Geospatial Content Server System Administration Guide

Release 2.3.0, January 2020

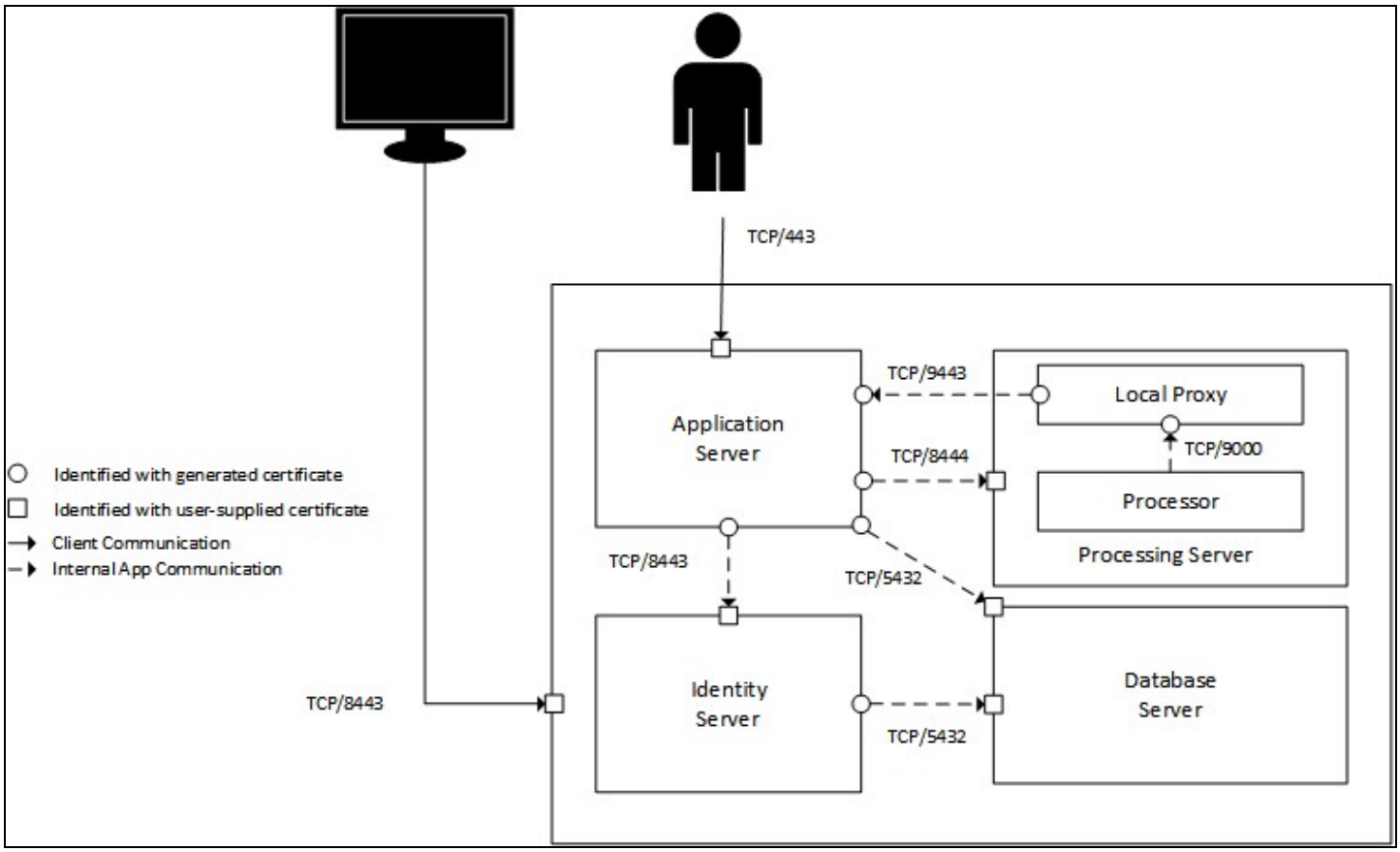
Table of Contents

1 Overview

Geospatial Content Server (GCS) provides a comprehensive enterprise solution for hosting, processing, serving, and analyzing terrain, imagery, and other heterogeneous 3D data. This document details how to install GCS on a single machine. If you want to install GCS on a distributed system, please contact AGI support.

1.1 Architecture

The diagram below depicts the GCS system architecture. The application server has two main responsibilities: to serve processed assets and to accept user requests to process assets. The actual processing of the data is performed in a separate process, the processing server, to allow for distributed scale-out with multi-machine installations. All communications between the GCS components are secured via TLS.



2 Prerequisites

Our installation assumes the use of the following technologies. If you use other technologies, please contact AGI support.

- Red Hat Enterprise Linux (RHEL) or CentOS 7, 64-bit
- firewalld (usually installed by default)
- Python 2 or 3 (usually installed by default)

Each server must also have an SSL Certificate generated by a trusted certificate authority. You are required to provide the certificate, private key, and certificate chain to the trusted root certificate authority when installing individual components (database, identity server, et al). In a single-machine scenario, the SSL certificate will be the same for all components.

2.1 Hardware Requirements

The following section details the **minimum** hardware required to install and run the Geospatial Content Server on a single machine.

Size	Number of Users	Hardware Requirements
Small	25 concurrent or up to 250 intermittent users	16 GB RAM 2x CPU (at least 2.0Ghz) 512 GB disk space

Size	Number of Users	Hardware Requirements
Medium	50 concurrent or up to 500 intermittent users	32 GB RAM 4x CPU (at least 2.0Ghz) 1 TB disk space
Large	100 concurrent or up to 1000 intermittent users	32 GB RAM 8x CPU (at least 2.0Ghz) 1 TB disk space

Note: The values above represent the minimum requirements. After processing, geospatial content can occupy up to four times as much space on the disk as the uploaded source files. If your organization processes large data files, you may require more disk space than is listed in the above requirements.

