Globus for SysAdmins

Vas Vasiliadis - vas@uchicago.edu
May 23, 2023
Our focus in this session

- Makes your storage accessible via Globus
- Software/tools installed and managed by sysadmin
- Native packaging Linux: DEB, RPM

[docs.globus.org/globus-connect-server]
Globus Connect Server v5 Overview
Globus Connect Server v5 Architecture

**Storage Gateways**
- POSIX Gateway
  - Project: /project
  - Scratch: /scratch
- Amazon Web Services (S3)

**Collections**
- Data access interface
  - GridFTP & HTTPS

**Endpoint**
- Management & config interface

**Data Transfer Nodes**
- Network & storage connected servers in ScienceDMZ

**Mapped**

**Guest**
High Assurance

**Connector**
- POSIX
- AWS S3
Installation register GCS client with Globus Auth; first-timers will need to create a Project to contain the registration.

Define Globus resources (gateways, collections, ...)

GCS Manager endpoint: abc.abc.data.globus.org
GCSv4 will be deprecated July 2023 and discontinued December 2023

Migration tools are available, please update!
GCS v5 install walkthrough

docs.globus.org/globus-connect-server

docs.globus.org/globus-connect-server/v5.4/quickstart
Key prerequisite: network accessibility

- Yes, you must have a chat with OpSec, NetOps, ...

- Control channel: must be on publicly routable IP
  - Default: port 443; configurable
  - Inbound and outbound traffic from all
  - Can be restricted to smaller CIDR block but limits functionality

- Data channel: can be on private network
  - Default: 50000-51000
  - Configurable, but *strongly advise against it*
1. Register a Globus Connect Server with Globus Auth
2. Install GCS packages on data transfer node (DTN)
3. Set up the endpoint and add node(s)
4. Create a POSIX storage gateway
5. Create a mapped collection
6. Associate endpoint with a subscription
7. Create a guest collection
8. Enable browser down/upload (HTTPS access)
9. Add other storage systems to the endpoint

Requires a Globus subscription
1. Install Globus Connect Server v5 packages

$ dpkg -i globus-repo_latest_all.deb
$ apt-key add /usr/share/globus-repo/RPM-GPG-KEY-Globus
$ apt-get update
$ apt-get --assume-yes install globus-connect-server54

Already done on your EC2 instances.
Endpoint creation and node setup
2. Set up endpoint and add node

$ globus-connect-server endpoint setup \
> "My Endpoint" \
> --organization "My Organization" \
> --contact-email me@uchicago.edu \
> --owner me@uchicago.edu

$ sudo globus-connect-server node setup

Note: endpoint setup command generates `deployment-key.json`
Use this file when setting up additional data transfer nodes

Identity must be known to Globus Auth; log in and confirm prior to endpoint setup
Set up endpoint and add a DTN

- **Access server:** ssh admin@tut.globusdemo.org
- **Switch to root:** sudo su
- **Run:** `globus-connect-server endpoint setup ...`
  
  - Ensure `--owner` is the identity you used to register the GCS
- **Run:** `globus-connect-server node setup ...`
- **Run:** `systemctl restart apache2`
- **Display endpoint details:**
  
  - `globus-connect-server login localhost`
  - `globus-connect-server endpoint show`

Cheatsheet: [bit.ly/gw-tut]
Our setup so far

Run `globus-connect-server` node setup to set up additional data transfer nodes

Copy deployment-key.json from original DTN

ENDPOINT
Management & config interface

DATA TRANSFER NODES
Network & storage connected servers in ScienceDMZ
Storage Gateways define a set of access policies

• **Authentication for local account-holders**
  – Which identity domain(s) are acceptable?
  – How are identities mapped from domain(s) to local accounts?

• **Policy scope**
  – Which parts of the storage system are accessible via Globus?
  – Which local accounts does this policy allow (or deny)?

