The CANFAR Science Platform

Brian Major, Canadian Astronomy Data Centre
IVOA Interop, May 2021

Agenda

- Quick overview, what's new
- Interoperability and Standard Potential
  - User Contributed Software
  - Groups for Allocations and Discovery

CANFAR Science Platform Overview

CADC: Canadian Astronomy Data Centre (Victoria, BC)
CANFAR: Canadian Advanced Network for Astronomical Research

- A lot of work over the last 18 months:
  - ARCADE Development Study - ALMA Reduction in the CANFAR Data Environment
  - CANARIE Research Software Program - Jupyter Notebooks and Image Registry
- CANFAR:
  - VMs on OpenStack - multi purpose and batch
  - **New: 100% Container-based, higher-level Science Platform**

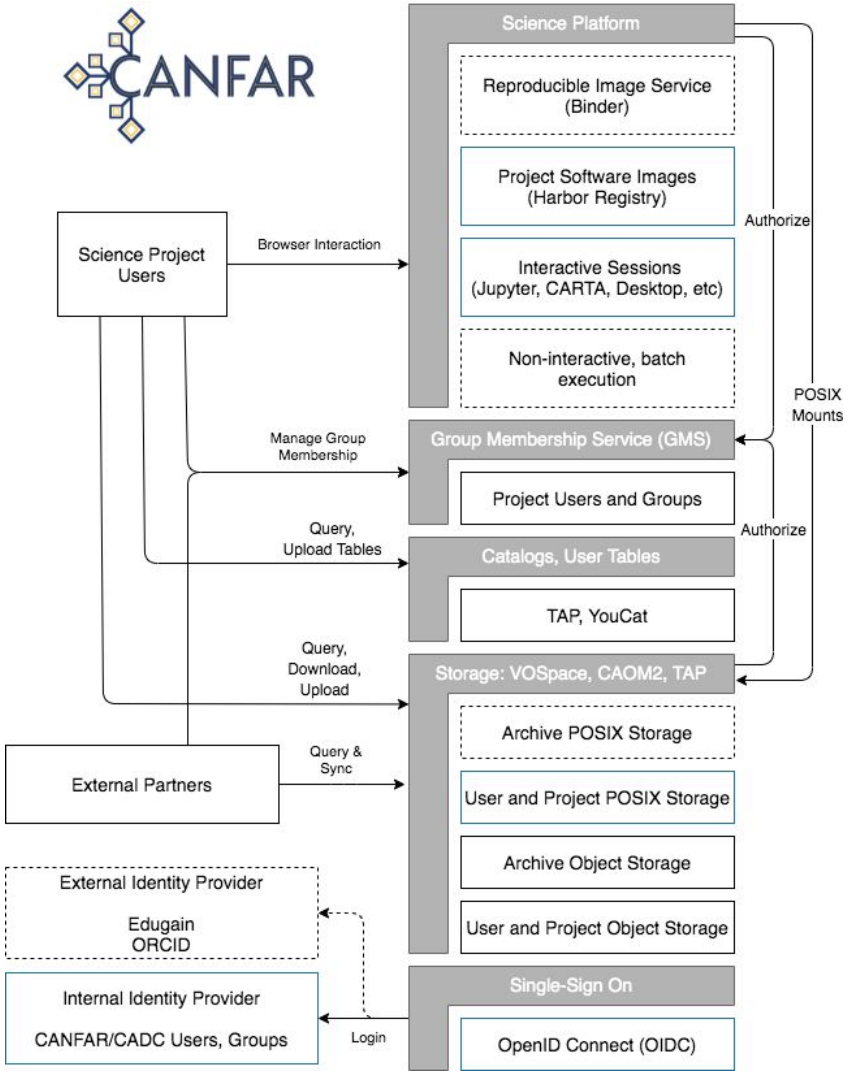
- Production deployment in place, scaling up users cautiously

- Kubernetes cluster provided by Compute Canada / UVic Research Computing.

CANFAR Architecture & Design

- Be guided by a science problem that someone wants solved.
- Look to solve the general problem rather than the specific issue (if possible)
- Accept that science users will see the general approach as longer to implement, but we will satisfy more users, eventually and create sustainable infrastructure.
- Use an IVOA (on top of whatever the latest shiny is) and help develop standards where new ones are needed.

