



USDF: What You Can Look Forward To

PCW 2022

Richard Dubois
8 August 2022

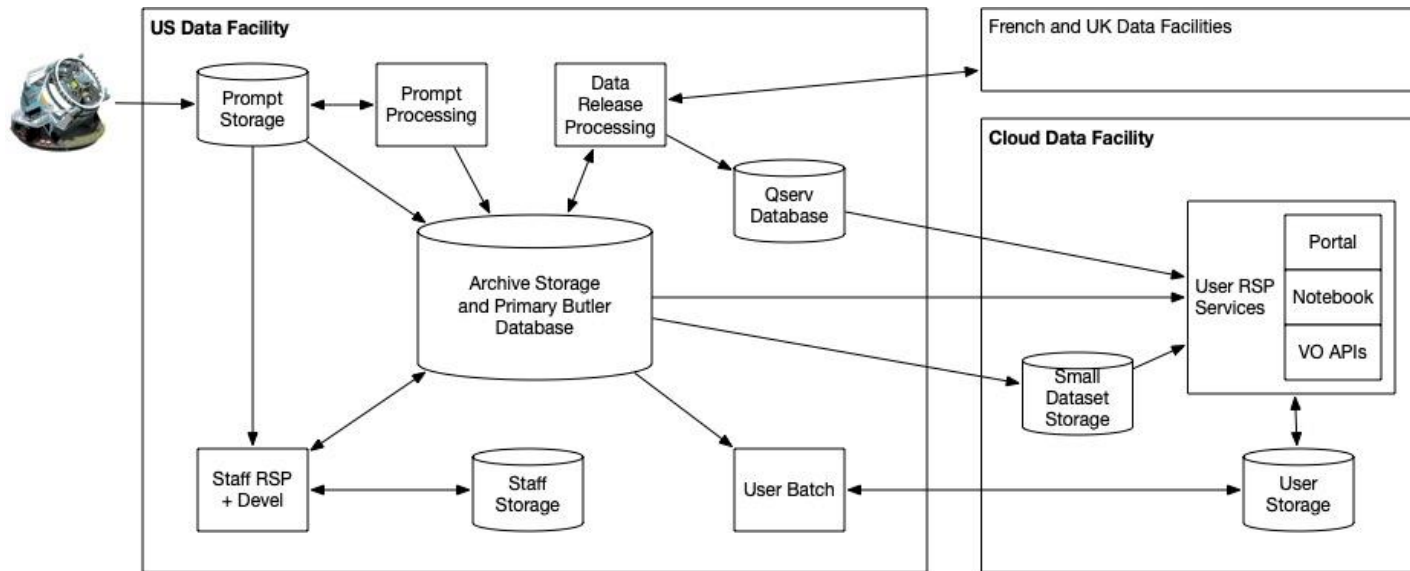


SLAC



U.S. DEPARTMENT OF
ENERGY

USDF: A Mix of On-prem and Cloud



Hybrid model: Data at SLAC but users on the Cloud.

Allows:

- Separation of security concerns
- Burst response
- Reduced risk

(see [DMTN-209](#))

