The Galaxy Toolshed

usegalaxy.org/toolshed
The Galaxy API
API Foundation

- Technologies
- Representational State Transfer (REST)
- Sessionless operations via HTTP
- JavaScript Object Notation (JSON)
Other API Interfaces

- Library permissions
- Forms
- Sample tracking requests and samples
- Manage users, roles, and quotas
- Execute tools and workflows
End-to-end pipelines

- Automatically upload data retrieved from instruments
- Start a workflow
- Move outputs to a data library
Beneath the Clouds
Building your own Galaxy production service
usegalaxy.org/production
Galaxy runs out of the box!

- Simple download, setup, and install design:
  
  ```
  % hg clone ...
  % sh run.sh
  ```

- Great for development!

- Not designed to support multiple users in a production environment with default configuration
Development-oriented defaults

- SQLite database
- One process
- Built-in HTTP server
- Local job execution
Start Fresh

• Don’t use an old Galaxy installation - check out a new copy
• Use a dedicated non-root user
• Start and stop with your OS’ system service method (e.g. init.d, service)
• Don’t share the database or database user
• Use a dedicated Python or virtualenv
• If you plan to use a cluster, put galaxy in a shared filesystem
Galaxy Config Basics

- Use the sample config (copy before starting):
  - `% cp universe_wsgi.ini.sample universe_wsgi.ini`
- Read the full sample config
- Set:
  - `use_interactive = False` - Not even safe (exposes config)
  - `debug = False` - You'll still be able to see tracebacks in the log file, doesn't load response in memory
# --- Galaxy -----------------------------------------------------------------

# Configuration of the Galaxy application.

[app:main]

# By default, Galaxy uses a SQLite database at 'database/universe.sqlite'. You
# may use a SQLAlchemy connection string to specify an external database
# instead. This string takes many options which are explained in detail in the
# config file documentation.
#database_connection = sqlite:///./database/universe.sqlite
database_connection = postgres:///galaxy

# -- Data Libraries

# These library upload options are described in much more detail in the wiki:
# http://wiki.g2.bx.psu.edu/Admin/Data%20Libraries/Uploading%20Library%20Files

# Add an option to the library upload form which allows administrators to
# upload a directory of files.
#library_import_dir = None
library_import_dir = /Users/nate/import
Administration

The menu on the left provides the following features

- **Security** – see the Data Security and Data Libraries section below for details
  - **Manage users** – provides a view of the registered users and all groups and non-private roles associated with each user.
  - **Manage groups** – provides a view of all groups along with the members of the group and the roles associated with each group (both private and non-private roles). The group names include a link to a page that allows you to manage the users and roles that are associated with the group.
  - **Manage roles** – provides a view of all non-private roles along with the role type, and the users and groups that are associated with the role. The role names include a link to a page that allows you to manage the users and groups that are associated with the role. The page also includes a view of the data library datasets that are associated with the role and the permissions applied to each dataset.

- **Data**
  - **Manage data libraries** – Data libraries enable a Galaxy administrator to upload datasets into a data library. Currently, only administrators can create data libraries. When a data library is first created, it is considered “public” since it will be displayed in the "Data Libraries" view for any user, even those that are not logged in. The Galaxy administrator can restrict access to a data library by associating roles with the data library’s "access library" permission. This permission will conservatively override the [dataset] "access" permission for the data library’s contained datasets.

    For example, if a data library's "access library" permission is associated with Role1 and the data library contains "public" datasets, the data library will still only be displayed to those users that have Role1. However, if the data library's "access library" permission is associated with both Role1 and Role2 and the data library contains datasets whose [dataset] "access" permission is associated with only Role1, then users that have Role2 will be able to access the library, but will not see those contained datasets whose [dataset] "access" permission is associated with only Role1.

    In addition to the "access library" permission, permission to perform the following functions on the data library (and it's contents) can be granted to users (a library item is one of: a data library, a library folder, a library dataset).

