services for GWS, they have the following minimal requirements:

PostgreSQL

- CPU: 2
- RAM: 8 GB
- HDD: 50 GB

Redis

- 2 nodes:
  - CPU: 2
  - RAM: 8 GB
  - HDD: 20 GB

Elasticsearch

- 3 "master" nodes:
  - CPU: 2
  - RAM: 8 GB
  - HDD: 20 GB
- 4 "data" nodes
  - CPU: 4
  - RAM: 16 GB
  - HDD: 20 GB

Network requirements

GWS ingress objects support Transport Layer Security (TLS) version 1.2 for a secure connection between Kubernetes cluster ingress and GWS ingress. TLS is disabled by default, but you can configure it for internal and external ingress by overriding the **entryPoints.internal.ingress.tls** and **entryPoints.external.ingress.tls** sections of the GWS ingress Helm chart.

For example:

```
entryPoints:
  external:
    ingress:
    tls:
      - secretName: gws-secret-ext
        hosts:
          - gws.genesys.com
```

In the example above:

- **secretName** is the name of the Kubernetes secret that contains the certificate. The secret is a prerequisite and must be created before you deploy GWS ingress.
- **hosts** is a list of the fully qualified domain names that should use the certificate. The list must be the same as the value configured for the **entryPoints.external.ingress.hosts** parameter.

Cookies

GWS components use cookies for following purposes:

- identify HTTP/HTTPS user sessions
- identify CometD user sessions
- support session stickiness

Genesys dependencies

Genesys Web Services and Applications must be deployed after Genesys Authentication.

For a look at the high-level deployment order, see Order of services deployment in the *Setting up Genesys Multicloud CX Private Edition* guide.

Next steps

- Configure GWS Services
- Deploy GWS Services
- Configure GWS Ingress
- Deploy GWS Ingress

Configure GWS Services

Contents

Learn how to configure GWS Services.

Related documentation:

-
-
-

RSS:

- For private edition

Create API clients

Use the Genesys Authentication operations API to create API clients for GWS services. Refer to the **API clients** table for the **name** and **client_id** values you must use in the API request. Make note of **encrypted_client_secret** in the responses - you need this value to set the related parameter in Override Helm chart values.

API clients

Service	name	client_id	Helm chart parameter
GWS Provisioning Service	gws-app-provisioning	gws-app-provisioning	secrets.gws-app-provisioning-client-secret
GWS Workspace Service	gws-app-workspace	gws-app-workspace	secrets.gws-app-workspace-client-secret
GWS Chat Service	gws-platform-chat	gws-platform-chat	secrets.gws-platform-chat-client-secret
GWS Configuration Service	gws-platform-configuration	gws-platform-configuration	secrets.gws-platform-configuration-client-secret
GWS Data Collector Service	gws-platform-datacollector	gws-platform-datacollector	secrets.gws-platform-datacollector-client-secret
GWS Interaction Service	gws-platform-ixn	gws-platform-ixn	secrets.gws-platform-ixn-client-secret
GWS OCS Service	gws-platform-ocs	gws-platform-ocs	secrets.gws-platform-ocs-client-secret
GWS Setting Service	gws-platform-setting	gws-platform-setting	secrets.gws-platform-setting-client-secret
GWS Statistics Service	gws-platform-statistics	gws-platform-statistics	secrets.gws-platform-statistics-client-secret

Service	name	client_id	Helm chart parameter
GWS UCS Service	gws-platform-ucs	gws-platform-ucs	secrets.gws-platform-ucs-client-secret
GWS Voice Service	gws-platform-voice	gws-platform-voice	secrets.gws-platform-voice-client-secret

Configure a secret to access JFrog

If you haven't done so already, create a secret for accessing the JFrog registry:

```
kubectl create secret docker-registry  --docker-server= --docker-username= --docker-password=
--docker-email=
```

Now map the secret to the default service account:

```
kubectl secrets link default  --for=pull
```

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 services.

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

Global 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 labels as "name: value"	{}
imageGlobals.registry	The Docker registry from which Kubernetes pulls images.	A valid registry URL	""
imageGlobals.pullPolicy	Specifies when Kubernetes pulls images from the registry on start up.	IfNotPresent or Always	"Always"
imageGlobals.imagePullSecrets	The secret Kubernetes uses to get credentials to pull images from the registry.	A valid secret	[]
deploymentGlobals.depl	A suffix for the names of	Any lowercase	"live"

Parameter	Description	Valid values	Default
oymentTag	Kubernetes objects created by the Helm chart.	alphanumeric value up to 8 characters long.	
deploymentGlobals.strategy	The strategy GWS uses to upgrade its containers.	RollingUpdate or Recreate	"RollingUpdate"
deploymentGlobals.location	Location of the deployment.	A valid location	"/USW1"
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 entry point of the container process.	A valid user ID or null	500
deploymentGlobals.securityContext.runAsGroup	The group ID to run the entry point of 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
serviceGlobals.type	The service type for all services.	ClusterIP, NodePort, or LoadBalancer	"ClusterIP"
serviceGlobals.labels	Custom labels to be added for all services.	A valid set of labels as "name: value"	{}
serviceGlobals.annotations	Custom annotations to be added for all services.	A valid set of annotations as "name: value"	{}
nodeSelector	The labels Kubernetes uses to assign pods to nodes.	Valid nodeSelector settings. See the Kubernetes documentation for details.	{}
tolerations	The tolerations Kubernetes uses for advanced pod scheduling.	Valid tolerations settings. See the Kubernetes documentation for details.	{}
dnsConfig	The DNS configuration for pods.	Valid DNS configuration settings. See the Kubernetes documentation for details.	{}
topologySpreadConstraints	In Kubernetes, topology spread constraints are used to control how Pods are spread across the cluster among failure-domains such as regions, zones, nodes,	Valid topology spread constraints settings. See the Kubernetes documentation for details.	{}

Parameter	Description	Valid values	Default
	and other user-defined topology domains. This helps to achieve high-availability as well as efficient resource utilization.		

GWS Provisioning Service parameters

Parameter	Description	Valid values	Default
gwsServices.gwsAppProvisioning.name	The name of the container deployment.	String	"gws-app-provisioning"
gwsServices.gwsAppProvisioning.appType	The type of application in this container.	nodejs, java, or frontend	"nodejs"
gwsServices.gwsAppProvisioning.image.registry	The Docker registry from which Kubernetes pulls images. If set, this parameter overrides imageGlobals.registry.	A valid registry URL	""
gwsServices.gwsAppProvisioning.livenessProbe.enable	Specifies whether to do a Kubernetes liveness probe to test if the container is running.	true or false	true
gwsServices.gwsAppProvisioning.livenessProbe.failureThreshold	Minimum consecutive failures for the probe to be considered failed after having succeeded.	1 or greater	3
gwsServices.gwsAppProvisioning.livenessProbe.successThreshold	Minimum consecutive successes for the probe to be considered successful after having failed. The default is 1, which is required for liveness and startup.	1 or greater	1
gwsServices.gwsAppProvisioning.livenessProbe.initialDelaySeconds	Number of seconds after the container has started before liveness probes are initiated.	Number	120
gwsServices.gwsAppProvisioning.livenessProbe.periodSeconds	How often (in seconds) to perform the probe.	1 or greater	30
gwsServices.gwsAppProvisioning.livenessProbe.timeoutSeconds	Number of seconds after which the probe times out.	1 or greater	10
gwsServices.gwsAppProvisioning.clientId	The ID of an encrypted client secret generated by Genesys Authentication for this component.	A valid ID	"gws-app-provisioning"

Parameter	Description	Valid values	Default
gwsServices.gwsAppProvisioning.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	""
gwsServices.gwsAppProvisioning.deployment.replicaCount	The number of pod replicas in this container deployment.	A number greater than 0	2
gwsServices.gwsAppProvisioning.resources.limits.cpu	The maximum amount of CPU Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of Kubernetes CPU	4
gwsServices.gwsAppProvisioning.resources.limits.memory	The maximum amount of memory Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of bytes	"5Gi"
gwsServices.gwsAppProvisioning.resources.requests.cpu	The guaranteed amount of CPU Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of Kubernetes CPU	1
gwsServices.gwsAppProvisioning.resources.requests.memory	The guaranteed amount of memory Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of bytes	"5Gi"
gwsServices.gwsAppProvisioning.postgres.address	The fully qualified domain name or IP of the PostgreSQL server for gws-app-provisioning.	A valid address	""
gwsServices.gwsAppProvisioning.postgres.port	The port of the PostgreSQL server for gws-app-provisioning.	A valid port	""
gwsServices.gwsAppProvisioning.postgres.db	The name of the PostgreSQL database for gws-app-provisioning.	A valid database name	""

Parameter	Description	Valid values	Default
gwsServices.gwsAppProvisioning.postgres.enableTls	Enable or disable a TLS connection to PostgreSQL for gws-app-provisioning. If true, you must configure the **secretsTls.postgresprovisioning.** parameters. See Configure connections with TLS and authentication for details.	true or false	false
gwsServices.gwsAppProvisioning.context.ports.server	The port for this container.	A valid port	48060
gwsServices.gwsAppProvisioning.context.ports.management	The management port for this container.	A valid port	48061
gwsServices.gwsAppProvisioning.context.loggingLevel	Specifies the logging level for this container.	ERROR, WARN, INFO, DEBUG, or TRACE	""
gwsServices.gwsAppProvisioning.context.env.GWS_SERVICE_AUTH_URL	**DEPRECATED** - Use gauth.authUrl instead. The internal service URI of the Genesys Authentication service. For example: http://gauth-auth.gauth.svc.cluster.local.:80	A valid URL	""
gwsServices.gwsAppProvisioning.context.env.GWS_SERVICE_CONF_URL	The internal service URI of the configuration service (part of GWS). For example: http://gws-service-proxy.gws.svc.cluster.local:80	A valid URL	""
gwsServices.gwsAppProvisioning.context.env.GWS_SERVICE_ENV_URL	**DEPRECATED** - Use gauth.envUrl instead. The internal service URI of the environment service (part of Genesys Authentication). For example: http://gauth-environment.gauth.svc.cluster.local.:80	A valid URL	""
gwsServices.gwsAppProvisioning.context.env.GWS_SERVICE_VOICEMAIL_URL	The URL of the voicemail server.	A valid URL	""
	The port for this server.	A valid port	80

Parameter	Description	Valid values	Default
gwsServices.gwsAppProvisioning.service.ports.server			
gwsServices.gwsAppProvisioning.service.ports.management	The management port for this server.	A valid port	81
gwsServices.gwsAppProvisioning.labels	Custom labels to be added for the container.	A valid set of labels as "name: value"	{}
gwsServices.AppProvisioning.annotations	Custom annotations to be added for the container.	A valid set of annotations as "name: value"	{}

GWS Workspace Service parameters

Parameter	Description	Valid values	Default
gwsServices.gwsAppWorkspace.name	The name of the container deployment.	String	"gws-app-workspace"
gwsServices.gwsAppWorkspace.appType	The type of application in this container.	nodejs, java, or frontend	"nodejs"
gwsServices.gwsAppWorkspace.image.registry	The Docker registry from which Kubernetes pulls images. If set, this parameter overrides imageGlobals.registry.	A valid registry URL	""
gwsServices.gwsAppWorkspace.livenessProbe.enable	Specifies whether to do a Kubernetes liveness probe to test if the container is running.	true or false	true
gwsServices.gwsAppWorkspace.livenessProbe.failureThreshold	Minimum consecutive failures for the probe to be considered failed after having succeeded.	1 or greater	3
gwsServices.gwsAppWorkspace.livenessProbe.successThreshold	Minimum consecutive successes for the probe to be considered successful after having failed. The default is 1, which is required for liveness and startup.	1 or greater	1
gwsServices.gwsAppWorkspace.livenessProbe.initialDelaySeconds	Number of seconds after the container has started before liveness probes are initiated.	Number	120
gwsServices.gwsAppWorkspace.livenessProbe.periodSeconds	How often (in seconds) to perform the probe.	1 or greater	30
gwsServices.gwsAppWorkspace.livenessProbe.ti	Number of seconds after which the probe times	1 or greater	10

