XSEDE Globus Connect Server v5.4 Installation Guide

Eric Blau

Version 1.2
February 2, 2022

The Globus Connect Server (GCS) v5 installation process is significantly different from GCS v4 and GridFTP installation processes. GCS v4 could be installed as a GridFTP server with additional configuration, however, GCS v5 must be installed using Globus provided packages and instructions for setting up an endpoint with one or more data transfer nodes. GCS v5 endpoints can be configured with multiple access policies, called *Storage Gateways*, and data spaces, called *Collections*. This document describes how to:

1. Install and configure a GCS v5.4 *endpoint* with one or more *data transfer nodes*
2. Define one or more *Storage Gateways* defining access policies for your endpoint
3. Define one or more *Collections* that access storage areas using Storage Gateway access policies

In this guide use the following XSEDE naming guidelines:

<Short Site Name> is a short name like “SDSC”

<Site Name> is a long site name like “San Diego Supercomputer Center (SDSC)” with the short site name in parenthesis

<Resource> is a short resource name like “Expanse”

The terminology around Globus Connect Server has changed since GCS v4. The Globus Connect version 5 terminology documentation gives a useful overview of terms used in this document.

0. Install the xsede-oauth-mapfile Package

XSEDE uses the GCS External Mapping functionality to map XSEDE OAuth identities to local accounts. The Globus Identity Mapping Guide in Section 4 External Mapping Programs Reference describes how external mapping programs work. The xsede-oauth-mapfile tool: generates the XSEDE mapfiles that GCS can use, and provides the wrapper script that GCS calls to lookup values in the XSEDE mapfiles.

Follow the xsede-oauth-mapfile INSTALL instructions as root:

a. You will need to reference the gcs-mapfile-lookup.json file from this install in steps 2.1.1, 2.2, and 2.3.1 below.

1. Creating an Endpoint

Follow Globus Connect Server V5 Installation Guide through section 4.7 to create and configure an endpoint referencing the following important notes:

a. **Using the root account**. Globus software installs, “endpoint setup”, and “node setup” discussed through Section “*4.4. Add Data Transfer Nodes to the endpoint*” must be done as root. All the remaining steps from “*4.5 Log into the endpoint*” and beyond do not require root and should be run from an admin or service account. XSEDE installation and configuration instructions below should not be run as root unless so noted.

b. GCS v5.4 by default relies on Let’s Encrypt certificates in the *.data.globus.org domain for servers and collections. Starting with v5.4.13 GCS can instead be configured to use a custom local hostname and certificate domain for endpoints, mapped collections, and/or guest collections using the instructions at Globus Connect Server Domain Guide - Configuration and Management of Custom Domains. If you want to have a local DNS hostname and domain, follow the Custom Domain instructions and obtain a wildcard SSL certificate per the instructions.

c. In Section “4.2.1. Create service credentials”:
Give your new client secret a description like “Server Setup Secret”.

d. In Section “4.2.2. Setup the Endpoint”:
Use the “**--keywords XSEDE”** option to aid in endpoint discovery.
Specify an --owner ID that can login to Globus. **<your_username>@xsede.org** is recommended.

Globus is implementing changes, at XSEDE’s request to make endpoints more easily discoverable by subscribing organization, but at present it is difficult to find a list of endpoints that are guaranteed to be "official" XSEDE endpoints. In GCS v4 and before XSEDE’s endpoints were owned by "xsede@globusid.org". Starting with GCS v5 XSEDE’s endpoints will be fully owned by the Service Provider, while still associated with XSEDE’s Globus subscription.

e. In Section “4.7. Set the endpoint as managed” submit a ticket to help@xsede.org to request that your endpoint be added to the XSEDE Globus subscription and provide the endpoint uuid (ID) in your ticket. Send this email instead of executing the command in this section because XSEDE service provider admins and testing staff are not XSEDE subscription managers.

f. Do nothing, this step was moved to “*0. Install the xsede-oauth-mapfile Package”* above.

