



JupyterHub + Globus: A Foundation for Interactive Data Science

Vas Vasiliadis
vas@uchicago.edu

NYSERNet– May 2, 2018



JupyterHub

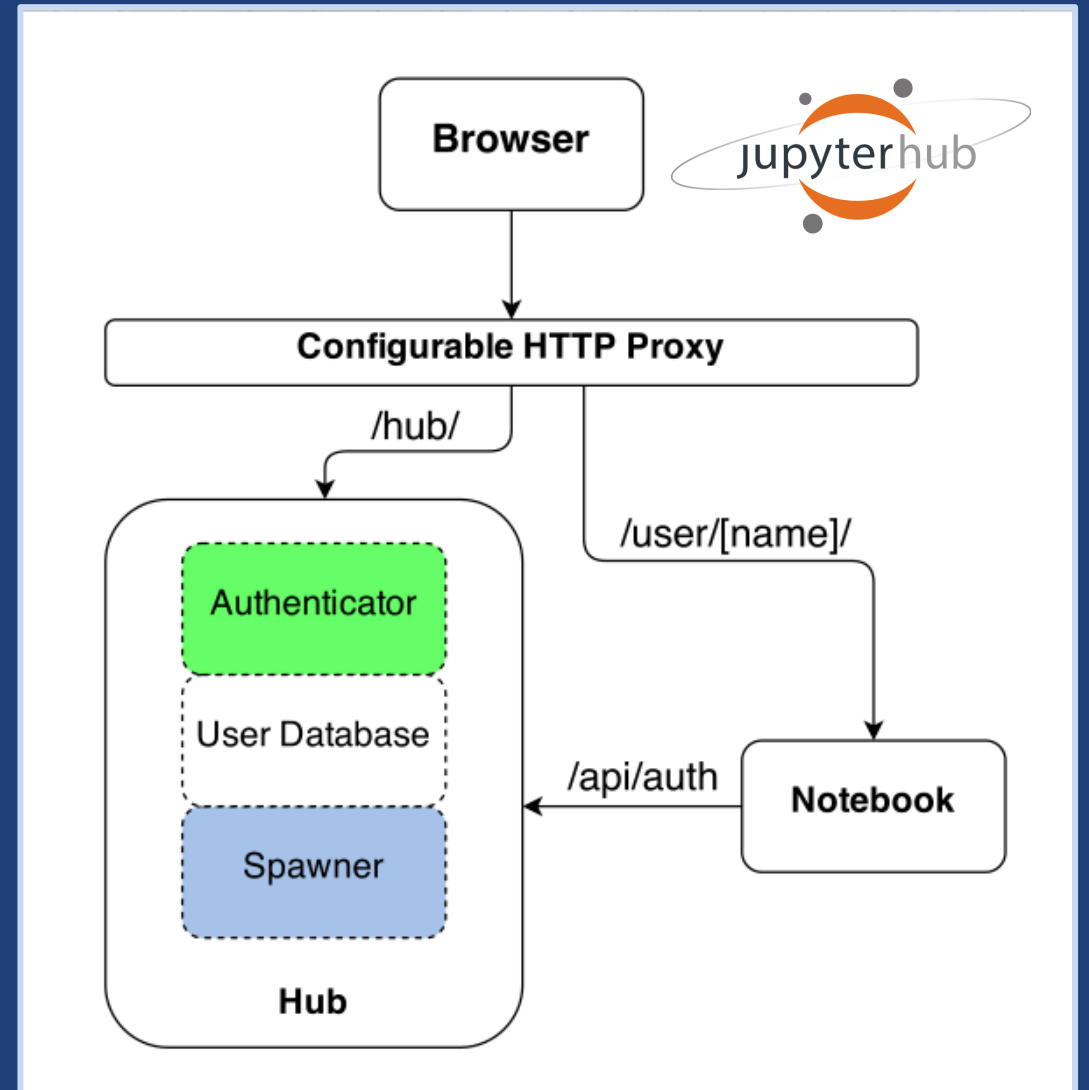


- **Multi-user Hub**
- **Manages multiple instances of the single-user Jupyter notebook server**
- **Serves notebooks to your research group, class of students, etc.**

jupyterhub.readthedocs.io/en/stable


JupyterHub Components

- **Multi-user Hub**
- **Configurable HTTP proxy**
- **Multiple single-user Jupyter notebook servers**
 - Python
 - R
 - ...



Securing JupyterHub with Globus Auth

- **Globus OAuth plugin**
 - Existing OAuth framework
 - Documentation covers app registration and config
 - Can restrict identity provider
 - Custom scopes
 - Tokens passed into notebook environment



The screenshot shows the GitHub interface for the `jupyterhub/oauthenticator` repository. At the top, there are navigation tabs for `Code`, `Issues` (21), `Pull requests` (5), `Projects` (0), and `Wiki`. Below the navigation, the current branch is `master`, and the file path is `oauthenticator / oauthenticator / globus.py`. A commit by `NickolausDS` is highlighted, with the message `Globus Auth: Added suggested change to reduce duplication`. Below the commit, it shows `3 contributors` with their avatars. The file statistics are `228 lines (187 sloc)` and `7.85 KB`. The code content is as follows:

```
1  """
2  Custom Authenticator to use Globus OAuth2 with JupyterHub
3  """
4  import os
5  import pickle
6  import base64
7
```



Securing JupyterHub with Globus Auth

Visit <https://developers.globus.org/> to set up your app. Ensure *Native App* is unchecked and make sure the callback URL looks like:

```
https://[your-host]/hub/oauth_callback
```