Parameter	Description	Valid values	Default
meoutSeconds	out.		
gwsServices.gwsAppWorkspace.clientId	The ID of an encrypted client secret generated by Genesys Authentication for this component.	A valid ID	"gws-app-workspace"
gwsServices.gwsAppWorkspace.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	""
gwsServices.gwsAppWorkspace.deployment.replicaCount	The number of pod replicas in this container deployment.	A number greater than 0	2
gwsServices.gwsAppWorkspace.resources.limits.cpu	The maximum amount of CPU Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of Kubernetes CPU	4
gwsServices.gwsAppWorkspace.resources.limits.memory	The maximum amount of memory Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of bytes	"5Gi"
gwsServices.gwsAppWorkspace.resources.requests.cpu	The guaranteed amount of CPU Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of Kubernetes CPU	1
gwsServices.gwsAppWorkspace.resources.requests.memory	The guaranteed amount of memory Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of bytes	"5Gi"
gwsServices.gwsAppWorkspace.context.ports.server	The port for this container.	A valid port	48050
gwsServices.gwsAppWorkspace.context.ports.ma	The management port for this container.	A valid port	48051

Parameter	Description	Valid values	Default
nagement			
gwsServices.gwsAppWorkspace.context.loggingLevel	Specifies the logging level for this container.	ERROR, WARN, INFO, DEBUG, or TRACE	""
gwsServices.gwsAppWorkspace.context.env.GWS_WORKSPACE_CONSUL_CACHE_TTL	The length of time, in milliseconds, that the GWS Workspace Service keeps service locations in cache locally.	Number	60000
gwsServices.gwsAppWorkspace.context.env.GWS_WORKSPACE_ENABLE_CHANGE_PASSWORD	Specifies whether the GWS Workspace Service allows the change password functionality.	true or false	true
gwsServices.gwsAppWorkspace.context.env.GWS_WORKSPACE_MEMORY_CACHE_ENABLED	Specifies whether the GWS Workspace Service should cache configuration data (such as agent groups) in memory.	true or false	true
gwsServices.gws-app-workspace.context.env.GWS_SECURE_COOKIE	Specifies whether the Workspace Service returns cookies with the Secure flag. Set this value to true if you configure GWS ingress to use TLS (see Network requirements for configuration details).	true or false	false
gwsServices.gwsAppWorkspace.context.env.GWS_SERVICE_AUTH_URL	**DEPRECATED** - Use gauth.authUrl instead. The internal service URI of the Genesys Authentication service. For example: http://gauth-auth.gauth.svc.cluster.local.:80	A valid URL	""
gwsServices.gwsAppWorkspace.context.env.GWS_SERVICE_ENV_URL	**DEPRECATED** - Use gauth.envUrl instead. The internal service URI of the environment service (part of Genesys Authentication). For example: http://gauth-environment.gauth.svc.cluster.local.:80	A valid URL	""
gwsServices.gwsAppWorkspace.service.ports.server	The port for this server.	A valid port	80
gwsServices.gwsAppWorkspace.service.ports.ma	The management port for this server.	A valid port	81

Parameter	Description	Valid values	Default
nagement			
gwsServices.gwsAppWorkspace.labels	Custom labels to be added for the container.	A valid set of labels as "name: value"	{}
gwsServices.gwsAppWorkspace.annotations	Custom annotations to be added for the container.	A valid set of annotations as "name: value"	{}

GWS Chat Service parameters

Parameter	Description	Valid values	Default
gwsServices.gwsPlatformChat.name	The name of the container deployment.	String	"gws-platform-chat"
gwsServices.gwsPlatformChat.enabled	Enables the component deployment.	true or false	false
gwsServices.gwsPlatformChat.appType	The type of application in this container.	nodejs, java, or frontend	"java"
gwsServices.gwsPlatformChat.image.registry	The Docker registry from which Kubernetes pulls images. If set, this parameter overrides imageGlobals.registry.	A valid registry URL	""
gwsServices.gwsPlatformChat.clientId	The ID of an encrypted client secret generated by Genesys Authentication for this component.	A valid ID	"gws-platform-chat"
gwsServices.gwsPlatformChat.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	""
gwsServices.gwsPlatformChat.deployment.replicaCount	The number of pod replicas in this container deployment.	A number greater than 0	2
gwsServices.gwsPlatformChat.resources.limits.cpu	The maximum amount of CPU Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of Kubernetes CPU	4
gwsServices.gwsPlatformChat.resources.limits.memory	The maximum amount of memory Kubernetes allocates for the container. See the	Units of bytes	"4Gi"

Parameter	Description	Valid values	Default
	Kubernetes documentation for details.		
gwsServices.gwsPlatformChat.resources.requests.cpu	The guaranteed amount of CPU Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of Kubernetes CPU	1
gwsServices.gwsPlatformChat.resources.requests.memory	The guaranteed amount of memory Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of bytes	"4Gi"
gwsServices.gwsPlatformChat.consul.enabled	Enables Consul registration for the component.	true or false	true
gwsServices.gwsPlatformChat.context.ports.server	The port for this container.	A valid port	48150
gwsServices.gwsPlatformChat.context.ports.management	The management port for this container.	A valid port	48151
gwsServices.gwsPlatformChat.context.loggingLevel	Specifies the logging level for this container.	ERROR, WARN, INFO, DEBUG, or TRACE	""
gwsServices.gwsPlatformChat.context.env	Environment variables for this container.		{}
gwsServices.gwsPlatformChat.livenessProbe.enable	Specifies whether to do a Kubernetes liveness probe to test if the container is running.	true or false	true
gwsServices.gwsPlatformChat.livenessProbe.failureThreshold	Minimum consecutive failures for the probe to be considered failed after having succeeded.	1 or greater	3
gwsServices.gwsPlatformChat.livenessProbe.successThreshold	Minimum consecutive successes for the probe to be considered successful after having failed. The default is 1, which is required for liveness and startup.	1 or greater	1
gwsServices.gwsPlatformChat.livenessProbe.initialDelaySeconds	Number of seconds after the container has started before liveness probes are initiated.	Number	120

Parameter	Description	Valid values	Default
gwsServices.gwsPlatformChat.livenessProbe.periodSeconds	How often (in seconds) to perform the probe.	1 or greater	30
gwsServices.gwsPlatformChat.livenessProbe.timeoutSeconds	Number of seconds after which the probe times out.	1 or greater	10
gwsServices.gwsPlatformChat.service.ports.server	The port for this server.	A valid port	80
gwsServices.gwsPlatformChat.service.ports.management	The management port for this server.	A valid port	81
gwsServices.gwsPlatformChat.labels	Custom labels to be added for the container.	A valid set of labels as "name: value"	{}
gwsServices.gwsPlatformChat.annotations	Custom annotations to be added for the container.	A valid set of annotations as "name: value"	{}

GWS Configuration Service parameters

Parameter	Description	Valid values	Default
gwsServices.gwsPlatformConfiguration.name	The name of the container deployment.	String	"gws-platform-configuration"
gwsServices.gwsPlatformConfiguration.appType	The type of application in this container.	nodejs, java, or frontend	"java"
gwsServices.gwsPlatformConfiguration.image.registry	The Docker registry from which Kubernetes pulls images. If set, this parameter overrides imageGlobals.registry.	A valid registry URL	""
gwsServices.gwsPlatformConfiguration.livenessProbe.enable	Specifies whether to do a Kubernetes liveness probe to test if the container is running.	true or false	true
gwsServices.gwsPlatformConfiguration.livenessProbe.failureThreshold	Minimum consecutive failures for the probe to be considered failed after having succeeded.	1 or greater	3
gwsServices.gwsPlatformConfiguration.livenessProbe.successThreshold	Minimum consecutive successes for the probe to be considered successful after having failed. The default is 1, which is required for liveness and startup.	1 or greater	1
gwsServices.gwsPlatformConfiguration.livenessProbe.initialDelaySecon	Number of seconds after the container has started before liveness	Number	120

Parameter	Description	Valid values	Default
ds	probes are initiated.		
gwsServices.gwsPlatformConfiguration.livenessProbe.periodSeconds	How often (in seconds) to perform the probe.	1 or greater	30
gwsServices.gwsPlatformConfiguration.livenessProbe.timeoutSeconds	Number of seconds after which the probe times out.	1 or greater	10
gwsServices.gwsPlatformConfiguration.clientId	The ID of an encrypted client secret generated by Genesys Authentication for this component.	A valid ID	"gws-platform-configuration"
gwsServices.gwsPlatformConfiguration.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	""
gwsServices.gwsPlatformConfiguration.deployment.replicaCount	The number of pod replicas in this container deployment.	A number greater than 0	2
gwsServices.gwsPlatformConfiguration.resources.limits.cpu	The maximum amount of CPU Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of Kubernetes CPU	4
gwsServices.gwsPlatformConfiguration.resources.limits.memory	The maximum amount of memory Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of bytes	"4Gi"
gwsServices.gwsPlatformConfiguration.resources.requests.cpu	The guaranteed amount of CPU Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of Kubernetes CPU	1
gwsServices.gwsPlatformConfiguration.resources.requests.memory	The guaranteed amount of memory Kubernetes allocates for the container. See the Kubernetes documentation for	Units of bytes	"4Gi"

Parameter	Description	Valid values	Default
	details.		
gwsServices.gwsPlatformConfiguration.context.ports.server	The port for this container.	A valid port	48030
gwsServices.gwsPlatformConfiguration.context.ports.management	The management port for this container.	A valid port	48031
gwsServices.gwsPlatformConfiguration.context.loggingLevel	Specifies the logging level for this container.	ERROR, WARN, INFO, DEBUG, or TRACE	""
gwsServices.gwsPlatformConfiguration.service.ports.server	The port for this server.	A valid port	80
gwsServices.gwsPlatformConfiguration.service.ports.management	The management port for this server.	A valid port	81
gwsServices.gwsPlatformConfiguration.labels	Custom labels to be added for the container.	A valid set of labels as "name: value"	{}
gwsServices.gwsPlatformConfiguration.annotations	Custom annotations to be added for the container.	A valid set of annotations as "name: value"	{}
gwsServices.gws-platform-configuration.context.env.GWS_CS_CLUSTER_SUPPORT	Specifies Configuration Server cluster support.	true or false	false
gwsServices.gws-platform-configuration.context.env.GWS_CONFIGURATION_common_discovery_tenants	Enable or disable Tenant discovery from Consul.	true or false	false
gwsServices.gws-platform-configuration.context.env.GWS_CONFIGURATION_common_discovery_ixn_intercept	Enable or disable multi-region support. To enable multi-region support, you must also set gwsServices.gws-platform-configuration.context.env.GWS_CONFIGURATION_common_discovery_tenants to true.	true or false	true