g. Create Globus Groups for your site's endpoint *administrators*, *activity managers*, and *activity monitors*

i. Follow https://docs.globus.org/how-to/managing-groups/ to create groups called:
   - ➢ <Resource> Administrators
   - ➢ <Resource> Activity Managers
   - ➢ <Resource> Activity Monitors

ii. Note the UUID for each group from the URL of its "about" page (step 5 in the previous document). The URL will look like https://app.globus.org/groups/{group UUID}/about

iii. *Administrators* can:
   - View or modify the endpoint, even if it is not public.
   - View, add, delete or modify GCS Manager nodes which provide access to the endpoint.
   - View, add, or delete the custom DNS name for mapped collections.
   - View, add, modify, or delete the storage gateways provided by the endpoint.
   - View (public information only) or delete the user credentials registered with the endpoint.
   - View, delete or modify collections hosted by the endpoint.
   - View, add, delete or modify other role assignments on the endpoint or any of its collections.

   Activity Managers can:
   - View the endpoint configuration, including storage gateways and their public policies.

   Activity Monitors can:
   - View the endpoint configuration, including storage gateways and their public policies.

iv. Run the command:
```
globus-connect-server endpoint role create \
      --principal-type group \
      administrator <administrator group UUID>
```
To set the administrator group for your endpoint to the group you just created

v. Repeat the previous step for activity manager and activity monitor roles:
```
globus-connect-server endpoint role create \
      --principal-type group \
      activity_manager <activity_manager group UUID>
globus-connect-server endpoint role create \
      --principal-type group \
      activity_monitor <activity_monitor group UUID>
```

vi. For additional information about these roles, see: https://docs.globus.org/globus-connect-server/v5/reference/endpoint/role/

2. Configuring the Endpoint for XSEDE use cases

For GCS 5.4 endpoints that provide access for XSEDE projects, we strongly recommend separate storage gateways and collections for XSEDE use, because XSEDE users have different access requirements and identity mappings than non-XSEDE users. (Specifically, for XSEDE use, an active XSEDE allocation is required and the XSEDE username is mapped to a local username.)

In a nutshell, the recommended XSEDE data access configuration is as follows.

- One collection for members of XSEDE-allocated projects.
- (optional) An additional collection that allows authorized individuals to create guest collections.
- (optional) An additional collection for use by XSEDE-allocated science gateways.
- (optional) Additional collections to support non-XSEDE uses.

The first two collections above can be combined if the local access policy permits it. Otherwise, each collection is distinct from the others.

Refer to the Design for XSEDE SP Deployment of Globus Connect Server 5.4 for a more in-depth description of the motivations of the recommended configuration

2.1 Configuring a Storage Gateway and Mapped Collection for Primary Access for XSEDE-allocated projects

These steps do not require root unless otherwise noted.

2.1.1 Creating a Storage Gateway: Primary Access for XSEDE allocated projects

Reference the Globus Connect Server V5 Data Access Admin Guide to create a storage gateway for primary access to data for XSEDE allocated projects.

Referring to GCS v5 Data Access Admin Guide Section 3.1:

1. Create a “posix” storage gateway to
2. Allowing authentication from “xsede.org”
3. Define the authentication timeout as the default 11 days
4. Use xsede-oauth-mapfile supplied mapfile lookup json configuration
5. Restrict access to filesystem paths that are designated for XSEDE use

This can be done by creating a filesystem path restrictions JSON file and executing one command.