Set scopes for authorization and transfer. The defaults include:

```
openid profile urn:globus:auth:scope:transfer.api.globus.org:all
```

Set the above settings in your `jupyterhub_config` :

```
# Tell JupyterHub to create system accounts
from oauthenticator.globus import LocalGlobusOAuthenticator
c.JupyterHub.authenticator_class = LocalGlobusOAuthenticator
c.LocalGlobusOAuthenticator.enable_auth_state = True
c.LocalGlobusOAuthenticator.oauth_callback_url = 'https://[your-host]/hub/oauth_callback'
c.LocalGlobusOAuthenticator.client_id = '[your app client id]'
c.LocalGlobusOAuthenticator.client_secret = '[your app client secret]'
```

github.com/jupyterhub/oauthenticator#globus-setup



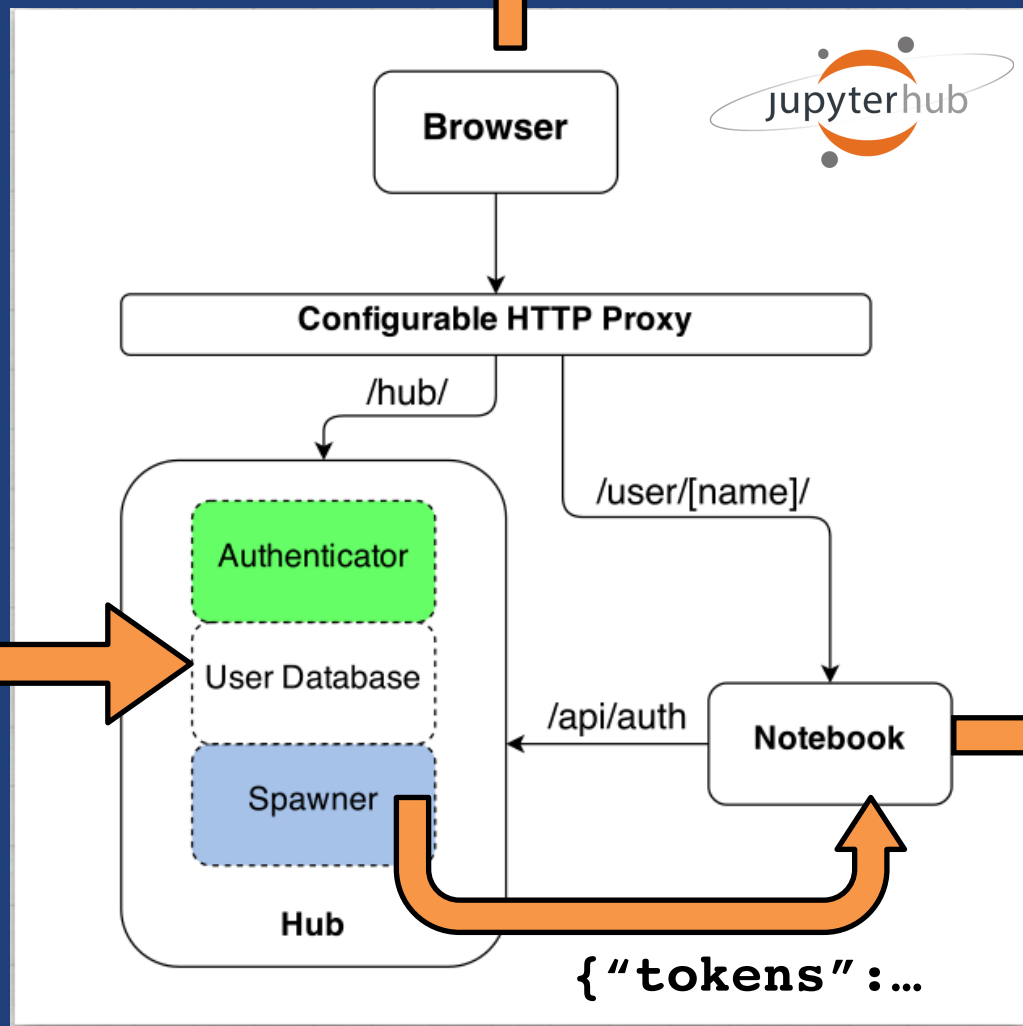
Tokens and Jupyter Notebooks

- **Tokens passed back to the JupyterHub server**
- **Stored as a secure attribute in database**
- **Passed into Notebook Server environment**
- **Can be pull into notebook or other code**
- **Use to communicate with...**
 - ...Globus services
 - ...other REST APIs secured with Globus Auth

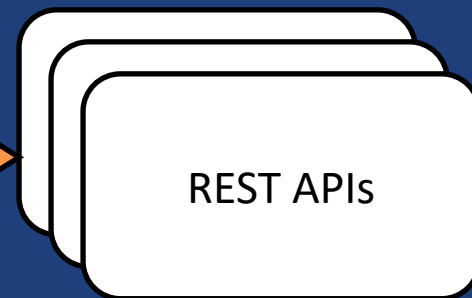


Tokens and Jupyter Notebooks

Login



Bearer a45cd...



REST APIs

`{"tokens": ...}`

`{"tokens": ...}`



Data Analysis Exercise

- **Log into our JupyterHub and launch Notebook Server**
- **Get tokens**
- **Access some Globus APIs**
- **Download some data**
- **Plot it**
- **PUT to an HTTPS endpoint**

Uses Zero to JupyterHub

zero-to-jupyterhub.readthedocs.io