• **High Assurance settings**

• **MFA requirements**
Authentication for local account-holders

- **Primary access (via a mapped collection) requires an account on the host system** *

- **Two-part authentication configuration:**
  1. Pick one or more identity domains
  2. Configure the method to map the authenticated identity to an account on your system

* You may allow primary users to share with others who don’t have accounts on your system
Picking identity domains

• **User must present identity from one of the configured domains**
  – On access attempts, linked identities will be scanned for a match
  – If no identity from the required domain(s), will be asked to link one

• **Identity domains may include…**
  – …any organization in Globus federated list
  – …your institution’s identity provider trusted by Globus
  – …a local OpenID Connect (OIDC) server using your PAM stack
Mapping identities to local accounts

• Default: Strip identity domain (everything after “@”)
  – e.g. `userX@globusdemo.org` maps to local account `userX`
  – Best for campus identities w/synchronized local accounts

• Use `--identity-mapping` option on storage gateway
  – Specify expression in a JSON document
  – Execute a custom script

[docs.globus.org/globus-connect-server/v5.4/identity-mapping-guide/]
Create a POSIX storage gateway
Creating a storage gateway

- Our storage gateway will access a POSIX system
  - This is the only type permitted without a subscription
- It will allow access to users with credentials from the globusid.org (or your own) domain
- Reauthentication will be required every 90 minutes

Cheatsheet: bit.ly/gw-tut
3. Create a storage gateway

$ globus-connect-server storage-gateway create posix \
> "My Storage Gateway" \
> --domain globusid.org \
> --authentication-timeout-mins 90

Allowed authentication domain

Duration of user session when accessing collections via this storage gateway
Our setup so far…

STORAGE GATEWAY
POSIX 1
/project
High Assurance

STORAGE GATEWAY
POSIX 2
/scratch

STORAGE GATEWAY
Amazon Web Services S3

STORAGE GATEWAY
CONNECTORS
POSIX
AWS S3
DATA TRANSFER
NODE 1
NODE 2

ENDPOINT
Management & config interface

DATA TRANSFER NODES
Network & storage connected servers in ScienceDMZ

POLICIES & CONFIGURATION

Create a mapped collection on the POSIX gateway
Creating a collection

• Our collection will use the default identity mapping
• It will be “rooted” at the user’s home directory
• Access will require authentication with an identity from the globusid.org (or your own) domain

Cheatsheet
bit.ly/gw-tut
4. Create a mapped collection

$ globus-connect-server collection create \
  > f77ff456-1f18-41d3-94a7-f3fd8858ea4d \
  > /\ 
  > "My Mapped Collection"

Collections are rooted at the specified base path

Specifying "/" as the base path sets the collection root to the local user’s home directory
Our setup so far…

COLLECTIONS
- Data access interface
  - GridFTP & HTTPS

STORAGE GATEWAY
- POSIX 1 /project
- POSIX 2 /scratch
- Amazon Web Services S3
  - High Assurance

ENDPOINT
- Management & config interface

DATA TRANSFER NODES
- Network & storage connected servers in ScienceDMZ
Understanding access to mapped collections
Access our mapped collection
We are using the default identity mapping, so...

- Create a local user account with the same name as your globusid.org (or other IdP) identity
  - e.g., for me@globusid.org create local account “me”
  - e.g., for me@orcid.org create local account “me”
- adduser --disabled-password --gecos 'me' me
- Access your mapped collection via the web app...
- ...and move some files, if you like
Common Collection configuration options

• Restrict access: local users, local groups
• Restrict sharing: paths, local users, local groups
• Allow guest collections → enables sharing
• Enable HTTPS access
• Force data channel encryption
Local account restrictions

• Note: These only apply to mapped collections
• A storage gateway’s allowed identity domains and identity mapping method determine the universe of local accounts that *may* access the mapped collection
• You can further narrow the access universe using…
  --user-allow
  --user-deny
  --posix-group-allow (POSIX storage gateways only)
  --posix-group-deny (POSIX storage gateways only)
Path restrictions

• Always use the narrowest *base path* possible for your storage gateway(s) and collection(s)
  – Storage gateway base specifies where collections may be created
  – Collection base specifies the base directory for the collection

• POSIX storage gateway
  – Use `--restrict_paths` to specify narrower read, read/write, or none access for specific paths
  – You provide a JSON doc that lists paths for each permission type
  – Note: These are absolute paths on the host system

• Collection: specify narrowest base path that satisfies the need
Restrict collection access to filesystem
Setting path restrictions

• A new storage gateway will limit access to /home
  – NB: No change to local permissions, only visibility via Globus

• We specify the path restrictions in paths.json
  – This file is in your admin user’s home directory

• Run: storage-gateway create command with the
  --restrict-paths option

• Create a new POSIX mapped collection

Cheatsheet
bit.ly/gw-tut
5. Create a restricted storage gateway, collection

$ globus-connect-server storage-gateway create posix \ 
> "My Storage Gateway - Restricted" \ 
> --domain globusid.org \ 
> --authentication-timeout-mins 90 \ 
> --restrict-paths file:/home/adminN/paths.json