2.2 Extra Packages for Enterprise Linux (EPEL)

In order to ensure compatibility between our software and your operating system, we install some packages from the Fedora Project's Extra Packages for Enterprise Linux (EPEL) during our installation process. EPEL is a collection of commonly used libraries and other software maintained by Red Hat and the Fedora community. Using EPEL ensures binary compatibility of third-party libraries and allows you to patch vulnerabilities in these libraries immediately, without requiring a new version of GCS. For more information on EPEL, please see their website: https://fedoraproject.org/wiki/EPEL.

Note: After you have installed EPEL, you may have to disable verification of metadata. You can find this setting either in the global configuration file (/etc/yum.conf), or in a repository-specific config file (a *.repo file in /etc/yum.repos.d):

```
gpgcheck=0
```

Alternatively, if you have set up GPG signing keys for your EPEL instance, you can enable this property:

```
gpgcheck=1
```

Note: EPEL must be installed on the machine you use to generate certificates and on the identity, application, and processing servers. You do not need to have it configured on the database server.

3 Pre-Installation

> **Note:** All scripts must be executed as a privileged user. Additionally, all of the installation scripts located at the installation root have a --help option. Running the scripts with this flag will print descriptions for all required and optional flags.

3.1 GCS License

In order to run GCS, you must obtain a license file from AGI. The application and processing server installation scripts expect the license to be in a "licenses" folder in the installation root. Yet, you may place your license in a custom location and specify its absolute path using the --license-file flag when running "install-app-server.sh" and "install-processing-server.sh" during the Installation Steps.

If you do not yet have a license and need to find the MAC address (host ID) of a machine, you can run print-host-id.sh from the "utils" directory of the installer:

```
bash utils/print-host-id.sh
```

There are two types of license files available: node-locked license files and network license files.

3.1.1 Node-locked (local) licenses

A node-locked license is the traditional license file. This license type does not require additional servers, but the license file itself must be maintained. If additional features are purchased or the license has expired, you will need to replace this file on each affected machine.

3.1.2 Networked licenses

Networked licenses are more flexible than traditional licenses. They can be maintained on a centralized license server, simplifying updates. This license type does require additional infrastructure to host the license server itself.

3.1.3 Deploying license files

If you are performing a new installation, follow the Installation Steps. If you have an existing installation that you are upgrading, follow the Upgrade Steps. If you want to update your license file without installing or upgrading, the license files should be copied to the application and processing servers under the /usr/local/tomcat/licenses/ folder.

> **Note:** On a single machine installation, the processing server license file can be found under /usr/local/gcs-processing/licenses folder

---

3.2 Complying with STIG requirements

If your installation needs to comply with Security Technical Implementation Guide (STIG) requirements you must provide the --harden flag to all server installation scripts. Visit the Defense Information Systems Agency website for more information.

> **Note:** You must specify new values for the user name and password via the `--keycloak-admin-user` and `--keycloak-admin-password` command-line arguments. Failure to do so will result in an insecure installation using default values.

3.3 Generating Internal GCS Certificates

Prior to installing, you need to create certificates by running "generate-certificates.sh". GCS uses these certificates for communication among its internal services. The script will create a "certificates" directory at the installation root containing these generated certificates.

```
bash generate-certificates.sh --app-server-internal-name=app.example.com
```

Generated certificates are unique to your environment, contain sensitive information, and cannot be replaced if lost. The certificates folder should be backed up securely with limited read access.

> **Note:** These certificates are for only GCS internal communication. You will still need machine-specific certificates for the server(s) on which you run the installer.

3.4 GCS User Roles and Permissions

The following table describes the different GCS user roles and permissions. The values you provide to the parameters --gcs-processing-group and --gcs-admin-group will be mapped to the gcs_processing and gcs_admin roles when installing the identity server.

User Permissions	Roles		
	(no role assigned)	gcs_ processing	gcs_ admin
User is able to view published assets	X	X	X
User is able to submit assets for processing		X	

User Permissions	Roles		
	(no role assigned)	gcs_ processing	gcs_ admin
User is able to perform administrative tasks, such as viewing system logs			X

4 Installation Worksheets

> **Note:** Unless otherwise specified, all parameters listed below are required for installation.

This section provides worksheets for collecting the information needed to install the software components as well as information that you will finalize as you perform the installations.

> **Note:** This section is for single machine installations. If your organization is installing for a distributed system, refer to the Appendix: Installation Worksheets (Distributed).

The tables below show the parameters required in the installer scripts along with a description of the parameters (the 'Parameter : Description' column), the example value used in the guide (the 'Example Value' column), and a space for you to enter your value (the 'Your Value' column).

> **Note:** As shown in the Installation Steps, the values in the 'Example Value' and the 'Your Value' columns should be surrounded by single quote characters when supplied to commands.

4.1 Common Parameters Worksheet

The following common parameters must be provided to each installation script:

Parameter : Description	Example Value	Your Value
--ssl-ca-file : PEM-formatted certificate authority file	customer-ca.crt	
--ssl-cert-file : PEM-formatted server certificate file	customer-server.crt	

Parameter : Description	Example Value	Your Value
--ssl-key-file : Unencrypted PEM RSA server certificate key file	customer-server.key	

4.2 Database Server Worksheet

Parameter : Description	Example Value	Your Value
--ip-network-mask : An IPv4 address range formatted in CIDR notation to determine which client hosts may attempt to connect to the database server. This parameter may be passed multiple times.	10.0.0.0/16	

4.3 Application Server Worksheet

4.3.1 Optional

Parameter : Description	Example Value	Default	Your Value
--ssl-protocols : The names of the protocols to support when communicating with clients. Supported protocol strings are defined by Apache (https://tomcat.apache.org/tomcat-9.0-doc/config/http.html).	TLSv1.2,TLSv1.3	TLSv1.2	
--app-server-client-auth-port : The port that listens for certificate authentication requests from the processing server.	9443	9443	

4.4 Processing Server Worksheet

4.4.1 Optional

Parameter : Description	Example Value	Default	Your Value
--app-server : The fully-qualified domain name of the application server on your internal network.	app.example.com	*hostname*	
--app-server-external : The user-facing fully-qualified domain name of the application server if different from the internal name.	public-gcs.example.com	*hostname*	
--proxy-port : The port number to be used by the internal proxy. This proxy encrypts communications between the local host (processing server) and the application server.	9000	9000	
--ssl-protocols : The names of the protocols to support when communicating with clients. Supported protocol strings are defined by Apache (https://tomcat.apache.org/tomcat-9.0-doc/config/http.html).	TLSv1.2,TLSv1.3	TLSv1.2	

4.5 Identity Server Worksheet

This document outlines the installation and configuration of an identity server using directory-based authentication (e.g., LDAP). Please contact AGI support if you do not have a directory server available.

