PureEngage - Genesys Cloud Services Hybrid Integrations Hybrid Deployments

Provisioning PureEngage Hybrid Integrations

1/22/2020

Contents

This article describes the essential provisioning steps to enable a hybrid integration between PureEngage On-Prem deployments and Genesys PureCloud services.

Supported Services

The following PureCloud services are supported and have supplementary documentation:

- Genesys Altocloud
  - Altocloud for Workspace Desktop Edition
  - Agent Pacing Service

About Provisioning

Before proceeding with the information in this article you should consult with the Genesys Professional Services team that you are working with to obtain the information needed to complete the provisioning.

To support the different authentication mechanisms in PureCloud Integration, you must create a transaction object in Genesys Configuration Server under the environment and associated script folder with the following data. PureEngage On-Premises Services, Components, and UIs will use this information to authenticate with the PureCloud Common Services and UIs.

Use Genesys Administrator Extension to manually create all of the PureCloud Common Service–related configuration information in Configuration Server at the Tenant level.

After you purchase a common cloud service, you will receive a welcome email to activate your admin accounts with PureCloud. With those credentials you can log in to the PureCloud Admin UI to perform the provisioning steps described below.

PureCloud Provisioning Steps

Perform the following steps using the PureCloud Admin UI or the PureCloud API.

1. As necessary, for each on-premises service, create OAuth Client Credentials grants:
   - Using the UI, follow these steps,

- Or using the API, reference these endpoints.

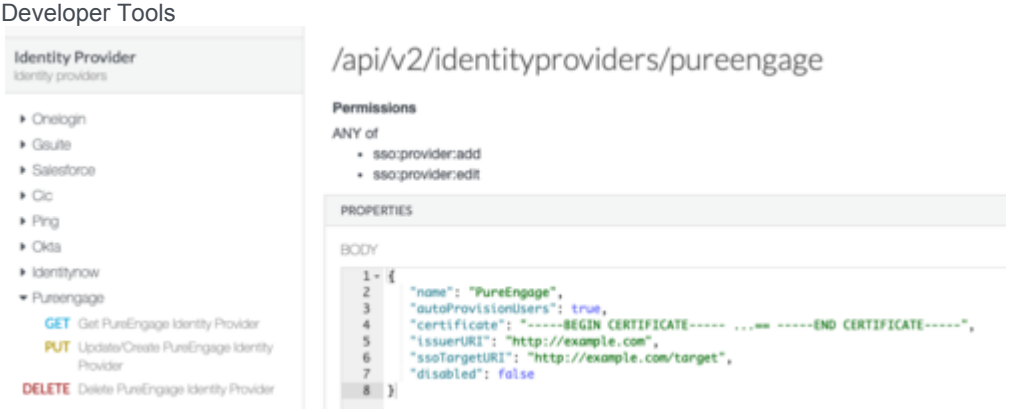
For more information about the different kinds of Client Grants, see the Authorization reference.
For more information about Permissions for Altocloud, see the Altocloud permissions overview.

2. Create a PureEngage Identity Provider (IDP). You can use the Identity Provider API via the PureCloud Developer Tools, SDKs, or Platform API.
   Sample Request:

```
PUT https://api.{{environment}}/api/v2/identityproviders/pureengage

{
   "name": "PureEngage",
   "autoProvisionUsers": true,
   "certificate": "-----BEGIN CERTIFICATE----- ...== -----END CERTIFICATE-----",
   "issuerURI": "http://www.genesys.com/pureengage",
   "ssoTargetURI": "http://example.com/target",
   "disabled": false
}
```

- Developer Tools



- SDKs

```
String publicCert = new String(Files.readAllBytes(Paths.get(publicCertFilename)));
IdentityProviderApi api = new IdentityProviderApi();
PureEngage pureEngage = new PureEngage(){{
    setCertificate(publicCert);
    setDisabled(false);
    setIssuerURI(issuer);
    setSsoTargetURI(targetUri);
    setAutoProvisionUsers(true);
}};
api.putIdentityprovidersPureengage(pureEngage);
```