$ globus-connect-server collection create \ 
> 3926bf02-6bc3-11e7-a9c6-22000bf2d287 \ 
> / \ 
> "My Mapped Collection - Restricted"

Fully qualified filename containing rule(s) for restricting access to specific filesystem paths
Revisit your mapped collections

• Your will need to authenticate on your new (restricted access) collection, and consent

• Note the access behavior differences between the two mapped collections
Subscriptions and Endpoint Roles

- Subscription(s) configured for your institution
- Multiple Subscription Managers per subscription
- Subscription Manager ties endpoint to subscription
  - Results in a “managed” endpoint
- Assign additional roles for endpoint management
  - Administrator, Manager, Monitor
Associate the endpoint with a subscription
Making your endpoint “Managed”

- Subscription managers can enable subscription features on an endpoint
- If you are not the subscription manager, send your endpoint ID to your subscription manager and ask them to add it.
Making your endpoint “Managed”

- Option A (for subscription managers): Run `globus-connect-server endpoint set-subscription-id`
- Option B: Put your endpoint ID in the spreadsheet and we will make it managed
- Confirm: `globus-connect-server endpoint show`

Cheatsheet: [bit.ly/gw-tut](bit.ly/gw-tut)
6. Associate endpoint with a subscription

```bash
$ globus-connect-server endpoint set-subscription-id DEFAULT
$ globus-connect-server endpoint set-subscription-id \
  > 39299902-6bc3-aa56-a9c6-22000bf2d287
```

Subscription managers can also set this via the web app Console page: app.globus.org/console (look under the Endpoints tab)
Be identity-, role-, and permission-aware

- Default: Only endpoint owner can configure an endpoint
- Delegate administrator role to other sysadmins
  - Best practice: Delegate to a Globus group, not individuals
- Check identity using the session command
- Check resource permissions on storage gateways and collections with `--include-private-policies` option

[docs.globus.org/globus-connect-server/v5.4/reference/role/](https://docs.globus.org/globus-connect-server/v5.4/reference/role/)
7. Create a guest collection

- Created by user, not endpoint administrator
- Grants access to specific Globus users without a mapped local account
- **“Guest”** users have the same (or more limited) permissions as the guest collection creator
  - Access logs show access by the collection creator*
- Guest collection’s root is relative to the mapped collection’s base path

* High Assurance collections log guest user identities to enable auditing
Sharing restrictions

• Guest collections may be created in any directory accessible by the collection, by any authorized local account

• You can restrict the authorized accounts…
  o --sharing-user-allow
  o --sharing-user-deny
  o --posix-sharing-group-allow
  o --posix-sharing-group-deny

• …and sharing paths…
  o --sharing-restrict-paths (specify JSON PathRestrictions)

• You can also set policies for specific user/path combinations
  o $ globus-connect-server sharing-policy create ...
Create and access a guest collection
Create and access a guest collection

- Enable creation of guest collections
- **Run:** `globus-connect-server collection update`
- Access the mapped collection; create `/projects`
- Create a guest collection on the `/projects` directory
- Grant read access to the “Tutorial Users” group
- Authenticate and browse guest collection
8. Enable web browser upload/download

- Authorized users can upload, download files via a browser
- Must have permissions to the collection
  - Collection configuration governs access
  - Web server is a different application (separate authentication)
Enable/disable file download/upload via browser
Enable HTTPS access

• **Run**: `globus-connect-server collection update`
• Access your mapped collection
• Upload a file from your laptop (and download it too!)

Using the management console
Things to do with the management console

- **Monitor current transfers on your endpoints**
  - See what’s going on at the transfer request level
  - Much better than watching individual file transfers
- **Pause (and later resume) a transfer in progress**
  - Sends a notice to the transfer owner
- **Set a pause rule for current and future transfers**
  - Ideal for maintenance mode
  - Notifies transfer owners,
  - Tasks resume when endpoint is un-paused

[docs.globus.org/management-console-guide/](https://docs.globus.org/management-console-guide/)
Resources

- GCSv5 Guides: docs.globus.org/globus-connect-server/
- Migration: docs.globus.org/globus-connect-server/migrating-to-v5.4/
- Globus support: support@globus.org