



GCS

Geospatial Content Server

Geospatial Content Server System Administration Guide

Release 2.10.0, November 2023

© 2023 Copyright ANSYS, Inc. All Rights Reserved.

Table of Contents

1 Overview	1
1.1 Architecture	1
1.1.1 GCS System Components	1
1.1.2 Single-Machine Deployment	3
1.1.3 Distributed Deployment	4
2 Requirements	6
2.1 Operating System	6
2.2 Hardware	6
2.2.1 Data Storage Requirements	6
2.2.2 Single-Machine Hardware Requirements	6
2.2.3 Distributed Hardware Requirements	7
2.2.4 Additional Considerations	7
2.3 Package Management	8
2.3.1 Package Repositories	8
2.3.2 Manual Package Install	9
2.4 SSL/TLS Certificates	10
2.4.3 Certificate Format Requirements	10
2.4.4 Single-Machine Deployment Certificates	10
2.4.5 Distributed Deployment Certificates	11
2.5 License Server	11

2.5.1 Windows Install	12
2.5.2 Linux Install	13
2.5.3 Managing the License Server on Linux	13
3 Installation Steps (Single-Machine)	15
4 Installation Steps (Distributed)	16
4.1 Installing Distributed GCS Servers	16
4.2 Installing Additional Terrain Analysis Servers	19
5 Getting Started	21
5.1 Configure GCS Users	21
5.1.1 Create GCS Test Users	21
5.1.2 Import Existing Users	22
5.2 Sign In to GCS	22
6 Managing Users and Roles	23
6.1 Accessing the AGI Identity Platform Administration Console	23
6.2 User Federation Integration	25
6.3 GCS User Roles	27
6.4 Mapping a Role to an Existing Group in your Directory	28
7 Upgrade Steps	30
7.1 GCS 2.9.0 to GCS 2.10.0 Migration Overview	30
7.2 GCS Configuration Report	31
7.3 Upgrade Steps (Single-Machine)	32
GCS 2.9.0 Migration Data Export and Configuration Report	32

GCS 2.10.0 Install and Migration Data Import	33
7.4 Upgrade Steps (Distributed)	34
GCS 2.9.0 Migration Data Export and Configuration Report	34
GCS 2.10.0 Install and Migration Data Import	35
8 Uninstallation Steps	38
8.1 Single Machine Uninstall	38
8.2 Distributed Environment Uninstall	38
9 Storage	41
9.1 Shared Locations	41
9.2 Server Locations	41
10 System Configuration	43
10.1 Terrain Analysis Configuration	43
10.1.1 Terrain Analysis Configuration in the UI	45
10.1.2 Terrain Analysis Tolerance	46
11 Maintenance	47
11.1 Get System Information	47
11.2 Updating GCS-Generated Certificates	47
11.2.1 Single-Machine Deployment	47
11.2.2 Distributed Deployment	48
11.3 Updating Your Certificates	49
11.4 Updating the Ansys License Manager Host	50
11.4.1 Single-Machine Deployment	50

11.4.2 Distributed Deployment	50
11.5 Service Maintenance	51
12 Migrating Data from STK Terrain Server	52
13 Earth Imagery	53
13.1 High-Resolution Imagery	53
14 Troubleshooting	54
14.1 System Administrator Log File Locations	54
14.2 Server Log File Locations	54
14.3 UI Issues	55
14.3.1 HTTP 401 - Cannot convert access token to JSON	55
14.4 Processing Issues	56
14.4.1 CityGML processing fails with "Failed to layer.json"	56
14.5 Performance Issues	57
14.5.1 Database Server Performance Issues	57
14.5.2 Terrain Analysis Server Performance Issues	60
A Release Notes	64
2.10.0	64
2.9.0	65
2.8.0	65
2.7.1	66
2.7.0	67
2.6.0	67

2.5.1	67
2.5.0	68
2.4.0	68
2.3.0	70
2.2.1	70
2.2.0	70
2.1.1	70
B Glossary	71
C Configuration Parameters	72
C.1 Required Parameters	72
C.1.1 All Deployments	72
C.1.2 Single-Machine Deployment	73
C.1.3 Distributed Deployment	74
C.2 Optional Parameters	75
C.2.1 Database Server	76
C.2.2 Identity Server	78
C.2.3 Application Server	78
C.2.4 Processing Server	79
C.2.5 Terrain Analysis Server	79
C.2.6 Imagery Server	80
C.2.7 Proxy Server	81
D Sideloaded Assets	82

E Data Migration	84
Source Environment Migration Data Export	84
Target Environment Migration Data Import	85

1 Overview

The Geospatial Content Server (GCS) provides a comprehensive enterprise solution for hosting, processing, serving, and analyzing terrain, imagery, and other heterogeneous 3D data, which includes:

- High resolution imagery, terrain, and 3D model files
- Support for STK and CesiumJS-based applications

This document details how to install GCS, including recommendations for deployment types based on your needs. It also contains maintenance and troubleshooting information. Please contact [AGI support](#) with any issues not addressed by this guide.

If you are upgrading an existing GCS installation, skip to [Upgrade Steps](#).

1.1 Architecture

The GCS system comprises multiple specialized server processes, hereafter referred to as components. There are two types of GCS deployments:

- **Single-Machine Deployments**, where all components of GCS are installed on the same physical or virtual server.
- **Distributed Deployments**, where components of GCS are installed across more than one physical or virtual server.

1.1.1 GCS System Components

Database server

- Stores data associated with stored assets in a PostgreSQL database
- Hosts binary terrain data as individual SQLite database files (with .terraindb extensions)

Identity server

- Responsible for providing user identities to the system
- Integrates with Active Directory/LDAP servers, or can store user accounts locally

Application server

- Hosts and serves the API and UI portions of the application
- Hosts and serves binary data for 3D models, imagery, and vector assets
- Serves binary data for terrain assets

Processing server

- Converts asset data to streaming-optimized formats, which are then hosted by the application server

Processing proxy server

- Installed alongside the processing server, this local proxy enables the processing server to request terrain data from the application server

Imagery server

- Hosts Sentinel-2 Earth imagery (provided with GCS data disk)

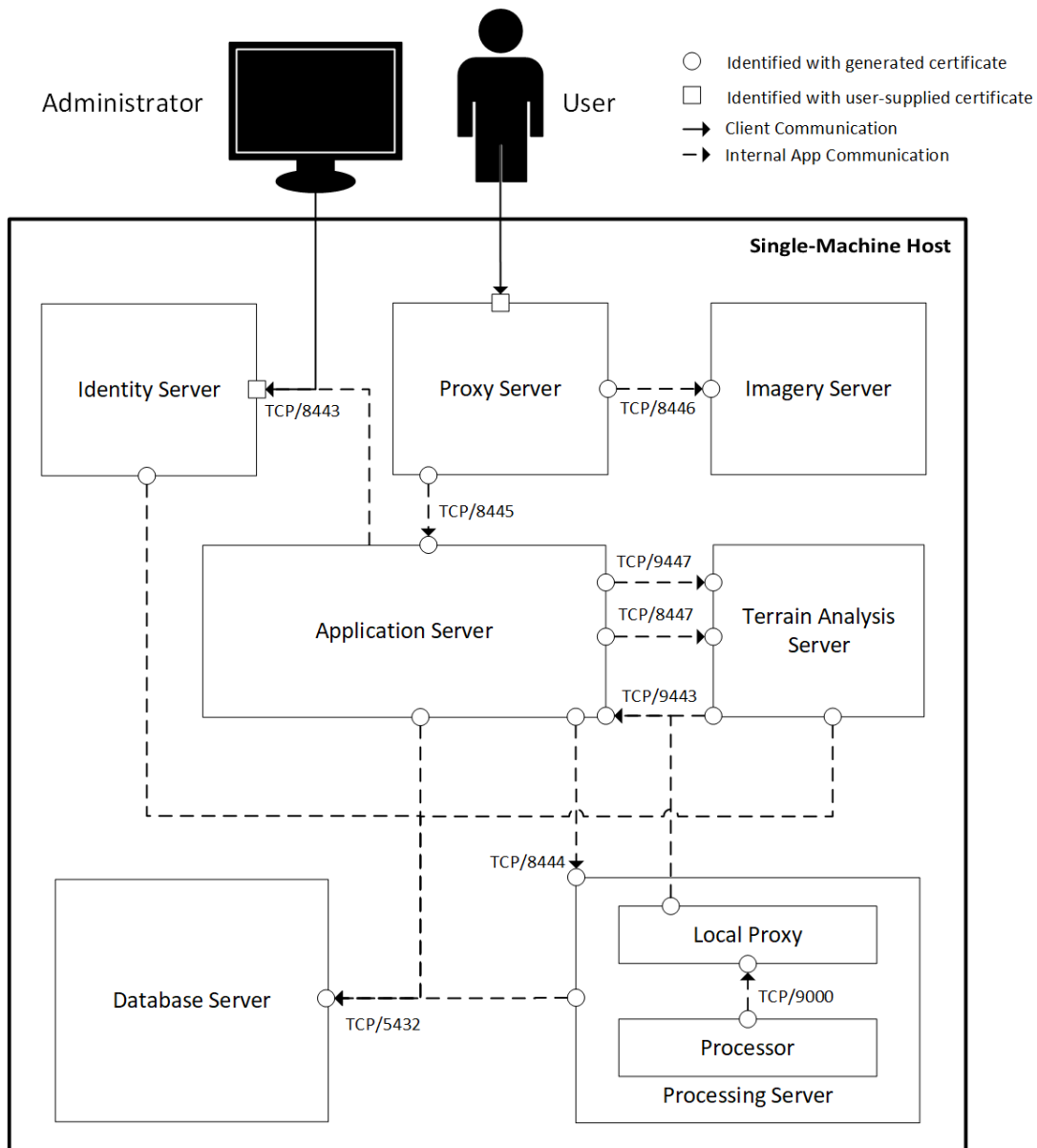
Proxy server

- Reverse proxy server through which end users access the system

1.1.2 Single-Machine Deployment

A single-machine deployment is easier to set up than a distributed deployment but limits your ability to scale up as load on the system increases. This deployment type is recommended for **non-production environments**. For production environments, we recommend a distributed deployment.

The diagram below depicts a single-machine deployment. The outer box represents the server hosting all the GCS components. The inner boxes represent each GCS component. All communications between GCS components are secured via TLS.



1.1.3 Distributed Deployment

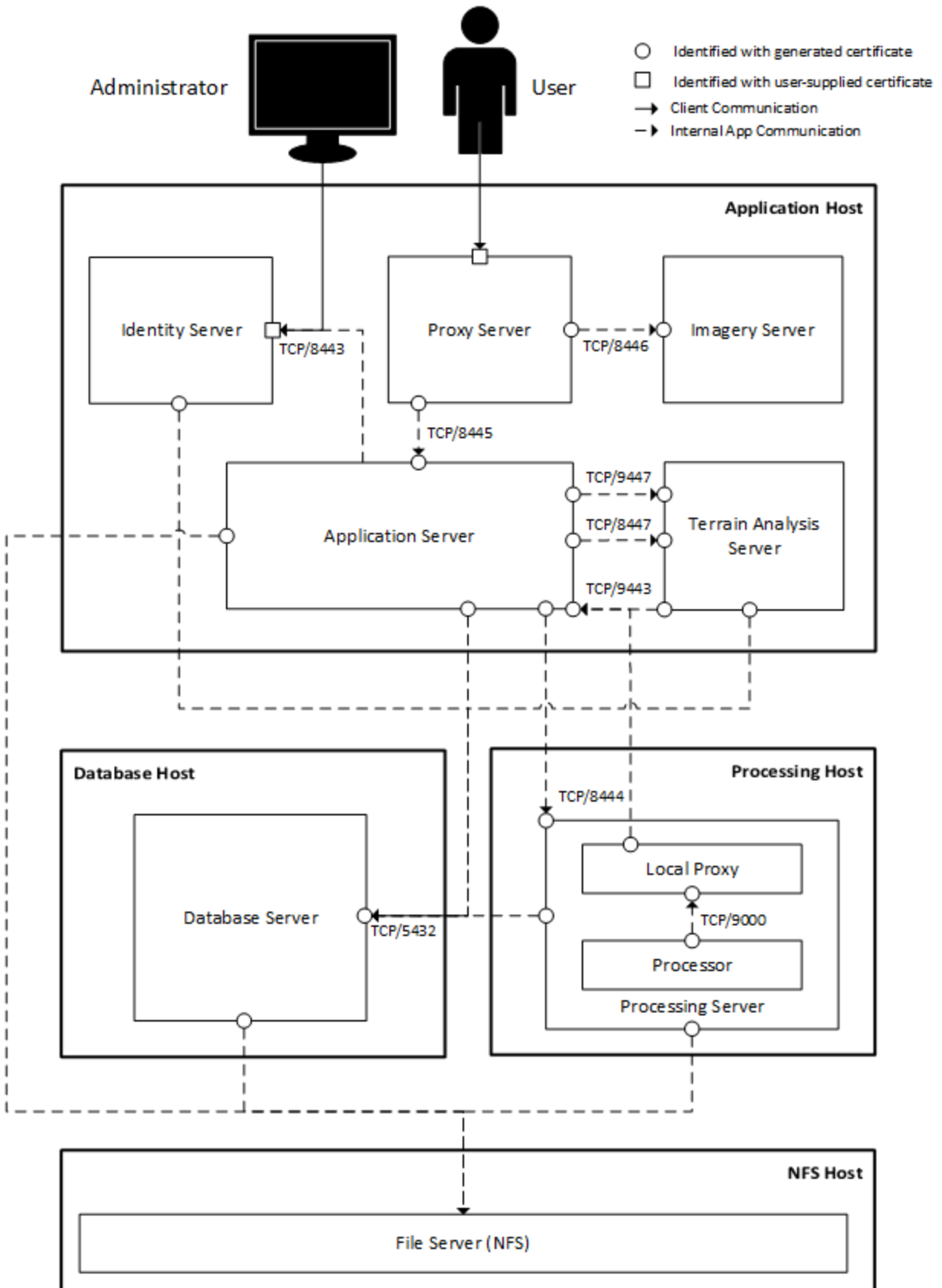
In this type of installation, not all GCS components are installed on the same virtual or physical machine. Distributed installs are intended for **production environments**, since it is easier to scale up individual servers as load increases. This deployment requires setting up a shared file server, since the application, processing, and database servers share file locations (see [Storage](#) for more details). You should also review the "Distributed Hardware Requirements" section of the [Requirements](#) page prior to installing.

Distributed deployments should distribute GCS components among machines in the following way:

- A dedicated host machine for the File Server (NFS) component
- A dedicated host machine for the Database Server component
- A dedicated host machine for the Processing Server component
- A dedicated host machine for the Application Server, Terrain Analysis Server, Identity Server, Imagery Server, and Proxy Server components

The diagram below depicts a distributed deployment. The outer boxes labeled with **Host** indicate different physical or virtual machines. The inner boxes represent each GCS component installed on that host. All communications between GCS components are secured via TLS.