GWS Data Collector Service parameters

Parameter	Description	Valid values	Default
gwsServices.gwsPlatformDatacollector.name	The name of the container deployment.	String	"gws-platform-datacollector"
gwsServices.gwsPlatformDatacollector.appType	The type of application in this container.	nodejs, java, or frontend	"java"

Parameter	Description	Valid values	Default
gwsServices.gwsPlatformDatacollector.image.registry	The Docker registry from which Kubernetes pulls images. If set, this parameter overrides imageGlobals.registry.	A valid registry URL	""
gwsServices.gwsPlatformDatacollector.livenessProbe.enable	Specifies whether to do a Kubernetes liveness probe to test if the container is running.	true or false	true
gwsServices.gwsPlatformDatacollector.livenessProbe.failureThreshold	Minimum consecutive failures for the probe to be considered failed after having succeeded.	1 or greater	3
gwsServices.gwsPlatformDatacollector.livenessProbe.successThreshold	Minimum consecutive successes for the probe to be considered successful after having failed. The default is 1, which is required for liveness and startup.	1 or greater	1
gwsServices.gwsPlatformDatacollector.livenessProbe.initialDelaySeconds	Number of seconds after the container has started before liveness probes are initiated.	Number	120
gwsServices.gwsPlatformDatacollector.livenessProbe.periodSeconds	How often (in seconds) to perform the probe.	1 or greater	30
gwsServices.gwsPlatformDatacollector.livenessProbe.timeoutSeconds	Number of seconds after which the probe times out.	1 or greater	10
gwsServices.gwsPlatformDatacollector.clientId	The ID of an encrypted client secret generated by Genesys Authentication for this component.	A valid ID	"gws-platform-datacollector"
gwsServices.gwsPlatformDatacollector.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	""
gwsServices.gwsPlatformDatacollector.deployment.replicaCount	The number of pod replicas in this container deployment.	A number greater than 0	2
gwsServices.gwsPlatformDatacollector.resources.limits.cpu	The maximum amount of CPU Kubernetes allocates for the container. See the	Units of Kubernetes CPU	4

Parameter	Description	Valid values	Default
	Kubernetes documentation for details.		
gwsServices.gwsPlatformDatacollector.resources.limits.memory	The maximum amount of memory Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of bytes	"5Gi"
gwsServices.gwsPlatformDatacollector.resources.requests.cpu	The guaranteed amount of CPU Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of Kubernetes CPU	1
gwsServices.gwsPlatformDatacollector.resources.requests.memory	The guaranteed amount of memory Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of bytes	"4Gi"
gwsServices.gwsPlatformDatacollector.context.ports.server	The port for this container.	A valid port	48180
gwsServices.gwsPlatformDatacollector.context.ports.management	The management port for this container.	A valid port	48181
gwsServices.gwsPlatformDatacollector.context.loggingLevel	Specifies the logging level for this container.	ERROR, WARN, INFO, DEBUG, or TRACE	""
gwsServices.gwsPlatformDatacollector.context.env.gws_datacollector_services_datacollector_distribution_enabled	Enables task distribution for the data collector.	true or false	true
gwsServices.gwsPlatformDatacollector.context.env.GWS_DATACOLLECTOR_SERVICES_DATACOLLECTOR_REINDEX_ENABLED	Enables background service for reindexing data.	true or false	true
gwsServices.gwsPlatformDatacollector.context.env.gws_datacollector_services_datacollector_reindex_onStart	Specifies whether to perform a reindex on start.	true or false	true
gwsServices.gwsPlatformDatacollector.context.	The period in minutes between scheduled	A time in minutes	30

Parameter	Description	Valid values	Default
env.GWS_DATACOLLECTOR_SERVICES_DATACOLLECTOR_REINDEX_PERIOD	reindex attempts.		
gwsServices.gwsPlatformDatacollector.context.env.GWS_DATACOLLECTOR_SERVICES_DATACOLLECTOR_STATISTICS_ENABLED	Enables statistics monitoring.	true or false	true
gwsServices.gwsPlatformDatacollector.context.env.GWS_DATACOLLECTOR_services_datacollector_statistics_period	Period in minutes between statistics checks.	A time in minutes	5
gwsServices.gwsPlatformDatacollector.service.ports.server	The port for this server.	A valid port	80
gwsServices.gwsPlatformDatacollector.service.ports.management	The management port for this server.	A valid port	81
gwsServices.gwsPlatformDatacollector.labels	Custom labels to be added for the container.	A valid set of labels as "name: value"	{}
gwsServices.gwsPlatformDatacollector.annotations	Custom annotations to be added for the container.	A valid set of annotations as "name: value"	{}

GWS Interaction Service parameters

Parameter	Description	Valid values	Default
gwsServices.gwsPlatformIxn.name	The name of the container deployment.	String	"gws-platform-ixn"
gwsServices.gwsPlatformIxn.appType	The type of application in this container.	nodejs, java, or frontend	"java"
gwsServices.gwsPlatformIxn.image.registry	The Docker registry from which Kubernetes pulls images. If set, this parameter overrides imageGlobals.registry.	A valid registry URL	""
gwsServices.gwsPlatformIxn.livenessProbe.enable	Specifies whether to do a Kubernetes liveness probe to test if the container is running.	true or false	true
gwsServices.gwsPlatformIxn.livenessProbe.failureThreshold	Minimum consecutive failures for the probe to be considered failed after having succeeded.	1 or greater	3
gwsServices.gwsPlatfor	Minimum consecutive	1 or greater	1

Parameter	Description	Valid values	Default
mIxn.livenessProbe.successThreshold	successes for the probe to be considered successful after having failed. The default is 1, which is required for liveness and startup.		
gwsServices.gwsPlatformIxn.livenessProbe.initialDelaySeconds	Number of seconds after the container has started before liveness probes are initiated.	Number	120
gwsServices.gwsPlatformIxn.livenessProbe.periodSeconds	How often (in seconds) to perform the probe.	1 or greater	30
gwsServices.gwsPlatformIxn.livenessProbe.timeoutSeconds	Number of seconds after which the probe times out.	1 or greater	10
gwsServices.gwsPlatformIxn.clientId	The ID of an encrypted client secret generated by Genesys Authentication for this component.	A valid ID	"gws-platform-ixn"
gwsServices.gwsPlatformIxn.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	""
gwsServices.gwsPlatformIxn.deployment.replicaCount	The number of pod replicas in this container deployment.	A number greater than 0	2
gwsServices.gwsPlatformIxn.resources.limits.cpu	The maximum amount of CPU Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of Kubernetes CPU	4
gwsServices.gwsPlatformIxn.resources.limits.memory	The maximum amount of memory Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of bytes	"4Gi"
gwsServices.gwsPlatformIxn.resources.requests.cpu	The guaranteed amount of CPU Kubernetes allocates for the container. See the	Units of Kubernetes CPU	1

Parameter	Description	Valid values	Default
	Kubernetes documentation for details.		
gwsServices.gwsPlatformIxn.resources.requests.memory	The guaranteed amount of memory Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of bytes	"4Gi"
gwsServices.gwsPlatformIxn.context.ports.server	The port for this container.	A valid port	48170
gwsServices.gwsPlatformIxn.context.ports.management	The management port for this container.	A valid port	48171
gwsServices.gwsPlatformIxn.context.loggingLevel	Specifies the logging level for this container.	ERROR, WARN, INFO, DEBUG, or TRACE	""
gwsServices.gwsPlatformIxn.context.env	Environment variables for this container.		{}
gwsServices.gwsPlatformIxn.service.ports.server	The port for this server.	A valid port	80
gwsServices.gwsPlatformIxn.service.ports.management	The management port for this server.	A valid port	81
gwsServices.gwsPlatformIxn.labels	Custom labels to be added for the container.	A valid set of labels as "name: value"	{}
gwsServices.gwsPlatformIxn.annotations	Custom annotations to be added for the container.	A valid set of annotations as "name: value"	{}

GWS OCS Service parameters

Parameter	Description	Valid values	Default
gwsServices.gwsPlatformOcs.name	The name of the container deployment.	String	"gws-platform-ocs"
gwsServices.gwsPlatformOcs.appType	The type of application in this container.	nodejs, java, or frontend	"java"
gwsServices.gwsPlatformOcs.image.registry	The Docker registry from which Kubernetes pulls images. If set, this parameter overrides imageGlobals.registry.	A valid registry URL	""
gwsServices.gwsPlatformOcs.livenessProbe.enable	Specifies whether to do a Kubernetes liveness probe to test if the	true or false	true

Parameter	Description	Valid values	Default
	container is running.		
gwsServices.gwsPlatformOcs.livenessProbe.failureThreshold	Minimum consecutive failures for the probe to be considered failed after having succeeded.	1 or greater	3
gwsServices.gwsPlatformOcs.livenessProbe.successThreshold	Minimum consecutive successes for the probe to be considered successful after having failed. The default is 1, which is required for liveness and startup.	1 or greater	1
gwsServices.gwsPlatformOcs.livenessProbe.initialDelaySeconds	Number of seconds after the container has started before liveness probes are initiated.	Number	120
gwsServices.gwsPlatformOcs.livenessProbe.periodSeconds	How often (in seconds) to perform the probe.	1 or greater	30
gwsServices.gwsPlatformOcs.livenessProbe.timeoutSeconds	Number of seconds after which the probe times out.	1 or greater	10
gwsServices.gwsPlatformOcs.clientId	The ID of an encrypted client secret generated by Genesys Authentication for this component.	A valid ID	"gws-platform-ocs"
gwsServices.gwsPlatformOcs.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	""
gwsServices.gwsPlatformOcs.deployment.replicaCount	The number of pod replicas in this container deployment.	A number greater than 0	2
gwsServices.gwsPlatformOcs.resources.limits.cpu	The maximum amount of CPU Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of Kubernetes CPU	4
gwsServices.gwsPlatformOcs.resources.limits.memory	The maximum amount of memory Kubernetes allocates for the container. See the Kubernetes	Units of bytes	"4Gi"

Parameter	Description	Valid values	Default
	documentation for details.		
gwsServices.gwsPlatformOcs.resources.requests.cpu	The guaranteed amount of CPU Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of Kubernetes CPU	1
gwsServices.gwsPlatformOcs.resources.requests.memory	The guaranteed amount of memory Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of bytes	"4Gi"
gwsServices.gwsPlatformOcs.context.ports.server	The port for this container.	A valid port	48090
gwsServices.gwsPlatformOcs.context.ports.management	The management port for this container.	A valid port	48091
gwsServices.gwsPlatformOcs.context.loggingLevel	Specifies the logging level for this container.	ERROR, WARN, INFO, DEBUG, or TRACE	""
gwsServices.gwsPlatformOcs.context.env.GWS_OCS_timeouts_requestTimeoutMs	Specifies the timeout, in milliseconds, for the GWS OCS Service to connect to OCS.	A time in milliseconds	5000
gwsServices.gwsPlatformOcs.service.ports.server	The port for this server.	A valid port	80
gwsServices.gwsPlatformOcs.service.ports.management	The management port for this server.	A valid port	81
gwsServices.gwsPlatformOcs.labels	Custom labels to be added for the container.	A valid set of labels as "name: value"	{}
gwsServices.gwsPlatformOcs.annotations	Custom annotations to be added for the container.	A valid set of annotations as "name: value"	{}

GWS Setting Service parameters

Parameter	Description	Valid values	Default
gwsServices.gwsPlatformSetting.name	The name of the container deployment.	String	"gws-platform-setting"
gwsServices.gwsPlatformSetting.appType	The type of application in this container.	nodejs, java, or frontend	"java"