> **Note:** Use the --keycloak-admin-user and --keycloak-admin-pass flags to change the username and password assigned to the administrative user for Keycloak. If you do not specify these flags, the default value of 'admin' will be used for both the username and password. After installation, you may change the password by logging into the AGI Identity Platform Administration Console.

> **Note:** The --keycloak-admin-ip-addresses parameter provides additional security and is necessary if you must comply with Security Technical Implementation Guidelines (STIG) requirements.

Parameter : Description	Example Value	Your Value
--gcs-admin-group : Common name (CN) of an LDAP group that has admin access to GCS	gcs_administrators	
--gcs-processing-group : Common name (CN) of an LDAP group that has processing access in GCS	gcs_processing	
--ldap-groups-dn : Distinguished name (DN) of a node where groups are defined in LDAP	ou=Groups,dc=example,dc=com	

Parameter : Description	Example Value	Your Value
--ldap-bind-dn : Bind distinguished name (DN) for LDAP	CN=LDAPUser,CN=Users,DC=example,DC=com	
--ldap-users-dn : Distinguished name (DN) where users are defined in LDAP	CN=users,DC=example,DC=com	
--ldap-bind-credentials : Password for the LDAP server	credentials123	
--ldap-connect-url : URL where the LDAP server is located	ldap://machineName.com:1234	
--keycloak-admin-ip-addresses : An IPv4 address range formatted in CIDR notation to determine which client hosts may access the Identity server admin console	192.168.0.0/24	

Parameter : Description	Example Value	Your Value
--keycloak-admin-user : Username for the Keycloak administrative user	systemadmin	
--keycloak-admin-pass : Password for the Keycloak administrative user	Passw0rd123!	
--app-server : The fully-qualified domain name of the application server on your internal network	app.example.com	

4.5.1 Optional

Parameter : Description	Example Value	Default	Your Value
--app-server-client-auth-port : The port that the application server uses to accept certificate-authenticated requests from the processing server	9443	9443	

5 Installation Steps

For a new installation, follow the steps outlined below, being sure to run the commands in the order shown. If you are upgrading an existing installation, please see Upgrade Steps.

When specifying parameter values, the equals operator must be included as shown in the examples. In this section, all values in green should be replaced with your values from the Installation Worksheets section.

> **Note:** This section is for single machine installations. If your organization is installing for a distributed system, refer to the Appendix: Installation Steps (Distributed).

> **Note:** All file locations should be specified as absolute paths.

1. Install the Database Server.

```
bash install-db-server.sh \
        --ssl-ca-file='/tmp/installer/certificates/customer-ca.crt' \
        --ssl-cert-file='/tmp/installer/certificates/gcs.yourdomain.com.crt' \
        --ssl-key-file='/tmp/installer/certificates/gcs.yourdomain.com.key' \
        --ip-network-mask='10.0.0.0/16'
```

2. Install the Identity Server.

```
bash install-id-server.sh \
        --ssl-ca-file='/tmp/installer/certificates/customer-ca.crt' \
        --ssl-cert-file='/tmp/installer/certificates/gcs.yourdomain.com.crt' \
        --ssl-key-file='/tmp/installer/certificates/gcs.yourdomain.com.key' \
        --gcs-admin-group='gcs_administrators' \
        --gcs-processing-group='gcs_processing' \
        --ldap-groups-dn='ou=Groups,dc=example,dc=com' \
        --ldap-bind-dn='CN=LDAPUser,CN=users,DC=example,DC=com' \
        --ldap-users-dn='CN=users,DC=example,DC=com' \
        --ldap-bind-credentials='credentials123' \
        --license-file='yourLicenseFile.lic'
```

3. Install the Application Server.

```
bash install-app-server.sh \
        --ssl-ca-file='/tmp/installer/certificates/customer-ca.crt' \
        --ssl-cert-file='/tmp/installer/certificates/gcs.yourdomain.com.crt' \
        --ssl-key-file='/tmp/installer/certificates/gcs.yourdomain.com.key' \
        --license-file='yourLicenseFile.lic'
```

4. Install the Processing Server.

> **Note:** You must install the processing server even if you have a serving-only license.

```
bash install-processing-server.sh \
        --ssl-ca-file='/tmp/installer/certificates/customer-ca.crt' \
        --ssl-cert-file='/tmp/installer/certificates/gcs.yourdomain.com.crt' \
        --ssl-key-file='/tmp/installer/certificates/gcs.yourdomain.com.key' \
        --license-file='yourLicenseFile.lic'
```

6 Post-Installation Verification

6.1 Database Server

1. Check the status of the service.

```
systemctl status postgresql-10
```

2. Check for the 'Active' state.

```
.postgresql-10.service - PostgreSQL 10 database server
Loaded: loaded (/etc/systemd/system/postgresql-10.service; enabled; vendor preset:
disabled)
Active: active (running) since Thu 2018-06-21 21:37:59 UTC; 3 days ago
Docs: https://www.postgresql.org/docs/10/static/
Main PID: 42446 (postmaster)
CGroup: /system.slice/postgresql-10.service
├─ ...
└─
Jun 21 21:37:58 machine-name systemd[1]: Starting PostgreSQL 10 database server...
Jun 21 21:37:58 machine-name postmaster[42446]: 2018-06-21 21:37:58.850 UTC
[42446] LOG: pgaudit extension initialized
Jun 21 21:37:58 machine-name postmaster[42446]: 2018-06-21 21:37:58.851 UTC
[42446] LOG: listening on IPv4 address "0.0.0.0", port 5432
...
Jun 21 21:37:59 machine-name systemd[1]: Started PostgreSQL 10 database server.
```