Geospatial Content Server



2 Requirements

2.1 Operating System

This installation supports 64-bit Enterprise Linux 8 and has been tested using the Rocky Linux 8 distribution. Some installation instructions and examples assume the use of Rocky Linux 8, but other Enterprise Linux 8-compatible distributions should work.

2.2 Hardware

2.2.1 Data Storage Requirements

GCS includes a Sentinel-2 world imagery dataset (660GB) and a world terrain dataset (765GB). Due to the size of these datasets, we suggest reserving approximately 1.5 TB of storage in addition to the space required for other assets that will be uploaded. The storage requirements listed in the sections below account for these dataset sizes. If you do not intend to use these datasets (for example, in the case of a test install or deployment to a non-production environment) you may reduce the storage requirements appropriately.

2.2.2 Single-Machine Hardware Requirements

The following table details the hardware required for a single-machine deployment of GCS. This deployment is for **non-production environments** only.

Host	CPUs	RAM	Storage Space	Storage Device
Single GCS Server	8	16 GB	2 TB	HDD

2.2.3 Distributed Hardware Requirements

The following table details the hardware required for a distributed deployment of GCS. This deployment type is recommended for **production environments**.

Host	CPUs	RAM	Storage Space	Storage Device
NFS Host	2	4 GB	2 TB	SSD
Database Host	8	16 GB	128 GB	HDD
Processing Host	2	4 GB	128 GB	HDD
Application Host	8	16 GB	128 GB	HDD
Additional Terrain Analysis Hosts *	8	16 GB	128 GB	HDD

* Terrain Analysis Servers can be scaled out on Additional Terrain Analysis Hosts to increase the computational throughput of the GCS Terrain Analysis services. See the "Installing Additional Terrain Analysis Servers" section of the [Installation Steps \(Distributed\)](#) steps.

2.2.4 Additional Considerations

Depending on your intended use of GCS, your hardware needs may differ from specifications defined above. Consider the following use cases:

- Using GCS to host large assets.
 - In this scenario, you may need to increase storage space on your NFS Host.
- Using GCS to frequently process new assets.
 - In this scenario, you may need to increase the RAM and number of CPU cores on your Processing Host.
- Using GCS to frequently stream tiles for visualization through a browser.
 - In this scenario, you may need to increase the RAM on your Application Host and Database Host.


- Using GCS to frequently do terrain analysis on large or high-resolution terrain assets.
 - In this scenario, you may need to increase the RAM on your Database Host or add additional CPUs on your Application Host.

2.3 Package Management

To ensure compatibility between our software and your operating system, we install some packages using RPM and DNF. Aside from ensuring binary compatibility among third-party libraries, using RPM packages enables you to patch vulnerabilities in these libraries immediately, without requiring a new version of GCS.

2.3.1 Package Repositories

Packages used by GCS come from the repositories listed below.

 **Note:** The AppStream and BaseOS repositories should already be configured. The URLs for the Rocky Linux AppStream and BaseOS repositories are shown for reference.

Repo Name	Repo ID	URL
AppStream	appstream	Distribution dependent Rocky Linux: https://dl.rockylinux.org/pub/rocky/8/AppStream/x86_64/os/
BaseOS	baseos	Distribution dependent Rocky Linux: https://dl.rockylinux.org/pub/rocky/8/BaseOS/x86_64/os/
EPEL	epel-release	https://dl.fedoraproject.org/pub/epel/8/Everything/x86_64/

PostgreSQL 15	N/A	https://download.postgresql.org/pub/repos/yum/15/redhat/rhel-8-x86_64/
------------------	-----	---

For information on configuring these package repositories, please see:

- <https://wiki.rockylinux.org/rocky/repo/>
- https://docs.fedoraproject.org/en-US/epel/#_quickstart
- <https://www.postgresql.org/download/linux/redhat/>

2.3.2 Manual Package Install

⚠ Important: Manually installing RPMs is an advanced use case. It makes patching more difficult and time-consuming. If possible, configure package repositories as listed in the section above.

If for policy or technical reasons you are unable to configure repositories as described above, you may have to manually install some packages before installing GCS. The list of packages directly used by GCS are listed below.

bash	iputils	postgresql15-contrib
bind-utils	java-11-openjdk	postgresql15-libs
curl	java-17-openjdk	postgresql15-server
dnf	jq	procps-ng
fcgi	libatomic	rpm
findutils	libcap	sed
firewalld	libxslt	shadow-utils
gawk	mesa-libGLU	sqlite_fdw_15
glibc	mod_ssl	systemd
glibc-common	openssl	tar

grep	pgaudit17_15	tomcat-native
haveged	policycoreutils	tzdata-java
httpd	policycoreutils-python-utils	unzip
iproute	postgresql15	zip

2.4 SSL/TLS Certificates

To establish proper trust on users' browsers, the proxy and the identity server components must use SSL certificates signed by a trusted certificate authority. Provide the certificate and private key when installing the proxy and the identity server components. Additionally, you must provide the certificate chain to the trusted root certificate authority when installing each individual component (database, identity server, et al). In a single-machine deployment, only one set of SSL certificates needs to be provided. See the [Architecture](#) section for a graphical representation.

All certificates for securing server-to-server communications are generated by the GCS installation scripts – you do not need to provide them.

2.4.3 Certificate Format Requirements

- Certificate Authority chain file must be in PEM format.
- Server certificate(s) must be in PEM format.
- Server certificate(s) must contain one of the machine's DNS names in the Subject or Subject Alternative Names (SAN).
- Key file(s) must be in unencrypted PEM RSA format.

2.4.4 Single-Machine Deployment Certificates

A single-machine deployment requires that you provide one server certificate, its key, and the certificate authority chain file that signed it. The list of files below are only examples. Your file names may differ.

- certificateAuthority.crt
- gcs-server.crt
- gcs-server.key

2.4.5 Distributed Deployment Certificates

A distributed deployment requires that you provide a certificate and key for the Identity server, a certificate and key for the Proxy servers, and the Certificate Authority chain file that signed them. The list of files below are only examples. Your file names may differ.

- certificateAuthority.crt
- gcs-identity-server.crt
- gcs-identity-server.key
- gcs-proxy-server.crt
- gcs-proxy-server.key

2.5 License Server

To run GCS, a license server is required. The Ansys License Manager, powered by FlexNet Manager, is required to serve your license locally or from a network. If you already have one set up, you can skip this section. If not, follow the instructions below. Note that the license server can be installed on either Windows or Enterprise Linux 8.

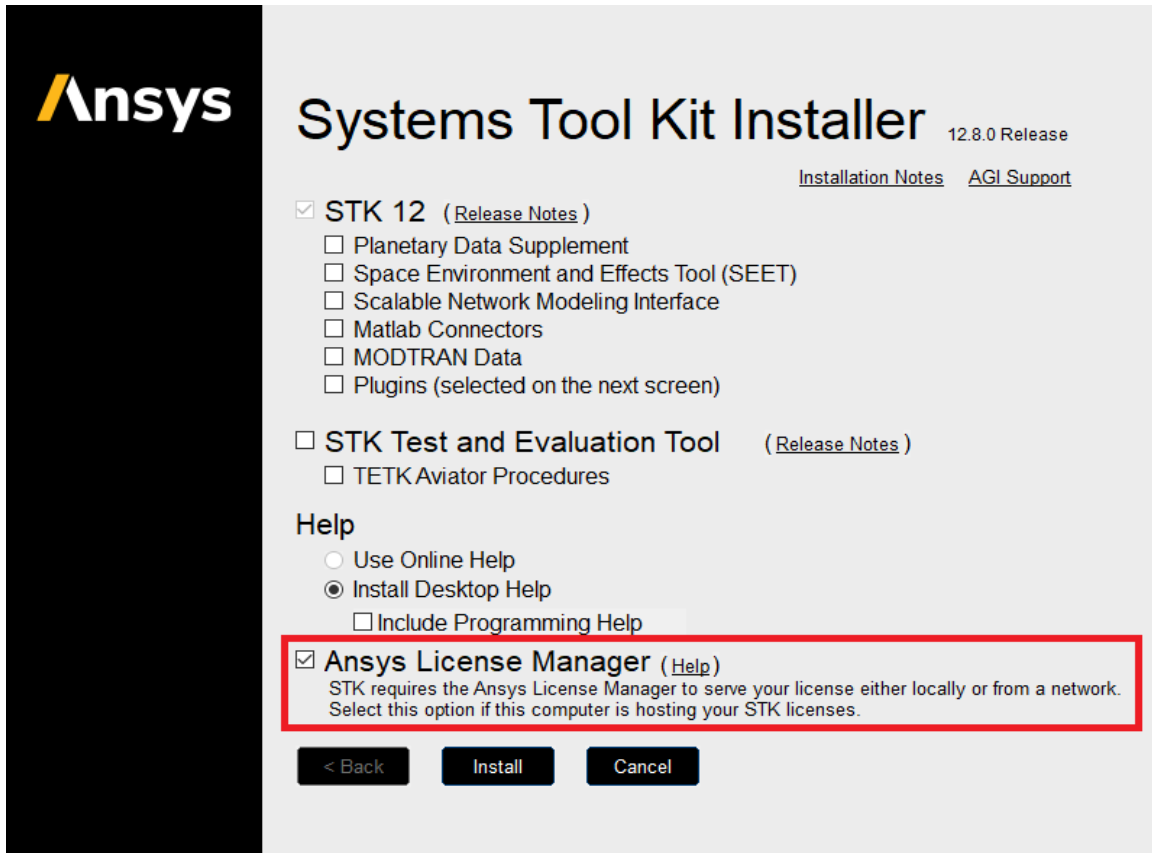
 **Reference:** You can find additional license server installation options and troubleshooting information in the Installation and Licensing Guide at [license-server/util/installHelp.pdf](#).

Note the following requirements:


- The Ansys License Manager requires that port 1055 is open. Ensure it is not in use and is exposed through any firewalls before installing the license server.

2.5.1 Windows Install

1. Run the Systems Tool Kit (STK) Installer.
2. Select the **Ansys License Manager** check box.



3. Click **Install**.
4. When the Ansys Installation Launcher appears, select **Install Ansys License Manager** to complete the installation.

 **Note:** For details on configuring your client-side license manager settings, select the **Getting Started - Licensing** PDF at the bottom of the Ansys Installation Launcher window.

2.5.2 Linux Install

1. Extract the GCS installer onto `/usr/local/gcs` on the intended License server machine and navigate to the **license-server** directory.

```
mkdir /usr/local/gcs

tar -xzf geospatial-content-installer-2.10.0-linux64.tgz -C /usr/local/gcs

cd /usr/local/gcs/geospatial-content-installer-2.10.0/license-server
```

2. Ensure that port 1055 is available and not blocked by firewall settings.
3. Ensure that the Linux Standards Base package, `redhat-lsb`, is installed. You can check this by running:

```
dnf list installed redhat-lsb
```

If it is not installed, you can install it by running:

```
dnf install redhat-lsb
```

4. Install the server.



Note: Additional prerequisites may be necessary to install the Ansys License Manager. Follow the instructions of the INSTALL output as needed. This command can also be used to upgrade an existing Ansys License Manager on Linux.

```
bash INSTALL -LM -silent
```

2.5.3 Managing the License Server on Linux

In order to manage the license server, you will need a way to launch a GUI on the license server. There are several ways to do this, including X11, VNC, and remote desktop using `xrdp`. The following instructions describe how to use X11 to do this.

SETTING UP X11 ON THE SERVER

1. Install the `xorg-x11-xauth` RPM:

```
dnf install xorg-x11-xauth
```

2. Check `/etc/ssh/sshd_config` to make sure the following setting is present and not commented out:

```
X11Forwarding yes
```

3. If you needed to modify `/etc/ssh/sshd_config`, run:

```
systemctl restart sshd
```

4. Make sure a web browser is installed. In this example, we will use Firefox. If Firefox is not already installed, you can install it as an RPM:

```
dnf install firefox
```

ACCESSING THE LICENSE SERVER

1. Install the following software on your local machine:

- An X11 server, such as Xming
- An SSH client, such as PuTTY

2. Start the X11 server on your desktop machine. If you are using Xming on Windows, you should see an "X" icon in the task bar after it starts.

3. Use SSH with X11 forwarding enabled to access the license server. Replace `user` with your username and `hostname` with the hostname of the license server.

```
ssh -X user@hostname
```

4. In your SSH session, open the web browser to access the license server management UI:

```
firefox http://localhost:1084
```

3 Installation Steps (Single-Machine)

For a new single-machine installation, follow all of the steps below. Be sure to run the commands in the order shown. If you are installing a distributed deployment, refer to [Installation Steps \(Distributed\)](#). If you are upgrading an existing installation, please see [Upgrade Steps](#).

Note:

- All scripts must be executed as a privileged user (conventionally named "root").
- All file locations should be specified as absolute paths.
- The installer relies on the `hostname -f` command to return a fully-qualified domain name.

1. Extract the GCS installer into `/usr/local/gcs` and navigate to the installer directory.

```
mkdir /usr/local/gcs  
tar -xzf geospatial-content-installer-2.10.0-linux64.tgz -C /usr/local/gcs  
cd /usr/local/gcs/geospatial-content-installer-2.10.0
```

2. Create a `conf/gcs.cfg` configuration file.

```
bash utils/create-config-file.sh --type='single'
```

3. Edit the `conf/gcs.cfg` file to define your GCS configuration. The configuration file allows you to customize your GCS installation. See [Configuration Parameters](#) for a full list of required and optional parameters.

4. Validate the configuration.

```
bash utils/validate-configuration.sh
```

This script checks for common configuration errors and outputs the results of each check. Some checks include:

- All required parameters are present
- Provided certificates are valid
- None of the specified ports conflict


5. Install GCS.

```
bash single-machine-install.sh
```

4 Installation Steps (Distributed)

4.1 Installing Distributed GCS Servers

Follow these steps to install a distributed deployment of GCS. Before starting, review the "Distributed Deployment" architecture in the [Overview](#) section. If you are upgrading an existing installation, please see [Upgrade Steps](#).

 **Note:**

- All scripts must be executed as a privileged user (conventionally named "root").
- All file locations should be specified as absolute paths.
- The installer relies on the `hostname -f` command to return a fully-qualified domain name on each machine in your deployment.

1. Extract the GCS installer into `/usr/local/gcs` on each host in your GCS deployment and navigate to the installer directory.

```
mkdir /usr/local/gcs  
  
tar -xzf geospatial-content-installer-2.10.0-linux64.tgz -C /usr/local/gcs  
  
cd /usr/local/gcs/geospatial-content-installer-2.10.0
```

2. On the Database Host, create a `conf/gcs.cfg` configuration file.

```
bash utils/create-config-file.sh --type='distributed'
```

3. On the Database Host, edit the `conf/gcs.cfg` file to define your GCS configuration. The configuration file enables you to customize your GCS installation. See [Configuration Parameters](#) for a full list of required and optional parameters.

