jupyter

Jupyter + Globus: The Foundation for Interactive Data Science

Rick Wagner
rick@globus.org

April 26, 2018

globus

Agenda

- **JupyterHub Background**
- **How to Secure JupyterHub with Globus Auth**
- **Accessing Web Services via Jupyter Notebooks**
- **More Information**

- JupyterHub Background
- How to Secure JupyterHub with Globus Auth
- Accessing Web Services via Jupyter Notebooks
- More Information

JupyterHub: *A multi-user Hub, spawns, manages, and proxies multiple instances of the single-user Jupyter notebook server.*

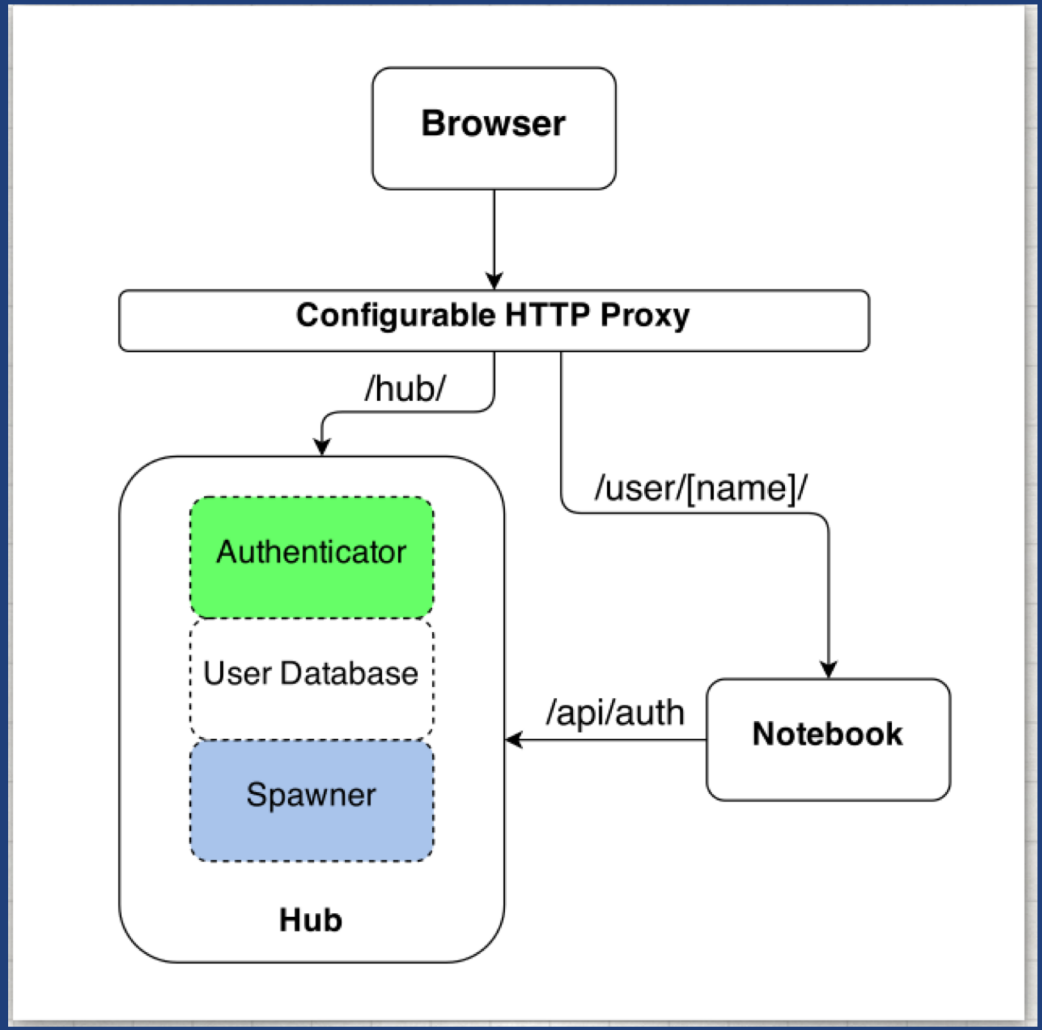
JupyterHub can be used to serve notebooks to a class of students, a corporate data science group, or a scientific research group.

