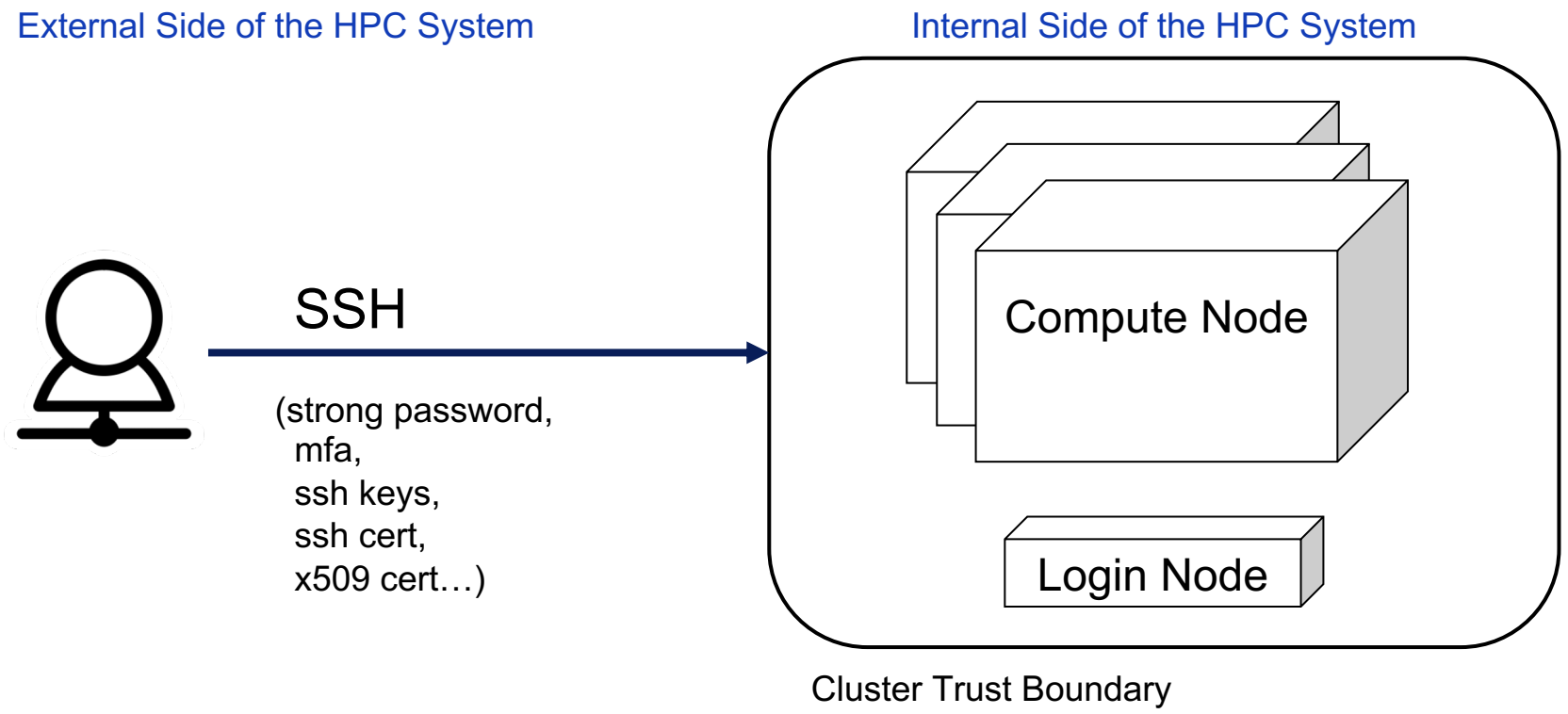
Running Secure Jupyter Notebooks Using the Satellite Proxy Service