First, create a path_restrictions_primary.json file that defines which filesystem paths can be used, like:

```
{
  "DATA_TYPE": "path_restrictions#1.0.0",
  "read": [
    "/public"
  ],
  "read_write": [
    "/home",
    "/projects"
  ],
  "none": [
    "/private"
  ]
}
```

Then execute a command similar to this:

```
globus-connect-server storage-gateway create \
    posix "XSEDE <Resource> Primary Storage Gateway" \
    --domain xsede.org \
    --identity-mapping \
    --authentication-timeout-mins $((60 * 24 * 11)) \
 file:/usr/local/share/utils/xsede_oauth_mapfile/etc/gcs-mapfile-lookup.json \
    --restrict-paths file:path_restrictions_primary.json
```

This command will display the newly created <Storage Gateway UUID>. To verify this storage-gateway is define use the command:
`globus-connect-server storage-gateway list`

The xsede-oauth-mapfile package installed above creates a mapfile in the default location of /etc/grid-security/xsede-oauth-mapfile. When Globus Connect Server calls gcs-mapfile-lookup to do the identity lookup it references a mapfile that includes the <Storage Gateway UUID>:
`/etc/grid-security/xsede-oauth-mapfile-<Storage Gateway UUID>`

To use the full XSEDE identity mapfile for this storage gateway create the symlink:
`sudo ln -s /etc/grid-security/xsede-oauth-mapfile \`
`/etc/grid-security/xsede-oauth-mapfile-<Storage Gateway UUID>`
otherwise, you can place custom mappings created manually or programmatically directly in:
`/etc/grid-security/xsede-oauth-mapfile-<Storage Gateway UUID>`

2.1.2 Creating a Mapped Collection: Primary Access for XSEDE allocated projects

A mapped collection allows access to data for users who have accounts in the storage gateway's user space (or local account). The collection uses the identity mapping method configured on the storage gateway to map the Globus account of the user accessing the

collection to an account in the Storage Gateway's user space. All accesses to the data on the collection are performed using the local account and (if needed for the storage gateway) the account's credentials.

Referring to https://docs.globus.org/globus-connect-server/v5/data-access-guide/#data_access_collection_create we will create a mapped collection with:

- A base path that is the shortest common path for the paths allowed by the storage gateway. (All paths allowed must be beneath the base path.)
- A display name and descriptive properties (description, contact information, web link) that make the purpose of the collection clear in search results.
- Guest collections are disabled.

In order to create this mapped collection, we need to know the UUID of the Storage Gateway created in the previous step. To get this, use the globus-connect-server storage-gateway list command:

```
globus-connect-server storage-gateway list
```

Make note of the UUID of the "XSEDE Primary Gateway" in the results of this command

Now, determine the base path for your mapped collection. It must be a single directory relative to the root of the storage gateway's virtual file system that will act as the root of the collection. All directories that will be accessible by the collection must fall under this single directory. The root directory, "/" can be used.

Choose a descriptive display name

For this primary mapped collection, it is recommended to use "XSEDE <Resource>" as the display name. Here, <Site Name> refers to the long version, e.g. "San Diego Supercomputing Center", <Short Site Name> would be, e.g. "SDSC", and <Resource> would be "Expanse".

```
globus-connect-server collection create \
      STORAGE_GATEWAY_ID \
      COLLECTION_BASE_PATH DISPLAY_NAME \
      --organization '<Site Name>' \
      --contact-email help@xsede.org \
     --description "Collection for XSEDE users to access data on <Resource>" \
      --keywords XSEDE,<Short Site Name>,<Resource> \
      --public
```

2.2 Enable project members to create Guest Collections (Optional)

Globus can be configured to enable Guest Collections, allowing authorized individuals who meet the criteria for accessing an XSEDE primary collection (XSEDE user with an active allocation and a local account on the resource) to share secure guest access to specific data with other Globus users. For example, a researcher who is a member of an XSEDE project might share read-only access to the results of a specific application run with a specific group of students, research assistants, or research partners who, themselves, are not part of the researcher's XSEDE project.

When the user wants to share something, they create a guest collection that references a specific directory. Once created, the guest collection allows the researcher to set Globus level permissions on directories within the guest collection that enable specific Globus individuals or groups to access it.