    • **add library item** – Users that have the role can add library items to this data library or folder and modify library items. Users that have the role can modify this library item.
Galaxy Admin Interface

Administration

The menu on the left provides the following features:

- **Security** – see the Data Security and Data Libraries section below for details
  - **Manage users** – provides a view of the registered users and all groups and non-private roles
  - **Manage groups** – provides a view of all groups along with the members of the group and the roles associated with each group (both private and non-private roles). The group names include a link to a page that allows you to manage the users and roles that are associated with the group.
  - **Manage roles** – provides a view of all non-private roles along with the role type, and the users and groups that are associated with the role. The role names include a link to a page that allows you to manage the users and groups that are associated with the role. The page also includes a view of the data library datasets that are associated with the role and the permissions applied to each dataset.

- **Data**
  - **Manage data libraries** – Data libraries enable a Galaxy administrator to upload datasets into a data library. Currently, only administrators can create data libraries.
    - When a data library is first created, it is considered “public” since it will be displayed in the “Data Libraries” view for any user, even those that are not logged in. The Galaxy administrator can restrict access to a data library by associating roles with the data library’s “access library” permission. This permission will conservatively override the [dataset] “access” permission for the data library’s contained datasets.
    - For example, if a data library’s “access library” permission is associated with Role1 and the data library contains “public” datasets, the data library will still only be displayed to those users that have Role1. However, if the data library’s “access library” permission is associated with both Role1 and Role2 and the data library contains datasets whose [dataset] “access” permission is associated with only Role1, then users that have Role2 will be able to access the library, but will not see those contained datasets whose [dataset] “access” permission is associated with only Role1.
    - In addition to the “access library” permission, permission to perform the following functions on the data library (and its contents) can be granted to users (a library item is one of: a data library, a library folder, a library dataset).
      - **add library item** – Users that have the role can add library items to this data library or folder
      - **modify library item** – Users that have the role can modify this library item

- **Tool versions**
- **Profile memory usage**
- **Profile job usage**
- **Manage installed tool shed repositories**
- **Search and browse tool sheds**
- **Manage form definitions**
- **Manage sample tracking**
- **Manage sequencers and external services**
- **Manage request types**
Galaxy Admin Interface

Administration

The menu on the left provides the following features:

- **Security** - see the Data Security and Data Libraries section below for details
  - **Manage users** - provides a view of the registered users and all groups and non-private roles
  - **Manage groups** - provides a view of all groups along with the members of the group and the roles associated with each group (both private and non-private roles). The group names include a link to a page that allows you to manage the users and groups that are associated with the role. The page also includes a view of the data library datasets that are associated with the role and the permissions applied to each dataset.

- **Manage roles** - provides a view of all non-private roles along with the role type, and the users and groups that are associated with the role. The role names include a link to a page that allows you to manage the users and groups that are associated with the role. The page also includes a view of the data library datasets that are associated with the role and the permissions applied to each dataset.

- **Data**
  - **Manage data libraries** - Data libraries enable a Galaxy administrator to upload datasets into a data library. Currently, only administrators can create data libraries.
    - When a data library is first created, it is considered "public" since it will be displayed in the "Data Libraries" view for any user, even those that are not logged in. The Galaxy administrator can restrict access to a data library by associating roles with the data library's "access library" permission. This permission will conservatively override the [dataset] "access" permission for the data library's contained datasets.
    - For example, if a data library's "access library" permission is associated with Role1 and the data library contains "public" datasets, the data library will still only be displayed to those users that have Role1. However, if the data library's "access library" permission is associated with both Role1 and Role2 and the data library contains datasets whose [dataset] "access" permission is associated with only Role1, then users that have Role2 will be able to access the library, but will not see those contained datasets whose [dataset] "access" permission is associated with only Role1.
    - In addition to the "access library" permission, permission to perform the following functions on the data library (and its contents) can be granted to users (a library item is one of: a data library, a library folder, a library dataset):
      - **add library item** - Users that have the role can add library items to this data library or folder
  - **Modify data libraries** - Users that have the role can modify this library.
Galaxy Admin Interface

- Fine-grained data access permissions
- Disk quotas and data libraries
- Job monitoring and management
Galaxy Admin Interface

Administration

The menu on the left provides the following features:

- **Security** – see the Data Security and Data Libraries section below for details
  - **Manage users** – provides a view of the registered users and all groups and non-private roles associated with each group (both private and non-private). The group names include a link to a page that lists all users associated with each group.
  - **Manage groups** – provides a view of all groups along with the members of the group and the roles associated with each group (both private and non-private). The group names include a link to a page that lists all users associated with the group.
  - **Manage roles** – provides a view of all non-private roles along with the role type, and the users and groups that are associated with the role. The role names include a link to a page that allows you to manage the associated users and groups.