https://github.com/opencadc/skaha

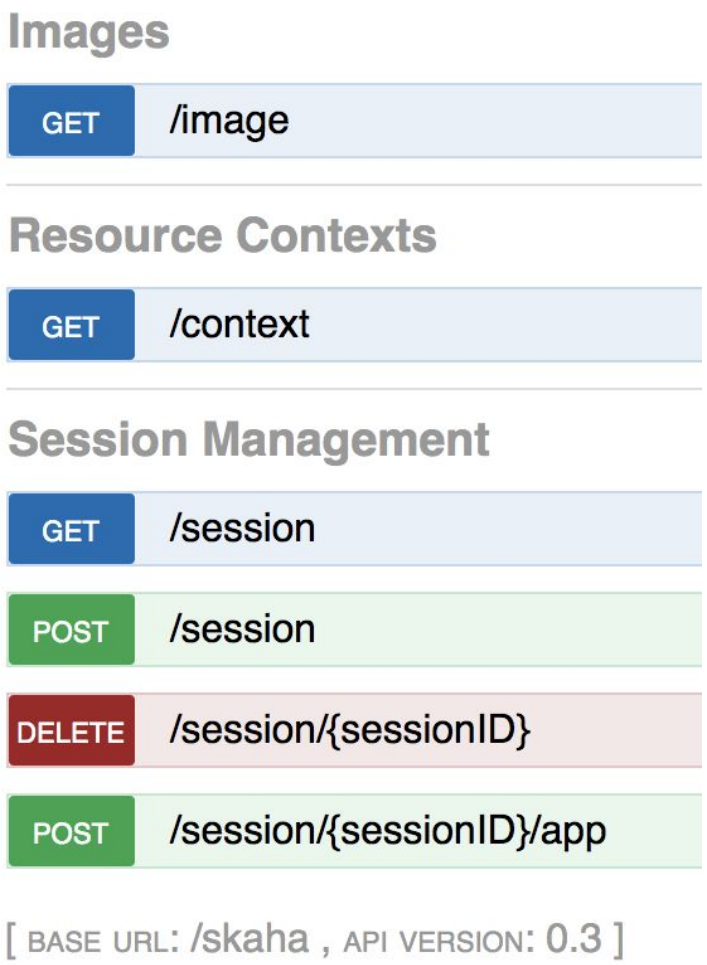


Platform API

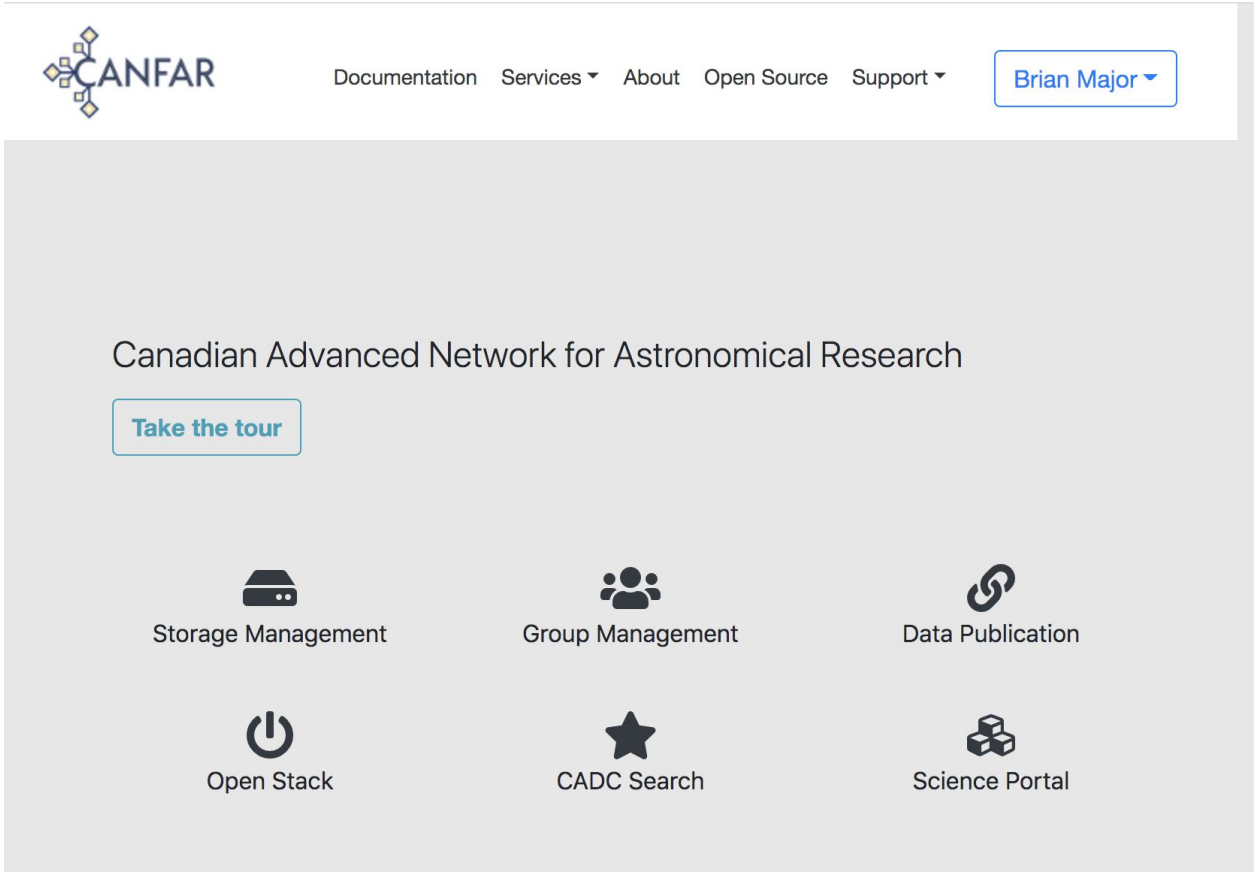
skaha is our platform API that pulls together:

- Authentication (SSO, OIDC Client)
- Groups and Authorization (GMS)
- POSIX VOSpace (cavern)
- User-controlled software container/image registry
- Kubernetes

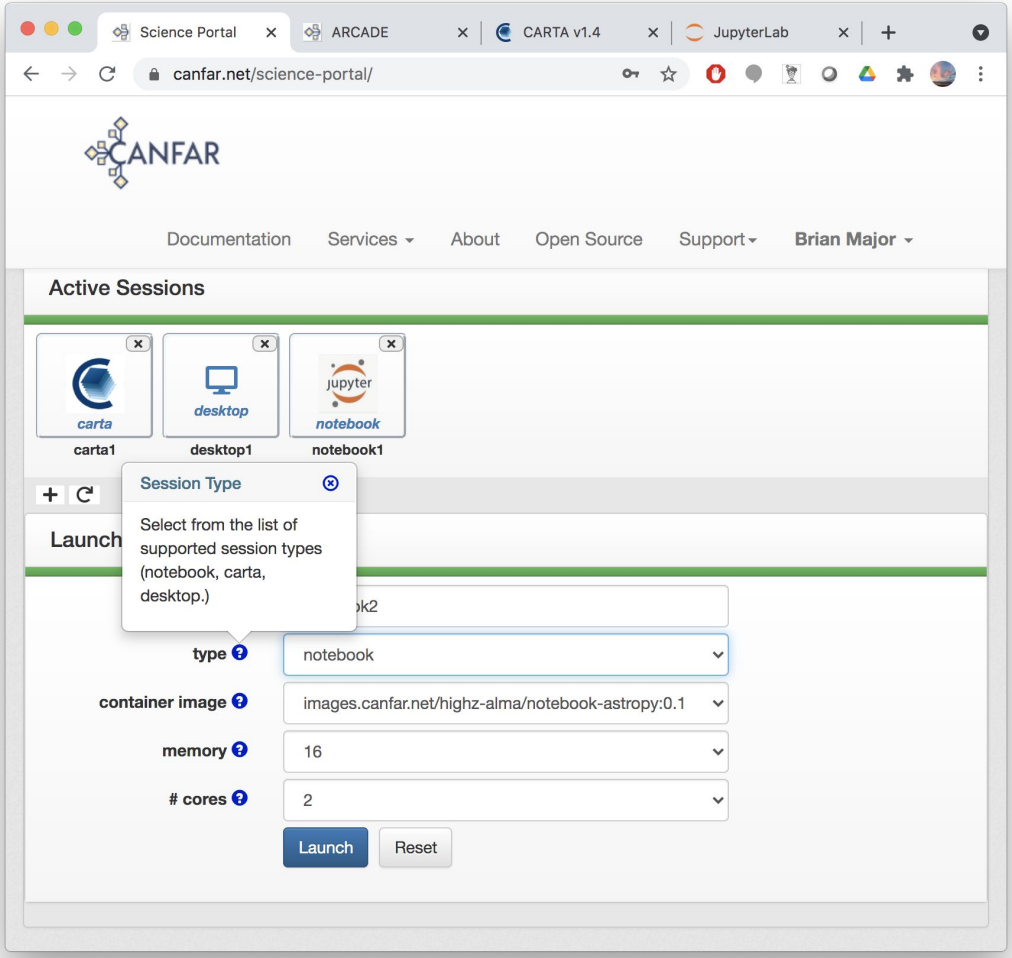
https://github.com/opencadc/skaha



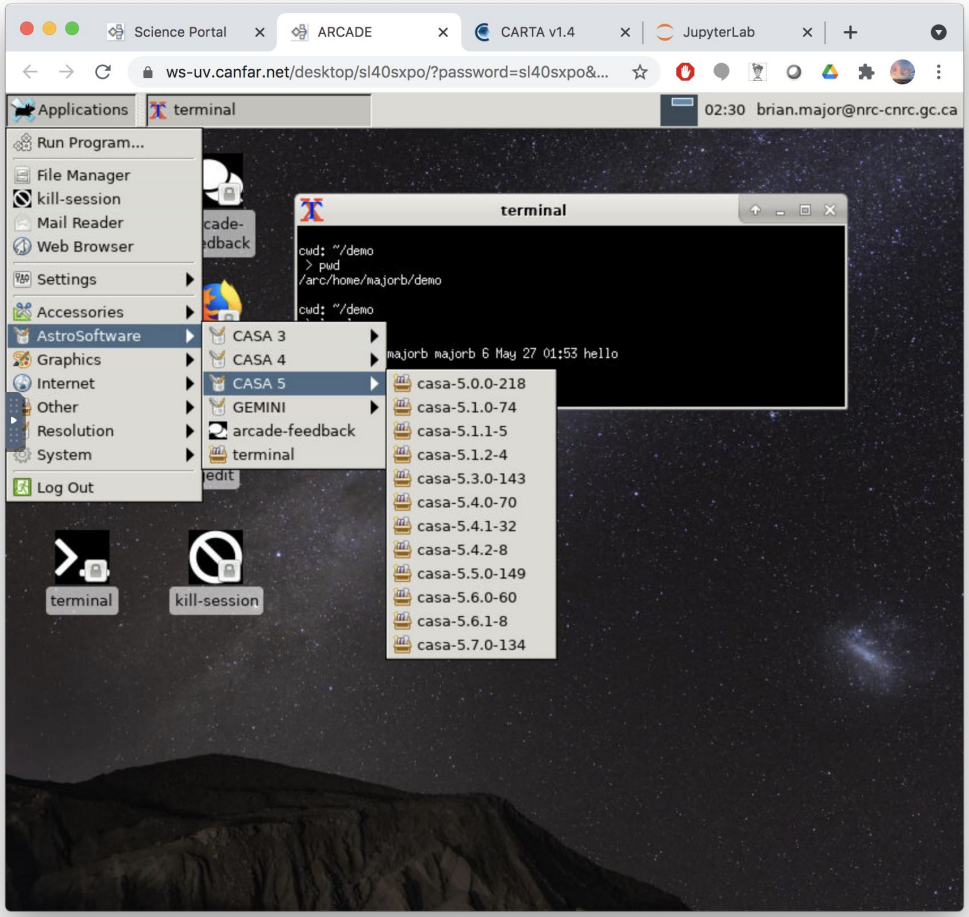
CANFAR Portal



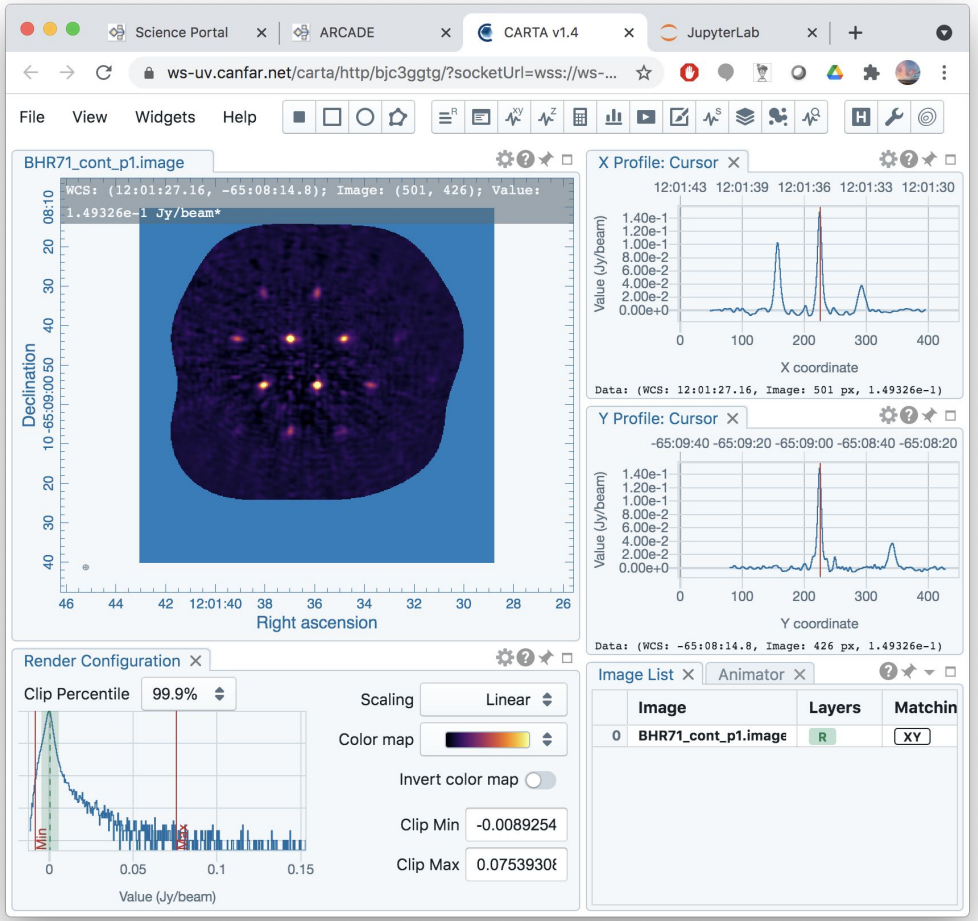
CANFAR Science Portal



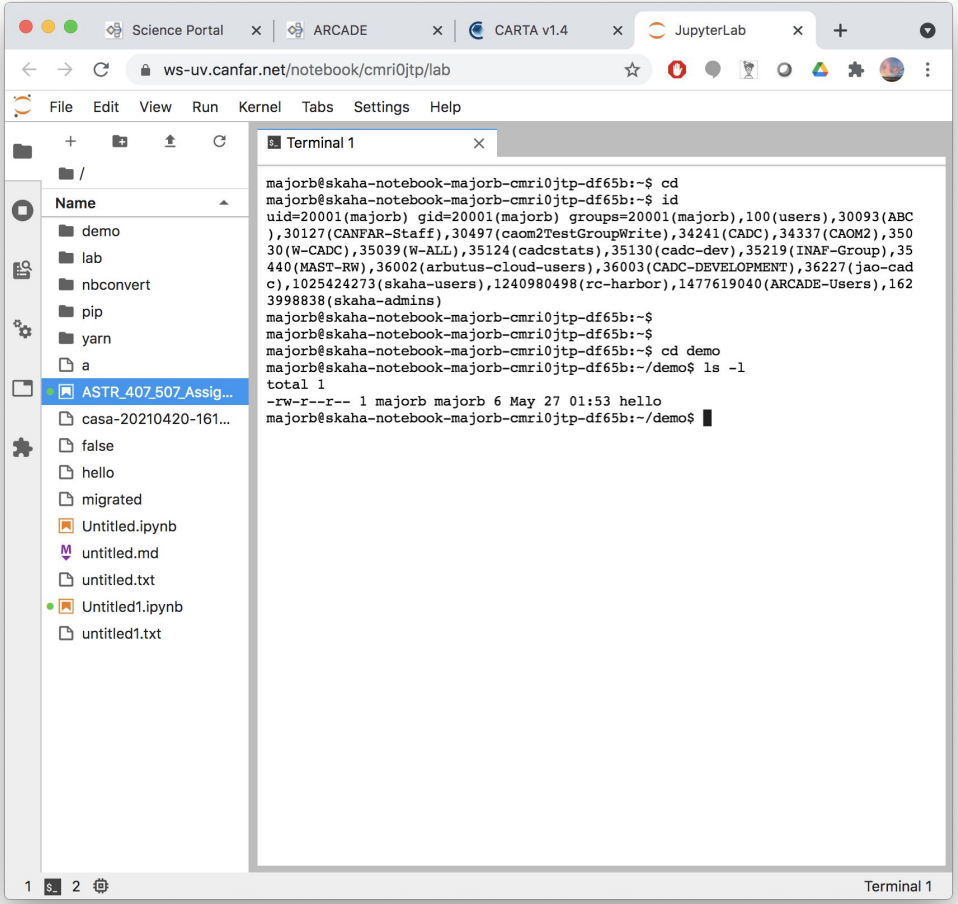
Desktop Session



CARTA Session



Notebook Session



Cavern Storage