4. On the Database Host, validate the configuration file.

```
bash utils/validate-configuration.sh
```

This script checks for common configuration errors and outputs the results of each check. Some checks include:

- All required parameters are present
- Provided certificates are valid
- None of the specified ports conflict

5. On the Database Host, generate certificates used for securing server-to-server communications. The following script will place these certificates in `/usr/local/gcs/share/ssl`.

```
bash generate-certificates.sh
```

6. Copy the `conf` folder from the Database Host installation root into the installation root of each host in your distributed deployment.
7. Copy the `/usr/local/gcs/share/ssl` folder from the Database Host onto each host in your distributed deployment.
8. Mount a network share that the Database, Application, and Processing Hosts can access. Components of these hosts need to share files under the `/var/lib/gcs` directory. There is no specific configuration required as long as all three hosts have access to the same directory. The following steps highlight one way to set up the shared file server using a Network File System (NFS).

a. Execute the following steps on the NFS Host machine that will host the files.

i. Configure the host to allow NFS traffic.

```
firewall-cmd --permanent --zone=public --add-service=nfs
firewall-cmd --reload
```

ii. Install and start the NFS server.

```
dnf install nfs-utils
systemctl enable --now nfs-server
```

iii. Create the shared location.

```
mkdir -p /export/gcs
```

iv. Create a `/etc/exports` file. Add the following content and replace `DB_SERVER`, `APP_SERVER`, and `PROC_SERVER` with each machine's DNS or IP address.

```
/export/gcs DB_SERVER(rw,sync,no_root_squash,no_all_squash) APP_SERVER(rw,sync,no_root_
squash,no_all_squash) PROC_SERVER(rw,sync,no_root_squash,no_all_squash)
```


- v. Reload the NFS service to apply configurations.

```
systemctl reload nfs-server
```

- b. Execute the following steps on each client machine that will access the files on the NFS Host. The Database, Application, and Processing Hosts all require access to `/var/lib/gcs`.

- i. Install autofs.

```
dnf install autofs
```

```
systemctl enable --now autofs
```

- ii. Create a `/etc/auto.master.d/gcs` file. Add the following content and replace `NFS_SERVER` with the NFS host machine's DNS or IP address.

```
/var/lib/gcs -fstype=nfs4,rw,nosuid,noexec NFS_SERVER:/export/gcs
```

- iii. Create a `/etc/auto.master.d/gcs.autofs` file and add the following content.

```
/- /etc/auto.master.d/gcs --timeout=600
```

- iv. Reload the autofs service to apply configurations.

```
systemctl reload autofs
```

9. On the Database Host machine:

- a. Install the Database Server.

```
bash install-db-server.sh
```

- b. Get the `gcs` group ID using the command below. You will need the `gcs` group ID when installing the Application and Processing Servers.

```
getent group gcs
```

The response should look like the following. In this example, the number 1002 is the group ID.

```
gcs:x:1002:
```

10. On the Application Host machine:

- a. Create the `gcs` group using the GID from the Database Host.

```
groupadd -g GID gcs
```

b. Install the Identity Server.

```
bash install-id-server.sh
```

c. Install the Application Server.

```
bash install-app-server.sh
```

d. Install the Imagery Server.

```
bash install-imagery-server.sh
```

e. Install the Terrain Analysis Server.

```
bash install-terrain-analysis-server.sh
```

f. Install the Proxy Server.

```
bash install-proxy-server.sh
```

11. On the Processing Host machine:**a. Create the `gcs` group using the GID from the Database Host.**

```
groupadd -g GID gcs
```

b. Install the Processing Server.

```
bash install-processing-server.sh
```

4.2 Installing Additional Terrain Analysis Servers

Additional Terrain Analysis Servers can be deployed to increase the computational throughput of the GCS Terrain Analysis services. Install additional Terrain Analysis Servers on Terrain Analysis Hosts that are separate from existing host machines in your GCS deployment. Execute the following steps:

1. On the Database Host:**a. Grant database access to the new host where you will install an additional Terrain Analysis Server.**

```
bash utils/grant-database-access-to-server.sh --ip-address=<TERRAIN_SERVER_IP_ADDRESS>
```

2. On a Terrain Analysis Host:

- a. Extract the GCS installer into `/usr/local/gcs` on and navigate to the installer directory.

```
mkdir /usr/local/gcs  
  
tar -xzf geospatial-content-installer-2.10.0-linux64.tgz -C /usr/local/gcs  
  
cd /usr/local/gcs/geospatial-content-installer-2.10.0
```

- b. Copy the `/usr/local/gcs/share/ssl` folder from the Database Host onto a Terrain Analysis Host.
- c. Copy the `/usr/local/gcs/geospatial-content-installer- 2.10.0/conf` folder from the Database Host onto a Terrain Analysis Host. If the original installation `conf` folder is not available, do the following:

- i. Create a `conf/gcs.cfg` configuration file.

```
bash utils/create-config-file.sh --type='distributed'
```

- ii. Edit the `conf/gcs.cfg` file to define your GCS configuration. See [Configuration Parameters](#) for a list of all required and optional configuration parameters.

- iii. Validate the configuration file.

```
bash utils/validate-configuration.sh
```

- d. Edit the `conf/gcs.cfg` file by setting `TERRAIN_ANALYSIS_SERVER` to the hostname of the new Terrain Analysis Host.
- e. Install a Terrain Analysis Server.

```
bash install-terrain-analysis-server.sh
```

5 Getting Started

After all the GCS components have been installed, it's time to start using the application.

5.1 Configure GCS Users

The Identity Server needs to be configured with users that can sign in to the GCS application. See the options below for ways to create or import users into the Identity Server.

5.1.1 Create GCS Test Users

GCS test users will help you explore GCS functionality in **non-production** environments. Create test users by running the following utility script on the machine hosting the Identity Server:

```
bash utils/create-gcs-test-users.sh
```

This script will create the following test users that can sign in to GCS. See the "GCS User Roles" section of [Managing Users and Roles](#) for descriptions of these roles.

Username	Password	Active Roles
gcs-guest	password	gcs_view_users
gcs-proc	password	gcs_processing
gcs-admin	password	gcs_admin
gcs-all	password	gcs_admin gcs-processing

⚠ Important: Test users are only intended for **non-production** environments. If you created them in a non-production environment that you want to convert into a production environment, make sure to **delete test users** through the Keycloak Administration Console.

5.1.2 Import Existing Users

Using the Keycloak Administration Console, you can integrate GCS with an existing User Federation to import users. Follow the directions in [Managing Users and Roles](#).

5.2 Sign In to GCS

Now that users have been configured in the Identity Server, you can use these users to sign in to GCS through a browser.

`https://PROXY_SERVER:PROXY_SERVER_PORT`

6 Managing Users and Roles

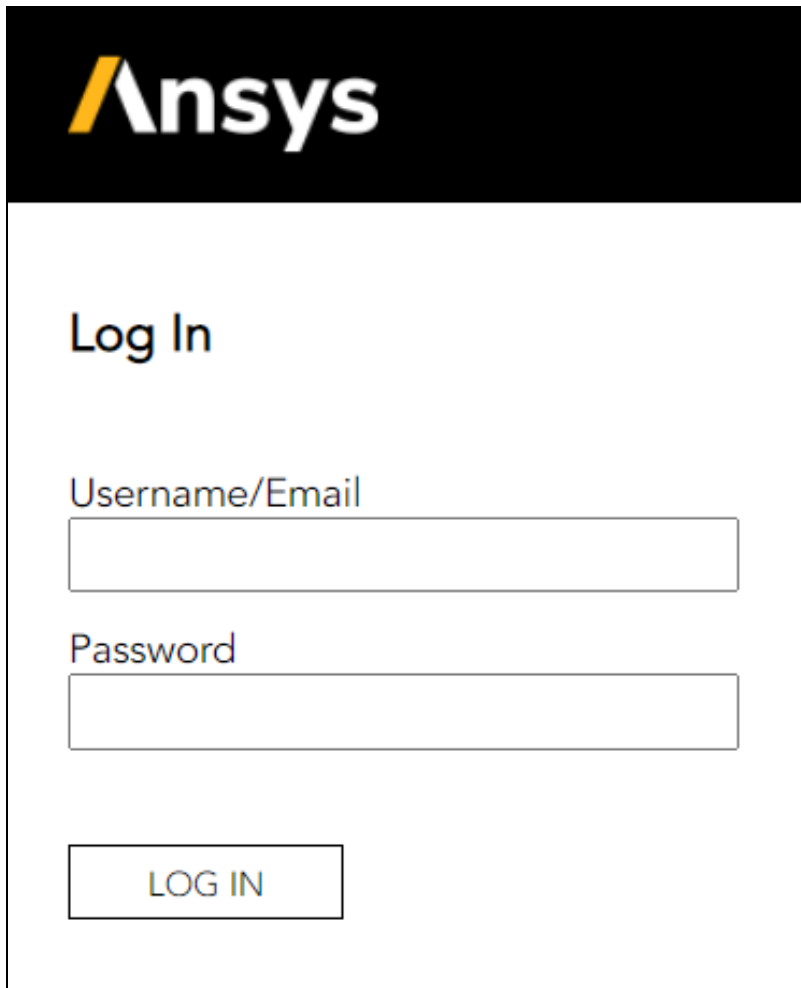
6.1 Accessing the AGI Identity Platform Administration Console

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, roles, role mappings, clients, and configurations. 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 **ID_SERVER** and **ID_SERVER_PORT** with the values used to install GCS:

`https://ID_SERVER:ID_SERVER_PORT/auth/admin`

2. Enter the username and password for the AIP admin user.
 - Username: Your **KEYCLOAK_ADMIN_USER** (default: admin)
 - Password: Your **KEYCLOAK_ADMIN_PASS**



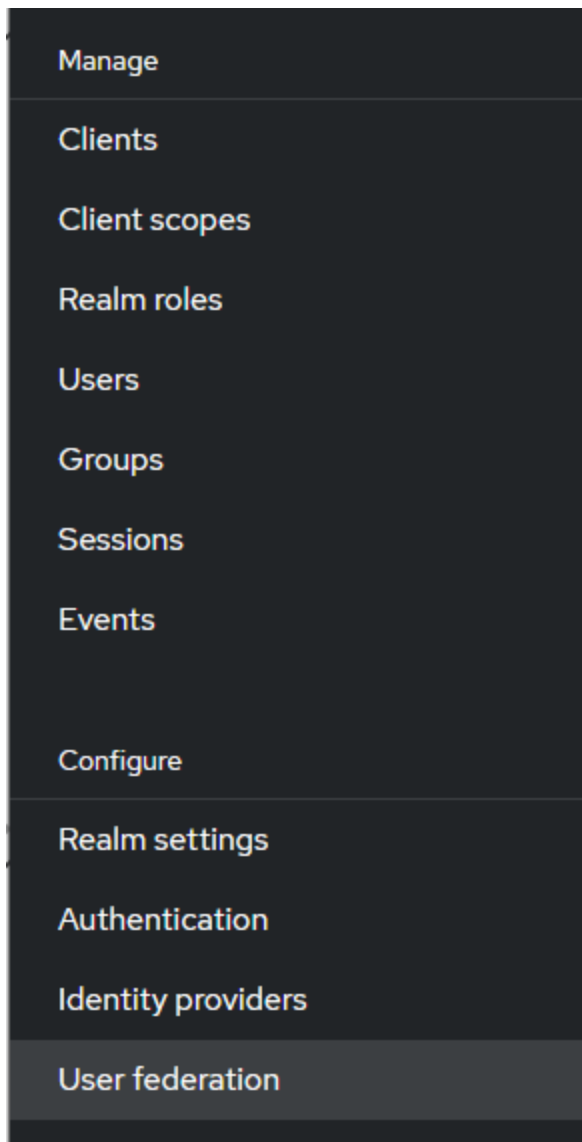
The image shows a login form for Ansys. At the top, there is a black header with the Ansys logo in white and yellow. Below the header, the text "Log In" is displayed in a large, bold, black font. Underneath, there are two input fields: the first is labeled "Username/Email" and the second is labeled "Password". Both fields are empty and have a thin black border. At the bottom of the form, there is a rectangular button with the text "LOG IN" in all caps, centered within the button.

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.

6.2 User Federation Integration

To sync users from Lightweight Directory Access Protocol (LDAP) and Active Directory servers, do the following in the Administration Console:

1. Click the **User Federation** link on the left menu.



2. Select the **Add LDAP providers** option. This will take you to the LDAP configuration page.

Add providers



Add Kerberos providers



Add Ldap providers

3. For information on how to configure LDAP, see the **Lightweight Directory Access Protocol (LDAP) and Active Directory** section of the Keycloak Server Administration Guide.

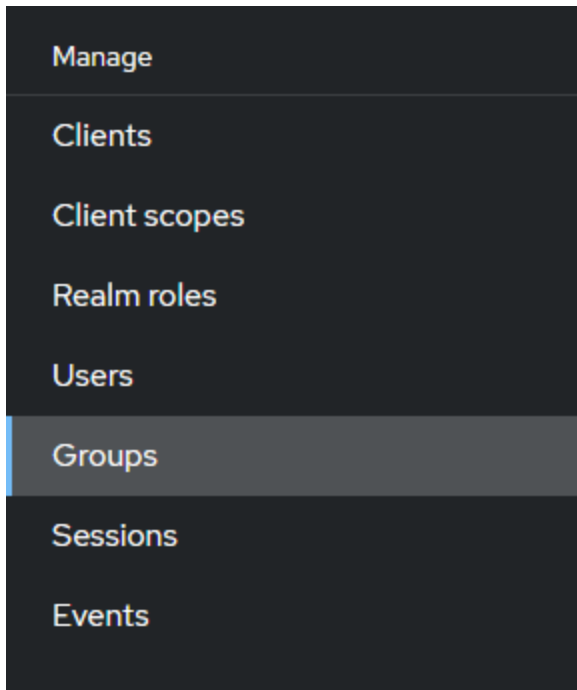
6.3 GCS User Roles

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

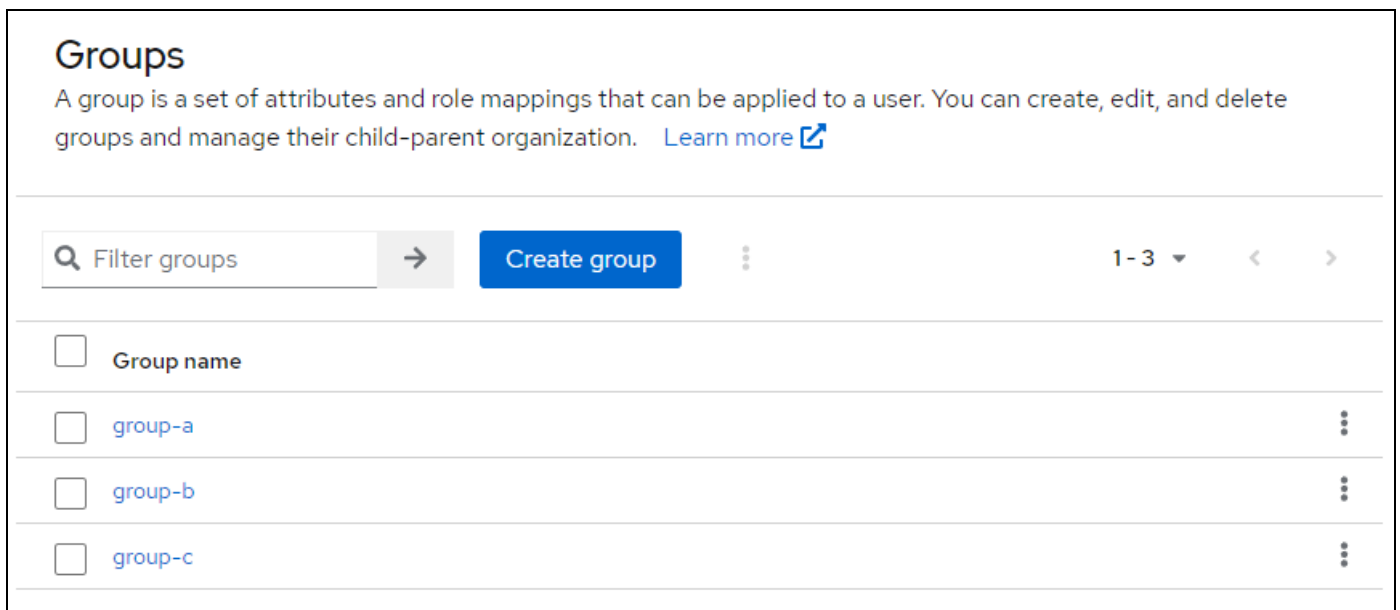
User Type	Role Name	Description
Viewer Users	<code>gcs_view_users</code>	Viewer users have read-only access to published assets. Authenticated users without any specific role mapping are equivalent to viewer users.
Processing Users	<code>gcs_processing</code>	Processing users have the gcs_processing 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. Processing users get the gcs_view_users role applied automatically.
Administrator Users	<code>gcs_admin</code>	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. Administrator users get the gcs_view_users role applied automatically.

