Web Services and Applications Private Edition Guide

9/19/2021

Table of Contents

Contents

Find links to all the topics in this guide.

Related documentation:

•

Early Adopter Program

Genesys Engage cloud private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

Genesys Web Services and Applications is a service available with the Genesys Engage cloud 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 Genesys Web Services and Applications
- Provision Genesys Web Services and Applications
- Deploy Genesys Web Services and Applications
- Deploy GWS Ingress
- Provision Agent Setup

- Upgrade, rollback, or uninstall

Operations

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

- Metrics
- Logging

About Genesys Web Services and Applications

Learn about Genesys Web Services and Applications and how it works in Genesys Engage cloud private edition.

Related documentation:

•

Early Adopter Program

Genesys Engage cloud private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

Genesys Web Services and Applications 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 Desktop, a custom desktop.

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

Architecture

Learn about Web Services and Applications architecture.

Related documentation:

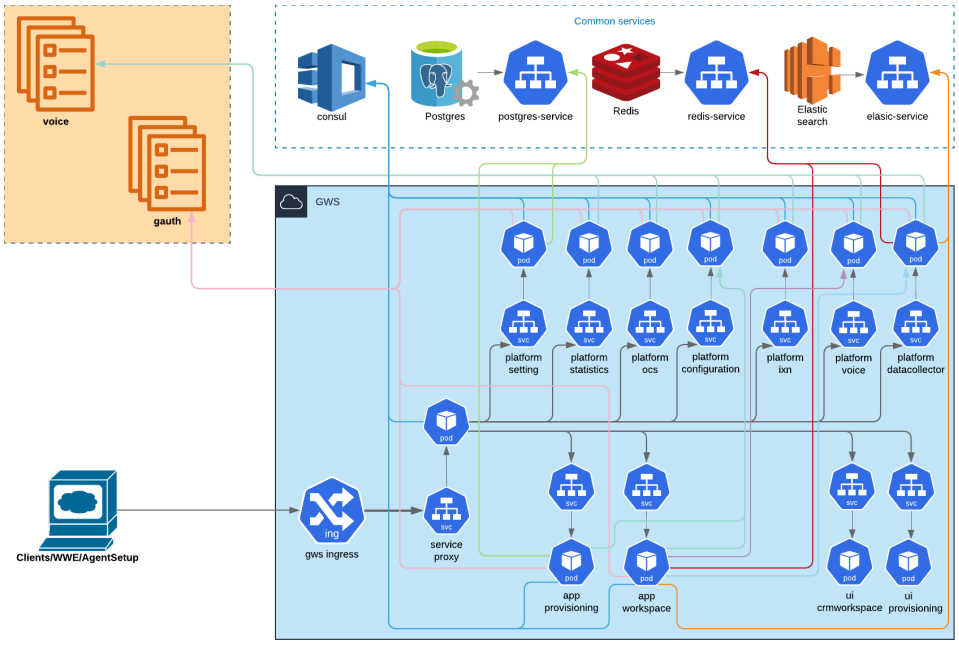
-

Early Adopter Program

Genesys Engage cloud private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

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.


Common services
consul
Postgres
postgres-service
Redis
redis-service
Elastic search
elasic-service
voice
gauth
GWS
platform setting
platform statistics
platform ocs
platform configuration
platform ixn
platform voice
platform datacollector
service proxy
gws ingress
app provisioning
app workspace
ui crmworkspace
ui provisioning
Clients/WWE/AgentSetup

High availability and disaster recovery

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

Related documentation:

-

Early Adopter Program

Genesys Engage cloud private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

See High Availability information for all services: [[PrivateEdition/Current/PEGuide/HADR#HAModes|]]

If you are using Workspace Web Edition, see High availability and disaster recovery.

Before you begin

Contents

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

Related documentation:

-

Early Adopter Program

Genesys Engage cloud private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

Limitations and assumptions

The common Authentication Service must be deployed prior to the Web Services and Applications deployment.

Download the Helm charts

You can download your GWS Helm charts from JFrog using your credentials.

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

For information about downloading Genesys Helm charts from JFrog Edge, see the suite-level documentation: Downloading your Genesys Engage containers.

Third-party prerequisites

Name	Version	Purpose	Shared/Dedicated
Elasticsearch	7.x	Text Searching & Indexing	Shared is Ok
Redis	6.x	Caching and Streaming service	Shared is Ok

Consul	1.8	Service Discovery	Shared Consul instance must be used to support tenant discovery
Postgres	11.x	Relational DB	Shared is Ok

All services are part of the platform list, and can be either dedicated or shared depending on deployment.

Additional options:

1. Redis server must run in cluster mode. Standalone mode is not supported.
2. Elasticsearch server requires additional options configured on each node for proper work of the "GWS Datacollector Service" component:
   ```
   action.auto_create_index: false
   thread_pool.write.queue_size: -1
   ```

Sizing

All third party servers can be shared with other Genesys solutions.