- Cavern Storage, accessible through:
  - POSIX mounts through all containers
  - VOSpace API
  - $HOME and $PROJECT dirs

- Each container **'runs as the user'** (with the **user's groups**) so plain POSIX permissions used in cavern

```
cwd: ~/demo
 > pwd
/arc/home/majorb/demo

cwd: ~/demo
 > ls -l
total 1
-rw-r--r-- 1 majorb majorb 6 May 27 01:53 hello

cwd: ~/demo
 >
```

```
Default
majorb@NRC-009195 ~ $ vls -l arc:/home/majorb/demo
-rw----r--  majorb                NONE              NONE
                    6 May 26 18:53 hello
majorb@NRC-009195 ~ $
```

```
Terminal 1
majorb@skaha-notebook-majorb-cmri0jtp-df65b:~$ id
uid=20001(majorb) gid=20001(majorb) groups=20001(majorb),100
97(caom2TestGroupWrite),34241(CADC),34337(CAOM2),35030(W-CAD
c-dev),35219(INAF-Group),35440(MAST-RW),36002(arbutus-cloud-
adc),1025424273(skaha-users),1240980498(rc-harbor),147761904
majorb@skaha-notebook-majorb-cmri0jtp-df65b:~$
majorb@skaha-notebook-majorb-cmri0jtp-df65b:~$ echo $HOME
/arc/home/majorb
```