2.2.1 Enabling sharing for all users and paths on Primary Mapped Collection

If local access policy allows guest collections to be created in all the filesystem paths accessible by the above *Storage Gateway for Primary Access for XSEDE-allocated projects*, then this access mode can be configured when creating those mapped collections using the --allow-guest-collections option. Or, guest collections can be enabled after the fact using:

```
# Find collection UUIDs
globus-connect-server collection list --filter "mapped-collections"

# Enable guest collections on a specific collection
globus-connect-server collection update COLLECTION_ID \
      --allow-guest-collections
```

2.2.2 Enabling Restricted Sharing on Primary Mapped Collection

If, however, guest collections are only permitted for a subset of filesystem paths accessible by the above *Storage Gateway for Primary Access for XSEDE-allocated projects*, or, you wish to restrict which users or posix groups can create guest collections, we recommend configuring your primary "XSEDE <resource>" mapped collection with any path restrictions and using the --sharing-user-allow, --sharing-user-deny, --posix-sharing-group-allow and/or --posix-sharing-group-deny options to control permission for creation of Guest Collections/sharing. For example:

If you want to restrict which paths can be shared, create a Path Restriction Document `path_restrictions_sharing.json` to define which filesystem paths are to be shared:

```
{  "DATA_TYPE": "path_restrictions#1.0.0",
  "read": [
    "/public"
  ],
  "read_write": [
```

```
    "/home",
    "/projects"
  ],
  "none": [
    "/private"
  ]
}
```

If you create a unix group on your resource containing the local usernames of the users authorized to create Guest Collections, you can use --posix-sharing-group-allow below:

```
 Find collection UUIDs
globus-connect-server collection list --filter "mapped-collections"

# Enable guest collections on a specific collection
globus-connect-server collection update COLLECTION_ID \
      --allow-guest-collections \
      --posix-sharing-group-allow <sharing group> \
      --sharing-user-allow <username> \
      --sharing-restrict-paths file:path_restrictions_sharing.json
```

Note: all options after --allow-guest-collections are optional. If you have no need to apply path restrictions, or to specify particular usernames/groups to receive or not receive permission to share, simply omit those options. Note also that --sharing-user-allow (or --sharing-user-deny) takes precedence over --posix-sharing-group-allow or --posix-sharing-group-deny, as described at the documentation for globus-connect-server collection update

More complex sharing permissions can be configured as described in Sharing Configuration and User Sharing Path Restrictions, however, having a one-to-one relationship between the directories available to XSEDE users in a mapped collection, and the files that are shareable makes it considerably more transparent for the user. See Sharing Policy Overview for more details.

2.2.3 Creating a new Storage Gateway for specific sharing access scenarios

Most likely, you will be able to use the methods outlined in sections 2.2.1 or 2.2.2 to enable sharing for XSEDE users. If you need to enable sharing access for a filesystem path that is not generally accessible on your primary storage gateway, or have other specific concerns, you can create a separate storage gateway with its own access controls, and a second mapped collection that can serve as the basis for Guest Collections. You do not need to do this if either scenario from 2.2.1 or 2.2.2 works for you.

When creating a new Storage Gateway, the --user-allow, --user-deny, --posix-group-allow and/or --posix-group-deny options can be used to control access. See Posix User Policies for more details. You can create a path restrictions document for use with the --restrict-paths option to specify which filesystem paths can be accessed in collections that use this storage gateway. See the PathRestrictions Document page for more details about path restriction documents.

If you create a unix group on your resource containing the local usernames of the users authorized to create Guest Collections, you can create a Storage Gateway to host those collections as follows:

```
globus-connect-server storage-gateway create \
      posix "XSEDE <Resource> Sharing Storage Gateway" \
      --domain xsede.org \
      --identity-mapping \
      file:/usr/local/share/utils/xsede_oauth_mapfile/etc/gcs-mapfile-lookup.js
      on \
      --restrict-paths file:path_restrictions_guests.json \
      --user-allow <sharing username>  \
      --posix-group-allow <sharing group>
```

Make sure that the path_restrictions_guests.json file used here allows access only to the directories you wish to be available for users to share. The options --user-allow and --posix-group-allow can be used multiple times to allow multiple users/groups.

