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
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

[BASE_URL: /skaha, API_VERSION: 0.3]
CANFAR Science Portal
Desktop Session
CARTA Session
Notebook Session

```
$ cd
$ id
uid=20001(majorb) gid=20001(majorb) groups=20001(majorb),100(users),30993(CADC),30127(CANFAR-Staff),30497(cm2m2TestGroupWrite),34241(CADC),34337(CADC),35030(CADC),35039(M-Ali),35124(casestate),35139(cadc-dev),35219(INMAP-Group),35440(MAST-MW),36002(abctools-cloud-users),36003(CADC-DEVELOPMENT),36227(jpa-cadc),1025424773(ekahaka-users),1240980498(rc-ecb),1477619400(ARCADE-Users),1623998383(ekahaka-admin)
```

```bash
$ majorb@ekahaka-notebook-majorb-cmr10jtp-df65b:-$ ls
 ASTRO_407_507_Assign...  casa-20210420-161...
     false               hello
     migrated            Untitled.ipynb
     Untitled.ipynb
     Untitled.md
     Untitled.txt
     Untitled.ipynb
```

```
$ majorb@ekahaka-notebook-majorb-cmr10jtp-df65b:-$ ls -l
-rw-rw-r-- 1 majorb majorb 6 May 27 01:53 hello
```

```
$ majorb@ekahaka-notebook-majorb-cmr10jtp-df65b:-$ echo
```
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

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

Select stack:
- images.canfar.net/highz-alma/notebook-astropy:0.1
- images.canfar.net/highz-alma/notebook-scipy:0.2
- images.canfar.net/k-pop/astromamba-notebook:0.2
- images.canfar.net/k-pop/kpop-notebook:1.0
- images.canfar.net/k-pop/kpop-notebook:1.1
- images.canfar.net/k-pop/isochrones-notebook:1.0
- images.canfar.net/k-pop/galpyorbits-notebook:1.0
- images.canfar.net/k-pop/graces-notebook:1.1
- images.canfar.net/k-pop/graces-notebook:1.2
- images.canfar.net/k-pop/graces-notebook:1.3
- images.canfar.net/new-new-earth/nesim:0.1.3
- images.canfar.net/new-new-earth/snap:0.5.0
- images.canfar.net/new-new-earth/snap:0.4.2
- images.canfar.net/new-new-earth/snap:0.4.1
- images.canfar.net/skaha/notebook-scipy:0.2
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 its 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