JupyterHub Docs

Basic Jupyter Components

JupyterHub components

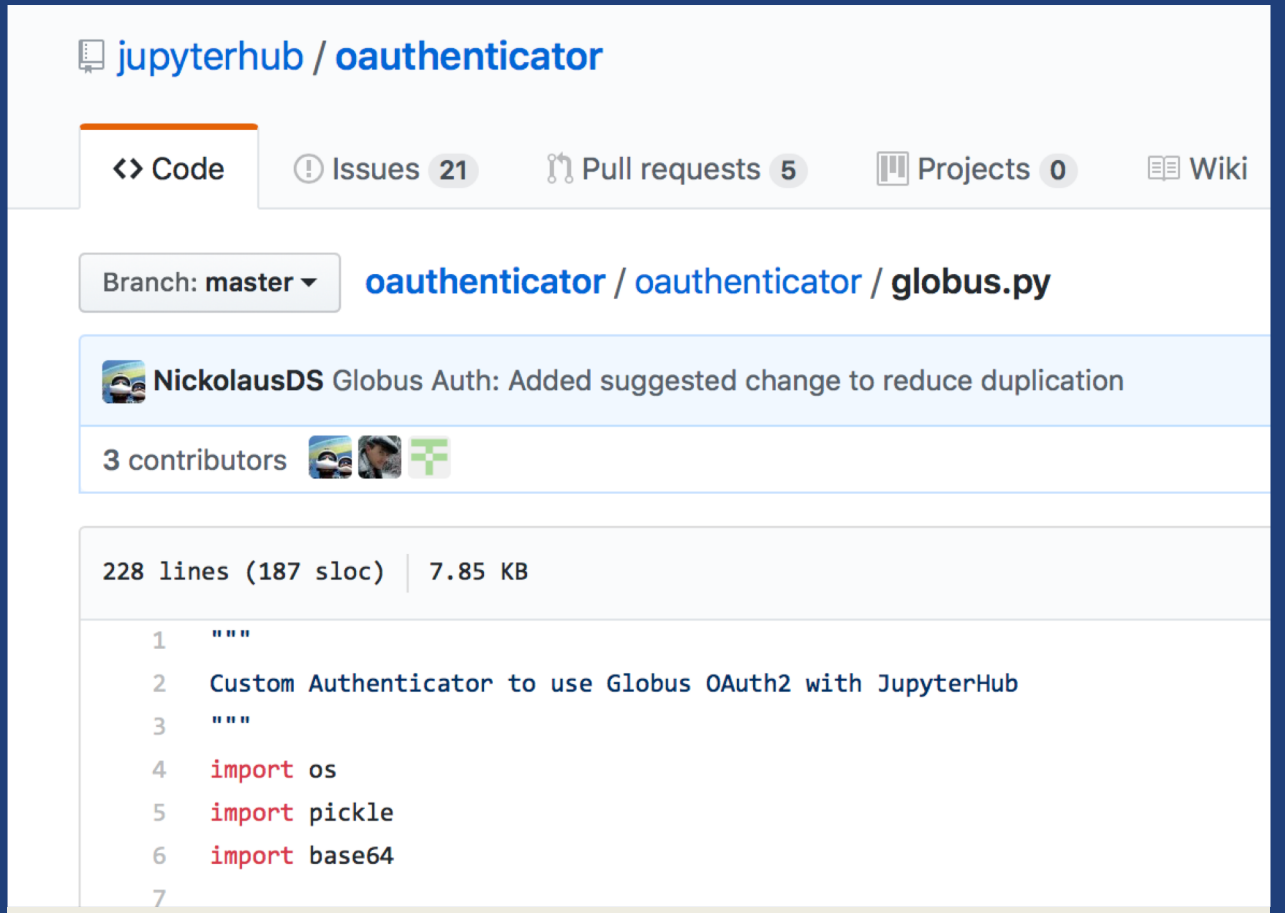
- **Multi-user Hub**
- **Configurable http proxy**
- **Multiple single-user Jupyter notebook servers**
  - Python
  - R
  - Etc.