However, Postges, Redis, and Elasticsearch can also be installed as dedicated for GWS. In that case, the minimal requirements are:

- **Postgres:**
  - 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
- **Elasticsearch (4 data nodes):**
  - CPU: 4
  - RAM: 16 GB
  - HDD: 20 GB

Storage requirements

There are no specific storage requirements.

Network requirements

Content coming soon

Browser requirements

You can use any of the following browsers for UIs:

- Chrome 75+
- Firefox 68+
- Firefox ESR 60.9
- Microsoft Edge

Genesys dependencies

Genesys Web Services and Applications (GWS) must be deployed along with Agent Setup and after Genesys Authentication is deployed.

See also Order of services deployment.

GDPR support

Content coming soon

Configure Genesys Web Services and Applications

Contents

Learn how to configure Genesys Web Services and Applications.

Related documentation:

-

Early Adopter Program

Genesys Engage cloud private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

Prerequisites

Deploy Genesys Authentication

The common Authentication Service must be deployed first.

Secret Configuration for Pulling Image

You might already have you secret created.

One of the way to do it is by using the following command:

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

You have to execute the following command to map the secret to the default service account:

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

Prepare your environment

Check the Cluster

Run the following command to get the version of the cluster:

```
oc get clusterversion
```

Create a New Project

Use the following command to create a new project:

```
oc new -project gws
```

Enable Security Context

Use the following command to enable the security context to the default service account:

```
oc adm policy add-scc-to-user genesys-restricted -z default -n gws
```

Download GWS Helm Charts

Download the GWS helm charts from JFrog using your credentials.

Create Two API Clients

Create two API clients on Genesys Authentication using the following procedure:

```
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
--data-raw '{"data": {
 "name": "external_api_client", -----------------
 "clientType": "CONFIDENTIAL",
 "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 ex
  "accessTokenExpirationTimeout": 43200,
  "contactCenterIds": [
  "*" ------------------
 ]
 }
}'
Result:
"status": {
  "code": 0
 },
 "data": {
  "clientType": "CONFIDENTIAL",
  "scope": [
   "*"
  ],
 "internalClient": false,
  "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.",
   ],
  "contactCenterIds": [
   "9350e2fc-a1dd-4c65-8d40-1f75a2e080dd"
  ],
  "accessTokenExpirationTimeout": 43200,
  "refreshTokenExpirationTimeout": 43200,
  "createdAt": 1619796576236,
  "name": "external_api_client",
  "client_id": "external_api_client",
  "client_secret": "secret",
  "encrypted_client_secret": "A34BOmXDedZwbTKrwmd4eA=="
 }
}
```


1. API Client for gws

- **name**: gws-app-workspace (Note: Name should not be changed)
- **client_Id**: gws-app-workspace (Note: Client ID should not be changed)
- **client_secret**: - default password is 'secret'

Record the 'encrypted_client_secret' as it is used to create your secret.

2. API Client for provisioning (Agent-setup)

- **name**: gws-app-provisioning (Note: Name should not be changed)
- **client_Id**: gws-app-provisioning (Note: Client ID should not be changed)
- **client_secret**:

Record the 'encrypted_client_secret' as it is used to create your secret.

Create Secrets

Add the following lines to the value override file to have Helm create secrets during deployment:

```
secrets:
 gws-consul-token:
 gws-postgres-username:
 gws-postgres-password:
 ops-user:
 ops-pass-encr:
 agentsetup-postgres-username:
 agentsetup-postgres-password:
 gws-app-workspace-encrypted:
 gws-app-provisioning-encrypted:
```

Update Parameters in values.yaml

In the values.yaml file provided by Genesys, update following parameters:

```
Image repo details:
 REGISTRY:
Postgres:
 POSTGRES_ADDR: Postgres service DB URL
 POSTGRES_DB: Postgres DB name for gws service
 POSTGRES_USER: Postgres user to access gws DB
 POSTGRES_PASS: Postgres Password
Redis:
 REDIS_ADDR: Address of the Redis cluster
 REDIS_PORT: Redis Port
elastic:
 ELASTICSEARCH_ADDR: Elastic search service master address
 ELASTICSEARCH_PORT: Port of ES service
Authentication service configurations:
 Add/update below variables in env section of all services under 'gwsServices'
  GWS_SERVICE_AUTH_URL: http://gauth-auth.gauth.svc.cluster.local.:80 // Genesys Authentication variable - point
  GWS_SERVICE_ENV_URL: http://gauth-environment.gauth.svc.cluster.local.:80 // Environment variable pointes to i
  GWS_WORKSPACE_SERVICES_ENV: http://gauth-environment.gauth.svc.cluster.local.:80 // Environment variable - poi
  GWS_WORKSPACE_SERVICES_AUTH: http://gauth-auth.gauth.svc.cluster.local.:80 // Genesys Authentication variable
  GWS_WORKSPACE_SERVICES_AUTH_FOR_REDIRECT: https://gauth..com //Genesys Authentication redirect variable - poin
```

Update the Value Overrides for Agent Setup

Agent Setup is part of the GWS deployment. It needs to be configured before the GWS deployment.

