

Scaling up JupyterHub for Education

Jess Hamrick

UC Berkeley
Project Jupyter
@jhamrick



The Problem

Jupyter notebooks for 200+ users

Students must have persistent files on a real filesystem (e.g. for file I/O exercises)

Students must not be able to see each others' files

Integration with services like nbgrader

- 1. Scaling out with Docker and NFS**
- 2. Integration with services**

1. Scaling out with Docker and NFS

2. Integration with services

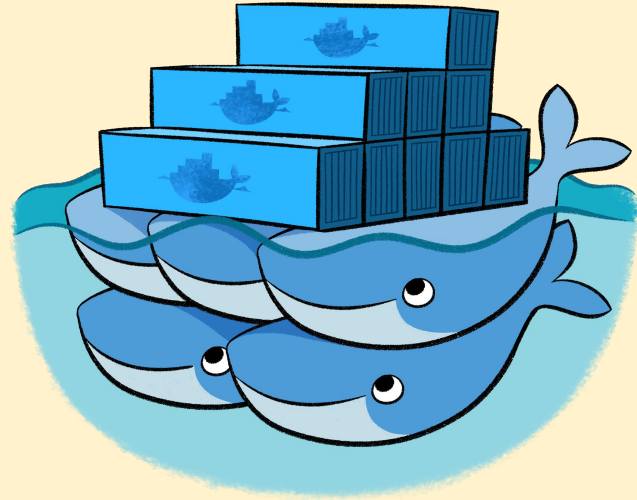
Challenge 1: Isolating users

Solution:



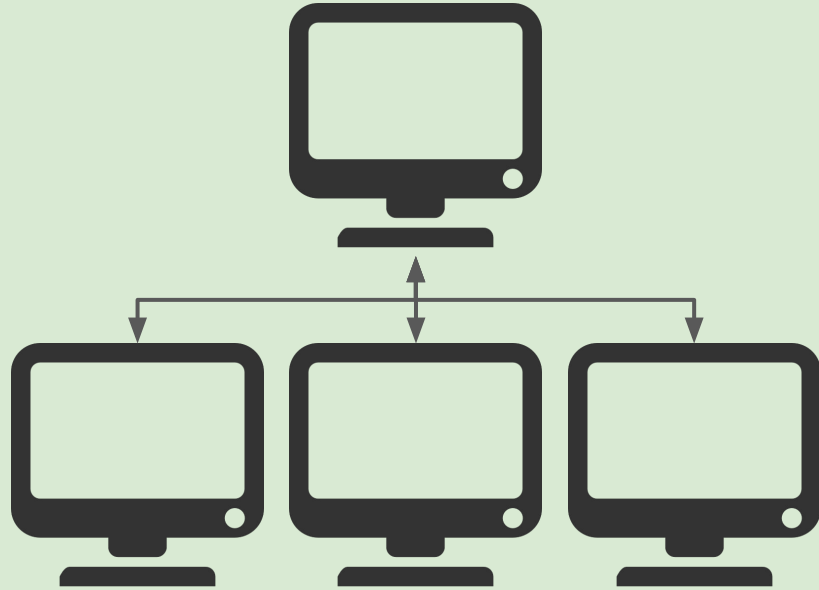
Challenge 2: Scaling up to ~220 users

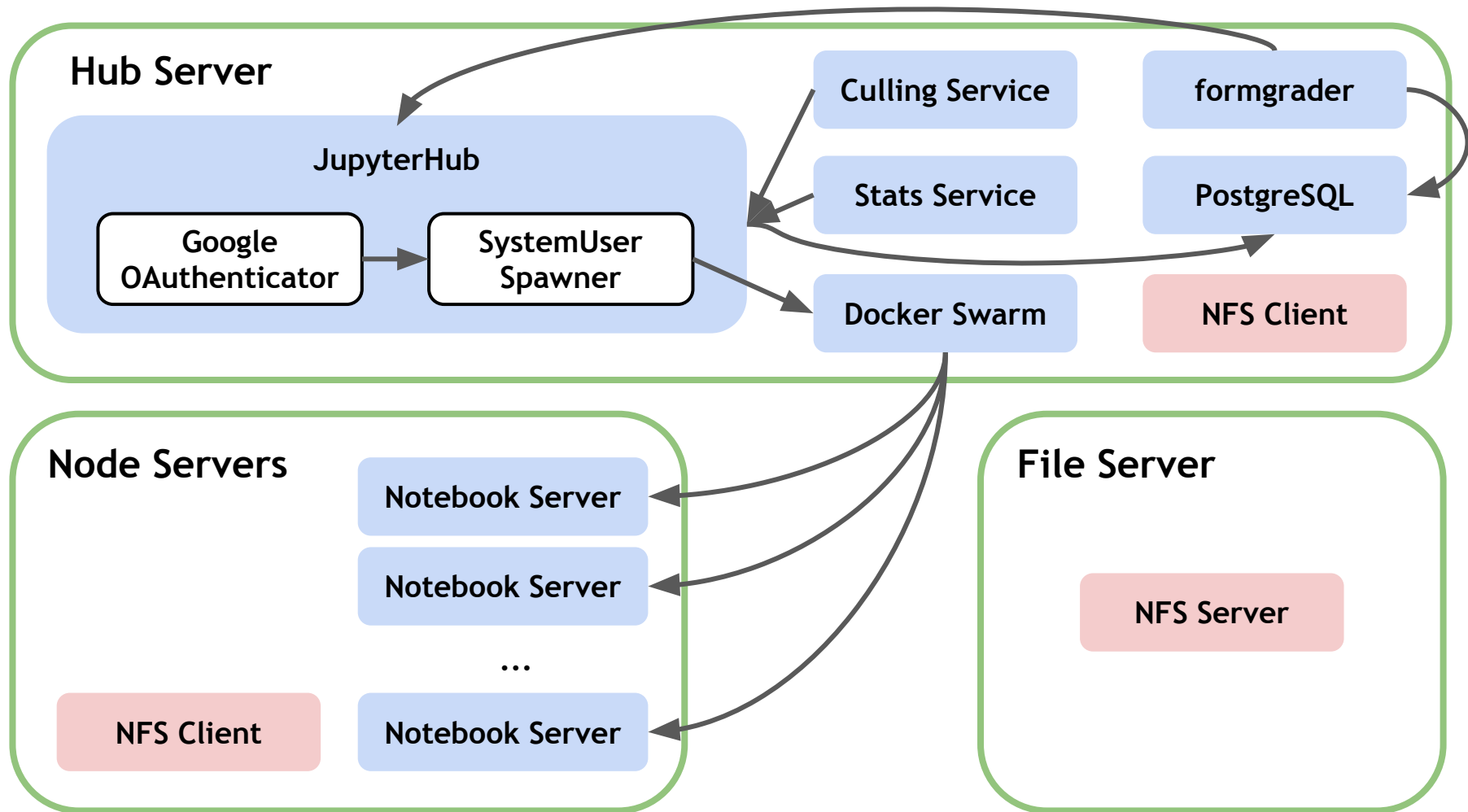
Solution:
Docker Swarm



Challenge 3: Persistent files

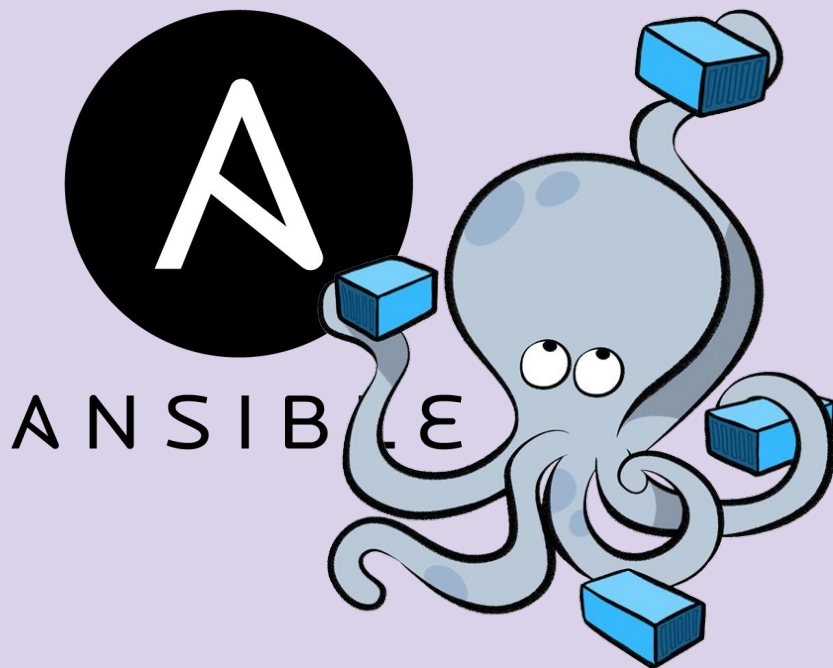
Solution:
NFS





Challenge 4: Managing everything 🤔

Solution:
Ansible &
Docker Compose



1. Scaling out with Docker and NFS

2. Integration with services

1. Scaling out with Docker and NFS

2. Integration with services

Culling Service

Shutting off idle single-user servers

Stats Service

Keeping track of who is using JupyterHub

Formgrader Service

Web application for grading Jupyter notebooks
(part of nbgrader)

Talking to JupyterHub

1. Get an API token:
 - jupyterhub token
2. Access the JupyterHub API, e.g.:
 - GET `http://<url>:8081/hub/api/users`
 - Headers: `{'authorization': 'token <hub_api_token>'}`
3. Note: the API is currently undocumented, use at your own risk!
 - <https://github.com/jupyterhub/jupyterhub/tree/master/jupyterhub/apihandlers>

Talking to JupyterHub

- See the stats and cull services for API examples:
 - Stats: https://github.com/minrk/log_jupyterhub_activity
 - Cull: https://github.com/jupyterhub/jupyterhub/blob/master/examples/cull-idle/cull_idle_servers.py

Getting a JupyterHub URL

1. Run your service at:

- `http://<service_url>:<service_port>`

2. Access the proxy API, e.g.:

- `POST http://<url>:8001/api/routes/myservice`