6.2 Identity Server

1. Check the status of the service.

```
systemctl status keycloak
```

2. Check for the 'Active' state.

```
.keycloak.service - WildFly Application Server for Keycloak
Loaded: loaded (/etc/systemd/system/keycloak.service; enabled; vendor preset:
disabled)
Active: active (running) since Thu 2018-06-21 21:41:32 UTC; 16h ago
Main PID: 46656 (standalone.sh)
CGroup: /system.slice/keycloak.service
├─46656 /bin/sh /usr/local/keycloak/bin/standalone.sh -Djboss.bind.address=0.0.0.0
└─47443 /usr/java/keycloak/bin/java -D[Standalone] -server -Xmx512m...
Jun 21 21:41:32 machine-name systemd[1]: Started WildFly Application Server for
Keycloak.
Jun 21 21:41:32 machine-name systemd[1]: Starting WildFly Application Server for
Keycloak...
```

6.3 Application Server

1. Check the status of the service.

```
systemctl status tomcat
```

2. Check for the 'Active' state.

```
        .tomcat.service - Apache Tomcat Java Servlet Container
        Loaded: loaded (/etc/systemd/system/tomcat.service; enabled; vendor preset:
disabled)
        Active: active (running) since Thu 2018-06-21 21:39:24 UTC; 17h ago
        Main PID: 43866 (java)
        CGroup: /system.slice/tomcat.service
        ├─43866 /usr/java/default/bin/java -
Djava.util.logging.config.file=/usr/local/tomcat/conf/logging.properties
        ├─ ...
        └─
        Jun 21 21:39:24 machine-name systemd[1]: Starting Apache Tomcat Java Servlet
Container...
        Jun 21 21:39:24 machine-name systemd[1]: Started Apache Tomcat Java Servlet
Container.
```

6.4 Processing Server

1. Check the status of the service.

```
systemctl status gcs-processing
```

2. Check for the 'Active' state.

```
        .gcs-processing.service - Apache Tomcat Java Servlet Container
        Loaded: loaded (/etc/systemd/system/gcs-processing.service; enabled; vendor
preset: disabled)
        Active: active (running) since Thu 2018-06-21 21:39:24 UTC; 17h ago
        Main PID: 43866 (java)
        CGroup: /system.slice/gcs-processing.service
        ├─43866 /usr/java/default/bin/java -Djava.util.logging.config.file=/usr/local/gcs-
processing/conf/logging.properties
        ├─ ...
        └─
        Jun 21 21:39:24 machine-name systemd[1]: Starting Apache Tomcat Java Servlet
Container...
        Jun 21 21:39:24 machine-name systemd[1]: Started Apache Tomcat Java Servlet
Container.
```

7 Upgrade Steps

When upgrading from an older version of GCS, you must upgrade existing database, identity, application, and processing servers. Existing assets and log data will be retained.

You do not need to execute the install*.sh scripts; you should only execute the scripts described below.

- The system will be unavailable during the upgrade process.
- You must backup your systems prior to performing an upgrade.
- You must supply the certificates generated during the original installation.
- Be mindful to use the new installer files when upgrading. Re-running the old installer bundle will not automatically find the new files.

7.1 Generating Certificates

> **Note:** GCS 2.3.0 requires additional certificates. This step should be executed once and the certificates should be used for all upgrade paths. When installing in a distributed environment, steps 1 and 2 should be performed for each machine and the generated certficates should be copied from the machine where *generate-certificates-for-upgrade.sh* was executed to */tmp/gcs-2.3.0/certificates*.

1. Create a new directory and copy the upgrade archive to this location.

```
mkdir /tmp/gcs-2.3.0

cp geospatial-content-installer-*.tgz /tmp/gcs-2.3.0
```

2. Extract the contents of the archive.

```
cd /tmp/gcs-2.3.0
```

```
tar xvf geospatial-content-installer-*.tgz
```

3. Copy the certificates backup to the upgrade folder.

```
cp -a {backup_location}/certificates /tmp/gcs-2.3.0
```

4. Execute the upgrade script.

```
bash generate-certificates-for-upgrade.sh
```

7.2 Database Server Upgrade

1. Execute the upgrade script.

```
bash upgrade-db-server.sh
```

7.3 Identity Server Upgrade

1. Execute the upgrade script.

```
bash upgrade-id-server.sh
```

7.4 Application Server Upgrade

> **Note:** You can update the license file during the upgrade by supplying the `--license-file` parameter. If you do not supply a value for this parameter, the existing license will be reused.

1. Execute the upgrade script.

```
bash upgrade-app-server.sh
```

7.5 Processing Server Upgrade

> **Note:** You can update the license file during the upgrade by supplying the `--license-file` parameter. If you do not supply a value for this parameter, the existing license will be reused.

1. Execute the upgrade script.

```
bash upgrade-processing-server.sh
```

8 Migrating Data from STK Terrain Server

In order to migrate data from STK Terrain Server to GCS, you must have access to the Terrain Server's local file system. The existing terrain data is stored in the db folder under the existing STK-terrain install location.

For each terraindb file to be imported:

1. Copy the file onto your machine.
2. Click the "Add a new asset" () button on the upper right of the user interface to create a new Terrain asset.
3. Enter a name for the new asset. The **Cesium Terrain Database** file type should be selected.
4. Upload the terraindb file. When the upload completes, click the **Finalize Asset** button to perform the import.
5. Click the **Globe View** button to make sure the terrain looks correct.
6. If you are satisfied with the results, click the **Publish** button to make the asset available to others.