/home/majorb/demo

Search Name... | Home | Up | Root | + Add | Download | Move

0.00 B remaining of 0.00 B (Request more from support@canfar.net)

Showing 1 to 1 of 1 entries

Name	Size	Last Modified (UTC)	Read/Write	Read
hello	6.00 B	2021-05-27 - 01:53:42		Public

https://github.com/opencadc/vos

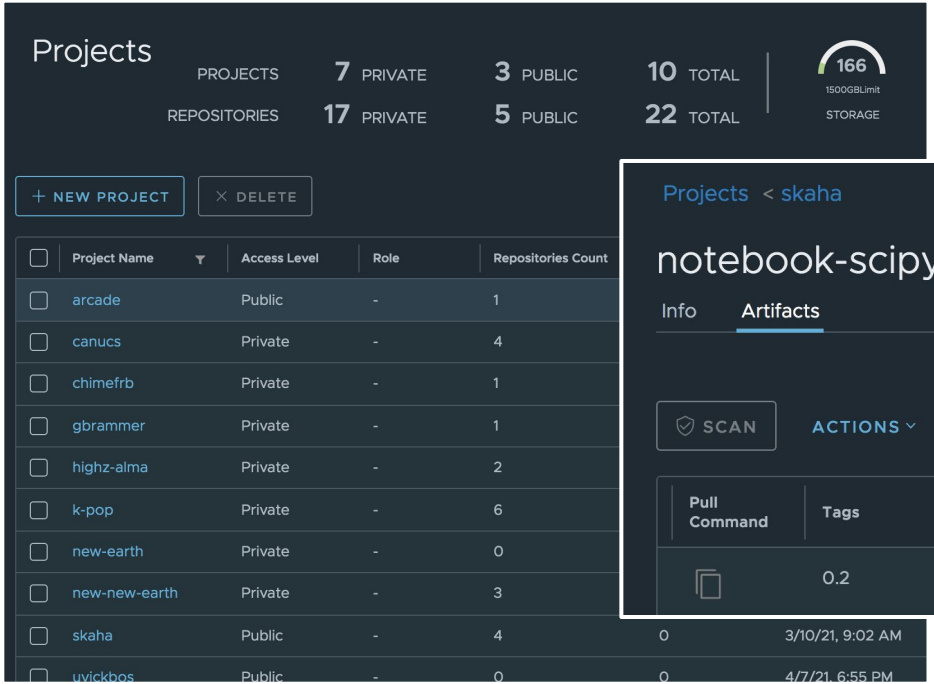
Future

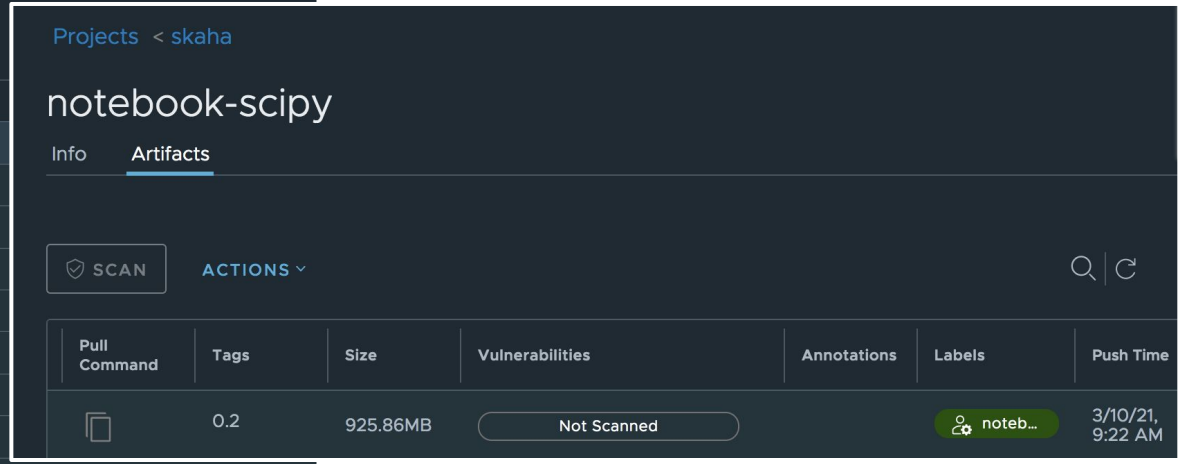
- Reproducibility with Binder: from Dockerfile to Container to paper.
- Cooperating containers: visualizers and notebooks exchanging data. Container-to-container SAMP?
- GPU integration into cluster (Nvidia vGUP V100D-8C)
- Batch scheduling