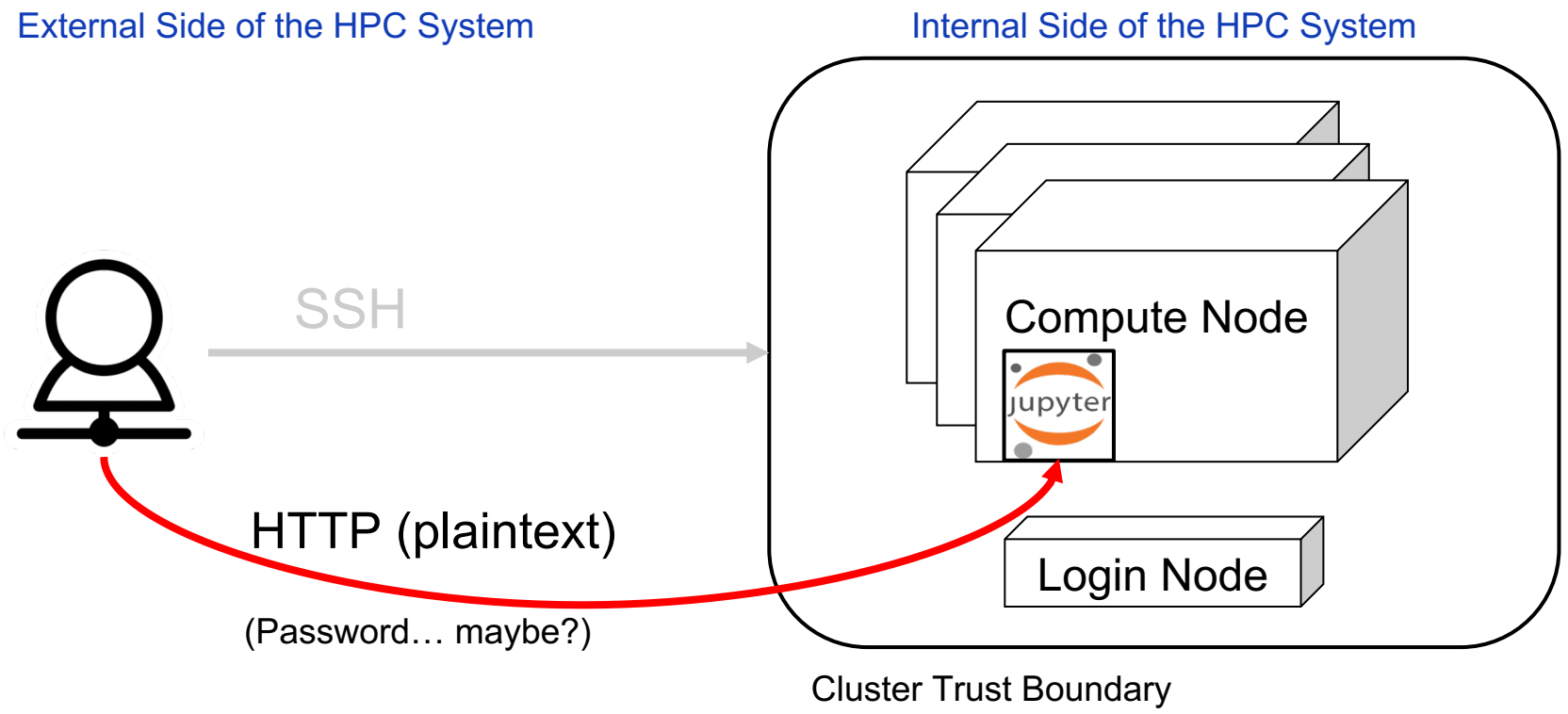
Presented by:
Mary Thomas (mpthomas at ucsd.edu)

CIML Summer Institute
6/18/2021

Introduction: Secure Connections to HPC Systems are Important



Introduction: Jupyter Notebooks are Popular



But provide a plaintext back-door to the system

Motivation: Make Doing A Right Thing Easier than The Wrong Things

A Wrong Thing: Plaintext to Compute Node

- Submit batch job.
- Wait till job runs.
- Figure out what node it's on.
- Point web browser at node.

A Right Thing: Improve secure access:

- Invoke the Satellite Reverse Proxy Service
- Point browser at secure, encrypted URL (HTTPS).
- (Wait until Jupyter Notebook shows up.)