6.4 Mapping a Role to an Existing Group in your Directory

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



2. Click the directory group name to be mapped to a role.



- Under the Role Mapping tab, click the **Assign role** button. Then select the desired role's checkbox (either **gcs_admin** or **gcs_processing**) from the list and click the **Assign** button to map that role to this group. You can select both roles in order to grant the rights assigned to each role.

Assign roles to group-a ✕

Filter by realm roles ▾ Search by role name → 1 - 10 ▾ < >

<input type="checkbox"/>	Name	Description
<input type="checkbox"/>	default-roles-agi01	`\${role_default-roles}`
<input checked="" type="checkbox"/>	gcs_admin	
<input type="checkbox"/>	gcs_metrics	
<input type="checkbox"/>	gcs_processing	
<input type="checkbox"/>	gcs_token	
<input type="checkbox"/>	gcs_view_users	
<input type="checkbox"/>	offline_access	`\${role_offline-access}`
<input type="checkbox"/>	segs_admin	
<input type="checkbox"/>	segs_data_consumer	
<input type="checkbox"/>	segs_data_manager	

1 - 10 ▾ < >

Assign Cancel



Note: The **gcs_view_users** role will be an inherited role when you assign either the **gcs_admin** or the **gcs_processing** role to a group. You can uncheck **Hide inherited roles** to view the hidden roles.

7 Upgrade Steps

This section describes how to upgrade existing GCS servers. If you are installing GCS for the first time, see [Installation Steps](#). Existing asset data and application tokens will be retained.

GCS Version	Can Upgrade From	Upgrade Procedure
2.10.0	2.9.0	New Install & Data Migration
2.9.0	2.7.0, 2.7.1, 2.8.0	In-Place
2.8.0	2.7.0, 2.7.1	In-Place
2.7.1	2.5.1, 2.6.0, 2.7.0	In-Place
2.7.0	2.5.1, 2.6.0	In-Place
2.6.0	2.5.1	In-Place
2.5.1	2.4.0, 2.5.0	In-Place
2.5.0	2.4.0	In-Place
2.4.0	2.3.0	In-Place
2.3.0	2.2.0, 2.2.1	In-Place
2.2.1	2.1.1, 2.2.0	In-Place

7.1 GCS 2.9.0 to GCS 2.10.0 Migration Overview

The process to upgrade from GCS 2.9.0 to GCS 2.10.0 differs significantly from earlier versions because the supported operating system changed from CentOS 7 to Enterprise Linux 8. To upgrade from GCS 2.9.0 to GCS 2.10.0, you must first install GCS 2.10.0 on an Enterprise Linux 8 machine. Make sure the installation configuration you use is correct for this deployment; for example, make sure the hostnames are the hostnames for the Enterprise Linux 8 machines and not the hostnames from the CentOS 7 machines. Then, follow the data migration procedure below.

Note: If your current install version is earlier than 2.9.0, upgrade your existing installation to 2.9.0 first, then perform the data migration.

The data migration procedure **will not** modify server configurations in your GCS 2.10.0 system. The only parameters specified in your installation configuration that will be overwritten are the Keycloak admin credentials (**KEYCLOAK_ADMIN_USER** and **KEYCLOAK_ADMIN_PASS**). In order to migrate installation configuration options from your GCS 2.9.0 system(s), you may use the configuration report scripts to identify any non-default settings and create your GCS 2.10.0 install configuration file based on the values identified. (See "GCS Configuration Report" below for more information.)

Important: Because the migration data import step will overwrite Keycloak admin credentials specified during the GCS 2.10.0 installation, it is important that you have access to your GCS 2.9.0 installation's Keycloak admin credentials before performing the migration. You will need these credentials to sign in to Keycloak following the data migration import step.

The migration exports the following types of data from your GCS 2.9.0 system:


- Assets
- Application tokens
- Keycloak user and realm configurations

These data will overwrite asset, application token, or Keycloak user/realm configurations present in your GCS 2.10.0 system.

7.2 GCS Configuration Report

The GCS installer bundle contains the `geospatial-content-el8-migration-tool-2.10.0-linux64.tgz` archive. Inside this archive are scripts that generate a configuration report, which can assist you in creating your installation configuration file for GCS 2.10.0. The upgrade steps below include instructions to generate the configuration report.

Note: The output of the GCS configuration report describes your existing GCS 2.9.0 system. This configuration may not be the best configuration for your new GCS 2.10.0 installation; for example, the hardware specifications of the new system, such as the number of CPUs or amount of RAM, may be different.

 Use the report as a starting point for a new configuration file, and review the values with your new environment in mind.

7.3 Upgrade Steps (Single-Machine)

Note:

- All scripts must be executed as a privileged user (conventionally named "root").
- All file locations should be specified as absolute paths.

GCS 2.9.0 Migration Data Export and Configuration Report

On the GCS 2.9.0 system:

1. Copy the `geospatial-content-el8-migration-tool-2.10.0-linux64.tgz` migration tool and extract it to `/usr/local/gcs/`.


```
tar -xzf geospatial-content-el8-migration-tool-2.10.0-linux64.tgz -C /usr/local/gcs
```

2. Export migration data. The files `oauth-client.properties` and `gcs-2.9.0-database.sql` will be exported to `/usr/local/gcs/migration-data/`.

```
cd /usr/local/gcs/el8-migration-tool
```

```
bash gcs-migration-export.sh
```

3. Back up the asset data files located in `/var/lib/gcs`, excluding `/var/lib/gcs/web/WEB-INF`.

 **Note:** The total size of the files could be very large depending on the assets being hosted by GCS. There are many ways to copy files between systems; consider how you want to copy these files to the target GCS 2.10.0 system in a later step.

4. Run the configuration report script. Consider whether you want to use the reported values in your new configuration file.

```
cd /usr/local/gcs/el8-migration-tool  
bash gcs-configuration-report.sh
```

GCS 2.10.0 Install and Migration Data Import

On the GCS 2.10.0 Enterprise Linux 8 system:

1. Follow [Installation Steps \(Single Machine\)](#). Make use of the GCS 2.9.0 configuration report when creating the configuration file for the new installation.
2. Copy backed-up asset data files to `/var/lib/gcs`.
3. Copy exported files `oauth-client.properties` and `gcs-2.9.0-database.sql` from `/usr/local/gcs/migration-data/` on the GCS 2.9.0 system to `/usr/local/gcs/migration-data/` on the GCS 2.10.0 system. Import migration data.

```
cd /usr/local/gcs/geospatial-content-installer-2.10.0  
bash data-migration/gcs-migration-import.sh
```

Update Keycloak configurations via the admin console:

- a. Sign in to the Keycloak admin console:
 - i. Open a web browser and navigate to the following URL, replacing `ID_SERVER` and `ID_SERVER_PORT` with the values used to install GCS:

`https://ID_SERVER:ID_SERVER_PORT/auth/admin`
 - ii. Enter the username and password for the AIP admin user. The credentials are the same as those from the GCS 2.9.0 install.
 - Username: Your `KEYCLOAK_ADMIN_USER` (default: admin)
 - Password: Your `KEYCLOAK_ADMIN_PASS`
- b. Select the **AGI01** realm.
- c. Click **Clients** in the left menu.

- d. For both the **gcs-web** and **gcs-app-default** clients, perform the following steps:
 - i. Click the client name to open its settings.
 - ii. In the **Access settings** section, update the **Valid web URIs** and **Web origins** properties with the new proxy server hostname and port:
 - Valid web URIs: `https://PROXY_SERVER: PROXY_SERVER_PORT/*`
 - Web origins: `https://PROXY_SERVER: PROXY_SERVER_PORT`
 - iii. In the **Capability config** section, turn off **Client Authentication**.
 - iv. Scroll to the bottom of the page and click **Save**.
 4. Sign in to your new GCS system and verify that all assets and tokens appear as they were in the GCS 2.9.0 system.
-

7.4 Upgrade Steps (Distributed)



Note:

- All scripts must be executed as a privileged user (conventionally named "root").
- All file locations should be specified as absolute paths.

GCS 2.9.0 Migration Data Export and Configuration Report

On the GCS 2.9.0 system:

1. Copy the `geospatial-content-el8-migration-tool-2.10.0-linux64.tgz` migration tool to each machine in your GCS distributed system and extract it to `/usr/local/gcs/`.

```
tar -xzf geospatial-content-el8-migration-tool-2.10.0-linux64.tgz -C /usr/local/gcs
```

2. On the database host:

- a.** Export migration data. The file `gcs-2.9.0-database.sql` will be exported to `/usr/local/gcs/migration-data/`.


```
cd /usr/local/gcs/el8-migration-tool
gcs-migration-export.sh
```

3. On the application host:

- a.** Export migration data. The file `oauth-client.properties` will be exported to `/usr/local/gcs/migration-data/`.

```
cd /usr/local/gcs/el8-migration-tool
bash gcs-migration-export.sh
```

- b.** Back up the asset data files located in `/var/lib/gcs`, excluding `/var/lib/gcs/web/WEB-INF`.

 **Note:** The total size of the files could be very large depending on the assets being hosted by GCS. Special consideration might be needed on the best way to copy these files to the target GCS 2.10.0 system in a later step.

- 4.** Run the configuration report script on each machine and consider using the report values in your new configuration file.

```
cd /usr/local/gcs/el8-migration-tool
bash gcs-configuration-report.sh
```

GCS 2.10.0 Install and Migration Data Import

On the GCS 2.10.0 Enterprise Linux 8 system:

- 1.** Follow [Installation Steps \(Distributed\)](#) for a new install of GCS 2.10.0. Make use of the GCS 2.9.0 configuration report when creating the configuration file for the new installation.

- 2.** On the identity host:

```
systemctl stop keycloak
```

3. On the database host:

- a. Copy backed up asset data files to `/var/lib/gcs`.
- b. Copy the exported `gcs-2.9.0-database.sql` file from the `/usr/local/gcs/migration-data/` folder on the GCS 2.9.0 database host to the `/usr/local/gcs/migration-data/` folder on the GCS 2.10.0 database host.
- c. Import migration data.

```
cd /usr/local/gcs/geospatial-content-installer-2.10.0  
  
bash data-migration/gcs-migration-import.sh
```

4. On the identity host:

```
systemctl start keycloak
```

5. On the application host:

- a. Copy the exported `oauth-client.properties` file from the `/usr/local/gcs/migration-data/` folder on the GCS 2.9.0 application host to the `/usr/local/gcs/migration-data/` folder on the GCS 2.10.0 application host.
- b. Run the migration import script.

```
cd /usr/local/gcs/geospatial-content-installer-2.10.0  
  
bash data-migration/gcs-migration-import.sh
```

6. On the processing host:

- a. Copy the exported `oauth-client.properties` file from the `/usr/local/gcs/migration-data/` folder on the GCS 2.9.0 application host to the `/usr/local/gcs/migration-data/` folder on the GCS 2.10.0 processing host.
- b. Run the migration import script.

```
cd /usr/local/gcs/geospatial-content-installer-2.10.0  
  
bash data-migration/gcs-migration-import.sh
```

7. If you have configured additional terrain analysis servers in the GCS 2.10.0 environment which are not on the same machine as the application host, perform the following steps on **each additional terrain analysis host:**

- a. Copy the exported `oauth-client.properties` file from the `/usr/local/gcs/migration-data/` folder on the GCS 2.9.0 application host to the `/usr/local/gcs/migration-data/` folder on the GCS 2.10.0 terrain analysis host.

b. Run the migration import script.

```
cd /usr/local/gcs/geospatial-content-installer-2.10.0  
bash data-migration/gcs-migration-import.sh
```

8. Update Keycloak configurations via the admin console:

a. Sign in to the Keycloak admin console:

- i. Open a web browser and navigate to the following URL, replacing **ID_SERVER** and **ID_SERVER_PORT** with the values used to install GCS:

`https://ID_SERVER:ID_SERVER_PORT/auth/admin`

- ii. Enter the username and password for the AIP admin user. The credentials are the same as those from the GCS 2.9.0 install.

- Username: Your **KEYCLOAK_ADMIN_USER** (default: admin)
- Password: Your **KEYCLOAK_ADMIN_PASS**

b. Select the **AGI01** realm.

c. Click **Clients** in the left menu.

d. For both the **gcs-web** and **gcs-app-default** clients, perform the following steps:

- i. Click the client name to open its settings.

- ii. In the **Access settings** section, update the **Valid web URIs** and **Web origins** properties with the new proxy server hostname and port:

- Valid web URIs: `https://PROXY_SERVER: PROXY_SERVER_PORT/*`
- Web origins: `https://PROXY_SERVER: PROXY_SERVER_PORT`

- iii. In the **Capability config** section, turn off **Client Authentication**.

- iv. Scroll to the bottom of the page and click **Save**.

9. Sign in to your new GCS system and verify that all assets and tokens appear as they were in the GCS 2.9.0 system.

8 Uninstallation Steps

8.1 Single Machine Uninstall

Navigate to the `/usr/local/gcs/geospatial-content-installer-2.10.0` folder, and uninstall GCS:

```
bash single-machine-uninstall.sh [OPTION]...
```

Optional Parameter	Description
<code>--keep-logs</code>	Keep server logs after GCS has been uninstalled.