This command will display the newly created <Storage Gateway UUID>.

To use the full XSEDE identity mapfile for this storage gateway create the symlink:
`sudo ln -s /etc/grid-security/xsede-oauth-mapfile \`
`/etc/grid-security/xsede-oauth-mapfile-<Storage Gateway UUID>`
otherwise, you can place custom mappings created manually or programmatically directly in:
`/etc/grid-security/xsede-oauth-mapfile-<Storage Gateway UUID>`

Next, create a mapped collection for sharing, making sure that the display name, description, and keywords make the purpose of the collection clear in search results.:

```
globus-connect-server collection create \
      STORAGE_GATEWAY_ID \
      COLLECTION_BASE_PATH "XSEDE <Resource> Sharing Collection" \
      --organization '<Site Name>' \
      --contact-email help@xsede.org \
      --description "This Collection can be used by XSEDE users who have been
specifically authorized to create Guest Collections in order to share data on
<Resource>" \
      --keywords XSEDE,<Short Site Name>,<Resource>,Sharing \
      --allow-guest-collections \
      --public
```

2.3 Enable XSEDE-allocated science gateways (Optional)

A science gateway is an application used by a community of researchers. The developer or operator of the application has been granted an allocation to use an XSEDE SP resource and is

assigned a community account by XSEDE. Researchers who use the application (usually) do not have accounts on the XSEDE SP resources. The community account is used to create a guest collection and grant permission (in the guest collection) to the science gateway's OAuth credentials. The science gateway can then access the guest collection as needed using the Globus Transfer API or direct web access (HTTPS).

For the storage gateway:

- The user must have an xsede.org identity.
- Authentication expires after 11 days.
- Local accounts are mapped from XSEDE identities using an external mapping script, but authorization is limited to community accounts.
- Data access is restricted to filesystem paths designated for XSEDE use by community accounts. (This may be a subset of the paths used in the previous section.)

For the mapped collection:

- The base path is the shortest common path for the paths allowed by the storage gateway. (All paths allowed must be beneath the base path.)
- The display name and descriptive properties (description, contact information, web link) make the purpose of the collection clear in search results.
- Guest collections are enabled.

The configuration of a storage gateway and mapped collection for Science Gateways is very similar to the storage gateway and mapped collection for Guest Collections in section 2.2.3 above. The largest differences are that in this case, the filesystem paths are restricted to only those that the Science Gateway(s) need to be able to share, and access is limited to only Science Gateway community accounts.

2.3.1 Creating a Storage Gateway: Science Gateways

Create a Storage Gateway with user access policies such that only Science Gateway community accounts have access. This can be best accomplished for posix connectors by using the --posix-group-allow and/or --posix-group-deny. See Posix User Policies for more details.

Create a group on your resource containing the local usernames of all Science Gateway community accounts authorized to share data on your resource. Then, you can create a Storage Gateway, adding this option:

```
globus-connect-server storage-gateway create \
       posix "XSEDE <Resource> Storage Gateway for Science Gateways" \
       --domain xsede.org \
       --identity-mapping \
```

```
      file:/usr/local/share/utils/xsede_oauth_mapfile/etc/gcs-mapfile-lookup.json.json \
      --restrict-paths file:path_restrictions_gateways.json \
      --user-allow <community account(s)>
```

Make sure that the path_restrictions_gateways.json file used here allows access only to the directories you wish to be available for science gateways.

This command will display the newly created <Storage Gateway UUID>.