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



[JupyterHub OAuthenticator]

Securing JupyterHub with Globus Auth

Globus Setup

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]'
```

https://github.com/jupyterhub/oauthenticator#globus-setup

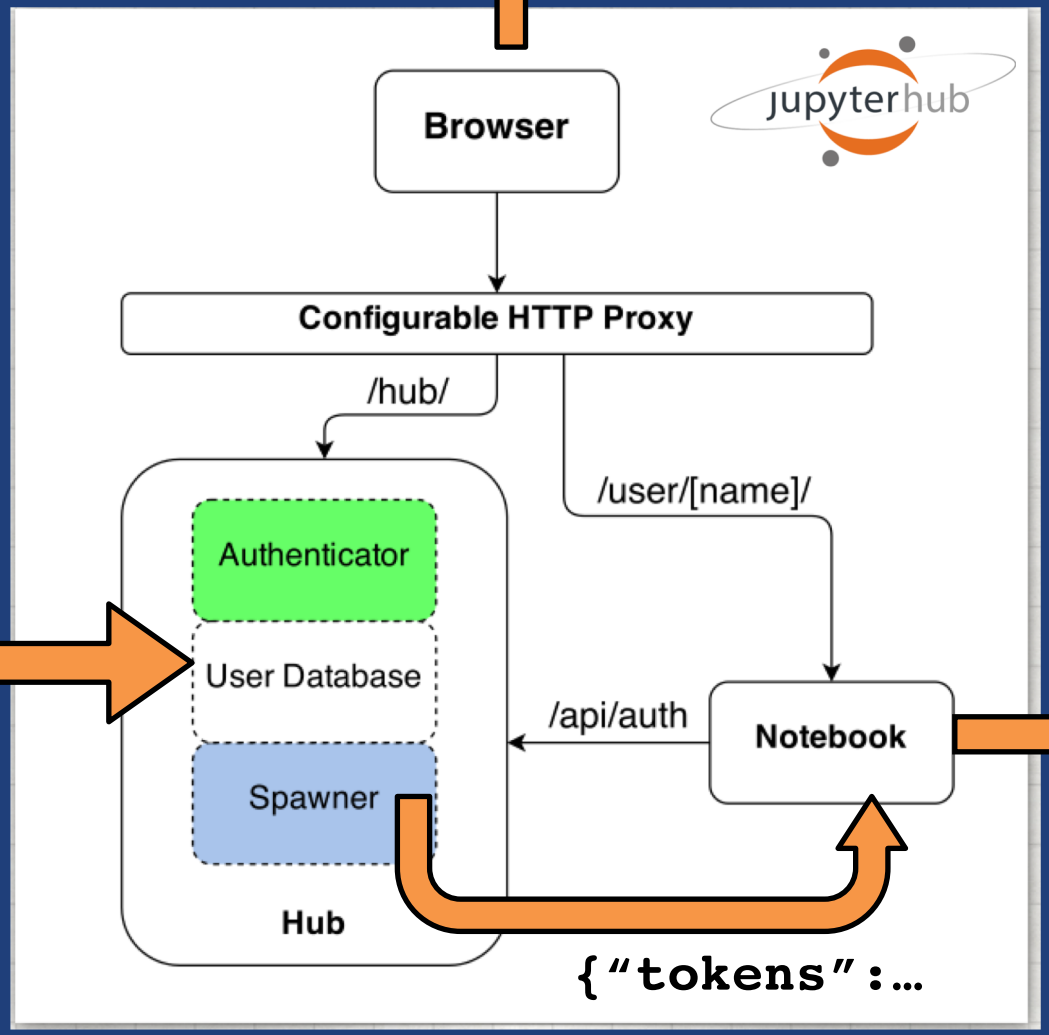
Tokens and Jupyter Notebooks

- Tokens are 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
- Used to talk to Globus and other REST APIs secured with Globus Auth

Tokens and Jupyter Notebooks

Login

globus



{ "tokens":…

Bearer a45cd…

REST APIs

Tutorial

What we're going to do:

- **Login into our GlobusWorld JupyterHub**
- **Launch (spawn) Notebook Server**
- **Get tokens**
- **Access some Globus APIs**
- **Download some data**
- **Plot it**
- **PUT it on an HTTPS endpoint**

Zero to JupyterHub: Fast JupyterHub on Kubernetes

`https://zero-to-jupyterhub.readthedocs.io`

Login to Start Tutorial

https://jupyter.demo.globus.org/

Future Goals: Containers