8.2 Distributed Environment Uninstall

The uninstall scripts are located on each server where a GCS server was installed under the `/usr/local/gcs/geospatial-content-installer-2.10.0` folder. Run the following commands from that location.

1. On the host of the Proxy Server, uninstall the Proxy Server:

```
bash uninstall-proxy-server.sh [OPTION]...
```

Optional Parameter	Description
<code>--keep-logs</code>	Keep HTTPD logs for the Proxy Server.

1. On the host of the Imagery Server, uninstall the Imagery Server:

```
bash uninstall-imagery-server.sh [OPTION]...
```

Optional Parameter	Description
<code>--keep-logs</code>	Keep HTTPD logs for the Imagery Server.

1. On the host of the Terrain Analysis Server, uninstall the Terrain Analysis Server:

```
bash uninstall-terrain-analysis-server.sh [OPTION]...
```

Optional Parameter	Description
<code>--keep-logs</code>	Keep Tomcat logs for the Terrain Analysis Server.

2. On the host of the Processing Server, uninstall the Processing Server:

```
bash uninstall-processing-server.sh [OPTION]...
```

Optional Parameter	Description
<code>--keep-logs</code>	Keep Tomcat and HTTPD logs for the Processing Server.

3. On the host of the Application Server, uninstall the Application Server:

```
bash uninstall-app-server.sh [OPTION]...
```

Optional Parameter	Description
<code>--keep-logs</code>	Keep Tomcat logs for the Application Server.

4. On the host of the Identity Server, uninstall the Identity Server:

```
bash uninstall-id-server.sh [OPTION]...
```

Optional Parameter	Description
<code>--keep-logs</code>	Keep Keycloak logs for the Identity Server.
<code>--skip-db-cleanup</code>	Skip database cleanup. Normally, both the ID and DB servers need to be running so the uninstaller can connect to the database and remove the Keycloak database. This flag is useful when either the ID or DB server is in a bad state.

5. On the host of the Database Server, uninstall the Database Server:

```
bash uninstall-db-server.sh [OPTION]...
```

Optional Parameter	Description
--keep-logs	Keep PostgreSQL logs for the Database Server.

9 Storage

Some components that make up GCS need to share files under the `/var/lib/gcs` directory. On a single-machine installation, this might be on the local disk. In a distributed installation, the location would be mounted from a network share.

9.1 Shared Locations

The amount of storage required for shared locations is based on the sizes of files user upload. We recommend erring on the side of caution and providing ample storage to minimize disruption caused by a lack of disk space.

Location	Description
<code>/var/lib/gcs/upload</code>	The upload destination.
<code>/var/lib/gcs/work</code>	Uploaded files are staged here for processing.
<code>/var/lib/gcs/web</code>	Non-terrain assets are moved here after processing.
<code>/var/lib/gcs/terrain-data</code>	Terrain data is moved here after processing.

9.2 Server Locations

Location	Description	Growth
<code>/usr/local/gcs-app</code>	App Server Home	Static ~300MB
<code>/var/log/gcs-app</code>	App Server Logs	Depends on log settings and log retention.

Location	Description	Growth
/etc/httpd-gcs-processing-proxy	Processing Proxy Server Configuration	Static ~100KB
/usr/local/gcs-processing	Processing Server Home	Static ~65MB
/var/log/httpd-gcs-processing-proxy	Processing Proxy Server Logs	Depends on log settings and log retention.
/var/log/gcs-processing	Processing Server Logs	Depends on log settings and log retention.
/usr/local/gcs-terrain-analysis	Terrain Analysis Server Home	Static ~70MB
/var/log/gcs-terrain-analysis	Terrain Analysis Server Logs	Depends on log settings and log retention.
/var/lib/postgresql	Application database and configuration	Increases with the number of users and assets.
/usr/local/keycloak	Identity Server and configuration	Static ~295MB
/var/log/keycloak	Keycloak Server logs	Depends on log settings and log retention.
/etc/httpd-gcs-proxy	Proxy Server Configuration	Static ~100KB
/var/log/httpd-gcs-proxy	Proxy Server Logs	Depends on log settings and log retention.
/etc/httpd-gcs-imagery	Imagery Server Configuration	Static ~100KB
/var/log/httpd-gcs-imagery	Imagery Server Logs	Depends on log settings and log retention.

10 System Configuration

This section documents a number of configuration files that you can modify to tune GCS to suit your environment. Before you change these files, we recommend using the administrator metrics tab in the GCS web UI to establish a performance baseline for the system. This will allow you to see how configuration changes affect the system.

10.1 Terrain Analysis Configuration

The parameters in the table below place limits on terrain analysis computations. These limits help prevent individual requests from overloading the server. Changes to the `info-services.properties` file take effect immediately.

<code>info-services.properties</code>	
Location:	<code>/usr/local/gcs-app/webapps/api/WEB-INF/classes/META-INF/info</code> , on the Application Server
<code>azel.maxRays</code>	The maximum number of rays that can be requested for an Azimuth Elevation computation. This value must be greater than 0. We recommend setting this to <i>30 * the number of terrain analysis servers</i> .
<code>azel.maxTimeoutSeconds</code>	The maximum timeout duration that can be requested for an Azimuth Elevation computation. This value must be greater than 0. Use this value to limit request durations so that the terrain analysis servers are not overwhelmed. Keep this value in sync with the timeout configuration of the GCS proxy server for <code>/api/analysis/*</code> requests in the file <code>/etc/httpd-gcs-proxy/conf.d/20-proxy.conf</code> .
<code>azel.timeoutSeconds</code>	The default duration that an Azimuth Elevation mask computation will run before timing out. This value must be greater than 0 and less than or equal to <code>azel.maxTimeoutSeconds</code> .
<code>azel.minTolerance</code>	The minimum acceptable angular error in radians that can be requested for an Azimuth Elevation mask computation. See "Terrain Analysis Tolerance" on page 46 for more information.

info-services.properties`azel.tolerance`

The default acceptable angular error in radians for an Azimuth Elevation mask computation. This value must be greater than or equal to `azel.minTolerance`.

`heights.maxLocations`

The maximum number of locations that can be requested for a Heights computation. This value must be greater than 0.

`los.maxLines`

The maximum number of lines of sight that can be requested for a Line of Sight computation. This value must be greater than 0. We recommend setting this to *512 * the number of terrain analysis servers*.

`los.maxTimeoutSeconds`

The maximum timeout duration that can be requested for a Line of Sight computation. This value must be greater than 0. Use this value to limit the request durations so that the terrain analysis servers are not overwhelmed. Keep this value in sync with the timeout configuration of the GCS proxy server for `/api/analysis/*` requests in file `/etc/httpd-gcs-proxy/conf.d/20-proxy.conf`.

`los.timeoutSeconds`

The default duration that a Line of Sight computation will run before timing out. This value must be greater than 0 and less than or equal to `los.maxTimeoutSeconds`.

`los.minTolerance`

The minimum acceptable angular error in radians that can be requested for a Line of Sight computation. See "Terrain Analysis Tolerance" on page 46 for more information.

`los.tolerance`

The default acceptable angular error in radians for a Line of Sight computation. This value must be greater than or equal to `los.minTolerance`.

10.1.1 Terrain Analysis Configuration in the UI

You can view the current terrain analysis configuration in the GCS web UI. After signing in, navigate to the "Admin" tab, then click "Terrain Dashboard". This shows the terrain analysis settings as pictured below.

GEOSPATIAL CONTENT SERVER

ASSETS TOKENS **ADMIN**

Admin Options	Terrain Analysis Settings	
Log Files	Az-El Mask	
Terrain Dashboard	Max az-el rays per request	30
All Metrics	Default az-el request timeout	600 seconds
	Max az-el request timeout	1800 seconds
	Default az-el angular tolerance	0.00050
	Minimum az-el angular tolerance	0.00050
	Line of Sight	
	Max lines per request	512
	Default line of sight request timeout	600 seconds
	Max line of sight request timeout	1800 seconds
	Default line of sight angular tolerance	0.00050
	Minimum line of sight angular tolerance	0.00050

10.1.2 Terrain Analysis Tolerance

Terrain analysis computations are extremely performance-intensive. A single computation can potentially load gigabytes of data, stressing I/O, memory, and compute resources. To limit the duration of terrain analysis requests, the "tolerance" parameters, `azel.tolerance` and `los.tolerance`, were introduced in GCS 2.9.0. These parameters allow GCS to limit the resolution of terrain data it uses as the distance from the observation point grows, resulting in order-of-magnitude speed increases with negligible differences in computed results.

RECOMMENDED SETTINGS - TERRAIN ANALYSIS TOLERANCE

To simplify the configuration of GCS, we recommend these settings for `azel.tolerance` and `los.tolerance` depending on your users' needs.

Recommended Settings - Terrain Analysis Tolerance	
0.0005	Performance (Default): Prioritizes speed of computations. Speed improvements of 10x or more vs. using a tolerance of 0.0 are typical. Analysis results have good fidelity to "ground truth" results (using a tolerance of 0.0) for most applications.
0.00008	Accuracy: Prioritizes accuracy of computations. Speed improvements of 2x vs. using a tolerance of 0.0 are typical. Analysis results rarely differ from "ground truth" results.

GCS only checks that the values provided for terrain analysis tolerance are valid. However, we suggest limiting the tolerance parameters between 0.002 and 0.00002. Values above 0.002 may introduce enough error to invalidate analysis results. Values below 0.00002 will slow analysis computations without increasing the accuracy of results.

11 Maintenance

11.1 Get System Information

Get system information, such as GCS server version and certificate expiration dates.

Execute the following script on each host in your GCS deployment:

```
bash maintenance/get-system-info.sh
```

11.2 Updating GCS-Generated Certificates

The following steps detail how to update the certificates used to secure server-to-server communications.

11.2.1 Single-Machine Deployment

We will be replacing the GCS-generated certificates in `/usr/local/gcs/share/ssl`. Run the following commands to update the certificates:

- 1. Backup existing certificate directory (optional):**

```
mv /usr/local/gcs/share/ssl /usr/local/gcs/share/ssl-old
```

- 2. Navigate to the installer directory:**


```
cd /usr/local/gcs/geospatial-content-installer-2.10.0
```

- 3. Generate new certificates:**

```
bash generate-certificates.sh
```

- 4. Update the GCS-generated certificates:**

```
bash maintenance/update-gcs-generated-certificates.sh
```

 **Note:** This script will stop and restart the applicable services.

11.2.2 Distributed Deployment

We will be replacing the GCS-generated certificates in `/usr/local/gcs/share/ssl` for each host in the distributed environment. First, go to the Database Host and navigate to the installation directory. Run the following commands to update the certificates:

1. Backup existing certificate directory (optional):

```
mv /usr/local/gcs/share/ssl /usr/local/gcs/share/ssl-old
```

2. Navigate to the installer directory:

```
cd /usr/local/gcs/geospatial-content-installer-2.10.0
```

3. Generate new certificates on the Database Host. This will create the new `/usr/local/gcs/share/ssl` directory and its contents. Ensure the `conf/gcs.cfg` config file contains proper hostname definitions for the Database, Application, Processing, and Imagery Server components.

```
bash generate-certificates.sh
```

4. Update Database Host with new certificates:

```
bash maintenance/update-gcs-generated-certificates.sh
```

5. On the Application Host, Processing Host, and each Terrain Analysis Host, do the following:

a. Copy the Database Host's `/usr/local/gcs/share/ssl` directory onto each host, overriding the existing directory.


b. Navigate to the installer directory:

```
cd /usr/local/gcs/geospatial-content-installer-2.10.0
```

c. For Terrain Analysis Hosts, edit the `conf/gcs.cfg` file by setting `TERRAIN_ANALYSIS_SERVER` to the server hostname of the current host.

d. Run the following command from the installation root:

```
bash maintenance/update-gcs-generated-certificates.sh
```


 **Note:** This script will stop and restart the applicable services.

11.3 Updating Your Certificates

The following steps detail how to update the certificates used to secure client-to-server communications. These certificates are unique to your organization. When updating your certificates you must provide your server certificate file and its key file, as well as your certificate authority file.

For a **Single Machine** deployment, perform the following steps on your single GCS host machine.

For a **Distributed** deployment, perform the following steps on your Application host machine.

 **Note:** When executing the steps below, replace the example certificate files with the **absolute path** to your new certificate file.

1. Stop the services.

```
systemctl stop gcs-app
systemctl stop keycloak
systemctl stop gcs-proxy
```

2. Replace certificates for the Proxy Server. If your certificate files' names changed, either rename them to the previous certificate names or edit the file references in `/etc/httpd-gcs-proxy/conf.d/10-ssl.conf`.

```
cp /path/to/your/new-ca.crt /etc/httpd-gcs-proxy/ssl
cp /path/to/your/new-server.crt /etc/httpd-gcs-proxy/ssl
cp /path/to/your/new-server.key /etc/httpd-gcs-proxy/ssl
```

3. From the installation root `/usr/local/gcs/geospatial-content-installer-2.10.0`, execute `maintenance/update-keystores.sh` with your new certificates.

```
bash maintenance/update-keystores.sh \
  --ssl-ca-file='/path/to/your/new-ca.crt' \
  --ssl-cert-file='/path/to/your/new-server.crt' \
```



```
--ssl-key-file='/path/to/your/new-server.key'
```

4. Restart the services.

```
systemctl restart gcs-app  
systemctl restart keycloak  
systemctl restart gcs-proxy
```

11.4 Updating the Ansys License Manager Host

The following steps describe the process used for updating the host and/or port of the Ansys License Manager used by GCS. This process may be necessary if your Ansys License Manager is migrated to a different machine or if its configuration is updated.

11.4.1 Single-Machine Deployment

1. Update the `/usr/local/gcs/licensing/shared_files/licensing/ansyslmd.ini` file with the new host and port information. The file should contain a single line with the format:

```
SERVER=PORT@HOSTNAME
```

2. Restart the GCS application and processing services:

```
systemctl restart gcs-app gcs-processing
```

11.4.2 Distributed Deployment

1. On the GCS Application Host:
 - a. Update the `/usr/local/gcs/licensing/shared_files/licensing/ansyslmd.ini` file with the new host and port information. The file should contain a single line with the format:

```
SERVER=PORT@HOSTNAME
```

- b.** Restart the application service:

```
systemctl restart gcs-app
```

- 2.** On the GCS Processing Host:

- a.** Update the `/usr/local/gcs/licensing/shared_files/licensing/ansyslmd.ini` file with the new host and port information. The file should contain a single line with the format:

```
SERVER=PORT@HOSTNAME
```

- b.** Restart the processing service:

```
systemctl restart gcs-processing
```

11.5 Service Maintenance

Since GCS servers are registered as `systemd` services, you can manage them using `systemctl`. You can find more information on `systemctl` by running `man systemctl`.