- **Data**
  - **Manage installed tool shed repositories**
  - **Search and browse tool sheds**

- **Install tools from the Tool Shed**

- **Fine-grained data access permissions**

- **Disk quotas and data libraries**

- **Job monitoring and management**

For example, if a data library's "access library" permission is associated with Role1 and the data library contains "public" datasets, the data library will still only be displayed to those users that have Role1. However, if the data library's "access library" permission is associated with both Role1 and Role2 and the data library contains datasets whose [dataset] "access" permission is associated with only Role1, then users that have Role2 will be able to access the library, but will not see those contained datasets whose [dataset] "access" permission is associated with only Role1.

In addition to the "access library" permission, permission to perform the following functions on the data library (and it's contents) can be granted to users (a library item is one of: a data library, a library folder, a library dataset).

- **add library item** – Users that have the role can add library items to this data library or folder
- **modify library items** – Users that have the role can modify this library item.
Galaxy Admin Interface

- Fine-grained data access permissions
- Disk quotas and data libraries
- Job monitoring and management
- Install tools from the Tool Shed
- Core facility sample tracking
Galaxy Admin Interface

Administration

The menu on the left provides the following features:

- **Security** – see the Data Security and Data Libraries section below for details
  - Manage users – provides a view of the registered users and all groups and non-private roles associated with each user.
  - Manage groups – provides a view of all groups along with the members of the group and the roles associated with each group (both private and non-private). The group names include a link to a page that lists all the users and roles associated with the group.
  - Manage roles – provides a view of all non-private roles along with the role type, and the users and groups that are associated with the role. The role names include a link to a page that allows you to manage the role settings.
- **Data**
  - Manage data libraries – allows you to manage and upload datasets into a data library. Currently, only administrators can create data libraries.
  - Install tools from the Tool Shed
- **Core facility sample tracking**

These features are only available once you define `admin_users`!
A few more options

- Enable browsers like UCSC, GBrowse and **Galaxy Trackster**
  - `ucsc_display_sites`, `gbrowse_display_sites`, `enable_tracks`
- “sudo” for Galaxy: `allow_user_impersonation`
- Publishing features: `enable_pages`
- Disk quotas: `enable_quotas`
Get a real database

- SQLite is serverless
- Galaxy is a heavy database consumer
- Locking will be an immediate issue
- Consumes Galaxy server process resources
- Migrating data is no fun
- Setup is very easy:
  
  ```
  database_connection = postgres://
  ```
victory# apt-get install postgresql
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
   libpq5 postgresql-9.1 postgresql-client-9.1 postgresql-client-common postgresql-common
Suggested packages:
   oidentd ident-server locales-all postgresql-doc-9.1
The following NEW packages will be installed:
   libpq5 postgresql postgresql-9.1 postgresql-client-9.1 postgresql-client-common postgresql-common
0 upgraded, 6 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/8,951 kB of archives.
After this operation, 22.6 MB of additional disk space will be used.
Do you want to continue [Y/n]? Y
   ... magic happens ...
victory# su - postgres
postgres@victory:~$ createuser -SDR galaxy
postgres@victory:~$ createdb -O galaxy galaxy
postgres@victory:~$
Offload the menial tasks: Proxy

- Directly serve static content faster than Galaxy's HTTP server
- Reduce load on the application
- Caching and compression
- Load balancing (more on that later)
- Hook your local authentication and authorization system
Proxy Options

- Server
  - nginx
    - Designed with proxy as the primary purpose
    - Has an upload module
    - The proxy used for usegalaxy.org and Cloud Galaxy
  - Apache
    - More authentication and other 3rd party modules
Downloading data from Galaxy

client

send me a dataset

data

proxy

can client access dataset?

yes, here's the data

galaxy

data

disk
Downloading data from the proxy

client

send me a dataset

data

proxy

can client access dataset?