  - Java
  - .NET
  - Python

- Platform APIs

```
PUT https://api.{{environment}}/api/v2/identityproviders/pureengage

{
   "name": "PureEngage",
   "autoProvisionUsers": true,
   "certificate": "-----BEGIN CERTIFICATE----- ...== -----END CERTIFICATE-----",
   "issuerURI": "http://www.genesys.com/pureengage",
   "ssoTargetURI": "http://example.com/target",
   "disabled": false
}
```

Authorization:

- Type: OAuth 2.0
- Access Token: request new token
- Add authorization data to: Request Headers

Troubleshooting:

- Ensure that the IDP is set with "autoProvisionUsers" = "true"
- Ensure that the issuer URI in your SAML assertion is the same as the issuer URI for the IDP.
- Ensure that you don't have multiple issuers with the same URI.

3. By default, Altocloud permissions are included in the Admin and AI Agent roles. You may grant Altocloud permissions to additional roles as needed.
4. (Optional as needed) Create additional Admin accounts by adding people to your organization and assigning them to the Admin role.

Transaction object for hybrid integrations

A transaction object is needed for Genesys components to authenticate with Genesys Cloud.

1. Create a transaction object (and alias) of type **list** named **hybrid_integration** in the **Script** folder of the **Environment** tenant.
   - Usage characteristics: the transaction object should be acquired at start-up and used until the component gets an error from a given API call. If an error occurs, your component should retrieve the hybrid_integration object from config server and try again. If the component still has problems, your component should end the associated processing with an error.
   - Tenant characteristics: the transaction object should be cretaed at the Environment level so it can be shared by multiple tenants. The transaction object allows for support of both single tenant and multi-tenant Configuration Servers. This can be overridden by putting the object in under a specific tenant level.
2. Create the following Object options in the **general** section:
   - **organization_sname**: The PureCloud organization short name for this tenant.

- **organization_id**: The PureCloud organization id for this tenant.
- **default_agent_role_name**: The default PureCloud agent role name.
- **default_supervisor_role_name**: The default PureCloud supervisor role name.
- **default_admin_role_name**: The default PureCloud admin role name.
- **base_auth_url**: The base auth URL that can be used for any PureCloud service; for example: **base_auth_url** should be https://[region_host]/oauth/token. [region_host] is the authentication-based FQDN for the region; the regions are listed on this page.
- **base_service_url**: The base URL that can be used for any PureCloud service; for example: **base_service_url** should be https://[region_host]/api/. [region_host] should be the API-based FQDN for the region; the regions are listed on this page. The rest of the url is PureCloud service and version specific; for example: **...v2/conversations**. The **base_service_url** and the service specific portion is combined in your componment code.

3. Create the following Object options in the **saml** section:
   - **issuer**: The SAML IDP URL.
   - **certificate**: The SAML related certificate.
   - **pkey**: The encrypted SAML related certificate private key.
   - **password**: The password to decrypt the private key.
   - **expire_time**: The expiration time for the access token.
4. To allow for better control and monitoring of the components using PureCloud Services, for each PureEngage Service that uses a common service you must create multiple sections, one for each OAuth client, in Configuration Server to allow for better control and monitoring of the components using the PureCloud Services and for different rate limiting per client. This does not mean that if you have *n* number of components on premises that are associated with one another, they cannot share a given client id.
   Genesys recommends that you consult architecture before performing this step.

   Create the following two options in each section:
   - **client_id**: The Client Credential Grant Client ID.
   - **password**: The Client Credential Grant Client secret.For example:
     - Create the following Object options in the **saml_auth** section for the OAuth client for SAML Authentication from the client (such as Workspace Desktop Edition):
       - **client_id**
       - **password**
     - For each Service or component using a Genesys API, such as the Agent Pacing Service, create the following Object options in the **ewt** section for the pacing engine to connect to PureCloud:
       - **client_id**
       - **password**

Opening Your Network

You must modify the permissions on our network to permit the PureEngage Components and UIs to access PureCloud Common APIs over your network and into the Internet. To do this, you must create a set of new firewall rules for the PureCloud Authentication and Common Services URLs.