The following services are created:

Service Name	Description
postgresql-15	Database server
keycloak	Identity server
gcs-app	Application server
gcs-processing	Processing server
gcs-processing-proxy	Processing server sidecar proxy
gcs-terrain-analysis	Terrain analysis server
gcs-imagery	Imagery server
gcs-proxy	Proxy server

12 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. Existing terraindb files are 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. Sign in to GCS and 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. If the terraindb file is small enough (a few gigabytes or less), upload the file through the web user interface. If the terraindb file is larger, it might be more efficient to upload the file by following the instructions in the [Sideload Assets](#) section.
5. When the upload completes, click the **Finalize Asset** button to perform the import.
6. Click the **Globe View** button to make sure the terrain looks correct.
7. If you are satisfied with the results, click the **Publish** button to make the asset available to others.

13 Earth Imagery

The GCS imagery server hosts Earth imagery from the Copernicus Sentinel-2 mission in both Web Map Service (WMS) and Web Map Tile Service (WMTS) formats. After the imagery server is installed, you can point applications to these URLs to use Sentinel-2 imagery:

- WMS: `https://PROXY_SERVER:PROXY_SERVER_PORT/mapcache`
- WMTS: `https://PROXY_SERVER:PROXY_SERVER_PORT/mapcache/wmts`

13.1 High-Resolution Imagery

GCS installs a low-resolution version of Sentinel-2 imagery as part of the Imagery Server installation. After installation, you can replace this with a high-resolution version provided on the Geospatial Content Data drive. You can find it on the drive at `./GcsDataFiles/Imagery/Sentinel-2/s2cloudless-2019_4326_v1.0.0.sqlite`.

Note:

- These steps assume you have successfully installed GCS.
- Specify all file locations as absolute paths.

1. Upload your SQLite database file to the Imagery Server. We recommend placing it in `/var/lib/gcs/mapserver/data`.
2. On the Imagery Server, run the utility script to change the imagery file served by this server:

```
cd /usr/local/gcs/geospatial-content-installer-2.10.0
bash utils/change-imagery-file.sh --imagery-file=FILE
```

14 Troubleshooting

14.1 System Administrator Log File Locations

Outputs from all install, upgrade, uninstall, and utility scripts are saved in `/usr/local/gcs/geospatial-content-installer-2.10.0/logs` for troubleshooting purposes. If you experience issues during these procedures, run the `utils/create-logs-package.sh` script to compress installer and server logs to send to [AGI support](#).

14.2 Server Log File Locations

Component	Log File Location
Database Server	<code>/var/log/postgres</code>
Identity Server	<code>/var/log/keycloak</code>
Application Server	<code>/var/log/gcs-app</code>
Processing Server	<code>/var/log/gcs-processing</code>
Processing Proxy Server	<code>/var/log/httpd-gcs-processing-proxy</code>
Terrain Analysis Server	<code>/var/log/gcs-terrain-analysis</code>
Imagery Server	<code>/var/log/httpd-gcs-imagery</code>
Proxy Server	<code>/var/log/httpd-gcs-proxy</code>

14.3 UI Issues

14.3.1 HTTP 401 - Cannot convert access token to JSON

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 Administration console, select the "AGI01" realm. Go to Realm Settings > Keys and look for the row with a "Type" of "RSA". In the "Public Keys" column, click "Public Key". On the Application Server, copy this value into each `oauth-client.properties` file under `/usr/local/gcs-app/webapps`. Replace the lines between

```
-----BEGIN PUBLIC KEY-----
```

and

```
-----END PUBLIC KEY-----
```

EXAMPLE:

```
jwt.public.key: -----BEGIN PUBLIC KEY-----\  
MIIBojANBgkqhkiG9w0BAQEFAAOCAQY8AMIIBigKCAyEA2uw3ZDkqhoc9uVybmkn\  
o67b4dFAzHxcRbZBpcxjwW+8j1822/4ATff4IvNBydU6efi7LK73v3KcEstQWloK\  
2cbrt+mupiKTuwoP5P1OhFR79SRedOCS0MMRVhFUGchvIy+F9Gy18K/1E+PMzNhc\  
k52QNLpm2WuXzwbZgBK2Kr1kzb5JRknvnnifBGBUipoQQBgq5opjRqc8P6lwQS4K\  
EP594DKbaZ49eu7kFIb3nBuDWgDEQRKqqlIgPrkPxK1F+aLIYrqlEy0vxxLFt84Z\  
/hkjLzQVFL3SarKH+7UWQqrPA8HzbCPOLdw4oFpkCA/pW3ePQnA2ofRbOSIXFiU5\  
fh1svvlmZrRFSrcvPXCZsrII3k8eQFiaTAMkp+TFPayZZEYf3ak0G/ISYzJJYVhg\  
uKVosOEroNg+o3zn06tced3m7waXNYkCCXDD92TH3kA+7ULxkVAskE2dQnqeMaOs\  
MqSdQRnqHBxhWxWeKN5vyNc1rwWT2TEOwJxIKP3WV4+rAgMBAAE=\  
-----END PUBLIC KEY-----
```

Repeat this process on the Processing Server and any Terrain Analysis Servers, using `/usr/local/gcs-processing/webapps` and `/usr/local/gcs-terrain-analysis/webapps` instead of `/usr/local/gcs-app/webapps`. Restart each service after updating `oauth-client.properties` so the changes take effect.

14.4 Processing Issues

14.4.1 CityGML processing fails with "Failed to layer.json"

CAUSE

The Processing Server cannot access terrain services running on the Application Server.

COMPONENTS AFFECTED

- GCS Processing Server

SOLUTION

Confirm that the terrain server URL in `/usr/local/gcs-processing/webapps/api/WEB-INF/classes/META-INF/processing/processing.properties` is correct. Next, verify that the Application Server is serving terrain by signing in to the GCS web application and previewing a terrain asset. If possible, preview the terrain referenced as the "Base Terrain" for the CityGML asset that is failing to process.

14.5 Performance Issues

Performance issues in GCS can appear for a variety of reasons. Depending on your system's hardware, deployment configuration, user traffic, and datasets, you might experience performance issues. While GCS attempts to install with a reasonable default configuration for optimizing performance, issues can still occur.

To address performance issues, you must calibrate your GCS environment in a way that works best for you. In this section, we go over which configurations you can change to affect performance outcomes.

14.5.1 Database Server Performance Issues

Certain GCS operations such as terrain tile fetching and terrain analysis computations can result in Database Server performance issues. These issues might manifest as slow or failed requests from a Database Client (e.g. a GCS Terrain Analysis Server) to the Database Server.

INSUFFICIENT MEMORY ON DATABASE SERVER

If the Database Server becomes critically low on memory, the Out Of Memory Killer kernel process will kill processes to free up memory.

HOW TO DIAGNOSE

Examine the output of the `journalctl -xe` command for Out Of Memory errors.

SOLUTION: INCREASE MEMORY ON DATABASE SERVER

Increase the available RAM on the machine running the Database Server. See the [Requirements](#) page for our hardware recommendations.

TOO MANY POSTGRESQL CONNECTION REQUESTS

PostgreSQL configures a maximum number of allowed connections from database clients. If clients try to establish more connections than this allowed number, PostgreSQL will not accept additional connection attempts.

HOW TO DIAGNOSE

Examine the logs of a Database Client (e.g. the `/var/log/gcs-terrain-analysis/stk-online-services.log` log file in a GCS Terrain Analysis Server) for an indication that connections to PostgreSQL were refused due to high volume.

SOLUTION: CHANGE POSTGRESQL MAX CONNECTIONS SETTING

If Database Clients are failing to connect to PostgreSQL because there are no available connections, you may want to increase the max connections PostgreSQL will handle. Use the `update-database-configuration.sh` utility script to do so.

1. Obtain PostgreSQL's current max connection limit by looking at the utility script's help output.

```
bash utils/update-database-configuration.sh --help
```

2. Determine a new PostgreSQL max connections value. A likely scenario is that you have installed numerous Terrain Analysis Servers that are using up all the PostgreSQL connections. Each Terrain Analysis Server has its own max workers property. A general recommendation is to set the PostgreSQL max connections to the sum of all Terrain Analysis Servers' max workers plus 100 (to account for all other GCS servers' connections). You can find the max workers of each Terrain Analysis Server by running this utility script's help option on each Terrain Analysis Server.

```
bash utils/update-terrain-analysis-configuration.sh --help
```

3. Update the PostgreSQL max connections value.



Note: This operation will restart PostgreSQL. Services will be temporarily unavailable.

```
bash utils/update-database-configuration.sh --postgres-max-connections=<value>
```

DATABASE AUDIT LOGS ARE TOO LARGE

A hardened installation of GCS configures database audit logging at the "read" level, which can result in excessive logging. Streaming terrain tiles for visualization and terrain analysis requires a very large number of reads from the Database Server. Logging all these reads affects performance and consumes a lot of disk space.

HOW TO DIAGNOSE

Run `du -h /var/log/postgresql`. If the disk space used by PostgreSQL logs appears very large, then database audits are logging every read.

SOLUTION: CHANGE DATABASE AUDITING SETTINGS

If streaming terrain tiles for visualization or analysis appears to be slow in a hardened installation, you may want to configure the Database Server to avoid auditing read access for the `terrain.tile` table in the `stk_online` database, or if possible, avoid auditing read access for all databases. You need to evaluate whether STIG/hardening requirements can be relaxed in your environment, and to what extent. To remove read level auditing for all databases, modify the following property in the file `~postgres/data/postgresql.stig.conf` in the Database Server:

```
pgaudit.log = 'ddl, role, write, function'
```

To enable read auditing of other tables, you can use pgAudit's Object Audit Logging. See [pgAudit's documentation](#) for more information. The following is an example of how to configure the Database Server to audit SELECT operations of all tables in the `stk_online` database, except for tables in the 'terrain' schema. In the Database Server, run

```
sudo -u postgres psql -d stk_online
```

At the prompt run the following script:

```
-- comment out the following line if the auditor role already exists
CREATE ROLE auditor;

-- grant SELECT permission of tables to the auditor role
DO
$$
DECLARE
entry RECORD;
BEGIN
FOR entry IN
SELECT nspname FROM pg_catalog.pg_namespace
WHERE nspname NOT IN ('terrain')
LOOP
EXECUTE 'GRANT SELECT ON ALL TABLES IN SCHEMA ' || entry.nspname || ' TO auditor';
END LOOP;
END;
$$ LANGUAGE plpgsql;
```

14.5.2 Terrain Analysis Server Performance Issues

Terrain analysis computations are prone to performance issues, especially if done on high resolution datasets, or if many users are issuing concurrent terrain analysis requests to your system. Terrain analysis performance issues can manifest as slow or unsuccessful terrain analysis requests.

INSUFFICIENT MEMORY ON THE TERRAIN ANALYSIS SERVER TOMCAT

If Tomcat becomes critically low on memory, the server cannot complete new terrain analysis calculations. On a Terrain Analysis Server, this manifests as `HTTP 500` error responses for terrain analysis requests that get interrupted by memory issues.

HOW TO DIAGNOSE

On a Terrain Analysis Server, examine the `/var/log/gcs-terrain-analysis/stk-online-services.log` log file for a 500 error. Additionally execute the `top` command to find how much memory the `gcs-terrain-analysis java` process is consuming. Compare the `RES` value against Tomcat's max memory setting. You can obtain the current configuration value by looking at our utility script's help output:

```
bash utils/update-terrain-analysis-configuration.sh --help
```

SOLUTION 1: INCREASE THE TERRAIN ANALYSIS SERVER TOMCAT MAX MEMORY SETTING

If there is more memory available on the machine, use the `update-terrain-analysis-configuration.sh` utility script to allocate additional memory to the Terrain Analysis Server.

```
bash utils/update-terrain-analysis-configuration.sh [OPTION]...
```

Configuration	Description
<code>--max-tomcat-memory</code>	The max memory allocated to the Terrain Analysis Server's Tomcat (e.g., <code>6G</code> for 6GB of max memory).

SOLUTION 2: INCREASE MEMORY ON TERRAIN ANALYSIS SERVER

Increase the available RAM on the machine running the Terrain Analysis Server. See the [Requirements](#) page for our hardware recommendations.

SOLUTION 3: DECREASE THE MAX TILE CACHE SIZE SETTING ON THE TERRAIN ANALYSIS SERVER

If increasing the Terrain Analysis max memory is not an option, you can also decrease the max tile cache size to decrease the likelihood of memory-related issues. You can obtain the current configuration value by looking at our utility script's help output:

```
bash utils/update-terrain-analysis-configuration.sh --help
```

Use the `update-terrain-analysis-configuration.sh` utility script to configure the behavior of the Terrain Analysis Server.

```
bash utils/update-terrain-analysis-configuration.sh [OPTION]...
```

Configuration	Description
<code>--max-tile-cache-size</code>	The max amount of memory allotted to tile caching.

INSUFFICIENT PROCESSING RESOURCES ON TERRAIN ANALYSIS SERVERS

Terrain analysis requests use all available CPU cores to parallelize computations. If there are not enough collective CPU cores amongst your Terrain Analysis Servers, your terrain analysis requests might be taking a long time or timing out entirely.

HOW TO DIAGNOSE

Use the `lscpu` utility to see how many CPU cores your Terrain Analysis Servers have. A small number of CPU cores can slow down terrain analysis requests.

SOLUTION 1: SCALE OUT TERRAIN ANALYSIS SERVERS

Horizontal scaling of Terrain Analysis Servers improves the speed of terrain analysis computations. See the "Installing Additional Terrain Analysis Servers" section of the [Distributed Installation Steps](#) for instructions. If a Terrain Analysis Server and Database Server are running on the same machine, you should uninstall the Terrain Analysis Server on this machine.

```
bash uninstall-terrain-analysis-server.sh
```

SOLUTION 2: INCREASE NUMBER OF CPU CORES ON TERRAIN ANALYSIS SERVERS

Increase the number of CPU cores on the machines running Terrain Analysis Servers. See the [Requirements](#) page for our hardware recommendations.

SOLUTION 3: UPDATE TERRAIN ANALYSIS SERVER MAX WORKERS SETTING

If your Terrain Analysis Server's hardware already meets or exceeds our hardware requirements, you can increase the max number of workers for terrain analysis computations. Use the `update-terrain-analysis-configuration.sh` utility script to configure the behavior of a Terrain Analysis Server.

1. Obtain the current configuration values of the Terrain Analysis Server by looking at the utility script's help output.

```
bash utils/update-terrain-analysis-configuration.sh --help
```

2. Update configuration values.

```
bash utils/update-terrain-analysis-configuration.sh [OPTION]...
```

Configuration	Description
<code>--max-workers</code>	The maximum number of workers for doing terrain analysis computations.

TERRAIN ANALYSIS REQUEST TIMEOUTS OR SERVER TOO BUSY

If the server is overloaded, you may receive a 429 HTTP Error response indicating a request timed out. If too many terrain analysis requests are issued to a Terrain Analysis Server, you may receive a 503 HTTP Error Response indicating that the server is too busy to process an incoming request.