SDSC Satellite Reverse Proxy Service

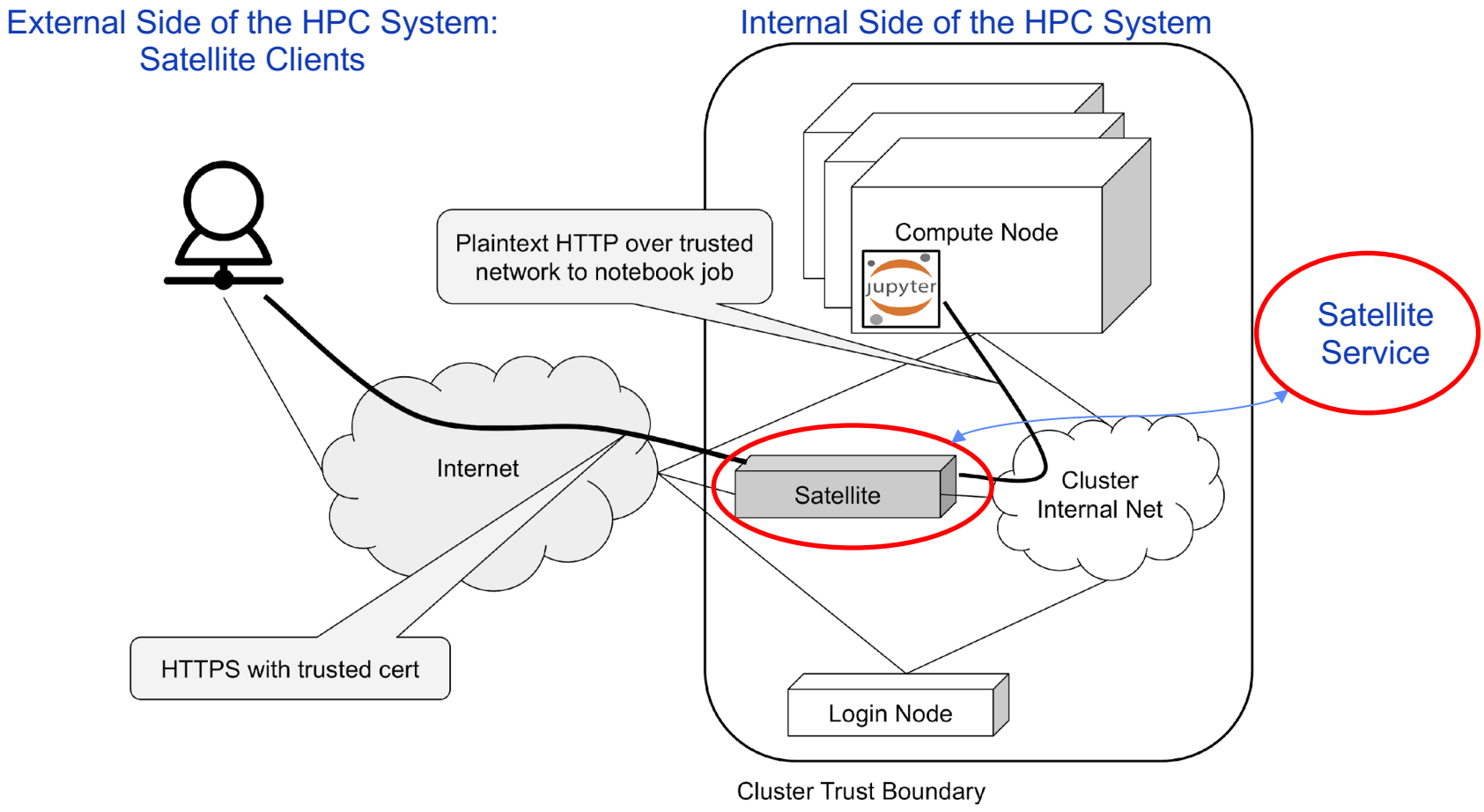
Just Two Components!

- Satellite: a self-service HTTP(s) reverse-proxy.
- Satellite Client: a shell-based utility to orchestrate a user's interaction with both Satellite and Slurm to start a Jupyter session within a batch job.

SDSC Satellite Proxy Service

GitHub Repo: https://github.com/sdsc-hpc-training-org/reverse-proxy

Jupyter Notebooks With Satellite



Satellite Proxy Service Design Goals/Requirements/…

- Satellite must serve proxied content over HTTPS, using a certificate signed by a CA approved by CA/Browser Forum.
- Users must be able to create and destroy their own services (proxy mappings), no admin/user support involved.
- Each instance (mapping) must be served from a distinct URL (sub-domain).
- Internal side of the proxy may be over HTTP, but only on a network already trusted for plaintext. (e.g. NFS home directories)
- Satellite code should be designed to minimize unmanaged dependencies. (Dependencies should be available through system package manager.)
- Satellite API must be usable with only wget/curl.

SDSC Satellite Clients

start-jupyter

- 1st generation shell utility developed to orchestrate a user's interaction with both Satellite and Slurm to start a Jupyter session within a batch job.