yes, here's the path

access path

data

galaxy

disk
Downloading data from the proxy

- The proxy server can send files much faster than Galaxy's internal HTTP server and file I/O methods
- Reduce load on the application, free the process
- Restartability
- Security is maintained: the proxy consults Galaxy for authZ
- Proxy server requires minimal config and then:
  - nginx: `nginx_x_accel_redirect_base = /_download`
  - Apache: `apache_xsendfile = True`
Uploading data to Galaxy
Uploading data
to the proxy (nginx)
Uploading data
to the proxy

- The proxy is also better at receiving files than Galaxy
- Again, reduce load on the application, free the process
- Again, restartability
- More reliable
- Slightly more complicated to set up, and nginx only
Uploading data
from a local filesystem

- Many browsers have file size limitations
- Interrupted uploads cannot be resumed
- You may want to upload directly from a server
- Perhaps your data is already on a filesystem locally accessible to the Galaxy server
Uploading data
from a local filesystem

- For data libraries
- For histories
Uploading data
from a local filesystem

• Non-admin users may also upload to libraries from the local filesystem if granted permission and user_library_import_dir is set
Uploading data
via FTP

- FTP not explicitly required, cp, scp, sftp, whatever method your users have to place data on the server
- Full config example for ProFTPD with user authentication against Galaxy available in Wiki
Caching data locally

• Some data (e.g. sequences and associated indexes) are useful to many but should automatically be recognized as available by tools
• Placing in a data library and requiring history import every time would be a nuisance
• Avoid duplication and wasted time repeatedly building indexes on the same sequences
• Manage locally cached data in Galaxy
Limitless tool resources:
Use a cluster (or two)

- Move intensive processing (tool execution) to other hosts
- Utilize existing resources
- No job interruption upon restart
- Per-tool cluster options
- DRMAA supports most other DRMs
- It's easy: Set `start_job_runners` and `default_cluster_job_runner` and go!
Per-tool Job Control

- `default_cluster_job_runner = pbs:///`
- `fastq_groomer = pbs://othercluster.example.org/groomerq`
- `bowtie_wrapper = pbs:///ngsq/-l nodes=1:ppn=8/`
Job users on the cluster

- By default, jobs run as the user Galaxy is started as.
- If your Galaxy users and cluster system users are identical, you may wish to run jobs on the cluster as the actual user.
- Galaxy uses `sudo` to change ownership of relevant files and submit the job to the cluster as the correct system user.
- Configurable for your specific environment.
Python and threading

- Galaxy is multi-threaded. No problem, right?
- Problem... Enter the Global Interpreter Lock

Galaxy Process

- Guido says: "run multiple processes instead of threads!"
Opening the bottleneck

- One job manager - responsible for dispatching jobs to handlers
- Many job handlers - responsible for preparing and finishing jobs, monitoring cluster queue(s)
- Many web servers
Defining extra servers is easy

[server:web_0]
port = 8000

[server:web_1]
port = 8001

[server:web_2]
port = 8002

...
Unload the Galaxy server

- Setting metadata is CPU intensive and will make Galaxy unresponsive
- Make a new process (better yet, run on the cluster!)
- All you need is: set_metadata_externally = True
- Run the data source tools on the cluster if they have access to the Internet
- Remove tool = local:// from config file
Data Management

- The Galaxy philosophy
  - Data is never overwritten
  - Data is never deleted
Data Management
filesystem choices

- Storage can easily be the bottleneck
- Your storage must scale with your cluster
- Transparent compression and deduplication can reduce usage drastically
- Suggestions
  - ZFS: usegalaxy.org relies on ZFS on Solaris
  - ZFS on FreeBSD stable, Native ZFS on Linux coming
  - Btrfs may be viable soon
Data Management

creating data

- By default, all Galaxy history and library datasets are assigned an ID and stored in `galaxy-dist/database/files/`
- Single directory = single massive filesystem
- Galaxy has a dataset abstraction layer to decouple from a single local filesystem: Object Store
  - Disk backend: single filesystem
  - Distributed backend: multiple filesystems
  - Amazon S3 backend in development
Data Management
creating data

object_store = distributed

<?xml version="1.0"?>
<backends maxpctfull="90">
  <backend id="pool1" type="disk" weight="5">
    <files_dir path="/pool/pool1/files"/>
    <extra_dir type="temp" path="/pool/pool1/tmp"/>
    <extra_dir type="job_work" path="/pool/pool1/work"/>
  </backend>
  <backend id="pool2" type="disk" weight="1">
    <files_dir path="/pool/pool2/files"/>
    <extra_dir type="temp" path="/pool/pool2/tmp"/>
    <extra_dir type="job_work" path="/pool/pool2/work"/>
  </backend>
</backends>
Data Management