HOW TO DIAGNOSE

Examine the `/var/log/gcs-terrain-analysis/stk-online-services.log` log file for this error.

SOLUTION: UPDATE TERRAIN ANALYSIS SERVER MAX QUEUE SIZE SETTING

The max queue size setting is in place to help prevent the server from getting overloaded. Increasing the value can lead to more requests getting accepted but may result in requests timing out if the server gets overloaded. A smaller queue size will drop more requests but will further prevent the server from overloading and can result in requests

completing as expected. Use the `update-terrain-analysis-configuration.sh` utility script to configure the behavior of a Terrain Analysis Server.

1. Obtain the current configuration values of the Terrain Analysis Server by looking at the utility script's help output.

```
bash utils/update-terrain-analysis-configuration.sh --help
```

2. Update configuration values.

```
bash utils/update-terrain-analysis-configuration.sh [OPTION]...
```

Configuration	Description
<code>--max-computation-queue-size</code>	The amount of queued terrain analysis computations this server allows. Consider increasing the queue size to avoid failed requests when the server is busy.

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

A Release Notes

2.10.0

- GCS now only supports Enterprise Linux 8, since maintenance support of CentOS 7 ends in June 2024. To upgrade an existing GCS installation, install GCS 2.10 and migrate your existing data into the new installation. See [Upgrade Steps](#) for more details.
- The computation of an Az-EI mask from terrain data has been improved to better sample along each azimuth. Users of these terrain analysis REST APIs should expect to see small differences in access results when using this improved Az-EI mask.
- Application, processing, and terrain analysis servers run on Java 17 instead of Java 11.
- Upgraded Apache Tomcat to version 10.1.15.
- Upgraded PostgreSQL to version 15.
- Upgraded Keycloak to version 22.0.5.
- Upgraded asset processing libraries to version 4.6.0.
- Processing progress is now displayed when an asset is finalized.
- The [Configuration Parameters](#) for installing GCS have been updated:
 - Removed `LOCALE_LANG` and `DB_CONNECTION_LIMIT` parameters.
 - Removed `TENANT_ID` parameter. New deployments will use 1 as the default value; existing deployments will continue using their existing tenant ID.
 - Added `POSTGRES_MAX_CONNECTIONS` parameter.
 - Replaced `TOMCAT_INITIAL_MEMORY` parameter with server-specific parameters `APP_TOMCAT_INITIAL_MEMORY`, `PROCESSING_TOMCAT_INITIAL_MEMORY`, and `TERRAIN_ANALYSIS_TOMCAT_INITIAL_MEMORY`. Also changed this default value to 256M.

- Replaced `WEB_MAX_THREADS` parameter with server-specific parameters `APP_WEB_MAX_THREADS`, `PROCESSING_WEB_MAX_THREADS`, and `TERRAIN_ANALYSIS_WEB_MAX_THREADS`.
 - Replaced `TOMCAT_LOG_LEVEL` parameter with server-specific parameters `APP_TOMCAT_LOG_LEVEL`, `PROCESSING_TOMCAT_LOG_LEVEL`, and `TERRAIN_ANALYSIS_TOMCAT_LOG_LEVEL`.
-


2.9.0

- Introduced application tokens. Users can now create and manage application tokens via the GCS UI. See the GCS User's Guide for further details.
 - Microsoft [announced on June 15, 2022](#) that it had ended support for Internet Explorer 11. As a result, GCS has also dropped support for Internet Explorer, therefore compatibility is no longer guaranteed. The GCS web interface will continue to work best on Microsoft Edge, Google Chrome, and Mozilla Firefox.
 - Modified terrain analysis computation algorithms to improve performance. The post-install scripts found under the Troubleshooting section of this document can be used to further tune your deployment.
 - Optional parameter `TOMCAT_MAX_MEMORY` from the install config file has been split out into three variables (`APPLICATION_TOMCAT_MAX_MEMORY`, `PROCESSING_TOMCAT_MAX_MEMORY`, `TERRAIN_ANALYSIS_TOMCAT_MAX_MEMORY`). This enables customization of each server individually at install time.
 - Updated MapCache to version 1.14.0. This resolves an issue where yum update commands were not completing successfully due to a dependency on armadillo 8.
 - Upgraded asset processing libraries to version 4.5.7. This new version includes fixes to allow processing of large OBJ assets.
-

2.8.0


- GCS now installs a dedicated terrain analysis server, which can be scaled out to meet higher workloads.
- Added support for georeferenced point cloud data in LASer (.las, .laz) format.

- Upgraded the packaged Ansys License Manager Linux installer to version 2022R2. You can upgrade your installed version by following the Ansys License Manager's Installation and Licensing Guide.
- Upgraded PostgreSQL to version 14.2.
- Upgraded asset processing libraries to version 4.5.5.
- Removed REST endpoints for archiving and unarchiving assets.
- Asset data files can now be directly transferred to the Application Host's file system, as a faster alternative to uploading them through the GCS web interface. See section "Sideloaded Assets" of the System Administration Guide.
- Configuration files are now the only supported strategy for supplying parameters to the installers.
- Single Machine Deployment installations now allow system administrators to skip providing certificates to deploy a non-production installation.
- Added a utility script to bundle application and installation logs into a single compressed file that can be sent to [AGI support](#) when reporting an issue.
- Added maintenance scripts for getting GCS system information, such as certificate expiration dates, and database connectivity status.

 **Note:** Microsoft [announced on June 15, 2022](#) that it had ended support for Internet Explorer 11. As a result, support for Internet Explorer will be removed in GCS version 2.9.0. The GCS web interface will continue to work best on Microsoft Edge, Google Chrome, and Mozilla Firefox.

2.7.1

- Fixed an issue introduced in GCS 2.6.0 that did not allow STK to use GCS assets.
- Upgraded Spring Framework to version 5.3.18 and Apache Tomcat to version 9.0.62.

 **Note:** Spring Framework 5.3.18 and Apache Tomcat 9.0.62 address CVE-2022-22965 (also known as "Spring4Shell"). While we do not believe GCS was vulnerable to this CVE, Spring and Tomcat were upgraded out of an abundance of caution.

2.7.0

- Updated Log4j to 2.17.1 to address CVE-2021-44832.
 - Distributed software installs now only require customer-provided certificates for the proxy and identity servers. All other servers use internal application-generated certificates.
 - Added certificate maintenance procedures to the System Administration Guide.
 - The Ansys License Manager Linux installer is now bundled with the GCS installation package.
-

2.6.0

- Updated Log4j to 2.16.0 to address CVE-2021-44228 and CVE-2021-45046.
 - Ansys License Manager is now required. Some installation procedures have changed. Please refer to the System Administrator Guide for more information.
 - STK Components license is no longer required.
 - Removed the use of weak cipher suites for secured communications in the database, application, processing, imagery, and proxy servers.
 - Added validation of customer-provided certificates during software installation.
 - Updated the Developer Guide documentation and code samples to show the proper way for a client application to connect to the AIP server.
-

2.5.1

- Fixed install of STK Enterprise Data Services (SEDS), which enables STK to discover GCS as a content provider.
- Fixed an issue where changing the database server port caused terrain processing to fail.

- Fixed an issue where changing the database server port prevented the `uninstall-database-server.sh` script from working.
 - Fixed line endings in AGI Identity Platform (AIP)/Keycloak scripts.
-

2.5.0

- Updated Sentinel-2 Earth imagery to 2019 version.
 - Updated PostgreSQL database to version 10.15 to address potential security issues.
 - Fixed an issue where the PostgreSQL database did not restart properly after a system reboot.
 - Log files are now deleted after 90 days.
-

2.4.0

- Updated asset processing libraries to version 4.4.0.
- Added analytical services for getting terrain heights.
- Added ability to layer terrain assets on top of existing terrain assets.
- Integrated Sentinel-2 imagery server on Linux into the Geospatial Content Server install.
- Fixed an issue in the identity server install scripts where hardened installs would not restrict Keycloak management from clients connecting from the provided IP addresses. Additionally, to comply with Security Technical Implementation Guidelines (STIG), the identity server upgrade installation will require the `--keycloak-admin-ip-addresses` parameter if the currently installed version of the identity server is detected to be hardened.
- Introduced configuration files as a way to manage install parameters for Geospatial Content Server.
- Some install parameters for the Geospatial Content Server install changed names. They are listed below. Please consider using the new configuration files to manage parameters for your installations. See section 4.1 of the GCS System Administration Guide for more information on configuration files.

Old Name	New Name
--id-db-server	--db-server
--id-db-server-port	--db-server-port
--proxy-port	--processing-proxy-port
--shutdown-port	--app-shutdown-port (for the app server) --processing-shutdown port (for the processing server)
--size	--db-machine-size
--allowed-host	Removed. Use --proxy-server instead.
--keycloak-bind-address	--id-server-bind-address
--ssl-cert-file	Use one of the following for the server you want to install: --app-server-ssl-cert-file --db-server-ssl-cert-file --id-server-ssl-cert-file --imagery-server-ssl-cert-file (new in 2.4.0) --processing-server-ssl-cert-file --proxy-server-ssl-cert-file (new in 2.4.0)
--ssl-key-file	Use one of the following for the server you want to install: --app-server-ssl-key-file --db-server-ssl-key-file --id-server-ssl-key-file --imagery-server-ssl-key-file (new in 2.4.0) --processing-server-ssl-key-file --proxy-server-ssl-key-file (new in 2.4.0)

2.3.0

- GCS now hosts its own terrain for processing and preview.
 - Added support for DTED and HRE formatted data.
 - Added developer documentation and REST API documentation.
 - Added license server support.
-

2.2.1

- Updated Geospatial Content Server to 2.2.1. This release includes bug fixes and performance improvements.
-

2.2.0

- Updated Geospatial Content Server to 2.2.0. This release includes processing and serving of terrain for visualization, as well as support for pre-processed 3D models, terrain, and imagery.
-

2.1.1

- Updated Geospatial Content Server to 2.1.1
- Updated the bundled version of Map Server for Windows to 4.0.0

B Glossary

Name	Definition
AIP	AGI Identity Platform. This Identity and Access Management (IdAM) solution, based on the open-source Keycloak project, provides user identities and access permissions to AGI applications.
Application Host	The host machine in a distributed deployment where the Application Server, Terrain Analysis Server, Identity Server, Imagery Server, and Proxy Server components are installed.
Database Host	The host machine in a distributed deployment where the Database Server is installed.
Distributed Deployment	Where components of GCS are installed across more than one physical or virtual server.
install root	The base directory of the installer after it is unarchived.
NFS Host	The host machine in a distributed deployment where the NFS File Server is installed.
privileged user	Normally the "root" user, this is a user on a Linux/Unix system with elevated access.
Processing Host	The host machine in a distributed deployment where the Processing Server is installed.
SEDS	STK Enterprise Data Services. This service, which comes bundled with GCS, allows STK to access GCS and other AGI Enterprise services.
Single-Machine Deployment	Where all components of GCS are installed on the same physical or virtual server.
upgrade root	Same as install root, but designates that the installer is being used to upgrade a previous installation.

C Configuration Parameters

When deploying GCS, configuration parameters can be defined in the `conf/gcs.cfg` configuration file. This section documents the valid configuration parameters.

C.1 Required Parameters

GCS requires a number of parameters to be set during the deployment process. Required parameters are different for single-machine and distributed deployments. There are also additional required parameters when conducting a STIG compliant deployment. The following sections list required parameters for each deployment type. Required parameters *must* be defined by the system administrator before deployment.

C.1.1 All Deployments

These parameters are **always required** for all GCS deployments. See the **additional required parameters** for Single-Machine and Distributed deployments based on your deployment type.

Configuration Parameter	Description	Default Value
LICENSE_SERVER	The fully-qualified domain name of the Ansys license server.	None
KEYCLOAK_ADMIN_PASS	Password for the Keycloak administrative user. If <code>HARDEN='true'</code> , the password must be 15 characters or longer.	None
DB_SERVER	The fully-qualified domain name of the host of the Database Server.	Output of <code>hostname -f</code>
ID_SERVER	The fully-qualified domain name of the host of the Identity Server.	Output of <code>hostname -f</code>

Configuration Parameter	Description	Default Value
APP_SERVER	The fully-qualified domain name of the host of the Application Server.	Output of <code>hostname -f</code>
PROCESSING_SERVER	The fully-qualified domain name of the host of the Processing Server.	Output of <code>hostname -f</code>
TERRAIN_ANALYSIS_SERVER	The fully-qualified domain name of the host of the Terrain Analysis Server.	Output of <code>hostname -f</code>
IMAGERY_SERVER	The fully-qualified domain name of the host of the Imagery Server.	Output of <code>hostname -f</code>
PROXY_SERVER	The fully-qualified domain name of the host of the Proxy Server. This is the domain name that users enter to access the application.	Output of <code>hostname -f</code>

C.1.2 Single-Machine Deployment

In addition to the required parameters listed above, these parameters are required for a single-machine GCS deployment.

Note:

- The `SSL_CA_FILE`, `SSL_CERT_FILE`, and `SSL_KEY_FILE` parameters are required to properly secure a production deployment.
- If not defined, GCS will generate temporary certificates. This is only intended for **non-production deployments**. You can replace these temporary certificates with your own after GCS is deployed. To update the certificates, follow the "Updating Customer-Provided Certificates" steps in the [Maintenance](#) section.
- Some browsers require you to import the temporary Certificate Authority into the browser's truststore to access GCS. The temporary CA can be found at `/usr/local/gcs/temporary-external-certs/gcs-temporary-ca.crt` after the deployment has completed. You may need to restart your browser after importing the CA.

Configuration Parameter	Description	Default Value
SSL_CA_FILE	Full path to a PEM-formatted certificate authority file. This is the certificate authority that signed the SSL certificate used to identify this server.	None
SSL_CERT_FILE	Full path to a PEM-formatted server certificate file that will be used to identify this server.	None
SSL_KEY_FILE	Full path to an unencrypted PEM RSA server certificate key file. This is the private key for SSL_CERT_FILE.	None

C.1.3 Distributed Deployment

In addition to the required parameters listed above, these parameters are required for a distributed GCS deployment.

Configuration Parameter	Description	Default Value
SSL_CA_FILE	Full path to a PEM-formatted certificate authority file. This is the certificate authority that signed the SSL certificate for the GCS proxy and identity server components.	None
ID_SERVER_SSL_CERT_FILE	Full path to a PEM-formatted certificate used to identify the GCS identity server.	None
ID_SERVER_SSL_KEY_FILE	Full path to the private key that pairs with the public key of ID_SERVER_SSL_CERT_FILE.	None

Configuration Parameter	Description	Default Value
PROXY_SERVER_SSL_CERT_FILE	Full path to a PEM-formatted certificate used to identify the GCS proxy server.	None
PROXY_SERVER_SSL_KEY_FILE	Full path to the private key that pairs with the public key of PROXY_SERVER_SSL_CERT_FILE.	None

C.2 Optional Parameters

The following parameters may be used to further customize your GCS deployment. They are not required for the deployment process, but are useful to resolve port conflicts, configure security settings, or tune performance.