- Key features in design:

  - User calls `start-jupyter` launch script, which requests token from Satellite, passes token to batch job script and submits the job to Slurm; token redeemed from batch job once it runs

  - Provided user with a prefabricated set of batch job scripts to choose from for certain popular applications on each system; user could modify/make their own custom batch job script

  - Small custom shell function library to make code more reusable

- Currently runs on: Expanse, Comet, TSCC, TSCC Stratus

https://github.com/sdsc-hpc-training-org/reverse-proxy

galyleo

- 2nd generation shell utility developed to orchestrate a user's interaction with both Satellite and Slurm to start a Jupyter session within a batch job.

- Developed while reviewing start-jupyter codebase to sort out how best to support Expanse (OOD) Portal and HPC User Services Group long-term; effectively recycled existing an SSH tunneling orchestration utility to use Satellite proxy service instead.

- Key features in design:

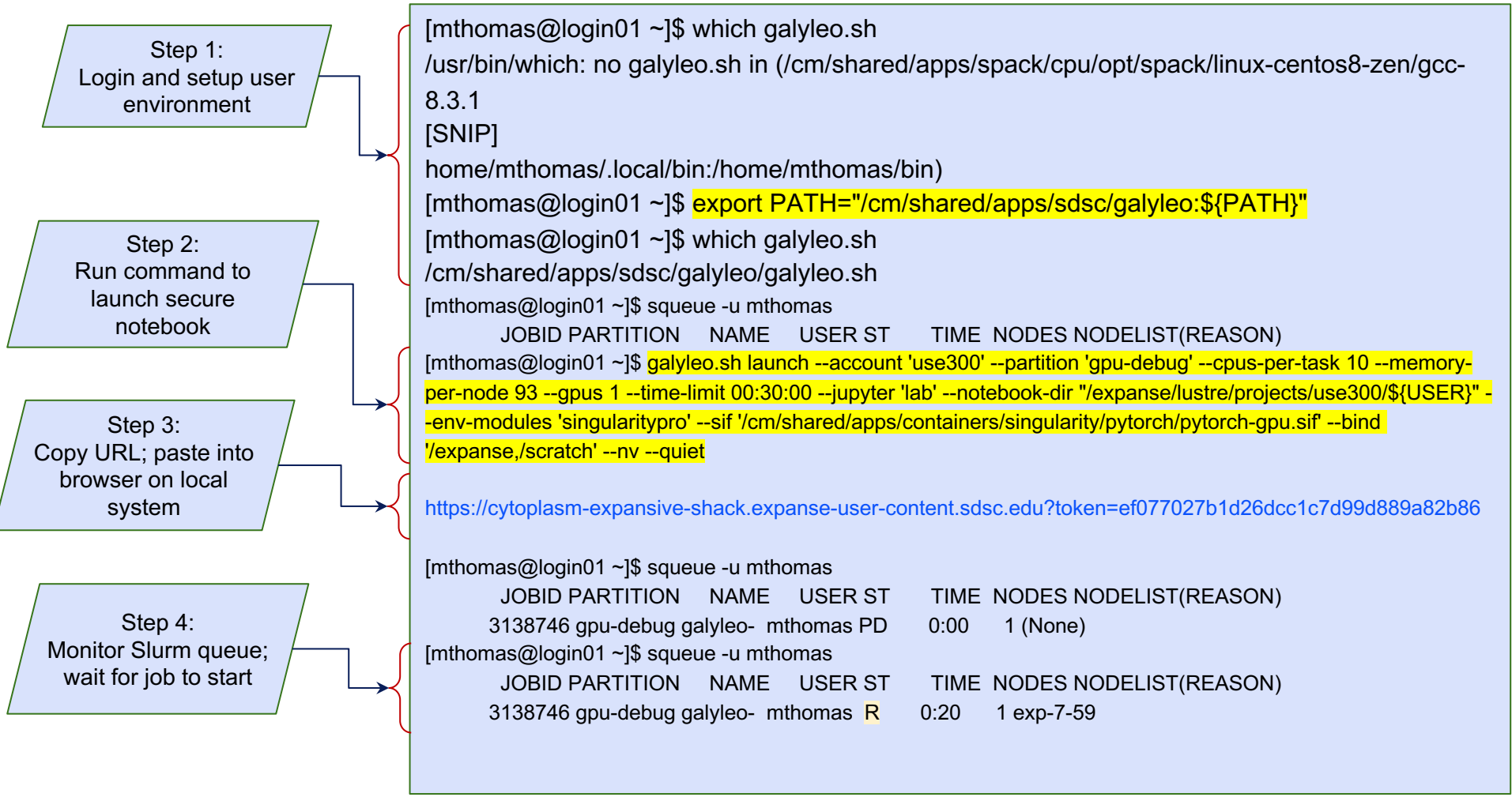
    - Recreate same interactions with Satellite service.
    - Increase flexibility for users to configure software environment; but also try to make it simpler for them to do themselves
    - Batch job script is generated completely on-the-fly.
    - Command-line argument driven.
    - Quiet mode for OOD portal

