Genesys Web Services and Applications Private Edition Guide

Provision Genesys Web Services and Applications

4/14/2026

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.