Configuration Parameter	Description	Default Value
LICENSE_SERVER_PORT	The HTTPS port for the Ansys license server.	1055
HARDEN	When <code>HARDEN='true'</code> , reduces the system's surface of vulnerability and significantly increases PostgreSQL log verbosity for auditing. <code>HARDEN='true'</code> is recommended for production deployments.	<code>true</code>
FIREWALL_ZONE	The firewall zone in which rich rules are applied.	<code>public</code>

Configuration Parameter	Description	Default 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-10.1-doc/config/http.html). Multiple protocols may be selected with a comma-separated list (e.g., 'TLSv1.2,TLSv1.3').	TLSv1.2

C.2.1 Database Server

Configuration Parameter	Description	Default Value
DB_SERVER_PORT	The HTTPS port for the Database Server.	5432
DB_BACKUP_DIR	Path to a directory where all database backups will be stored.	/usr/local/postgres
DB_BACKUP_CRON_EXPRESSION	A cron expression to schedule when backup of database cluster will occur. The default schedule runs every night at midnight.	0 0 * * *

Configuration Parameter	Description	Default Value
DB_MACHINE_SIZE	The size of the machine that the deployment is running on. This will be used to tune the database settings for optimal performance. The available options are M1, L1, S3, M3 and L3. M1 specifies a single machine deployment on a machine with approximately 8GB of RAM. L1 specifies a single machine deployment on a machine with approximately 16GB of RAM. S3 specifies a multi-machine deployment, where the database is on a machine with approximately 4GB of RAM. M3 and L3 specify multi-machine deployments, with the same amount of RAM specified in M1 and L1, respectively.	M1
POSTGRES_MAX_CONNECTIONS	The maximum number of allowed concurrent connections to the postgres server.	250
IP_NETWORK_MASKS	One or more IPv4 address ranges formatted in CIDR notation to control which clients may attempt to connect to the database server.	Default configuration creates the IP network masks by determining the IP addresses of the server hostnames provided in the configuration file.

C.2.2 Identity Server

Configuration Parameter	Description	Default Value
ID_SERVER_PORT	The HTTPS port for the Identity Server.	8443
KEYCLOAK_ADMIN_USER	Username for the Keycloak administrative user.	admin

C.2.3 Application Server

Configuration Parameter	Description	Default Value
APP_SERVER_PORT	The HTTPS port for the Application Server.	8445
APP_SERVER_CLIENT_AUTH_PORT	The HTTPS port used to accept processing server requests containing client certificates.	9443
APP_TOMCAT_INITIAL_MEMORY	The initial memory allocated to the Application server's Tomcat (e.g., 256M for 256MB of initial memory).	256M
APP_TOMCAT_MAX_MEMORY	The max memory allocated to the Application server's Tomcat (e.g., 4G for 4GB of max memory).	4G
APP_TOMCAT_LOG_LEVEL	Specify Tomcat log level for the Application server. Options: SEVERE, WARNING, INFO, CONFIG, FINE, FINER, FINEST, ALL.	INFO
APP_WEB_MAX_THREADS	The maximum number of request processing threads to be created by the Application Tomcat server.	200

C.2.4 Processing Server

Configuration Parameter	Description	Default Value
PROCESSING_SERVER_PORT	The HTTPS port for the Processing Server.	8444
PROCESSING_TOMCAT_INITIAL_MEMORY	The initial memory allocated to the Processing server's Tomcat (e.g., 256M for 256MB of initial memory).	256M
PROCESSING_TOMCAT_MAX_MEMORY	The max memory allocated to the Processing server's Tomcat (e.g., 4G for 4GB of max memory).	4G
PROCESSING_TOMCAT_LOG_LEVEL	Specify Tomcat log level for the Processing server. Options: SEVERE, WARNING, INFO, CONFIG, FINE, FINER, FINEST, ALL.	INFO
PROCESSING_WEB_MAX_THREADS	The maximum number of request processing threads to be created by the Processing Tomcat server.	200

C.2.5 Terrain Analysis Server

Configuration Parameter	Description	Default Value
TERRAIN_ANALYSIS_SERVER_PORT	The HTTPS port for the Terrain Analysis server.	8447
TERRAIN_ANALYSIS_SERVER_CLIENT_AUTH_PORT	The HTTPS port used to accept terrain analysis internal server requests containing client certificates.	9447

Configuration Parameter	Description	Default Value
TERRAIN_ANALYSIS_TOMCAT_INITIAL_MEMORY	The initial memory allocated to the Terrain Analysis server's Tomcat (e.g., 256M for 256MB of initial memory).	256M
TERRAIN_ANALYSIS_TOMCAT_MAX_MEMORY	The max memory allocated to the Terrain Analysis server's Tomcat (e.g., 6G for 6GB of max memory).	6G
TERRAIN_ANALYSIS_TOMCAT_LOG_LEVEL	Specify Tomcat log level for the Terrain Analysis server. Options: SEVERE, WARNING, INFO, CONFIG, FINE, FINER, FINEST, ALL.	INFO
TERRAIN_ANALYSIS_WEB_MAX_THREADS	The maximum number of request processing threads to be created by the Terrain Analysis Tomcat server.	200
TERRAIN_ANALYSIS_MAX_QUEUE_SIZE	The amount of queued terrain analysis computations this server allows.	4096
TERRAIN_ANALYSIS_MAX_TILE_CACHE_SIZE	The max amount of memory allotted to tile caching, in bytes.	2147483648
TERRAIN_ANALYSIS_MAX_WORKERS	The maximum number of workers for doing terrain analysis computations.	40

C.2.6 Imagery Server

Configuration Parameter	Description	Default Value
IMAGERY_SERVER_PORT	The HTTPS port for the Imagery Server.	8446

C.2.7 Proxy Server

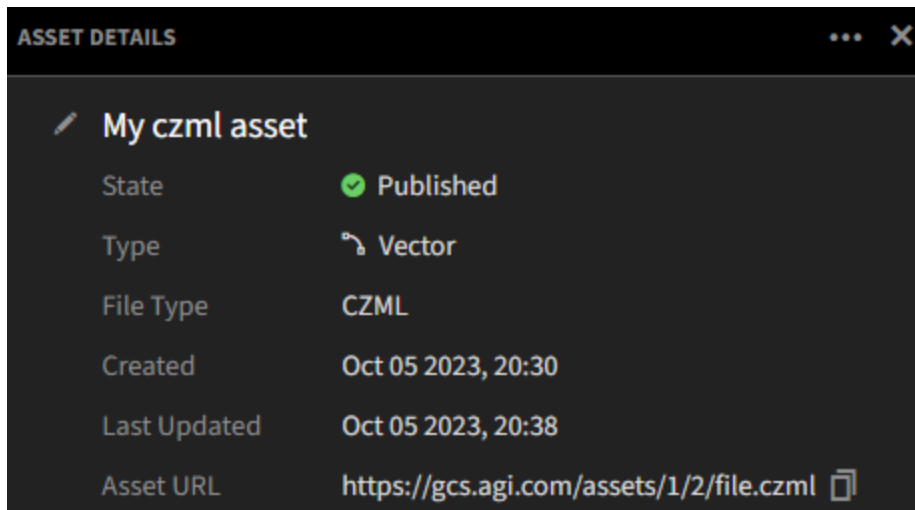
Configuration Parameter	Description	Default Value
PROXY_SERVER_PORT	The HTTPS port for the reverse Proxy Server.	443
PROXY_SERVER_BIND_ADDRESS	The IP address from which the proxy server will accept connections.	0.0.0.0

D Sideload Assets

Sometimes it is impractical to upload very large files through the GCS web UI. In these cases, files can be "sideloaded", which is a procedure where a system administrator uploads files directly to a special folder in the Application Host's file system. Follow these steps to sideload asset data files:

1. Have the user create an asset using the GCS web UI. The asset must be in a DRAFT state, with no files associated with it. The name of the asset needs to be unique among all DRAFT assets. The best practice is to prefix the name with the user's logon id, and use only letters, numbers, underscores, and dashes. The asset can later be renamed to a more appropriate name.
2. On the Application Host, upload the asset file to the `/var/lib/gcs/sideload/<tenant-id>/<asset-name>/`. You can find the tenant-id value in one of two ways:
 - a. If you've already uploaded and processed an asset on your installation then you can fetch it from the existing asset's URL in the asset details panel. In the following image, the asset URL format is:

`https://<server>/assets/<tenant-id>/<asset-id>/file.czml.`



- b. Find it in the Keycloak Administration console.

- i. Sign into the Keycloak admin console

1. Open a web browser and navigate to the following URL, replacing `ID_SERVER` and `ID_SERVER_PORT` with the values used to install GCS:

`https://ID_SERVER:ID_SERVER_PORT/auth/admin`

2. Enter the username and password for the AIP admin user.

- Username: Your **KEYCLOAK_ADMIN_USER** (default: admin)
- Password: Your **KEYCLOAK_ADMIN_PASS**

ii. Select the **AGI01** realm.

iii. Click **Clients** in the left menu.

iv. In the table of clients, click the **gcs-web** client.

v. Select the **Client scopes** tab.

vi. Select the **gcs-web-dedicated** client scope.

vii. Select the **tenant_id** mapper.

viii. The value for **Claim value** property is your tenant id.

3. Make sure the files and folders created during this process are all owned by `gcs-app:gcs`.

4. Once the asset data file is finished uploading, rename the asset folder to have a ".ready" suffix to start the process.

5. The system will automatically detect the file to sideload. The asset folder suffix will change to ".inprogress" while the file is being associated to the correct asset and moved to the right location. The asset folder suffix changes to ".error" when the process fails (usually due to file permissions).

6. The asset folder suffix changes to ".done" when the asset data file is sideloaded successfully. At this point, after refreshing the GCS web UI, the user is free to rename and finalize the asset.

E Data Migration

This section provides instructions on populating a new (target) GCS environment with a copy of all the assets and application tokens from another (source) GCS environment of the same version. This can be used to transfer data from a test or evaluation GCS environment to a new production environment.

When performing this migration, any data and configuration exported from the source GCS environment will overwrite the data and configuration of the target GCS environment. Options specified in the installation configuration file will not be overwritten, with the exception of the Keycloak admin credential parameters (**KEYCLOAK_ADMIN_USER** and **KEYCLOAK_ADMIN_PASS**).

⚠ Important: Because the migration data import step will overwrite Keycloak admin credentials specified during the target GCS environment installation, it is important that you have access to your source GCS environment's Keycloak admin credentials before performing the migration. These credentials will be necessary to sign in to Keycloak and to perform configuration following the data migration import step.

Source Environment Migration Data Export

On the source GCS environment:

1. If GCS system is a single machine install:
 - a. Export migration data. The files **oauth-client.properties** and **gcs-2.10.0-database.sql** will be exported to **/usr/local/gcs/migration-data/**.

```
cd /usr/local/gcs/geospatial-content-installer-2.10.0
```

```
bash data-migration/gcs-migration-export.sh
```

2. If GCS system is a distributed install:
 - a. On the database host:
 - i. Export migration data. The file **gcs-2.10.0-database.sql** will be exported to **/usr/local/gcs/migration-data/**.

```
cd /usr/local/gcs/geospatial-content-installer-2.10.0
```

```
bash data-migration/gcs-migration-export.sh
```

b. On the application host:

- i.** Export migration data. The file `oauth-client.properties` will be exported to `/usr/local/gcs/migration-data/`.

```
cd /usr/local/gcs/geospatial-content-installer-2.10.0
```

```
bash data-migration/gcs-migration-export.sh
```

3. On the application host, back up the asset data files located in `/var/lib/gcs`.



Note: The total size of the files could be very large depending on the assets being hosted by GCS. Special consideration might be needed on the best way to copy these files to the target GCS environment in a later step.

Target Environment Migration Data Import

On the target GCS environment:

1. On the identity host:

```
systemctl stop keycloak
```

2. On the database host:

- a.** Copy backed up asset data files to `/var/lib/gcs`.

- b.** Copy exported files `oauth-client.properties` and `gcs-2.10.0-database.sql` from folder `/usr/local/gcs/migration-data/` on the source environment to folder `/usr/local/gcs/migration-data/` on the target environment.

- c.** Import migration data.

```
cd /usr/local/gcs/geospatial-content-installer-2.10.0
```

```
bash data-migration/gcs-migration-import.sh
```

3. On the identity host:

```
systemctl start keycloak
```

4. Update Keycloak configurations via the admin console:
 - a. Sign in to the Keycloak admin console:
 - i. Open a web browser and navigate to the following URL, replacing **ID_SERVER** and **ID_SERVER_PORT** with the values used to install GCS:

`https://ID_SERVER:ID_SERVER_PORT/auth/admin`
 - ii. Enter the username and password for the AIP admin user. The credentials are the same as those from the source GCS environment.
 - Username: Your **KEYCLOAK_ADMIN_USER** (default: admin)
 - Password: Your **KEYCLOAK_ADMIN_PASS**
 - b. Select the **AGI01** realm.
 - c. Click **Clients** in the left menu.
 - d. For both the **gcs-web** and **gcs-app-default** clients, perform the following steps:
 - i. Click the client name to open its settings.
 - ii. In the **Access settings** section, update the **Valid web URIs** and **Web origins** properties with the new proxy server hostname and port:
 - Valid web URIs: `https://PROXY_SERVER: PROXY_SERVER_PORT/*`
 - Web origins: `https://PROXY_SERVER: PROXY_SERVER_PORT`
 - iii. In the **Capability config** section, turn off **Client Authentication**.
 - iv. Scroll to the bottom of the page and click **Save**.
5. If the target GCS 2.10.0 environment is a distributed install:
 - a. On the application host:
 - i. Copy the exported **oauth-client.properties** file from the `/usr/local/gcs/migration-data/` folder on the source environment application host to the `/usr/local/gcs/migration-data/` folder on the target GCS 2.10.0 application host.

ii. Run the migration import script.

```
cd /usr/local/gcs/geospatial-content-installer-2.10.0  
bash data-migration/gcs-migration-import.sh
```

b. On the processing host:

i. Copy the exported **oauth-client.properties** file from the **/usr/local/gcs/migration-data/** folder on the source environment application host to the **/usr/local/gcs/migration-data/** folder on the target GCS 2.10.0 processing host.

ii. Run the migration import script.

```
cd /usr/local/gcs/geospatial-content-installer-2.10.0  
bash data-migration/gcs-migration-import.sh
```

c. If you have additional terrain analysis servers in the target environment which are not on the same machine as the application host, perform the following steps on **each** additional terrain analysis host:

i. Copy the exported **oauth-client.properties** file from the **/usr/local/gcs/migration-data/** folder on the source environment application host to the **/usr/local/gcs/migration-data/** folder on the target GCS 2.10.0 terrain analysis host.

ii. Run the migration import script.

```
cd /usr/local/gcs/geospatial-content-installer-2.10.0  
bash data-migration/gcs-migration-import.sh
```

6. Verify install and data migration for correctness. You should now be able to log in to your target environment and inspect that all assets and tokens are as they were in the source environment.