https://github.com/mkandes/galyleo

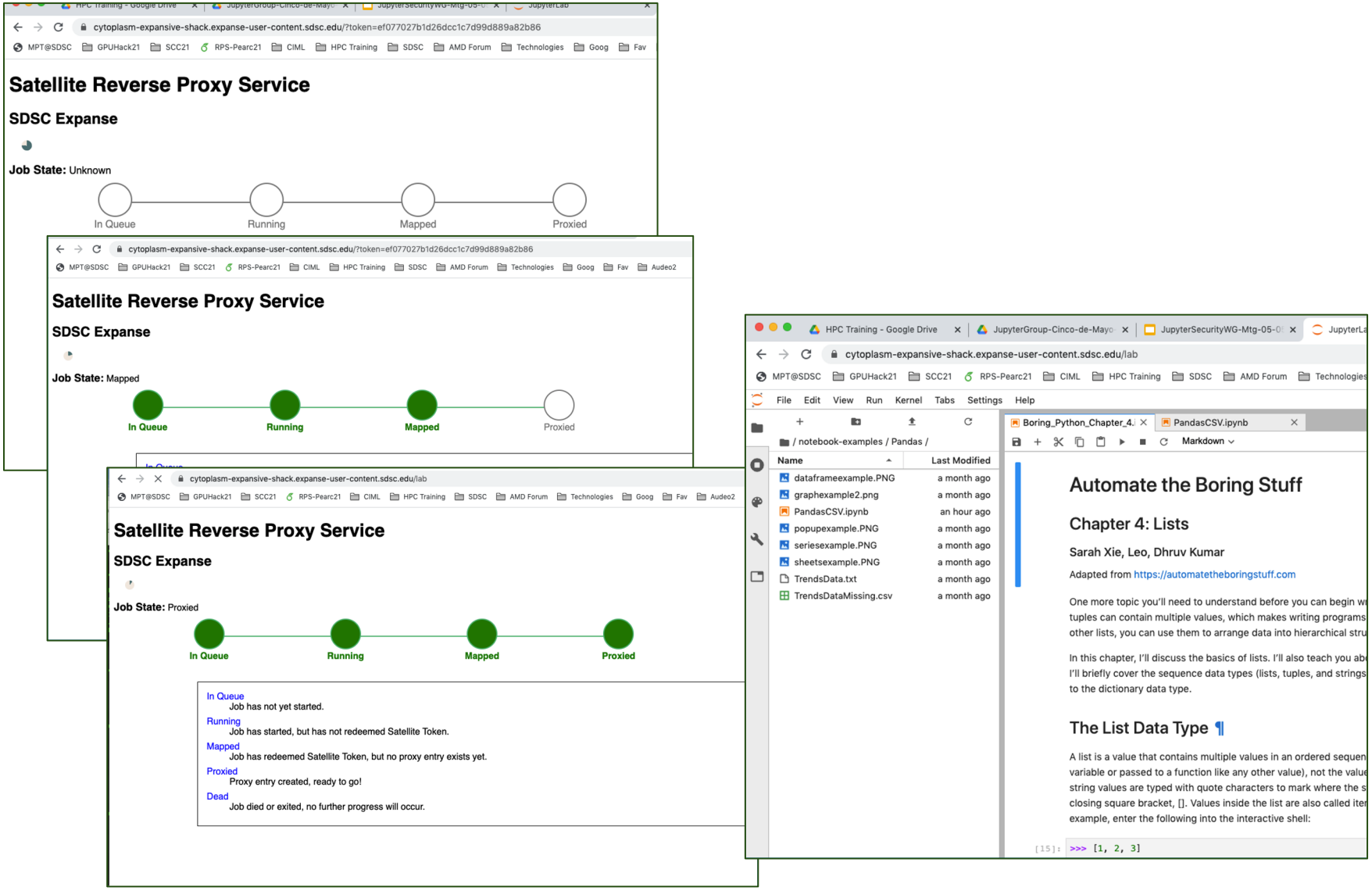
galyleo demo examples on Expanse

```
# Location of galyleo directory on Expanse

export PATH="/cm/shared/apps/sdsc/galyleo:${PATH}"

# Example 1: Launch a Jupyter Notebook session on a single CPU core in the 'debug'
# partition on Expanse using the 'base' Anaconda3 software environment provided as part
# of Expanse's standard software modules.

/cm/shared/apps/sdsc/galyleo/galyleo.sh launch --account 'use300' --partition 'debug'  --
cpus-per-task 1 --memory-per-node 1 --time-limit 00:30:00 --jupyter 'notebook' --notebook-
dir "/expanse/lustre/projects/use300/${USER}" --env-modules 'cpu,gcc,anaconda3' --
conda-env 'base' --quiet


# Example 2: Launch a JupyterLab session on a single GPU in the 'gpu-debug' partition
# on Expanse using the latest PyTorch Singularity container available.

galyleo.sh launch --account 'use300' --partition 'gpu-debug' --cpus-per-task 10 --memory-
per-node 93 --gpus 1 --time-limit 00:30:00 --jupyter 'lab' --notebook-dir
"/expanse/lustre/projects/use300/${USER}" --env-modules 'singularitypro' --sif
'/cm/shared/apps/containers/singularity/pytorch/pytorch-gpu.sif' --bind '/expanse,/scratch' --
nv --quiet
```