- Headers: `{'authorization': 'token <proxy_api_token>'}`
- Body: `{'target': 'http://<service_url>:<service_port>'}`

3. Access your service at:

- `http://<url>:8000/myservice`

Getting a JupyterHub URL

- Config proxy API documentation:
 - <https://github.com/jupyterhub/configurable-http-proxy>

Authenticating with JupyterHub

1. Get an API token:
 - jupyterhub token
2. Use the HubAuth service:
 - <http://jupyterhub.readthedocs.io/en/latest/api/services.auth.html>
3. Example use case: nbgrader formgrade

DockerSpawner and SystemUserSpawner:
<https://github.com/jupyterhub/dockerspawner>

Google OAuthenticator:
<https://github.com/ryanlovett/jh-google-oauthenticator>

Reference deployment with Docker:
<https://github.com/jupyterhub/jupyterhub-deploy-docker>

Ansible deployment (out of date!!):

<https://github.com/compmodels/jupyterhub-deploy>

Docker Compose template (out of date!!):

<https://github.com/compmodels/jupyterhub-deploy/blob/master/roles/hub/templates/docker-compose.yml>

JupyterHub Docker image (out of date!!):
<https://github.com/compmodels/jupyterhub>

Single-user server Docker image (out of date!!):
<https://github.com/compmodels/systemuser>

Stats service Docker image (out of date!!):

<https://github.com/compmodels/stats>

Culling service Docker image (out of date!!):

<https://github.com/compmodels/cull>

Formgrader service Docker image (out of date!!):

<https://github.com/compmodels/formgrader>