9 Managing Roles

9.1 Accessing the AGI Identity Platform Adminstration Console

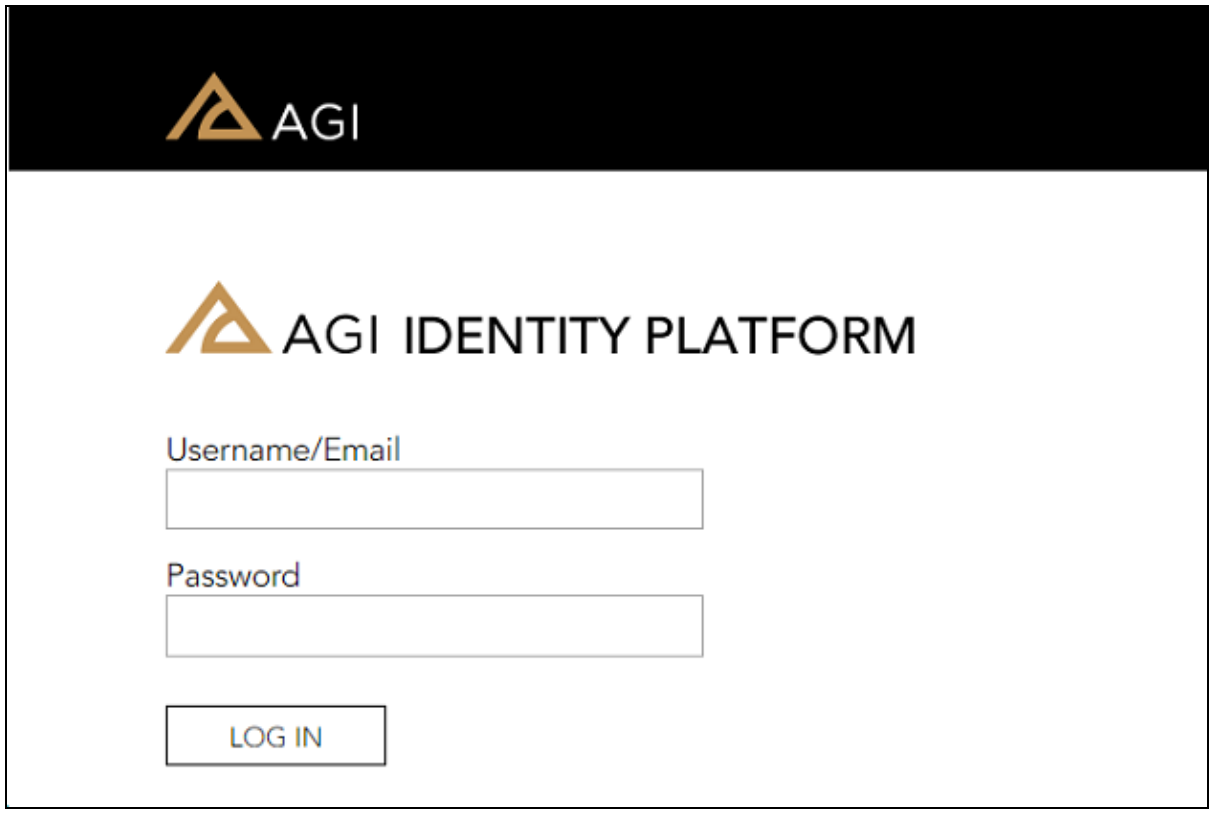
The AGI Identity Platform (AIP) is part of the identity server installation shown in the Architecture diagram in the Overview. Using the AGI Identity Platform Administration console, administrators can manage users and sessions and can configure the mapping to a user federation provider. Use of the console requires password authentication and a web browser that supports HTML5 and JavaScript.

1. Open a web browser and navigate to the following URL, replacing <authority> with your AIP host and port number separated by a colon (e.g. id.yourdomain.com:8443) :

   https://<authority>/auth/admin

2. Enter the username and password specified in the Identity Server Worksheet section. If a username and password were not specified, use the default values:

   - Username: admin
   - Password: admin



Users, attributes, and group members are maintained in the directory server specified during installation. The AGI Identity Platform uses Keycloak to integrate with your directory and authenticate users within GCS. The identity server installer creates a one-way mapping between your directory and its user store. When a user is authenticated, the user's attributes and group membership are copied to the Keycloak user store.

9.2 Mapping Directory Groups to GCS Roles

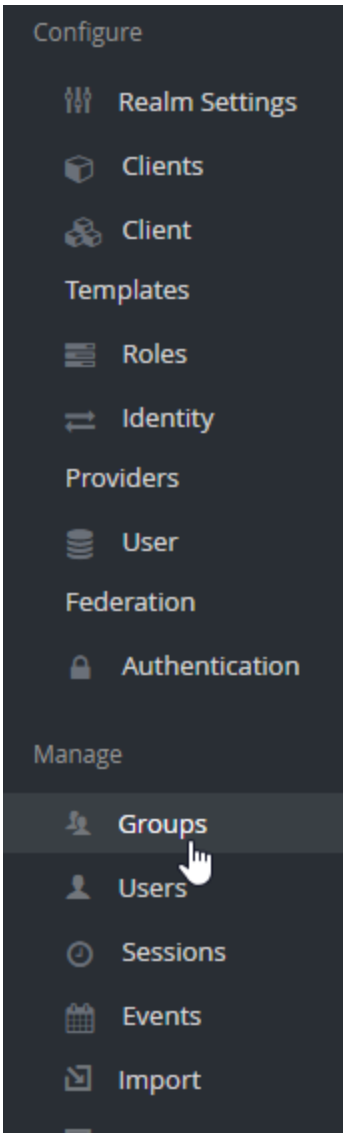
GCS uses role mappings to define a user's access to the system.

Viewer Users	Authenticated users without any specific role mapping are viewer users with read-only access to published assets.