Running GPU notebook using galyleo.sh



- Paste the HTTPS URL into a web browser


cytoplasm-expansive-shack.expanse-user-content.sdsc.edu/?token=ef077027b1d26dcc1c7d99d889a82b86
Satellite Reverse Proxy Service
SDSC Expanse
Job State: Unknown
In Queue
Running
Mapped
Proxied
Job State: Mapped
Job State: Proxied
cytoplasm-expansive-shack.expanse-user-content.sdsc.edu/lab
In Queue
Job has not yet started.
Running
Job has started, but has not redeemed Satellite Token.
Mapped
Job has redeemed Satellite Token, but no proxy entry exists yet.
Proxied
Proxy entry created, ready to go!
Dead
Job died or exited, no further progress will occur.
HPC Training - Google Drive
cytoplasm-expansive-shack.expanse-user-content.sdsc.edu/lab
File Edit View Run Kernel Tabs Settings Help
/ notebook-examples / Pandas /
Name
Last Modified
dataframeexample.PNG
graphexample2.png
PandasCSV.ipynb
popupexample.PNG
seriesexample.PNG
sheetsexample.PNG
TrendsData.txt
TrendsDataMissing.csv
Boring_Python_Chapter_4.
PandasCSV.ipynb
Markdown
Automate the Boring Stuff
Chapter 4: Lists
Sarah Xie, Leo, Dhruv Kumar
Adapted from https://automatetheboringstuff.com
The List Data Type ¶
[15]:
>>> [1, 2, 3]

Summary and Future Work

- Satellite:
  - Daemon / Systemd unit for more frequent mapping updates. Current cron-driven process only updates at the top of the minute.
  - Investigate internal PKI to operate over untrusted networks. Requires service support for TLS. (Jupyter Notebooks do!)
  - Develop usage metrics
- galyleo:
  - Add system config function to make deployment process simple for other systems, especially when deploying upgrade of galyleo
  - Add user config function to make subsequent calls to the same software environment and resource request simpler; i.e., templates
  - Complete integration with Expanse (OOD) portal; add a mechanism to allow OOD to track open notebook sessions
  - Secure other web-based tools through Satellite. e.g., TensorBoard
  - Extend to other HPC systems at SDSC/XSEDE, etc (in collaboration with Juypter project).
- Jupyter Working Group @ SDSC:
  - Mary Thomas, Scott Sakai, Marty Kandes (SDSC); Rick Wagner (UCSD); James McDougall (intern)

Questions?