Parameter	Description	Valid values	Default
gwsServices.gwsPlatformSetting.image.registry	The Docker registry from which Kubernetes pulls images. If set, this parameter overrides imageGlobals.registry.	A valid registry URL	""
gwsServices.gwsPlatformSetting.livenessProbe.enable	Specifies whether to do a Kubernetes liveness probe to test if the container is running.	true or false	true
gwsServices.gwsPlatformSetting.livenessProbe.failureThreshold	Minimum consecutive failures for the probe to be considered failed after having succeeded.	1 or greater	3
gwsServices.gwsPlatformSetting.livenessProbe.successThreshold	Minimum consecutive successes for the probe to be considered successful after having failed. The default is 1, which is required for liveness and startup.	1 or greater	1
gwsServices.gwsPlatformSetting.livenessProbe.initialDelaySeconds	Number of seconds after the container has started before liveness probes are initiated.	Number	120
gwsServices.gwsPlatformSetting.livenessProbe.periodSeconds	How often (in seconds) to perform the probe.	1 or greater	30
gwsServices.gwsPlatformSetting.livenessProbe.timeoutSeconds	Number of seconds after which the probe times out.	1 or greater	10
gwsServices.gwsPlatformSetting.clientId	The ID of an encrypted client secret generated by Genesys Authentication for this component.	A valid ID	"gws-platform-setting"
gwsServices.gwsPlatformSetting.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	""
gwsServices.gwsPlatformSetting.deployment.replicaCount	The number of pod replicas in this container deployment.	A number greater than 0	2
gwsServices.gwsPlatformSetting.resources.limits.cpu	The maximum amount of CPU Kubernetes allocates for the container. See the	Units of Kubernetes CPU	4

Parameter	Description	Valid values	Default
	Kubernetes documentation for details.		
gwsServices.gwsPlatformSetting.resources.limits.memory	The maximum amount of memory Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of bytes	"4Gi"
gwsServices.gwsPlatformSetting.resources.requests.cpu	The guaranteed amount of CPU Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of Kubernetes CPU	1
gwsServices.gwsPlatformSetting.resources.requests.memory	The guaranteed amount of memory Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of bytes	"4Gi"
gwsServices.gwsPlatformSetting.context.ports.server	The port for this container.	A valid port	48140
gwsServices.gwsPlatformSetting.context.ports.management	The management port for this container.	A valid port	48141
gwsServices.gwsPlatformSetting.context.loggingLevel	Specifies the logging level for this container.	ERROR, WARN, INFO, DEBUG, or TRACE	""
gwsServices.gwsPlatformSetting.context.env.GWS_SETTING_DB_INIT_DB	Enables database initialization in PostgreSQL. Set this parameter to true in regions with the primary PostgreSQL server and false in regions with PostgreSQL replicas.	true or false	true
gwsServices.gwsPlatformSetting.service.ports.server	The port for this server.	A valid port	80
gwsServices.gwsPlatformSetting.service.ports.management	The management port for this server.	A valid port	81
gwsServices.gwsPlatformSetting.labels	Custom labels to be added for the container.	A valid set of labels as "name: value"	{}
gwsServices.gwsPlatfor	Custom annotations to	A valid set of	{}

Parameter	Description	Valid values	Default
mSetting.annotations	be added for the container.	annotations as "name: value"	

GWS Statistics Service parameters

Parameter	Description	Valid values	Default
gwsServices.gwsPlatformStatistics.name	The name of the container deployment.	String	"gws-platform-statistics"
gwsServices.gwsPlatformStatistics.appType	The type of application in this container.	nodejs, java, or frontend	"java"
gwsServices.gwsPlatformStatistics.image.registry	The Docker registry from which Kubernetes pulls images. If set, this parameter overrides imageGlobals.registry.	A valid registry URL	""
gwsServices.gwsPlatformStatistics.livenessProbe.enable	Specifies whether to do a Kubernetes liveness probe to test if the container is running.	true or false	true
gwsServices.gwsPlatformStatistics.livenessProbe.failureThreshold	Minimum consecutive failures for the probe to be considered failed after having succeeded.	1 or greater	3
gwsServices.gwsPlatformStatistics.livenessProbe.successThreshold	Minimum consecutive successes for the probe to be considered successful after having failed. The default is 1, which is required for liveness and startup.	1 or greater	1
gwsServices.gwsPlatformStatistics.livenessProbe.initialDelaySeconds	Number of seconds after the container has started before liveness probes are initiated.	Number	120
gwsServices.gwsPlatformStatistics.livenessProbe.periodSeconds	How often (in seconds) to perform the probe.	1 or greater	30
gwsServices.gwsPlatformStatistics.livenessProbe.timeoutSeconds	Number of seconds after which the probe times out.	1 or greater	10
gwsServices.gwsPlatformStatistics.clientId	The ID of an encrypted client secret generated by Genesys Authentication for this component.	A valid ID	"gws-platform-statistics"
gwsServices.gwsPlatformStatistics.priorityClassName	The class name Kubernetes uses to determine the priority of the pods for this	A valid priority class name	""

Parameter	Description	Valid values	Default
	container deployment relative to other pods. See the Kubernetes documentation for details.		
gwsServices.gwsPlatformStatistics.deployment.replicaCount	The number of pod replicas in this container deployment.	A number greater than 0	2
gwsServices.gwsPlatformStatistics.resources.limits.cpu	The maximum amount of CPU Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of Kubernetes CPU	4
gwsServices.gwsPlatformStatistics.resources.limits.memory	The maximum amount of memory Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of bytes	"4Gi"
gwsServices.gwsPlatformStatistics.resources.requests.cpu	The guaranteed amount of CPU Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of Kubernetes CPU	1
gwsServices.gwsPlatformStatistics.resources.requests.memory	The guaranteed amount of memory Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of bytes	"4Gi"
gwsServices.gwsPlatformStatistics.context.ports.server	The port for this container.	A valid port	48070
gwsServices.gwsPlatformStatistics.context.ports.management	The management port for this container.	A valid port	48071
gwsServices.gwsPlatformStatistics.context.loggingLevel	Specifies the logging level for this container.	ERROR, WARN, INFO, DEBUG, or TRACE	""
gwsServices.gwsPlatformStatistics.context.env	Environment variables for this container.		{}
gwsServices.gwsPlatformStatistics.service.ports.server	The port for this server.	A valid port	80

Parameter	Description	Valid values	Default
gwsServices.gwsPlatformStatistics.service.ports.management	The management port for this server.	A valid port	81
gwsServices.gwsPlatformStatistics.labels	Custom labels to be added for the container.	A valid set of labels as "name: value"	{}
gwsServices.gwsPlatformStatistics.annotations	Custom annotations to be added for the container.	A valid set of annotations as "name: value"	{}

GWS UCS Service parameters

Parameter	Description	Valid values	Default
gwsServices.gwsPlatformUcs.name	The name of the container deployment.	String	"gws-platform-ucs"
gwsServices.gwsPlatformUcs.enabled	Enables the component deployment.	true or false	false
gwsServices.gwsPlatformUcs.appType	The type of application in this container.	nodejs, java, or frontend	"java"
gwsServices.gwsPlatformUcs.image.registry	The Docker registry from which Kubernetes pulls images. If set, this parameter overrides imageGlobals.registry.	A valid registry URL	""
gwsServices.gwsPlatformUcs.clientId	The ID of an encrypted client secret generated by Genesys Authentication for this component.	A valid ID	"gws-platform-ucs"
gwsServices.gwsPlatformUcs.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	""
gwsServices.gwsPlatformUcs.deployment.replicaCount	The number of pod replicas in this container deployment.	A number greater than 0	2
gwsServices.gwsPlatformUcs.resources.limits.cpu	The maximum amount of CPU Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of Kubernetes CPU	4
gwsServices.gwsPlatformUcs.resources.limits.m	The maximum amount of memory Kubernetes	Units of bytes	"4Gi"

Parameter	Description	Valid values	Default
emory	allocates for the container. See the Kubernetes documentation for details.		
gwsServices.gwsPlatformUcs.resources.requests.cpu	The guaranteed amount of CPU Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of Kubernetes CPU	1
gwsServices.gwsPlatformUcs.resources.requests.memory	The guaranteed amount of memory Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of bytes	"4Gi"
gwsServices.gwsPlatformUcs.consul.enabled	Enables Consul registration for the component.	true or false	true
gwsServices.gwsPlatformUcs.context.ports.server	The port for this container.	A valid port	48080
gwsServices.gwsPlatformUcs.context.ports.management	The management port for this container.	A valid port	48081
gwsServices.gwsPlatformUcs.context.loggingLevel	Specifies the logging level for this container.	ERROR, WARN, INFO, DEBUG, or TRACE	""
gwsServices.gwsPlatformUcs.context.env	Environment variables for this container.		{}
gwsServices.gwsPlatformUcs.livenessProbe.enable	Specifies whether to do a Kubernetes liveness probe to test if the container is running.	true or false	true
gwsServices.gwsPlatformUcs.livenessProbe.failureThreshold	Minimum consecutive failures for the probe to be considered failed after having succeeded.	1 or greater	3
gwsServices.gwsPlatformUcs.livenessProbe.successThreshold	Minimum consecutive successes for the probe to be considered successful after having failed. The default is 1, which is required for liveness and startup.	1 or greater	1
gwsServices.gwsPlatformUcs.livenessProbe.initi	Number of seconds after the container has	Number	120

Parameter	Description	Valid values	Default
alDelaySeconds	started before liveness probes are initiated.		
gwsServices.gwsPlatformUcs.livenessProbe.periodSeconds	How often (in seconds) to perform the probe.	1 or greater	30
gwsServices.gwsPlatformUcs.livenessProbe.timeoutSeconds	Number of seconds after which the probe times out.	1 or greater	10
gwsServices.gwsPlatformUcs.service.ports.server	The port for this server.	A valid port	80
gwsServices.gwsPlatformUcs.service.ports.management	The management port for this server.	A valid port	81
gwsServices.gwsPlatformUcs.labels	Custom labels to be added for the container.	A valid set of labels as "name: value"	{}
gwsServices.gwsPlatformUcs.annotations	Custom annotations to be added for the container.	A valid set of annotations as "name: value"	{}

GWS Voice Service parameters

Parameter	Description	Valid values	Default
gwsServices.gwsPlatformVoice.name	The name of the container deployment.	String	"gws-platform-voice"
gwsServices.gwsPlatformVoice.appType	The type of application in this container.	nodejs, java, or frontend	"java"
gwsServices.gwsPlatformVoice.image.registry	The Docker registry from which Kubernetes pulls images. If set, this parameter overrides imageGlobals.registry.	A valid registry URL	""
gwsServices.gwsPlatformVoice.livenessProbe.enable	Specifies whether to do a Kubernetes liveness probe to test if the container is running.	true or false	true
gwsServices.gwsPlatformVoice.livenessProbe.failureThreshold	Minimum consecutive failures for the probe to be considered failed after having succeeded.	1 or greater	3
gwsServices.gwsPlatformVoice.livenessProbe.successThreshold	Minimum consecutive successes for the probe to be considered successful after having failed. The default is 1, which is required for liveness and startup.	1 or greater	1
gwsServices.gwsPlatfor	Number of seconds after	Number	120