Data is never removed from disk unless

- *allow_user_dataset_purge = True*
- users click “delete permanently”
- Solution: cleanup_datasets.py

- Run from cron to remove data from disk that has been deleted by the user (but not “deleted permanently”)
- Configurable deletion policy allows removal after data has been deleted for a specified number of days
Monitoring

- Monitor Galaxy

- Provided methods:
  - With cron/email using `galaxy-dist/cron/check_galaxy.sh`
  - With Nagios using `galaxy-dist/contrib/nagios`

- The provided scripts upload files and run jobs
Collecting Statistics

- The Galaxy Reports webapp
  - Modify `galaxy-dist/reports_wsgi.ini` for your environment
  - Start the webapp with `sh run_reports.sh`
### All Jobs for May 2012

Click Total Jobs to see jobs for that day

<table>
<thead>
<tr>
<th>Day</th>
<th>Date</th>
<th>User Jobs</th>
<th>Monitor Jobs</th>
<th>Total Jobs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Saturday</td>
<td>May 26, 2012</td>
<td>1115</td>
<td>155</td>
<td>1270</td>
</tr>
<tr>
<td>Friday</td>
<td>May 25, 2012</td>
<td>5138</td>
<td>193</td>
<td>5331</td>
</tr>
<tr>
<td>Thursday</td>
<td>May 24, 2012</td>
<td>4600</td>
<td>153</td>
<td>4753</td>
</tr>
<tr>
<td>Wednesday</td>
<td>May 23, 2012</td>
<td>4384</td>
<td>188</td>
<td>4572</td>
</tr>
<tr>
<td>Tuesday</td>
<td>May 22, 2012</td>
<td>5852</td>
<td>195</td>
<td>6047</td>
</tr>
<tr>
<td>Monday</td>
<td>May 21, 2012</td>
<td>3905</td>
<td>198</td>
<td>4103</td>
</tr>
<tr>
<td>Sunday</td>
<td>May 20, 2012</td>
<td>1423</td>
<td>193</td>
<td>1616</td>
</tr>
<tr>
<td>Saturday</td>
<td>May 19, 2012</td>
<td>1158</td>
<td>182</td>
<td>1340</td>
</tr>
<tr>
<td>Friday</td>
<td>May 18, 2012</td>
<td>4546</td>
<td>190</td>
<td>4736</td>
</tr>
<tr>
<td>Thursday</td>
<td>May 17, 2012</td>
<td>4679</td>
<td>199</td>
<td>4878</td>
</tr>
<tr>
<td>Wednesday</td>
<td>May 16, 2012</td>
<td>7378</td>
<td>185</td>
<td>7563</td>
</tr>
<tr>
<td>Tuesday</td>
<td>May 15, 2012</td>
<td>5962</td>
<td>200</td>
<td>6162</td>
</tr>
<tr>
<td>Monday</td>
<td>May 14, 2012</td>
<td>4803</td>
<td>179</td>
<td>4982</td>
</tr>
<tr>
<td>Sunday</td>
<td>May 13, 2012</td>
<td>1563</td>
<td>196</td>
<td>1759</td>
</tr>
<tr>
<td>Saturday</td>
<td>May 12, 2012</td>
<td>1555</td>
<td>151</td>
<td>1706</td>
</tr>
<tr>
<td>Friday</td>
<td>May 11, 2012</td>
<td>5778</td>
<td>57</td>
<td>5835</td>
</tr>
<tr>
<td>Thursday</td>
<td>May 10, 2012</td>
<td>4506</td>
<td>0</td>
<td>4506</td>
</tr>
<tr>
<td>Wednesday</td>
<td>May 09, 2012</td>
<td>4812</td>
<td>0</td>
<td>4812</td>
</tr>
<tr>
<td>Tuesday</td>
<td>May 08, 2012</td>
<td>5172</td>
<td>0</td>
<td>5172</td>
</tr>
</tbody>
</table>
usegalaxy.org/production
The Team