Satellite Client Example: start-jupyter

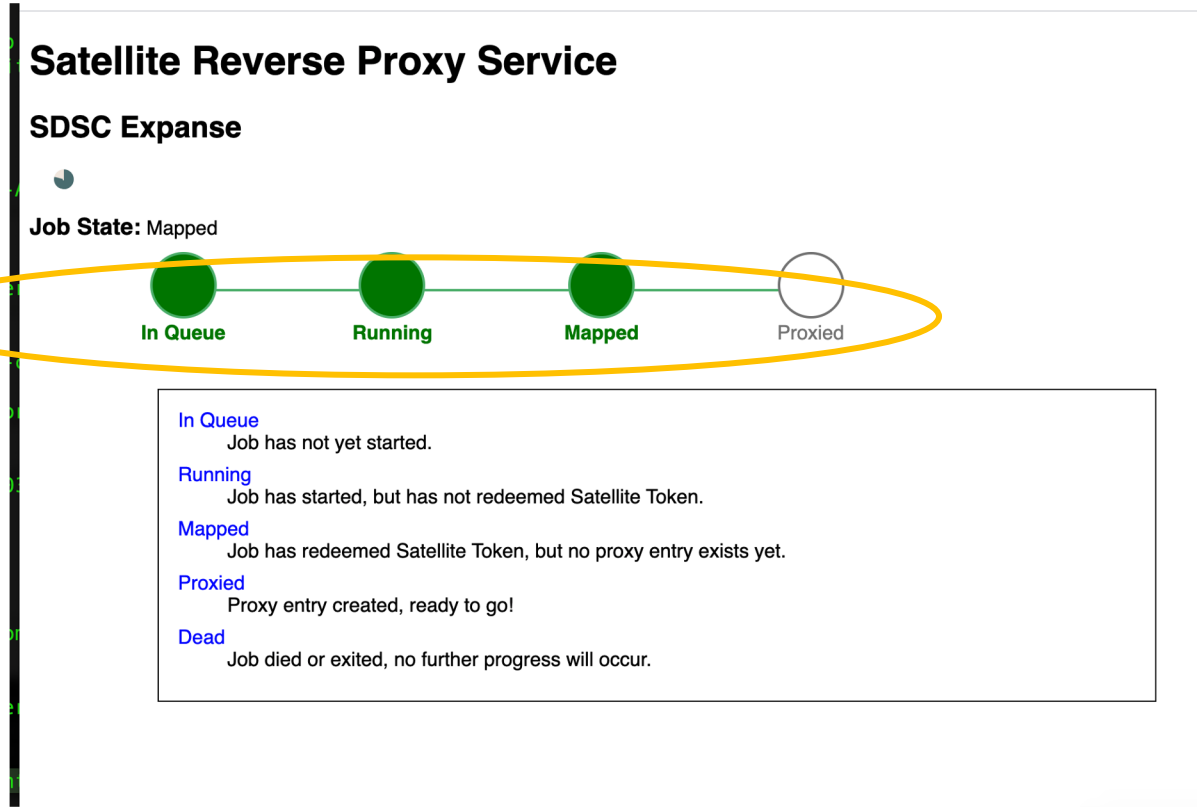
Running start-notebook

```
(base) [username@login02 reverse-proxy-branch-james-dev]$ ./start-jupyter -A abc123
Your notebook is here:
https://annuity-headphone-aptitude.expanse-user-content.sdsc.edu?token=ae0ff01b6780aa32893d6673976769cf
If you encounter any issues, please email help@xsede.org and mention the Reverse Proxy Service.
No time given. Default is 30 mins
Using ./slurm-expanse/notebook.sh
Your job id is 670505
You may occasionally run the command 'squeue -j 670505' to check the status of your job
(base) [username@login02 reverse-proxy-branch-james-dev]$ squeue -u username -u username
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
            670505   compute notebook  username  R       0:37      1 exp-1-17
(base) [username@login02 reverse-proxy-branch-james-dev]$ cat  slurm-670505.out
[I 09:36:06.377 NotebookApp] Serving notebooks from local directory: /home/username
[I 09:36:06.377 NotebookApp] Jupyter Notebook 6.1.4 is running at:
[I 09:36:06.377 NotebookApp] http://exp-1-17.eth.cluster:8888/?token=...
[I 09:36:06.377 NotebookApp]  or http://127.0.0.1:8888/?token=...
[I 09:36:06.377 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100     9  100     9    0     0     52      0 --:--:-- --:--:-- --:--:--    52
Success!
[I 09:37:14.362 NotebookApp] 302 GET /?token=ae0ff01b6780aa32893d6673976769cf (10.21.0.30) 0.36ms
```