Parameter	Description	Valid values	Default
mVoice.livenessProbe.initialDelaySeconds	the container has started before liveness probes are initiated.		
gwsServices.gwsPlatformVoice.livenessProbe.periodSeconds	How often (in seconds) to perform the probe.	1 or greater	30
gwsServices.gwsPlatformVoice.livenessProbe.timeoutSeconds	Number of seconds after which the probe times out.	1 or greater	10
gwsServices.gwsPlatformVoice.clientId	The ID of an encrypted client secret generated by Genesys Authentication for this component.	A valid ID	"gws-platform-voice"
gwsServices.gwsPlatformVoice.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	""
gwsServices.gwsPlatformVoice.deployment.replicaCount	The number of pod replicas in this container deployment.	A number greater than 0	2
gwsServices.gwsPlatformVoice.resources.limits.cpu	The maximum amount of CPU Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of Kubernetes CPU	4
gwsServices.gwsPlatformVoice.resources.limits.memory	The maximum amount of memory Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of bytes	"4Gi"
gwsServices.gwsPlatformVoice.resources.requests.cpu	The guaranteed amount of CPU Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of Kubernetes CPU	1
gwsServices.gwsPlatformVoice.resources.requests.memory	The guaranteed amount of memory Kubernetes allocates for the container. See the	Units of bytes	"4Gi"

Parameter	Description	Valid values	Default
	Kubernetes documentation for details.		
gwsServices.gwsPlatformVoice.context.ports.server	The port for this container.	A valid port	48040
gwsServices.gwsPlatformVoice.context.ports.management	The management port for this container.	A valid port	48041
gwsServices.gwsPlatformVoice.context.loggingLevel	Specifies the logging level for this container.	ERROR, WARN, INFO, DEBUG, or TRACE	""
gwsServices.gwsPlatformVoice.context.env	Environment variables for this container.		{}
gwsServices.gwsPlatformVoice.service.ports.server	The port for this server.	A valid port	80
gwsServices.gwsPlatformVoice.service.ports.management	The management port for this server.	A valid port	81
gwsServices.gwsPlatformVoice.labels	Custom labels to be added for the container.	A valid set of labels as "name: value"	{}
gwsServices.gwsPlatformVoice.annotations	Custom annotations to be added for the container.	A valid set of annotations as "name: value"	{}

Gplus Adapter for Salesforce parameters

Parameter	Description	Valid values	Default
gwsServices.gwsUiCrmworkspace.name	The name of the container deployment.	String	"gws-ui-crmworkspace"
gwsServices.gwsUiCrmworkspace.appType	The type of application in this container.	nodejs, java, or frontend	"frontend"
gwsServices.gwsUiCrmworkspace.image.registry	The Docker registry from which Kubernetes pulls images. If set, this parameter overrides imageGlobals.registry.	A valid registry URL	""
gwsServices.gwsUiCrmworkspace.livenessProbe.enable	Specifies whether to do a Kubernetes liveness probe to test if the container is running.	true or false	false
gwsServices.gwsUiCrmworkspace.livenessProbe.failureThreshold	Minimum consecutive failures for the probe to be considered failed after having succeeded.	1 or greater	3
gwsServices.gwsUiCrmw	Minimum consecutive	1 or greater	1

Parameter	Description	Valid values	Default
orkspace.livenessProbe.successThreshold	successes for the probe to be considered successful after having failed. The default is 1, which is required for liveness and startup.		
gwsServices.gwsUiCrmworkspace.livenessProbe.initialDelaySeconds	Number of seconds after the container has started before liveness probes are initiated.	Number	120
gwsServices.gwsUiCrmworkspace.livenessProbe.periodSeconds	How often (in seconds) to perform the probe.	1 or greater	30
gwsServices.gwsUiCrmworkspace.livenessProbe.timeoutSeconds	Number of seconds after which the probe times out.	1 or greater	10
gwsServices.gwsUiCrmworkspace.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	""
gwsServices.gwsUiCrmworkspace.deployment.replicaCount	The number of pod replicas in this container deployment.	A number greater than 0	2
gwsServices.gwsUiCrmworkspace.resources.limits.cpu	The maximum amount of CPU Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of Kubernetes CPU	1
gwsServices.gwsUiCrmworkspace.resources.limits.memory	The maximum amount of memory Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of bytes	"0.5Gi"
gwsServices.gwsUiCrmworkspace.resources.requests.cpu	The guaranteed amount of CPU Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of Kubernetes CPU	0.1
gwsServices.gwsUiCrmworkspace.resources.requ	The guaranteed amount of memory Kubernetes	Units of bytes	"0.5Gi"

Parameter	Description	Valid values	Default
ests.memory	allocates for the container. See the Kubernetes documentation for details.		
gwsServices.gwsUiCrmworkspace.context.ports.server	The port for this container.	A valid port	50070
gwsServices.gwsUiCrmworkspace.context.ports.management	The management port for this container.	A valid port	50070
gwsServices.gwsUiCrmworkspace.context.env	Environment variables for this container.		{}
gwsServices.gwsUiCrmworkspace.service.ports.server	The port for this server.	A valid port	80
gwsServices.gwsUiCrmworkspace.service.ports.management	The management port for this server.	A valid port	81
gwsServices.gwsUiCrmworkspace.labels	Custom labels to be added for the container.	A valid set of labels as "name: value"	{}
gwsServices.gwsUiCrmworkspace.annotations	Custom annotations to be added for the container.	A valid set of annotations as "name: value"	{}

Agent Setup parameters

Parameter	Description	Valid values	Default
gwsServices.gwsUiProvisioning.name	The name of the container deployment.	String	"gws-ui-provisioning"
gwsServices.gwsUiProvisioning.appType	The type of application in this container.	nodejs, java, or frontend	"frontend"
gwsServices.gwsUiProvisioning.image.registry	The Docker registry from which Kubernetes pulls images. If set, this parameter overrides imageGlobals.registry.	A valid registry URL	""
gwsServices.gwsUiProvisioning.livenessProbe.enable	Specifies whether to do a Kubernetes liveness probe to test if the container is running.	true or false	false
gwsServices.gwsUiProvisioning.livenessProbe.failureThreshold	Minimum consecutive failures for the probe to be considered failed after having succeeded.	1 or greater	3
gwsServices.gwsUiProvisioning.livenessProbe.successThreshold	Minimum consecutive successes for the probe to be considered	1 or greater	1

Parameter	Description	Valid values	Default
	successful after having failed. The default is 1, which is required for liveness and startup.		
gwsServices.gwsUiProvisioning.livenessProbe.initialDelaySeconds	Number of seconds after the container has started before liveness probes are initiated.	Number	120
gwsServices.gwsUiProvisioning.livenessProbe.periodSeconds	How often (in seconds) to perform the probe.	1 or greater	30
gwsServices.gwsUiProvisioning.livenessProbe.timeoutSeconds	Number of seconds after which the probe times out.	1 or greater	10
gwsServices.gwsUiProvisioning.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	""
gwsServices.gwsUiProvisioning.deployment.replicaCount	The number of pod replicas in this container deployment.	A number greater than 0	2
gwsServices.gwsUiProvisioning.resources.limits.cpu	The maximum amount of CPU Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of Kubernetes CPU	1
gwsServices.gwsUiProvisioning.resources.limits.memory	The maximum amount of memory Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of bytes	"0.5Gi"
gwsServices.gwsUiProvisioning.resources.requests.cpu	The guaranteed amount of CPU Kubernetes allocates for the container. See the Kubernetes documentation for details.	Units of Kubernetes CPU	0.1
gwsServices.gwsUiProvisioning.resources.requests.memory	The guaranteed amount of memory Kubernetes allocates for the container. See the	Units of bytes	"0.5Gi"

Parameter	Description	Valid values	Default
	Kubernetes documentation for details.		
gwsServices.gwsUiProvisioning.context.ports.server	The port for this container.	A valid port	50040
gwsServices.gwsUiProvisioning.context.ports.management	The management port for this container.	A valid port	50040
gwsServices.gwsUiProvisioning.service.ports.server	The port for this server.	A valid port	80
gwsServices.gwsUiProvisioning.service.ports.management	The management port for this server.	A valid port	81
gwsServices.gwsUiProvisioning.labels	Custom labels to be added for the container.	A valid set of labels as "name: value"	{}
gwsServices.gwsUiProvisioning.annotations	Custom annotations to be added for the container.	A valid set of annotations as "name: value"	{}

Genesys services parameters

Parameter	Description	Valid values	Default
gauth.authUrl	The URL of the Authentication Service (part of Genesys Authentication). For example: http://gauth-auth.gauth.svc.cluster.local.:80 Note: If a value is set for context.env.GWS_SERVICE_AUTH_URL, it overrides this parameter.	A valid URL	""
gauth.envUrl	The URL of the Environment Service (part of Genesys Authentication). For example: http://gauth-environment.gauth.svc.cluster.local.:80 If a value is set for context.env.GWS_SERVICE_ENV_URL, it overrides this parameter.	A valid URL	""

Third-party services parameters

Parameter	Description	Valid values	Default
postgres.address	The fully qualified domain name or IP of the PostgreSQL server.	A valid address	""
postgres.db	The name of the PostgreSQL database.	A valid database name	""
postgres.enableTls	Enable or disable a TLS connection to PostgreSQL. If true, you must configure the **secretsTls.postgres.** parameters. See Configure connections with TLS and authentication for details.	true or false	false
elasticSearch.address	The fully qualified domain name or IP of the Elasticsearch cluster.	A valid address	""
elasticSearch.port	The Elasticsearch port.	A valid port	9200
elasticSearch.enableTls	Enable or disable TLS connection to the Elasticsearch cluster. If true, you must configure the **secretsTls.elasticsearch.** parameters. See Configure connections with TLS and authentication for details.	true or false	false
elasticSearch.username	The username for the Elasticsearch cluster. The password is set in secrets.gws-elasticsearch-password.	A valid username	""
redis.address	The Redis cluster host name.	A valid address	""
redis.port	The Redis port.	A valid port	6379
redis.enableTls	Enable or disable a TLS connection to the Redis cluster. If true, you must configure the **secretsTls.redis.** parameters. See Configure connections with TLS and authentication for details.	true or false	false

Parameter	Description	Valid values	Default
redis.verifyPeer	Enable or disable validation of the Redis certificate against the list of supplied Certificate Authorities.	true or false	true
consul.port	The port of the local Consul agent.	A valid port	8500
consul.kv_prefix	The prefix used to locate GWS data in the Consul KV datastore.	String	"gws"
prometheus.metricServer.enabled	Enable annotation-based discovery to scrape metrics.	true or false	false

Secrets parameters

Parameter	Description	Valid values	Default
secrets.gws-redis-password	The password to access the Redis cluster.	A valid password	""
secrets.gws-consul-token	The API token to access Consul.	A valid API token	""
secrets.gws-postgres-username	The username to access the PostgreSQL database.	A valid username	""
secrets.gws-postgres-password	The password to access the PostgreSQL database	A valid password	""
secrets.agentsetup-postgres-username	The username to access the PostgreSQL database for gws-app-provisioning.	A valid username	""
secrets.agentsetup-postgres-password	The password to access the PostgreSQL database for gws-app-provisioning.	A valid password	""
secrets.gws-app-provisioning-client-secret	The encrypted client secret generated by Genesys Authentication for the gws-app-provisioning component. See Create API clients.	A valid client secret	""
secrets.gws-app-workspace-client-secret	The encrypted client secret generated by Genesys Authentication for the gws-app-workspace component. See Create API clients.	A valid client secret	""
secrets.gws-platform-	The encrypted client	A valid client secret	""