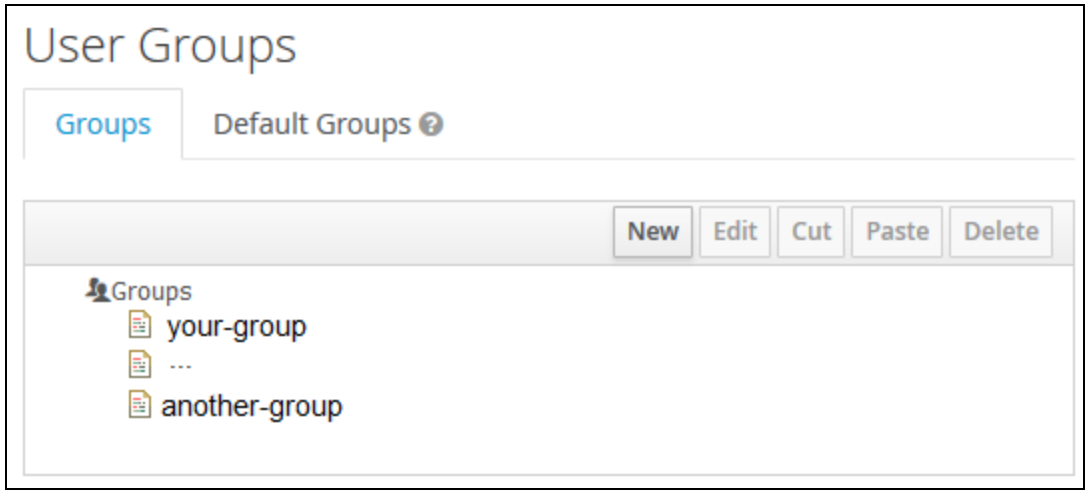
Processing Users	Processing users have the **gcs_processing** role applied either directly to their user account or indirectly through their group membership. Processing users can upload and process assets. After an asset has been processed, processing users can publish the asset to make it visible to other users.
Administrator Users	Administrator users have the **gcs_admin** role applied either directly to their user account or indirectly through their group membership. Administrator users can see published assets as well as an administrative tab with recent log data.

9.3 Mapping a Role to an Existing Group in your Directory

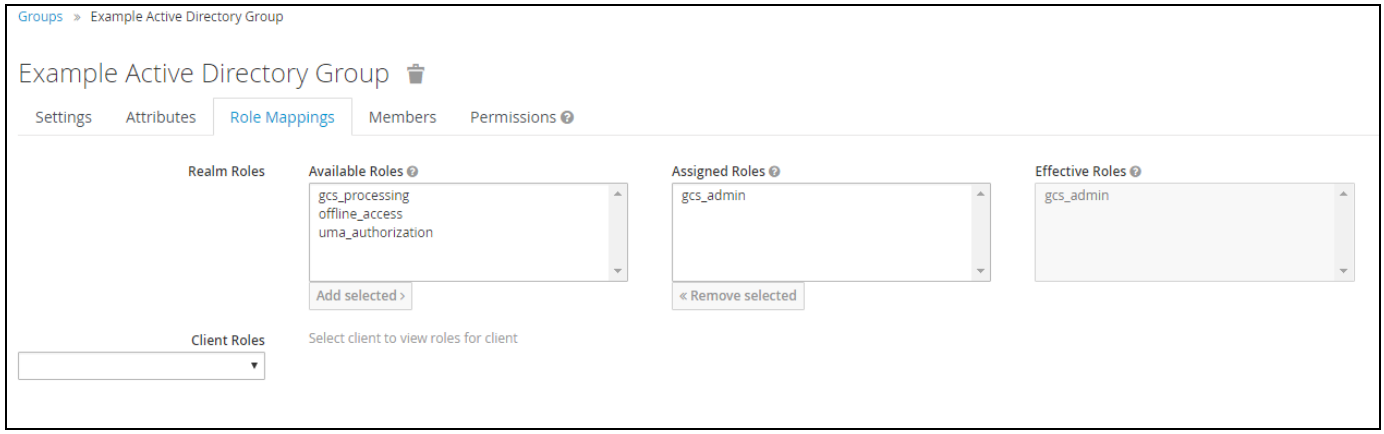
1. Click the **Groups** link on the left menu.



2. Double-click the directory group name to be mapped to a role.



3. Select the desired role (either **gcs_admin** or **gcs_processing**) from the **Available Roles** list and click the Add Selected > button to map that role to this group. The role name moves to the **Assigned Roles** list. Both roles may be selected and grant the rights assigned to each role.



10 Troubleshooting

10.1 Log File Locations

Component	Log File Location
Database Server	/var/log/postgres
Identity Server	/usr/local/keycloak/standalone/log
Application Server	/var/log/tomcat
Processing Server	/var/log/gcs-processing

10.2 Common Issues

Symptom:	*Cannot convert access token to JSON* (accompanied by a HTTP 401 response from the server).
Cause:	The public key from Keycloak does not match the oauth-client.properties file in Tomcat.

Components Affected:	• GCS Application Server • GCS Processing Server

Solution:	Copy the public key from Keycloak. In the Keycloak Adminstration console, select the "agigcs01" realm. Go to Realm Settings > Keys and look for the row with a "Type" of "RSA". In the "Public Keys" column, click "Public Key". Copy this value into each oauth-client.properties file under ~tomcat/webapps (there may be several). Replace the lines between
	`-----BEGIN PUBLIC KEY-----`
	and
	`-----END PUBLIC KEY-----`
	
	EXAMPLE:
	