From the gws-services helm charts, update the following lines in the value overrides under the `gwsServices` > `appProvisioning` > `context` > `env` section before installing GWS:

- GWS_SERVICE_AUTH_URL: Auth internal service URI from gauth namespace (for example, http://gauth-auth.gauth.svc.cluster.local.:80)
- GWS_SERVICE_ENV_URL: Environment internal service URI from gauth namespace (for example, http://gauth-environment.gauth.svc.cluster.local.:80)
- GWS_SERVICE_CONF_URL: gws internal service URI from gws namespace (for example, http://gws-service-proxy.gws.svc.cluster.local:80)
- GWS_PROVISIONING_SERVICES_AUTH_FOR_REDIRECT : External https ingress URLS from gauth service(ex: https://gauth.)
- GWS_PROVISIONING_OBJECTCACHE_POSTGRES_USER:
- GWS_PROVISIONING_OBJECTCACHE_POSTGRES_PASSWORD:
- GWS_PROVISIONING_OBJECTCACHE_POSTGRES_HOST:
- GWS_PROVISIONING_OBJECTCACHE_POSTGRES_PORT:

Create or Update versions.yaml

Create/update the versions.yaml file with the latest docker versions. See Updated Helm Charts and Containers.

Provision Genesys Web Services and Applications

Contents

- Administrator

Learn how to provision Genesys Web Services and Applications.

Related documentation:

-

Early Adopter Program

Genesys Engage cloud private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

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
--data-raw '{"data": {
 "name": "external_api_client", -----------------
 "clientType": "CONFIDENTIAL",
 "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 ex
 "accessTokenExpirationTimeout": 43200,
 "contactCenterIds": [
 "*" ------------------
 ]
 }
}'
Result:
 "status": {
  "code": 0
 },
"data": {
  "clientType": "CONFIDENTIAL",
  "scope": [
    "*"
  ],
  "internalClient": false,
  "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.",
  ],
  "contactCenterIds": [
     "9350e2fc-a1dd-4c65-8d40-1f75a2e080dd"
  ],
  "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
--data-raw '{
   "data": {
      "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 previou
        "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.

Deploy Genesys Web Services and Applications

Contents

Learn how to deploy Genesys Web Services and Applications.

Related documentation:

-

Early Adopter Program

Genesys Engage cloud private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

Deploying Genesys Web Services and Applications in OpenShift

Login to OpenShift Cluster

Use the following command to log in to OpenShift cluster from the host where you will run deployment:

```
oc login --token  --server
```

Select your GWS Project

Use the following command to select the default GWS project that was created as a prerequisite:

```
oc project gws
```

Render Templates

To ensure that resources are created correctly, you can render the templates for debugging purposes within installing them. Use the following command to render the templates:

```
helm template --debug /gws-services-1.0.18.tgz -f values-gws-new.yaml -f gws-versions_ltst.yaml
```

Kubernetes descriptors are displayed. The values are generated from Helm templates, based on settings from values-gws-new.yaml and gws-versions_ltst.yaml. Ensure that no errors are displayed. This configuration will be applied to your Kubernetes cluster.

Deploy GWS

Use the following command to deploy GWS:

```
helm install gws-services./gws-services-1.0.18.tgz -f values-gws-new.yaml -f gws-versions_ltst.yaml
```

Validate OpenShift Deployment

Use the following command to check the installed Helm release:

```
helm list -n gws
```

The result should show gws-service deployment details similar to the following:

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

To check the Genesys Web Services and Applications project status, use the following command:

```
helm status gws-services
```

The result should display the details with `STATUS: deployed`:

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

To check the GWS OpenShift objects created by Helm, use the following command:

```
oc get all -n gws
```

Deploy GWS Ingress

Contents

Learn how to deploy GWS Ingress.

Related documentation:

-

Early Adopter Program

Genesys Engage cloud private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

Prerequisites

GWS project must be created and 'gws-services' must be installed.