User contributed software

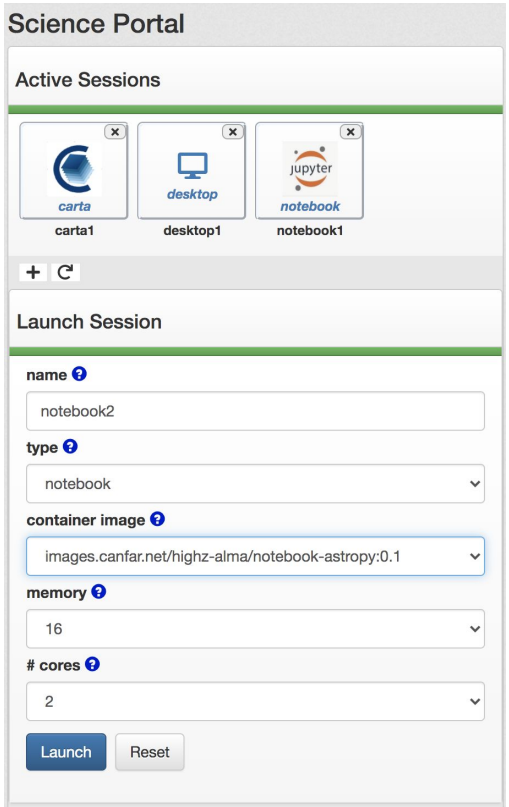
Harbor image registry

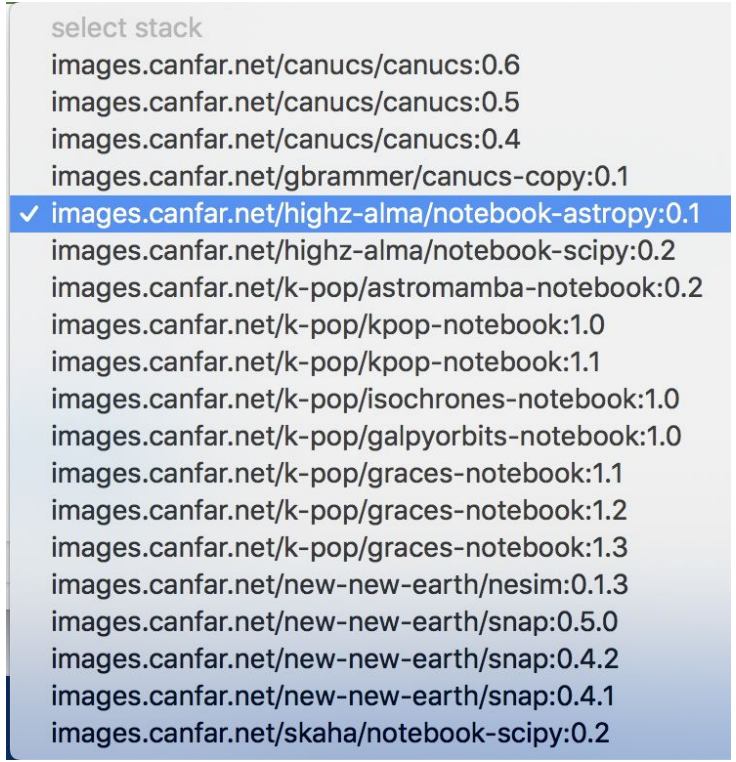
```
> docker login images.canfar.net
> docker push images.canfar.net/skaha/notebook-scipy:0.2
```





Harbor images you are allowed to use





Projects bring their containers

- Each project creates their own containers or use public ones already available
- Projects maintain, update, version, and curate their software (containers)
- Agreement between skaha and containers
  - Must be able to run as **non-root** user
  - Must fit one of the **Session Types**
  - Aiming to reduce the number of types (generalize), or;
  - Allow users to define new session types
- Aim to for little or no skaha-specific container requirements.

Session/Container Types

Browser Sessions - HTML5/websocket containers

- Jupyter Notebooks, Desktop (NoVNC), CARTA Vis, Pluto, …
- Connect to session canfar.net domain with port 443
- Each uses different **ports**
- Path rewriting usually required and supported by browser apps