	`jwt.public.key: -----BEGIN PUBLIC KEY-----\`
	`MIIBojANBgkqhkiG9w0BAQEFAAOCAY8AMIIBigKCAYEA2uw3ZDkqhoc9uVybmkGn\`
	`o67b4dFAzHxcRbZBpcxjwW+8j1822/4ATFf4IvNBydU6efi7LK73v3KcEsTQWloK\`
	`2cbrt+mupiKTuwoP5PlOhFR79SRedOCS0MMRVhFUGchvIy+F9Gy18K/1E+PMzNhc\`
	`k52QNLPm2WuXzwBZgBK2Kr1kzb5JRknvnnifBGbUipoQQBgq5opjRqc8P6lwQS4K\`
	`EP594DKbaZ49eu7kFIb3nBuDWgDEQRKqqlIgPrkPxK1F+aLIYrqley0vxxLFt84Z\`
	`/hkjLzQVFL3SarKH+7UWQqrPA8HzbCPOLdw4oFpkCA/pW3ePQnA2ofRbOSIXFiU5\`
	`fh1svvlmZrRFSrcvPXCZsrII3k8eQFiaTAMkp+TFPayZZEyf3ak0G/ISYzJJYVhg\`
	`uKVosOEroNg+o3zN06tced3m7waXNYkCCXDd92TH3kA+7ULxkVAskE2dQnqeMaOs\`
	`MqSdQRnqHBxhWxWeKN5vyNc1rwWT2TEOwJxIKP3WV4+rAgMBAAE=\`
	`-----END PUBLIC KEY-----`

Symptom:	The GCS Application Server or the GCS Processing Server does not start. *Error calling refresh* appears in the catalina.out log file.
Cause:	The license file is invalid. This may be due to: • An expired license • Windows line endings on a Linux system • No recognized GCS features • An invalid version
Components Affected:	• GCS Application Server • GCS Processing Server

Solution:	• Run dos2unix on the license file located under tomcat/licenses and gcs-processing/licenses and restart tomcat with:
	`systemctl restart tomcat`
	or
	`systemctl restart gcs-processing`
	• Manually inspect the contents of the license files, verifying that they contain valid features, are not expired, and have a major version matching the installed version (2.0)
	• Request a new license file from support@agi.com, if needed

In the event that you encounter a different symptom, or the provided solutions do not fix your problem, contact AGI support.

11 Uninstallation Steps

Once GCS is installed, the uninstall scripts will be located at `/usr/local/gcs/uninstall`. Run the following commands from that location.

1. Uninstall the Processing Server:

   ```
   bash uninstall-processing-server.sh
   ```

   Optional flags:

   ```
   --keep-logs
   ```

   Prevents processing server logs from being removed.

2. Uninstall the Application Server:

   ```
   bash uninstall-app-server.sh
   ```

   Optional flags:

   ```
   --keep-logs
   ```

   Prevents application server logs from being removed.

   ```
   --remove-assets
   ```

   By default, asset data is not removed. Passing this option forces the removal of assets.

3. Uninstall the Identity Server:

   ```
   bash uninstall-id-server.sh
   ```

4. Uninstall the Database Server:

   ```
   bash uninstall-db-server.sh
   ```

   Optional flags:

   ```
   --keep-logs
   ```

   Prevents database server logs from being removed.

`--remove-database`

By default, the database is not removed. Passing this option forces the removal of the database.

`--db-backup-dir=PATH`

If a custom backup location was specified during installation, this option must be specified to delete that backup location.

A Installation Worksheets (Distributed)

> **Note:** Unless otherwise specified, all parameters listed below are required for installation.

This section provides worksheets for collecting the information needed to install the software components as well as information that you will finalize as you perform the installations.

The tables below show the parameters required in the installer scripts along with a description of the parameters (the 'Parameter : Description' column), the example value used in the guide (the 'Example Value' column), and a space for you to enter your value (the 'Your Value' column).

> **Note:** As shown in the Installation Steps (Distributed), the values in the 'Example Value' and the 'Your Value' columns should be surrounded by single quote characters when supplied to commands.

A.1 Common Parameters Worksheet

The following common parameters must be provided to each installation script:

Parameter : Description	Example Value	Your Value
--ssl-ca-file : PEM-formatted certificate authority file	customer-ca.crt	
--ssl-cert-file : PEM-formatted server certificate file	customer-server.crt	
--ssl-key-file : Unencrypted PEM RSA server certificate key file	customer-server.key	

A.2 Database Server Worksheet

Parameter : Description	Example Value	Your Value
--ip-network-mask : An IPv4 address range formatted in CIDR notation to determine which client hosts may attempt to connect to the database server. This parameter may be passed multiple times.	10.0.0.0/16	

A.2.1 Optional

Parameter : Description	Example Value	Default Value	Your Value
--install-app-db : Installs the application database. Multiple 'install-*-db' flags may be provided. If no 'install-*-db' flag is provided, all databases will be installed.			
--install-id-db :Installs the identity database. Multiple 'install-*-db' flags may be provided. If no 'install-*-db' flag is provided, all databases will be installed.			
--db-server-port : The port this database server will listen to for connections.	5432	5432	

A.3 Application Server Worksheet

Parameter : Description	Example Value	Default	Your Value
--db-server : The fully qualified domain name of the application's database server.	db.example.com	*hostname*	
--id-server : The fully qualified domain name of the identity server.	id.example.com	*hostname*	
--processing-server : The fully qualified domain name of the asset processing server.	processing.example.com	*hostname*	

A.3.1 Optional

Parameter : Description	Example Value	Default	Your Value
--ssl-protocols : The names of the protocols to support when communicating with clients. Supported protocol strings are defined by Apache (https://tomcat.apache.org/tomcat-9.0-doc/config/http.html).	TLSv1.2,TLSv1.3	TLSv1.2	

Parameter : Description	Example Value	Default	Your Value
--app-server-client-auth-port : The port that listens for certificate authentication requests from the processing server.	9443	9443	
--db-server-port : The port number for the database.	5432	5432	
--id-server-port : The identity server port.	443	443	
--processing-server-port : The processing server port.	443	443	

A.4 Processing Server Worksheet

Parameter : Description	Example Value	Default Value	Your Value
--app-server : The fully-qualified domain name of the application server on your internal network.	app.example.com	*hostname*	
--db-server : The fully qualified domain name of the application's database server.	db.example.com	*hostname*	