Installation Steps

- Use project gws

```
oc project gws
```

- Credential mapping to default service account

```
oc adm policy add-scc-to-user genesys-restricted -z default-restricted -n gws
```

- Download the gws-ingress helm charts from following repo

```
https://pureengage.jfrog.io/ui/packages/helm:%2F%2Fgws-ingress?name=gws&type=packages
```

- Create Secret for consul token

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

- Copy the values.yaml file from the gws-ingress folder and update the value for Host parameters:

```
REGISTRY: pureengage-docker-staging.jfrog.io/gws
entryPoints:
  internal:
    service:
      type: LoadBalancer
      annotations: {}
    ingress:
      path: /
      annotations: {}
      tlsEnable: false
      secretName: gws-secret-int
      hostName: gws01-int.yourclusterdomain.com   ------------------------------- http internal end point for a
  internalTest:
    service:
      type: LoadBalancer
      annotations: {}
    ingress:
      path: /
      annotations: {}
      tlsEnable: false
      secretName: gws-secret-int
      hostName: gws01-test.yourclusterdomain.com   ------------------------------- http test end point for acce

  external:
    service:
      type: ClusterIP
      annotations: {}
        #service.beta.kubernetes.io/aws-load-balancer-internal: "true"
        #service.beta.kubernetes.io/aws-load-balancer-type: nlb
    ingress:
      path: /
      annotations: {}
      tlsEnable: false
      secretName: gws-secret-ext
      hostName: gws01.yourclusterdomain.com              ------------------------------- http end point for acc
      hostNameTemp: gws-temp.yourclusterdomain.com       ------------------------------- http test end point fo

  externalTest:
    service:
      type: ClusterIP
      annotations: {}
        #service.beta.kubernetes.io/aws-load-balancer-internal: "true"
        #service.beta.kubernetes.io/aws-load-balancer-type: nlb
    ingress:
      path: /
      annotations: {}
        #appgw.ingress.kubernetes.io/connection-draining: "true"
        #appgw.ingress.kubernetes.io/connection-draining-timeout: "30"
        #appgw.ingress.kubernetes.io/cookie-based-affinity: "true"
        #appgw.ingress.kubernetes.io/ssl-redirect: "false"
        #cert-manager.io/cluster-issuer: letsencrypt-prod
        #ingress.kubernetes.io/ssl-redirect: "false"
        #kubernetes.io/ingress.class: azure/application-gateway
      tlsEnable: false
      secretName: gws-secret-ext
      hostName: gws.apps.yourclusterdomain.com  ------------------------------- http end point for accessing GW

Version Details:
gws-system-nginx: 9.0.000.14
```

Copy the above file and the gws-ingress helm package to the installation location.

- Install gws-ingress

```
helm install gws-ingress ./gws-ingress-0.2.7.tgz -f values.yaml
```

Test Installation

To check the ingress installation, run the following commands:

- Check the pod

```
oc get pod
```

It should return gws-service-proxy and the status should be running. Example:

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

- Check the service

oc get svc The result should display the service name. Example:

```
y ClusterIP 10.202.55.20  80/TCP,81/TCP,85/TCP,86/TCP 4d13h
```

Create HTTPS Routes

Run the following command to create https routes to access externally:

The recommended Hostname format is gws.. For example, VCE cluster (https://console-openshift-console.yourclusterdomain.com/), the host name should be gws.yourclusterdomain.com

```
oc create route edge --service=gws-service-proxy --hostname=
```

Test Routes

Access the following URL in a browser. It should navigate you to the login page.

https:///ui/wwe/index.html

Provision Agent Setup

Contents

- Administrator

Learn how to provision Agent Setup.

Related documentation:

-

Early Adopter Program

Genesys Engage cloud private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

Prerequisites

Agent Setup is used to manage the controls and settings that run the contact center and enable the users within it to handle and manage interactions.

Agent Setup is part of Genesys Web Services and Applications (GWS) and must be enabled when deploying GWS services.

Provisioning

To provision Agent Setup you must run the following commands in order to create https access routes for both `gws-app-provisioning` and `gws-ui-provisioning`.

```
oc create route edge --service=gws-ui-provisioning-blue-srv --hostname= --path /ui/provisioning
```

```
oc create route edge --service=gws-app-provisioning-blue-srv --hostname= --path /provisioning
```

Genesys recommends the hostname format prov.. Example: For VCE cluster (https://console-openshift-console.apps..com/), the host name should be prov.apps..com

Next Step

Launch Agent Setup using the URL **https:///ui/provisioning**.

Refer to Get started with Agent Setup for more information.

Upgrade, rollback, or uninstall

Contents

Learn how to upgrade, rollback or uninstall Genesys Web Services and Applications.

Related documentation:

-

Early Adopter Program

Genesys Engage cloud private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

Upgrade Genesys Web Services and Applications

Content coming soon

Rollback Genesys Web Services and Applications

Content coming soon

Uninstall Genesys Web Services and Applications

Content coming soon

Metrics

Learn which metrics you should monitor for Web Services and Applications and when to sound the alarm.

Related documentation:

-

Early Adopter Program

Genesys Engage cloud private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

Content coming soon

Logging

Related documentation:

-

Early Adopter Program

Genesys Engage cloud private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

Genesys Web Services and Applications output logs to stdout.