Parameter	Description	Valid values	Default
chat-client-secret	secret generated by Genesys Authentication for the gws-platform-chat component. See Create API clients.		
secrets.gws-platform-configuration-client-secret	The encrypted client secret generated by Genesys Authentication for the gws-platform-configuration component.	A valid client secret	""
secrets.gws-platform-datacollector-client-secret	The encrypted client secret generated by Genesys Authentication for the gws-platform-datacollector component.	A valid client secret	""
secrets.gws-platform-ixn-client-secret	The encrypted client secret generated by Genesys Authentication for the gws-platform-ixn component.	A valid client secret	""
secrets.gws-platform-ocs-client-secret	The encrypted client secret generated by Genesys Authentication for the gws-platform-ocs component.	A valid client secret	""
secrets.gws-platform-setting-client-secret	The encrypted client secret generated by Genesys Authentication for the gws-platform-setting component.	A valid client secret	""
secrets.gws-platform-statistics-client-secret	The encrypted client secret generated by Genesys Authentication for the gws-platform-statistics component.	A valid client secret	""
secrets.gws-platform-ucs-client-secret	The encrypted client secret generated by Genesys Authentication for the gws-platform-ucs component. See Create API clients.	A valid client secret	""
secrets.gws-platform-voice-client-secret	The encrypted client secret generated by Genesys Authentication for the gws-platform-voice component.	A valid client secret	""
secrets.ops-username	The username of an operational user.	A valid username	""
secrets.ops-password	The encrypted password	A valid password	""

Parameter	Description	Valid values	Default
	of the operational user.		
secrets.gws-elasticsearch-password	The password for the Elasticsearch cluster. The username is set in elasticSearch.username.	A valid password	""

Create or update the versions file

Create or update the **versions.yaml** file with the latest container versions for your deployment. See Updated Helm Charts and Containers for Genesys Web Services and Applications for the full list of versions.

For example:

```
gws-app-provisioning: 9.0.000.93
gws-app-workspace: 9.0.000.90
gws-platform-configuration: 9.0.000.79
gws-platform-datacollector: 9.0.000.50
gws-platform-ixn: 9.0.000.43
gws-platform-ocs: 9.0.000.46
gws-platform-setting: 9.0.000.52
gws-platform-statistics: 9.0.000.61
gws-platform-voice: 9.0.000.66
gws-system-nginx: 9.0.000.16
gws-ui-crmworkspace: 9.0.000.62
gws-ui-provisioning: 9.0.000.84
```

Configure Kubernetes

GWS services stores sensitive data, such as credentials for third-party services, as Kubernetes secrets. For details, see Secrets parameters and Configure connections with TLS and authentication.

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 Global parameters table above.

Pod priority

You can configure pod priority by overriding the **priorityClassName** option for each of the GWS services components - see Override Helm chart values. For example:

```
gwsServices:
  gwsPlatformConfiguration:
    priorityClassName: genesysengage-high-priority
```

Genesys recommends the following priority for GWS pods:

Critical priority pods

- gws-app-provisioning
- gws-app-workspace
- gws-platform-voice

High priority pods

- gws-platform-configuration
- gws-platform-datacollector
- gws-platform-ixn
- gws-platform-ocs
- gws-platform-setting
- gws-platform-statistics
- gws-system-nginx
- gws-ui-crmworkspace
- gws-ui-provisioning

Next steps

- Deploy GWS Services
- Configure GWS Ingress
- Deploy GWS Ingress

Configure connections with TLS and authentication

Contents

Learn how to configure Transport Layer Security and authentication for connections to third-party services and non-containerized Genesys servers.

Related documentation:

-
-
-

RSS:

- For private edition

Genesys Web Services and Applications (GWS) supports secure connections to third-party services and legacy Genesys servers using Transport Layer Security (TLS) version 1.2.

To enable TLS, you must download and unpack the **gws-services** Helm chart locally. Next, create any required certificates for the services and put the truststores under the **gws-services** directory of the unpacked chart. For example: **gws-services/crts/gwsPlatformSettingPostgresTrustore.p12.** Following this example, the setting for the PostgreSQL truststore would be: `secretsTls.postgres.truststores.gws-plaftom-setting-postgres-truststore: crts/gwsPlatformSettingPostgresTrustore.p12`

> **Important**
> When you Deploy GWS Services, make sure to point to your local files during the installation.

Next, configure TLS by overriding Helm chart values in the **values.yaml** file. See TLS for third-party services and TLS for legacy Genesys servers for details.

TLS for third-party services

GWS supports TLS connections to the third-party services Redis, PostgreSQL, and Elasticsearch. To enable TLS for these services, set the following parameters in the **values.yaml** file:

- gwsServices.gwsAppProvisioning.postgres.enableTls
- postgres.enableTls
- elasticSearch.enableTls

- redis.enableTls

You must also define the following truststore paths and passwords in the **values.yaml** file:

Redis

Parameter	Description	Valid values	Default
secretsTls.redis.enabled	Specifies whether a Kubernetes secret is created for the TLS connection to the Redis cluster.	true or false	false
secretsTls.redis.truststores.gws-platform-datacollector-redis-truststore	The Redis client truststore path for the GWS Data Collector Service.	A valid path to the truststore file, relative to the **gws-services** directory.	""
secretsTls.redis.truststores.gws-platform-ixn-redis-truststore	The Redis client truststore path for the GWS Interaction Service.	A valid path to the truststore file, relative to the **gws-services** directory.	""
secretsTls.redis.truststores.gws-app-workspace-redis-truststore	The Redis client truststore path for the GWS Workspace Service.	A valid path to the truststore file, relative to the **gws-services** directory.	""
secretsTls.redis.truststores.gws-app-provisioning-redis-truststore	The Redis client truststore path for Agent Setup.	A valid path to the truststore file, relative to the **gws-services** directory.	""
secretsTls.redis.truststores.gws-platform-voice-redis-truststore	The Redis client truststore path for the GWS Voice Service.	A valid path to the truststore file, relative to the **gws-services** directory.	""
secretsTls.redis.passwords.gws-platform-datacollector-redis-truststore-password	The Redis client truststore password for the GWS Data Collector Service.	A valid password	""
secretsTls.redis.passwords.gws-platform-ixn-redis-truststore-password	The Redis client truststore password for the GWS Interaction Service.	A valid password	""
secretsTls.redis.passwords.gws-app-workspace-redis-truststore-password	The Redis client truststore password for the GWS Workspace Service.	A valid password	""
secretsTls.redis.passwords.gws-app-provisioning-redis-truststore-password	The Redis client truststore password for Agent Setup.	A valid password	""
secretsTls.redis.passwor	The Redis client	A valid password	""

Parameter	Description	Valid values	Default
ds.gws-platform-voice-redis-truststore-password	truststore password for the GWS Voice Service.		

PostgreSQL

Parameter	Description	Valid values	Default
secretsTls.postgres.enabled	Specifies whether a Kubernetes secret is created for the TLS connection to PostgreSQL.	true or false	false
secretsTls.postgres.truststores.gws-platform-setting-postgres-truststore	The PostgreSQL client truststore path for the GWS Setting Service.	A valid path to the truststore file, relative to the **gws-services** directory.	""
secretsTls.postgres.passwords.gws-platform-setting-postgres-truststore-password	The PostgreSQL client truststore password for the GWS Setting Service.	A valid password	""
secretsTls.postgresprovisioning.enabled	Specifies whether a Kubernetes secret is created for the Agent Setup TLS connection to PostgreSQL.	true or false	false
secretsTls.postgresprovisioning.truststores.gws-app-provisioning-postgres-truststore	The PostgreSQL client truststore path for Agent Setup.	A valid path to the truststore file, relative to the **gws-services** directory.	""
secretsTls.postgresprovisioning.passwords.gws-app-provisioning-postgres-truststore-password	The PostgreSQL client truststore password for Agent Setup.	A valid password	""

Elasticsearch

Parameter	Description	Valid values	Default
secretsTls.elasticsearch.enabled	Specifies whether a Kubernetes secret is created for the TLS connection to the Elasticsearch cluster.	true or false	false
secretsTls.elasticsearch.truststores.gws-platform-datacollector-elasticsearch-truststore	The Elasticsearch client truststore path for the GWS Data Collector Service.	A valid path to the truststore file, relative to the **gws-services** directory.	""
secretsTls.elasticsearch.	The Elasticsearch client	A valid password	""

Parameter	Description	Valid values	Default
passwords.gws-platform-datacollector-elasticsearch-truststore-password	truststore password for the GWS Data Collector Service.		

TLS for legacy Genesys servers

GWS supports TLS connections to legacy Genesys servers in a mixed mode environment. GWS uses the Platform SDK to connect to legacy Genesys servers, such as Configuration Server, Interaction Server, T-Server, Universal Contact Server, Stat Server, Chat Server, and Outbound Contact Server.

GWS services use upgrade mode ports for TLS connections between Platform SDK and legacy Genesys services, which means you cannot enable TLS in the GWS **values.yaml** file. Instead, configure the TLS parameters in Configuration Server.

You must also define the following truststore paths and passwords in the GWS **values.yaml** file:

Truststore paths

Parameter	Description	Valid values	Default
psdk.enabled	Specifies whether a Kubernetes secret is created for TLS connections to legacy Genesys servers.	true or false	false
psdk.truststores.gws-platform-configuration-psdk-truststore	The PSDK client truststore path for the GWS Configuration Service.	A valid path to the truststore file, relative to the **gws-services** directory.	""
psdk.truststores.gws-platform-ixn-psdk-truststore	The PSDK client truststore path for the GWS Interaction Service.	A valid path to the truststore file, relative to the **gws-services** directory.	""
psdk.truststores.gws-platform-chat-psdk-truststore	The PSDK client truststore path for the GWS Chat Service.	A valid path to the truststore file, relative to the **gws-services** directory.	""
psdk.truststores.gws-platform-ucs-psdk-truststore	The PSDK client truststore path for the GWS UCS Service.	A valid path to the truststore file, relative to the **gws-services** directory.	""
psdk.truststores.gws-platform-voice-psdk-truststore	The PSDK client truststore path for the GWS Voice Service.	A valid path to the truststore file, relative to the **gws-services** directory.	""
psdk.truststores.gws-platform-statistics-psdk-	The PSDK client truststore path for the	A valid path to the truststore file, relative	""

Parameter	Description	Valid values	Default
truststore	GWS Statistics Service.	to the **gws-services** directory.	
psdk.truststores.truststores.gws-platform-datacollector-psdk-truststore	The PSDK client truststore path for the GWS Data Collector Service.	A valid path to the truststore file, relative to the **gws-services** directory.	""
psdk.gws-platform-ocs-psdk-truststore	The PSDK client truststore path for the GWS OCS Service.	A valid path to the truststore file, relative to the **gws-services** directory.	""

Truststore passwords

Parameter	Description	Valid values	Default
psdk.passwords.gws-platform-configuration-psdk-truststore-password	The PSDK client truststore password for the GWS Configuration Service.	A valid password	""
psdk.passwords.gws-platform-ixn-psdk-truststore-password	The PSDK client truststore password for the GWS Interaction Service.	A valid password	""
psdk.passwords.gws-platform-chat-psdk-truststore-password	The PSDK client truststore password for the Chat Service.	A valid password	""
psdk.passwords.gws-platform-ucs-psdk-truststore-password	The PSDK client truststore password for the UCS Service.	A valid password	""
psdk.passwords.gws-platform-voice-psdk-truststore-password	The PSDK client truststore password for the GWS Voice Service.	A valid password	""
psdk.passwords.gws-platform-statistics-psdk-truststore-password	The PSDK client truststore password for the GWS Statistics Service.	A valid password	""
psdk.passwords.gws-platform-datacollector-psdk-truststore-password	The PSDK client truststore password for the GWS Data Collector Service.	A valid password	""
psdk.passwords.gws-platform-ocs-psdk-truststore-password	The PSDK client truststore password for the GWS OCS Service.	A valid password	""

Deploy GWS Services

Contents

Learn how to deploy GWS Services 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.