Non-browser containers

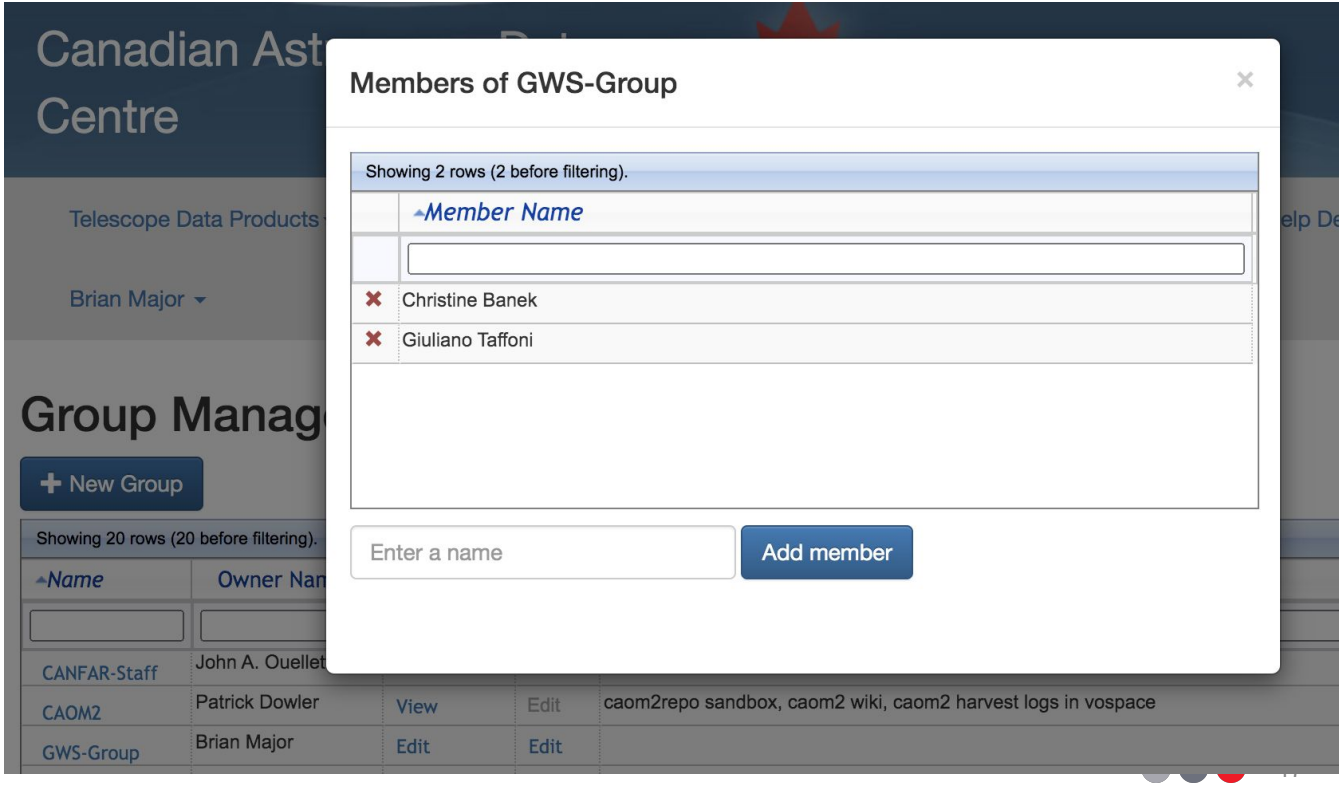
- Headless -- just run the container as is, no proxy necessary
  - API allows override of entrypoint and parameters
  - Step towards batch
- Desktop container
  - Entrypoint is always xterm
  - Window appears in Desktop session, but is it's own container running elsewhere.

Reproducibility:

- Avoid installing software in $HOME (cavern)

Users manage their team

- Anonymous access to a science platform isn't likely.
- Projects and Users arrange for authorization and allocations with resources.



Groups for allocations

- Allocations to CANFAR are assigned to **projects**
- Projects are represented by a **group**
- Membership checks in the groups are done through the Group Membership Service (**GMS**)
- CADC extension of GMS allows project admins to **control user membership**

- Resources assigned to projects/groups:
  - Storage space with quota in POSIX mounted VOspace (cavern)
  - Storage space with quota in object store VOSpace (vault)
  - Storage space with quota in docker image registry
  - CPU core and ram allocations in kubernetes cluster
  - Project database schema in user-managed TAP (youcat)

Groups for discovering platform capabilities

- Storage space with quota in POSIX mounted VOspace (cavern)
- Storage space with quota in object store VOSpace (vault)
- Storage space with quota in docker image registry
- CPU cores and ram allocations in kubernetes cluster
- Project database schema in user-managed TAP (youcat)

Question: What can this platform do?

More Pertinent Question: **What can I do** in this platform and how do I do it?

Answer: For all the groups in which you are a member: where are those groups assigned?

A reverse GMS/group lookup.

Groups for discovering platforms capabilities

Example: Group A is assigned to a project in an **image registry** with properties:

- Access URL
- Space used, space remaining
- Client protocols supported
- Authentication mechanism supported

Example: Group A is assigned to a project in **POSIX VOSpace** with properties:

- Service ID/IVOID (leads to Access URL)
- Space used, space remaining
- Mount point used in containers (eg /cavern)
- Remaining details provided by VOSpace itself

Pattern: Where groups are assigned says what the members can do:

- Registry/Discovery (what is it granting?)
- StandardID for the resource (how do I use it?)
- StandardID-specific details

Discovering all IVOA platform capabilities via Group Grants

Question: What can this platform do?

Revised: What can I do in this platform and how do I do it?

Bigger Question: What can I do with **all available IVOA platforms**?

Answer: Where have my groups been set across all platforms?

- Platform A Container Registry
- Platform A Storage
- Platform A Compute
- Platform B Compute

Thoughts:

- How to do find all the group grants? Nothing 'central' in IVOA except the Registry.
- Platforms open up access to other public/private container registries.
- GMS groups don't have institutional boundaries so are a good fit.

THANK YOU

brian.major@nrc-cnrc.gc.ca