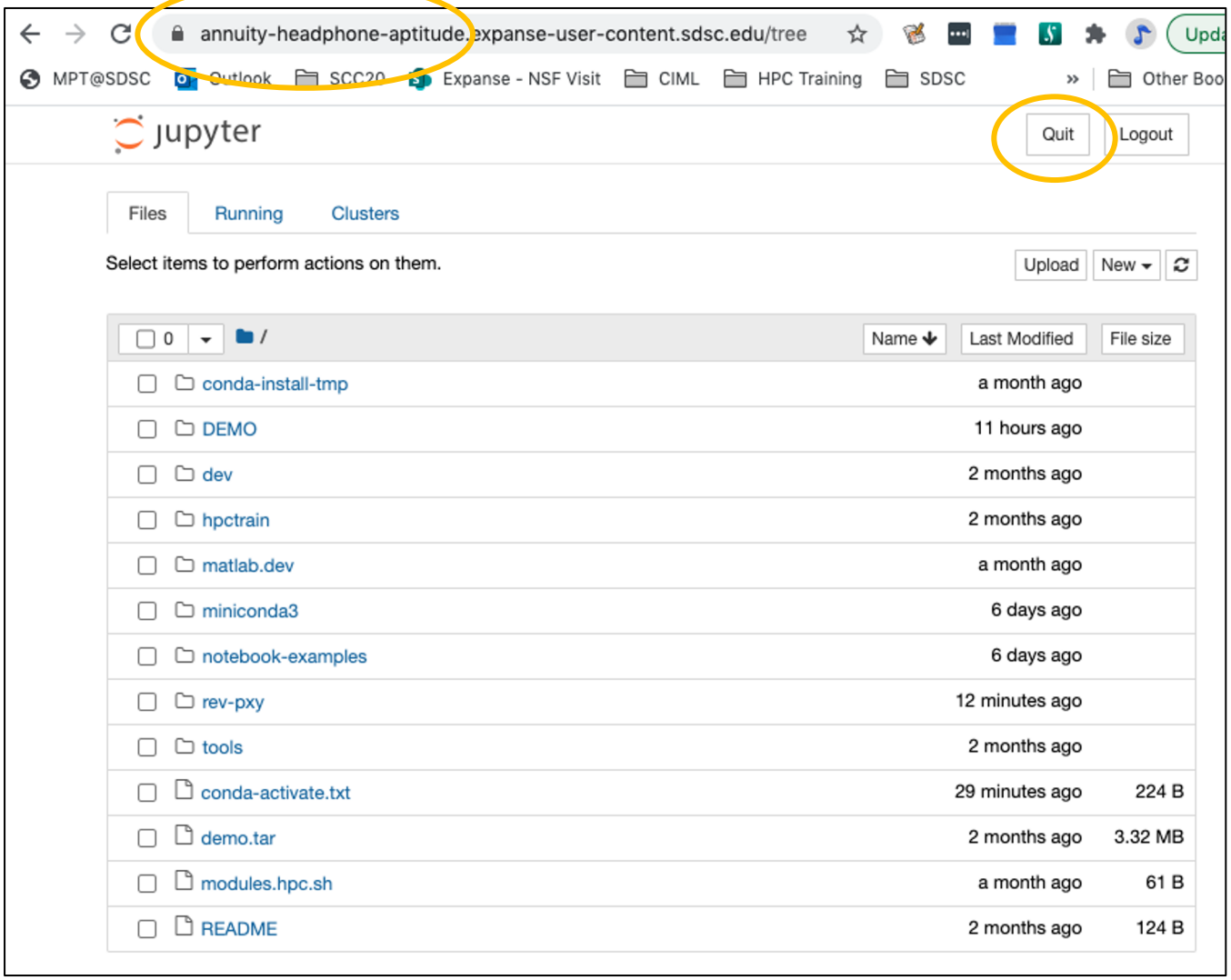
- Paste the HTTPS URL into a web browser

Satellite Server Pening Page

- Load notebook URL in browser; wait for it to launch
- Monitor pending page
- Run the “squeue” command on the HPC system to check job status
- If the job queue is busy, it may take a while to launch the notebook
- Treat Jupyter Notebook URL as a password



Your notebook is launched



When done with the notebook be sure to shut it down by quitting the notebook