> **Important**
>
> Make sure to review Before you begin for the full list of prerequisites required to deploy Genesys Web Services and Applications.

Prepare your environment

To prepare your environment for the deployment, complete the steps in this section for Google Kubernetes Engine (GKE) or Azure Kubernetes Service (AKS).

GKE

Log in to the GKE cluster from the host where you will run the deployment:

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

Create a new namespace for Genesys Web Services and Applications with a JSON file that specifies the namespace metadata. For example, **create-gws-namespace.json**:

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

Execute the following command to create the namespace:

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

Confirm the namespace was created:

```
kubectl describe namespace gws
```

AKS

Log in to the AKS cluster from the host where you will run the deployment:

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

Create a new namespace for Genesys Web Services and Applications with a JSON file that specifies the namespace metadata. For example, **create-gws-namespace.json**:

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

Execute the following command to create the namespace:

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

Confirm the namespace was created:

```
kubectl describe namespace gws
```

Deploy

To deploy GWS Services, you'll need the Helm package and your override files. Copy **values.yaml**, **versions.yaml** and the Helm package (**gws-services-.tgz**) to the installation location. For debugging purposes, use the following command to render templates without installing so you can check that resources are created properly:

```
helm template --debug /gws-services-.tgz -f values.yaml -f versions.yaml
```

The result shows Kubernetes descriptors. The values you see are generated from Helm templates, and based on settings from **values.yaml** and **versions.yaml**. Ensure that no errors are displayed; you will later apply this configuration to your Kubernetes cluster.

Now you're ready to deploy GWS Services:

> **Important**
>
> If you have configured TLS for connections to third-party services or legacy Genesys servers, make sure to point to your local files in the `helm upgrade` command.

```
helm upgrade --install gws-services /gws-services --version= -n gws -f ./override.gws-
services.values.yaml -f ./versions.yaml
```

Validate the deployment

First check the installed Helm release:

```
helm list –n gws
```

The result should show the **gws-services** deployment details. For example:

```
NAME            NAMESPACE   REVISION    UPDATED
STATUS      CHART                   APP VERSION
gws-services    gws         1           2021-05-19  11:49:49.2243107 +0530 +0530
deployed    gws-services-1.0.18     1.0
```

Check the **gws-services** status:

```
helm status gws-services
```

The result should show the namespace details with a status of deployed:

```
NAME: gws-services
LAST DEPLOYED: Wed May 19 11:49:49 2021
NAMESPACE: gws
STATUS: deployed
REVISION: 1
TEST SUITE: None
```

Check the GWS Kubernetes objects created by Helm:

```
kubectl get all -n gws
```

The result should show all the created pods, services, ConfigMaps, and so on.

Finally, confirm Agent Setup is accessible by navigating to **gws./ui/provisioning** in a web browser.

Next steps

- Configure GWS Ingress
- Deploy GWS Ingress
- Provision Genesys Web Services and Applications

Configure GWS Ingress

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
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"
sessionCookieName	The cookie name for sticky sessions.	A valid cookie name	"GWSSESSIONID"
entryPoints.internal.ingress.enabled	Specifies whether internal ingress is enabled. Set this value to false if you are deploying Genesys Web Services and	true or false	true

Parameter	Description	Valid values	Default
	Applications in a single namespace.		
entryPoints.internal.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.internal.ingress.annotations	Custom annotations for internal ingress.	A valid set of annotations as "name: value"	{}
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.hosts	List of external ingress hostnames.	Valid hostnames	["gws.genesys.com"]
entryPoints.external.ingress.tls	List of TLS configurations for external ingress. See	Valid TLS configurations	[]

Parameter	Description	Valid values	Default
	Network requirements for an example configuration.		
gwsServices.gwsAppProvisioning.name	Specifies the name of the **GWS Provisioning Service** deployment.	Value of the **gwsServices.gwsAppProvisioning.name** parameter as described in Configure GWS Services.	"gws-app-provisioning"
gwsServices.gwsAppProvisioning.ingress.enabled	Specifies whether ingress is enabled for the component.	true or false	true
gwsServices.gwsAppProvisioning.service.ports.server	Specifies the service port of the component.	Value of the **gwsServices.gwsAppProvisioning.service.ports.server** parameter as described in Configure GWS Services.	80
gwsServices.gwsAppWorkspace.name	Specifies the name of the **GWS Workspace Service** deployment.	Value of the **gwsServices.gwsAppWorkspace.name** parameter as described in Configure GWS Services.	"gws-app-workspace"
gwsServices.gwsAppWorkspace.ingress.enabled	Specifies whether ingress is enabled for the component.	true or false	true
gwsServices.gwsAppWorkspace.service.ports.server	Specifies the service port of the component.	Value of the **gwsServices.gwsAppWorkspace.service.ports.server** parameter as described in Configure GWS Services.	80
gwsServices.gwsPlatformChat.name	Specifies the name of the **GWS Chat Service** deployment.	Value of the **gwsServices.gwsPlatformChat.name** parameter as described in Configure GWS Services.	gws-platform-chat
gwsServices.gwsPlatformChat.ingress.enabled	Specifies whether ingress is enabled for the component.	true or false	false
gwsServices.gwsPlatformChat.service.ports.server	Specifies the service port of the component.	Value of the **gwsServices.gwsPlatformChat.service.ports.server** parameter as described in Configure GWS Services.	80
gwsServices.gwsPlatformConfiguration.name	Specifies the name of the **GWS Configuration Service** deployment.	Value of the **gwsServices.gwsPlatformConfiguration.name** parameter as described in Configure GWS Services.	"gws-platform-configuration"
gwsServices.gwsPlatformConfiguration.ingress.enabled	Specifies whether ingress is enabled for the component.	true or false	true

Parameter	Description	Valid values	Default
gwsServices.gwsPlatformConfiguration.service.ports.server	Specifies the service port of the component.	Value of the **gwsServices.gwsPlatformConfiguration.service.ports.server** parameter as described in Configure GWS Services.	80
gwsServices.gwsPlatformDatacollector.name	Specifies the name of the **GWS Data Collector Service** deployment.	Value of the **gwsServices.gwsPlatformDatacollector.name** parameter as described in Configure GWS Services.	"gws-platform-datacollector"
gwsServices.gwsPlatformDatacollector.ingress.enabled	Specifies whether ingress is enabled for the component.	true or false	true
gwsServices.gwsPlatformDatacollector.service.ports.server	Specifies the service port of the component.	Value of the **gwsServices.gwsPlatformDatacollector.service.ports.server** parameter as described in Configure GWS Services.	80
gwsServices.gwsPlatformIxn.name	Specifies the name of the **GWS Interaction Service** deployment.	Value of the **gwsServices.gwsPlatformIxn.name** parameter as described in Configure GWS Services.	"gws-platform-ixn"
gwsServices.gwsPlatformIxn.ingress.enabled	Specifies whether ingress is enabled for the component.	true or false	true
gwsServices.gwsPlatformIxn.service.ports.server	Specifies the service port of the component.	Value of the **gwsServices.gwsPlatformIxn.service.ports.server** parameter as described in Configure GWS Services.	80
gwsServices.gwsPlatformOcs.name	Specifies the name of the **GWS OCS Service** deployment.	Value of the **gwsServices.gwsPlatformOcs.name** parameter as described in Configure GWS Services.	"gws-platform-ocs"
gwsServices.gwsPlatformOcs.ingress.enabled	Specifies whether ingress is enabled for the component.	true or false	true
gwsServices.gwsPlatformOcs.service.ports.server	Specifies the service port of the component.	Value of the **gwsServices.gwsPlatformOcs.service.ports.server** parameter as described in Configure GWS Services.	80
gwsServices.gwsPlatformSetting.name	Specifies the name of the **GWS Setting Service** deployment.	Value of the **gwsServices.gwsPlatformSetting.name** parameter as described in Configure GWS Services.	"gws-platform-setting"
gwsServices.gwsPlatformSetting.ingress.enabled	Specifies whether	true or false	true

Parameter	Description	Valid values	Default
	ingress is enabled for the component.		
gwsServices.gwsPlatformSetting.service.ports.server	Specifies the service port of the component.	Value of the **gwsServices.gwsPlatformSetting.service.ports.server** as parameter described in Configure GWS Services.	80
gwsServices.gwsPlatformStatistics.name	Specifies the name of the **GWS Statistics Service** deployment.	Value of the **gwsServices.gwsPlatformStatistics.name** parameter as described in Configure GWS Services.	"gws-platform-statistics"
gwsServices.gwsPlatformStatistics.ingress.enabled	Specifies whether ingress is enabled for the component.	true or false	true
gwsServices.gwsPlatformStatistics.service.ports.server	Specifies the service port of the component.	Value of the **gwsServices.gwsPlatformStatistics.service.ports.server** parameter as described in Configure GWS Services.	80
gwsServices.gwsPlatformUcs.name	Specifies the name of the **GWS UCS Service** deployment.	Value of the **gwsServices.gwsPlatformUcs.name** parameter as described in Configure GWS Services.	"gws-platform-ucs"
gwsServices.gwsPlatformUcs.ingress.enabled	Specifies whether ingress is enabled for the component.	true or false	false
gwsServices.gwsPlatformUcs.service.ports.server	Specifies the service port of the component.	Value of the **gwsServices.gwsPlatformUcs.service.ports.server** parameter as described in Configure GWS Services.	80
gwsServices.gwsPlatformVoice.name	Specifies the name of the **GWS Voice Service** deployment.	Value of the **gwsServices.gwsPlatformVoice.name** parameter as described in Configure GWS Services.	"gws-platform-voice"
gwsServices.gwsPlatformVoice.ingress.enabled	Specifies whether ingress is enabled for the component.	true or false	true
gwsServices.gwsPlatformVoice.service.ports.server	Specifies the service port of the component.	Value of the **gwsServices.gwsPlatformVoice.service.ports.server** parameter as described in Configure GWS Services.	80

Next steps

- Deploy GWS Ingress
- Provision Genesys Web Services and Applications

Deploy GWS Ingress

Contents

Learn how to deploy GWS Ingress 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.

> **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.

Prerequisites

Before you deploy GWS ingress, you must first Deploy GWS Services and Configure GWS Ingress.