A.4.1 Optional

Parameter : Description	Example Value	Default	Your Value
--app-server-client-auth-port : The port number to use for certificate authentication with the app server.	9443	9443	
--app-server-external : The user-facing fully-qualified domain name of the application server if different from the internal name.	public-gcs.example.com	hostname	
--app-server-port : The port number for the application server.	443	443	
--db-server-port : The port number for the database server.	5432	5432	
--proxy-port : The port number to be used by the internal proxy. This proxy encrypts communications between the local host (processing server) and the application server.	9000	9000	
--ssl-protocols : The names of the protocols to support when communicating with clients. Supported protocol strings are defined by Apache (https://tomcat.apache.org/tomcat-9.0-doc/config/http.html).	TLSv1.2,TLSv1.3	TLSv1.2	

A.5 Identity Server Worksheet

This document outlines the installation and configuration of an identity server using directory-based authentication (e.g., LDAP). Please contact AGI support if you do not have a directory server available.

> **Note:** Use the --keycloak-admin-user and --keycloak-admin-pass flags to change the username and password assigned to the administrative user for Keycloak. If you do not specify these flags, the default value of 'admin' will be used for both the username and password. After installation, you may change the password by logging into the AGI Identity Platform Administration Console.

> **Note:** The --keycloak-admin-ip-addresses parameter provides additional security and is necessary if you must comply with Security Technical Implementation Guidelines (STIG) requirements.

Parameter : Description	Example Value	Your Value
--gcs-admin-group : Common name (CN) of an LDAP group that has admin access to GCS	gcs_administrators	
--gcs-processing-group : Common name (CN) of an LDAP group that has processing access in GCS	gcs_processing	
--ldap-groups-dn : Distinguished name (DN) of a node where groups are defined in LDAP	ou=Groups,dc=example,dc=com	

Parameter : Description	Example Value	Your Value
--ldap-bind-dn : Bind distinguished name (DN) for LDAP	CN=LDAPUser,CN=Users,DC=example,DC=com	
--ldap-users-dn : Distinguished name (DN) where users are defined in LDAP	CN=users,DC=example,DC=com	
--ldap-bind-credentials : Password for the LDAP server	credentials123	
--ldap-connect-url : URL where the LDAP server is located	ldap://machineName.com:1234	
--keycloak-admin-ip-addresses : An IPv4 address range formatted in CIDR notation to determine which client hosts may access the Identity server admin console	192.168.0.0/24	

Parameter : Description	Example Value	Your Value
--keycloak-admin-user : Username for the Keycloak administrative user	systemadmin	
--keycloak-admin-pass : Password for the Keycloak administrative user	Passw0rd123!	
--app-server : The fully-qualified domain name of the application server on your internal network	app.example.com	
--id-db-server : The Fully Qualified Domain Name for the identity database server when installing on a distributed system	db.example.com	

A.5.1 Optional

Parameter : Description	Example Value	Default	Your Value
--app-server-client-auth-port : The port that the application server uses to accept certificate-authenticated requests from the processing server	9443	9443	
--id-db-server-port : The port number for the database server hosting the identity database.	5432	5432	

B Installation Steps (Distributed)

For a new installation, follow the steps outlined below, being sure to run the commands in the order shown. If you are upgrading an existing installation, please see Upgrade Steps.

When specifying parameter values, the equals operator must be included as shown in the examples. In this section, all values in green should be replaced with your values from the Installation Worksheets (Distributed) section.

> **Note:** All file locations should be specified as absolute paths.

1. Install the Database Server.

```
bash install-db-server.sh \
    --ssl-ca-file='/tmp/installer/certificates/customer-ca.crt' \
    --ssl-cert-file='/tmp/installer/certificates/db.yourdomain.com.crt' \
    --ssl-key-file='/tmp/installer/certificates/db.yourdomain.com.key' \
    --ip-network-mask='10.0.0.0/16'
```

2. Install the Identity Server.

```
bash install-id-server.sh \
    --ssl-ca-file='/tmp/installer/certificates/customer-ca.crt' \
    --ssl-cert-file='/tmp/installer/certificates/id.yourdomain.com.crt' \
    --ssl-key-file='/tmp/installer/certificates/id.yourdomain.com.key' \
    --gcs-admin-group='gcs_administrators' \
    --gcs-processing-group='gcs_processing' \
    --ldap-groups-dn='ou=Groups,dc=yourdomain,dc=com' \
    --ldap-bind-dn='CN=LDAPUser,CN=users,DC=yourdomain,DC=com' \
    --ldap-users-dn='CN=users,DC=yourdomain,DC=com' \
    --ldap-bind-credentials='credentials123' \
    --license-file='yourLicenseFile.lic' \
    --id-db-server='db.yourdomain.com' \
    --app-server='app.yourdomain.com'
```

3. Install the Application Server.

```
bash install-app-server.sh \
    --ssl-ca-file='/tmp/installer/certificates/customer-ca.crt' \
    --ssl-cert-file='/tmp/installer/certificates/app.yourdomain.com.crt' \
    --ssl-key-file='/tmp/installer/certificates/app.yourdomain.com.key' \
    --license-file='yourLicenseFile.lic' \
    --id-server='id.yourdomain.com' \
    --processing-server='processing.yourdomain.com' \
    --db-server='db.yourdomain.com'
```

4. Install the Processing Server.

> **Note:** You must install the processing server even if you have a serving-only license.

```
bash install-processing-server.sh \
    --ssl-ca-file='/tmp/installer/certificates/customer-ca.crt' \
    --ssl-cert-file='/tmp/installer/certificates/processing.yourdomain.com.crt' \
    --ssl-key-file='/tmp/installer/certificates/processing.yourdomain.com.key' \
    --license-file='yourLicenseFile.lic' \
    --app-server='app.yourdomain.com' \
    --db-server='db.yourdomain.com'
```