To use the full XSEDE identity mapfile for this storage gateway create the symlink:
`sudo ln -s /etc/grid-security/xsede-oauth-mapfile \`
`/etc/grid-security/xsede-oauth-mapfile-<Storage Gateway UUID>`
otherwise, you can place custom mappings created manually or programmatically directly in:
`/etc/grid-security/xsede-oauth-mapfile-<Storage Gateway UUID>`

2.3.2 Creating a Mapped Collection: Science Gateways

Next, create a mapped collection for sharing, making sure that the display name, description, and keywords make the purpose of the collection clear in search results.:

```
globus-connect-server collection create \
      STORAGE_GATEWAY_ID \
      COLLECTION_BASE_PATH "XSEDE <Resource> Collection for Science Gateways" \
      --organization '<Site Name>' \
      --contact-email help@xsede.org \
      --description "This Collection can be used by XSEDE Science Gateways that have been specifically authorized to create Guest Collections in order to share data on <Resource>" \
      --keywords XSEDE,<Short Site Name>,<Resource>,"Science Gateways" \
      --allow-guest-collections \
      --public
```

2.4 Announcing Production GCS v5 Availability

Per the XSEDE GCS 5.4 Deployment Plan, section *E. Coordination and Communication*, when a GCS v5 endpoint production availability date is set, or when it becomes available in production, the SP must notify XSEDE of the production date, and the file-system paths supported for standard data transfers, guest collections, and science gateways. On the production date the SP should also announce GCS v5 availability through XSEDE User News.

Appendix: Useful Resources

- ➔ Globus Connect Server v5.4 Installation Guide

- ➔ Globus Connect Server v5.4 Authorization and Authentication Guide
- ➔ Globus Connect Server Troubleshooting Guide

Appendix: De-installation Steps

How to fully delete a GCS v5.4 endpoint on a RHEL derived distro server.
Reference: https://docs.globus.org/globus-connect-server/v5.4/reference/node/cleanup/

Step 1:

To remove ANY node from the endpoint, simply run the following command on that node:

```
globus-connect-server node cleanup
```

To decommission an endpoint completely, including storage gateways and hosted collections, run this command on all data mover nodes in the endpoint.

```
globus-connect-server endpoint cleanup --client-id YOUR_CLIENT_ID
--deployment-key /PATH/TO/YOUR/deployment-key.json
```

It is very important to perform this step before Step 4, as there is no way for an admin to properly cleanup their endpoint themselves if the Auth client used to setup their endpoint has been deleted prior to the cleanup command being run. In such a case, you'll have to contact Globus support.

Step 2: Delete RPMs

You can remove the Globus software packages with one of these commands, as appropriate for your rhel distro:

```
yum remove \*globus\*
dnf remove \*globus\*
```

Step 3: Delete <any other directories>

As a precaution, you'll want to run these commands to delete any residual config that might be left on your rhel system:

```
rm -r /etc/gridftp.d/
rm -r /etc/globus/
rm -r /var/lib/globus-connect-server/
rm /etc/gridftp.conf
rm /etc/globus-connect-server.conf
rm /etc/httpd/conf.d/gcs_manager.conf
rm /etc/httpd/conf.d/gcs_manager.conf.gcs
```

```
rm /etc/httpd/conf.d/mod-globus.conf
rm /etc/httpd/conf.d/tls-mod-globus.conf
rm /etc/httpd/conf.d/tls_gcs_manager.conf.gcs
rm /etc/httpd/conf.modules.d/10-globus.conf
```

Some or all of the 'rm' commands may produce 'No such file or directory' messages. This is fine, as this step is just a precaution.

Step 4: Delete / Rename Registered App Service Credentials

Once you've completed the cleanup, go to the URL below and rename the Auth client that was used when you initially setup this endpoint:

https://auth.globus.org/v2/web/developers

You'll want to rename the client so that you know never to use it again for any purpose. It is important to never re-use an Auth client that was used to setup one endpoint to setup another endpoint. This is not supported and can lead to various hard to solve problems. If the endpoint deletion process encountered no issues, then you can just go ahead and delete the Auth client instead of renaming it. Renaming the client is a more cautious approach, but the client can simply be deleted if the endpoint was successfully cleaned up.