Deploy

To deploy GWS ingress, you need the GWS ingress Helm package and override file. Copy **values.yaml** and the Helm package (**gws-ingress-.tgz**) to the installation location. Run the following command to deploy GWS ingress:

```
helm upgrade --install gws-ingress /gws-ingress --version= -n gws -f ./override.gws-
ingress.values.yaml -f ./versions.yaml
```

Validate the deployment

First, check that the pod is running:

```
kubectl get pod
```

The result should show that gws-service-proxy is running. For example:

```
gws-service-proxy-d5997957f-m4kcg 1/1 Running 0 4d13h
```

Check the service:

```
kubectl get svc
```

The result should display the service name, gws-service-proxy. For example:

```
gws-service-proxy ClusterIP 10.202.55.20  80/TCP,81/TCP,85/TCP,86/TCP 4d13h
```

Check the **gws-ingress** status:

```
helm status gws-ingress -n gws
```

The result should show the namespace details with a status of deployed:

```
NAME: gws-ingress
LAST DEPLOYED: Fri Sep 17 11:54:31 2021
NAMESPACE: gws
STATUS: deployed
REVISION: 1
TEST SUITE: None
```

Check the installed Helm release:

```
helm list –n gws
```

The result should show the **gws-services** and **gws-ingress** deployment details. For example:

```
NAME            NAMESPACE    REVISION    UPDATED                                   STATUS
CHART                  APP VERSION

gws-ingress     gws          1           2021-09-17 11:54:31.339091 -0300 ADT    deployed
gws-ingress-0.2.7      1.0

gws-services    gws          1           2021-09-17 11:43:50.0692273 -0300 ADT   deployed
gws-services-1.0.55    1.0
```

Check the GWS Kubernetes objects created by Helm:

```
kubectl get all -n gws
```

The result should show all the created pods, services, ConfigMaps, and so on.

Next steps

- Provision Genesys Web Services and Applications

Provision Genesys Web Services and Applications

Contents

- Administrator

Learn how to provision Genesys Web Services and Applications.

Related documentation:

-
-
-

RSS:

- For private edition

Prerequisites

- You have installed the Genesys Authentication services and the following URLs are accessible:
  - /auth/v3/oauth/token
  - /environment/v3/environments
- You have the ops credentials (admin_username and admin_password) from the **values_gauth.yaml** file.
- Genesys Web Services and Applications services are accessible.
- You have Configuration Server details such as hostname or IP, port, username, password, and cloud application name.

Create API Client

curl --location --request POST '/auth/v3/ops/clients' \

```
--header 'Content-Type: application/json' \
--user ops:ops \ ---------------------------- Cloud ops credentials () from
values_gauth.yaml. The default value is ops:ops
--data-raw '{"data": {
 "name": "external_api_client", -----------------
 "clientType": "CONFIDENTIAL",
 "internalClient": true,
 "refreshTokenExpirationTimeout": 43200,
 "client_id": "external_api_client", -----------------
 "client_secret": "", -------------------------
 "authorities": ["ROLE_INTERNAL_CLIENT"],
 "scope": ["*"],
 "authorizedGrantTypes": ["client_credentials", "authorization_code", "refresh_token",
"password"],
 "redirectURIs": ["https://gauth.","https://wwe.","https://gws.","https://prov."], ----->
```

```
should add gws/prov external URLS here
 "accessTokenExpirationTimeout": 43200
 }
}'
Result:
 "status": {
  "code": 0
 },
"data": {
  "clientType": "CONFIDENTIAL",
  "scope": [
    "*"
  ],
  "internalClient": true,
  "authorizedGrantTypes": [
  "refresh_token",
  "client_credentials",
  "password",
  "authorization_code",
  "urn:ietf:params:oauth:grant-type:token-exchange",
  "urn:ietf:params:oauth:grant-type:jwt-bearer"
 ],
"authorities": [
   "ROLE_INTERNAL_CLIENT"
 ],
  "redirectURIs": [
    "https://gauth.",
    "https://gws.",
    "https://prov.",
  ],
  "accessTokenExpirationTimeout": 43200,
  "refreshTokenExpirationTimeout": 43200,
  "createdAt": 1619796576236,
  "name": "external_api_client",
  "client_id": "external_api_client",
  "client_secret": "secret",
  "encrypted_client_secret": "A34BOmXDedZwbTKrwmd4eA=="
 }
}
```

Create Authentication Token

curl --location --user external_api_client:secret --request POST '/auth/v3/oauth/token' \ ----- user is the API client created in the previous step

```
--data-urlencode 'username=ops' \
--data-urlencode 'client_id=external_api_client' \ ------------------ client ID created in
the previous step
--data-urlencode 'grant_type=password' \
--data-urlencode 'password=ops'

 Result

{
    "access_token": "5f1ecb33-5c63-4606-8e30-824e494194c6",
    "token_type": "bearer",
    "refresh_token": "f0c7eed6-cc55-426f-9594-7ae14903e749",
    "expires_in": 43199,
    "scope": "*"
```

```
}
```

Add Genesys Tenant/Environment

> **Warning**
> Complete this step after installing the Tenant service.

```
curl --location --request POST '/environment/v3/environments' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer f3aa2109-8889-4182-b2b7-d86917c53e4e' \ ----- access token
generated in previous step
--data-raw '{
   "data": {
      "id" : , which is used while deploying the Tenant service
      "username": "default", --------------------------- Configuration Server username
      "password": "password", ------------------------- Configuration Server password
      "connectionProtocol": "addp",
      "remoteTimeout": 7,
      "appName": "Cloud", ------------------------- Cloud app
      "traceMode": "CFGTMBoth",
      "tlsEnabled": false,
      "configServers": [{
         "primaryPort": 2020, ---------------------- Configuration Server port
         "readOnly": false,
         "primaryAddress": "172.24.132.84", ------ Configuration Server IP
         "locations": "/USW1"
      }],
      "localTimeout": 5,
      "tenant": "Environment"
  }
}'

Result
{
   "status": {
      "code": 0
   },
   "path": "/environments/d0fb6386-236c-4739-aec0-b9c1bd6173df" - Environment ID
}
```

Add Contact Center

> **Warning**
> Complete this step after installing the Tenant service.

```
curl --location --request POST '/environment/v3/contact-centers' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer 9901f8d6-0351-47f8-b718-7db992f53a02' \
--data-raw '{
    "data": {
        "domains": ,
        "environmentId": "343dd264-7c26-4f9e-82c5-26baedbcb797", ------- > Environment ID
created in the previous step
        "auth": "configServer",
        "id" : , which is used while deploying Tenant service
    }
}'

Result
{
    "status": {
        "code": 0
    },
    "path": "/contact-centers/ed4c03f3-6275-4419-8b2b-11d14af10655" - Contact center ID
```

Record the contact center ID (also known as CCID) from the POST request above – you need it to provision other Genesys services. Now, open a web browser, navigate to the GWS URL and try to log in using any agent available in Configuration Server.

Update CORS settings

Please follow the Provision Genesys Authentication instructions for CORS settings.

Create an Agent Setup admin user

Complete the steps in this section to create an admin user for Agent Setup.

> ### Important
> The Tenant service should be running and able to access Configuration Server.

1. Log in to Configuration Manager.
2. Create a **Person** (uncheck **isAgent** Checkbox) with **userName**: `AgentAdmin`.
3. Add the created user to the **Users** access group as well as to the **Agent Setup Administrators** group.

Launch Agent Setup using the URL **gws./ui/provisioning** and log in with the AgentAdmin user.

Refer to Get started with Agent Setup for more information.

Upgrade, roll back, or uninstall

Contents

Learn how to upgrade, roll back, or uninstall GWS.

Related documentation:

-
-
-

RSS:

- For private edition

> **Important**
>
> The instructions on this page assume you have deployed the services in service-specific 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.

Supported upgrade strategies

Genesys Web Services and Applications supports the following upgrade strategies:

Service	Upgrade Strategy	Notes
	• Rolling Update	
	• Rolling Update	

The upgrade or rollback process to follow depends on how you deployed the service initially. Based on the deployment strategy adopted during initial deployment, refer to the corresponding upgrade or rollback section on this page for related instructions.

For a conceptual overview of the upgrade strategies, refer to Upgrade strategies in the Setting up Genesys Multicloud CX Private Edition guide.

Timing

A regular upgrade schedule is necessary to fit within the Genesys policy of supporting N-2 releases, but a particular release might warrant an earlier upgrade (for example, because of a critical security fix).

If the service you are upgrading requires a later version of any third-party services, upgrade the third-party service(s) before you upgrade the private edition service. For the latest supported versions of third-party services, see the Software requirements page in the suite-level guide.

Scheduling considerations

Genesys recommends that you upgrade the services methodically and sequentially: Complete the upgrade for one service and verify that it upgraded successfully before proceeding to upgrade the next service. If necessary, roll back the upgrade and verify successful rollback.

Monitoring

Monitor the upgrade process using standard Kubernetes and Helm metrics, as well as service-specific metrics that can identify failure or successful completion of the upgrade (see Observability in Genesys Web Services and Applications).

Genesys recommends that you create custom alerts for key indicators of failure — for example, an alert that a pod is in pending state for longer than a timeout suitable for your environment. Consider including an alert for the absence of metrics, which is a situation that can occur if the Docker image is not available. Note that Genesys does not provide support for custom alerts that you create in your environment.

Preparatory steps

Ensure that your processes have been set up to enable easy rollback in case an upgrade leads to compatibility or other issues.

Each time you upgrade a service:

1. Review the release note to identify changes.
2. Ensure that the new package is available for you to deploy in your environment.
3. Ensure that your existing **-values.yaml** file is available and update it if required to implement changes.

Rolling Update

Rolling Update: Upgrade

Execute the following command to upgrade :

```
helm upgrade --install  -f -values.yaml  -n
```

Tip: If your review of Helm chart changes (see Preparatory Step 3) identifies that the only update you need to make to your existing **-values.yaml** file is to update the image version, you can pass the image tag as an argument by using the `--set` flag in the command:

```
helm upgrade --install  -f -values.yaml  --set .image.tag=
```

GWS example:

```
helm upgrade -f values.yaml -f versions.yaml gws-services ./gws-services
```

GWS Ingress example:

```
helm upgrade -f values.yaml -f versions.yaml gws-ingress ./gws-ingress
```

Rolling Update: Verify the upgrade

Follow usual Kubernetes best practices to verify that the new service version is deployed. See the information about initial deployment for additional functional validation that the service has upgraded successfully.

Rolling Update: Rollback

Execute the following command to roll back the upgrade to the previous version:

```
helm rollback
```

or, to roll back to an even earlier version:

```
helm rollback
```

Alternatively, you can re-install the previous package:

1. Revert the image version in the `.image.tag` parameter in the **-values.yaml** file. If applicable, also revert any configuration changes you implemented for the new release.
2. Execute the following command to roll back the upgrade:

   ```
   helm upgrade --install  -f -values.yaml
   ```

   Tip: You can also directly pass the image tag as an argument by using the `--set` flag in the command:

   ```
   helm upgrade --install  -f -values.yaml  --set .image.tag=
   ```

GWS Services examples

An example using `helm rollback`:

```
helm rollback gws-services
```

An example using `helm upgrade`:

```
helm upgrade -f previous-values.yaml -f previous-versions.yaml gws-services ./gws-services
```

GWS Ingress examples

An example using `helm rollback`:

```
helm rollback gws-ingress
```

An example using `helm upgrade`:

```
helm upgrade -f previous-values.yaml -f previous-versions.yaml gws-ingress ./gws-ingress
```

Rolling Update: Verify the rollback

Verify the rollback in the same way that you verified the upgrade (see Rolling Update: Verify the upgrade).

Uninstall

> **Warning**
> Uninstalling a service removes all Kubernetes resources associated with that service. Genesys recommends that you contact Genesys Customer Care before uninstalling any private edition services, particularly in a production environment, to ensure that you understand the implications and to prevent unintended consequences arising from, say, unrecognized dependencies or purged data.

Execute the following command to uninstall :

```
helm uninstall  -n
```

GWS example

```
helm uninstall gws-services
```

GWS Ingress example

```
helm uninstall gws-ingress
```