RSP = Rubin Science Platform

Cloud-SLAC Division of Scope

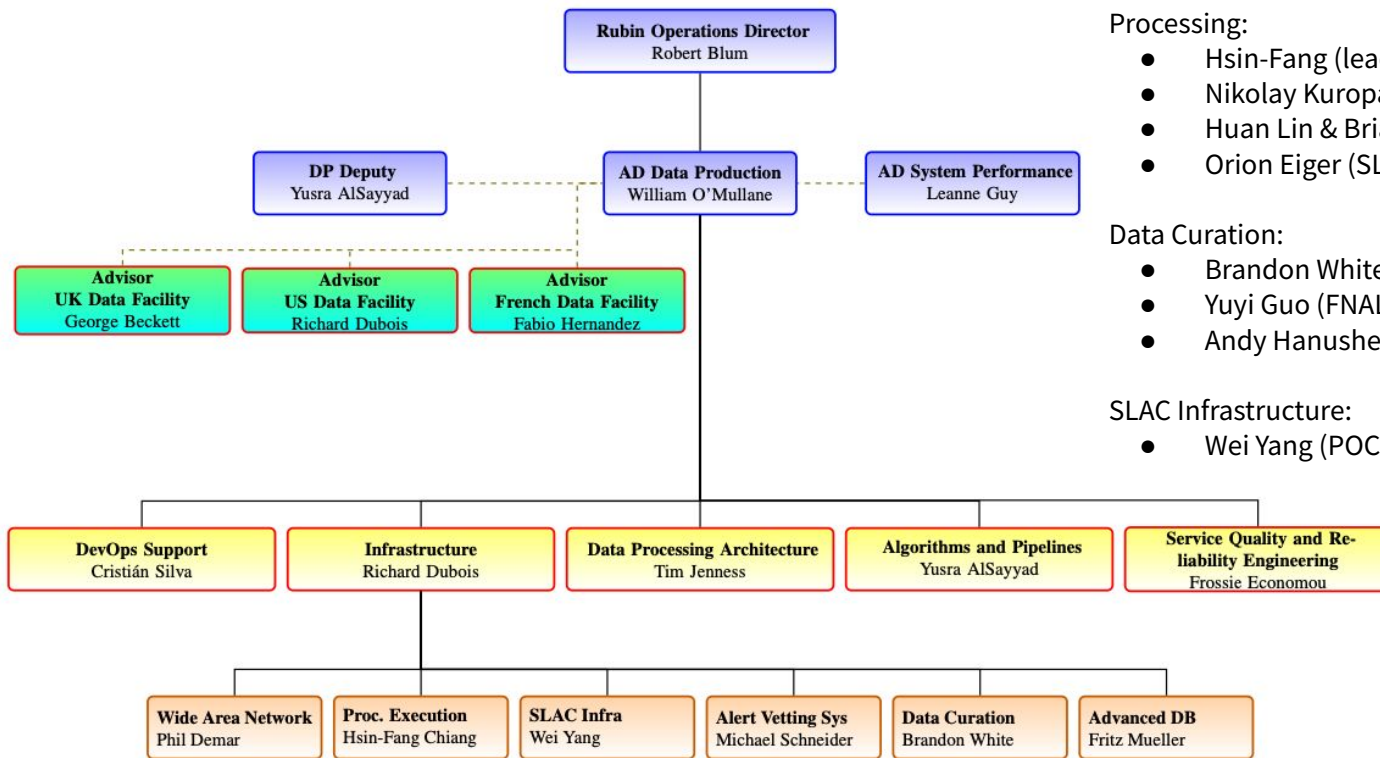
Cloud

- Science users, with access provided by the Rubin Science Platform
- Provide personal storage/CPU + cloud access to coadds
 - Effective 500 cores added each year + 2 PB → 10 PB storage
- Services run here to take advantage of elasticity, but storage is limited to user-generated data, small portions of DRP datasets, and caches.

SLAC

- Prompt and DRP processing
- Qserv catalogue server
- Storage archive for all data
- Serving alerts to the community
- Home for developers and staff (and commissioners)

USDF Who? Presented to You by Ops



Processing:

- Hsin-Fang (lead)
- Nikolay Kuropatkin, Jen Adelman-McCarthy (FNAL)
- Huan Lin & Brian Yanny filling in from V&V in DP0.2
- Orion Eiger (SLAC, new)

Data Curation:

- Brandon White (lead, FNAL)
- Yuyi Guo (FNAL)
- Andy Hanushevsky

SLAC Infrastructure:

- Wei Yang (POC to SLAC SCS/TID)

PanDA (BNL):

- Eddie Karavakis
- Wen Guan
- Zhaoyu Yang
- Shuwei Ye (through Sept)

Postgres DBA (FNAL):

- Olga Vlasova
- Chris Dollinger

Core & Networking: Network design & implementation, 'ground zero' services (DNS, NTP, etc)

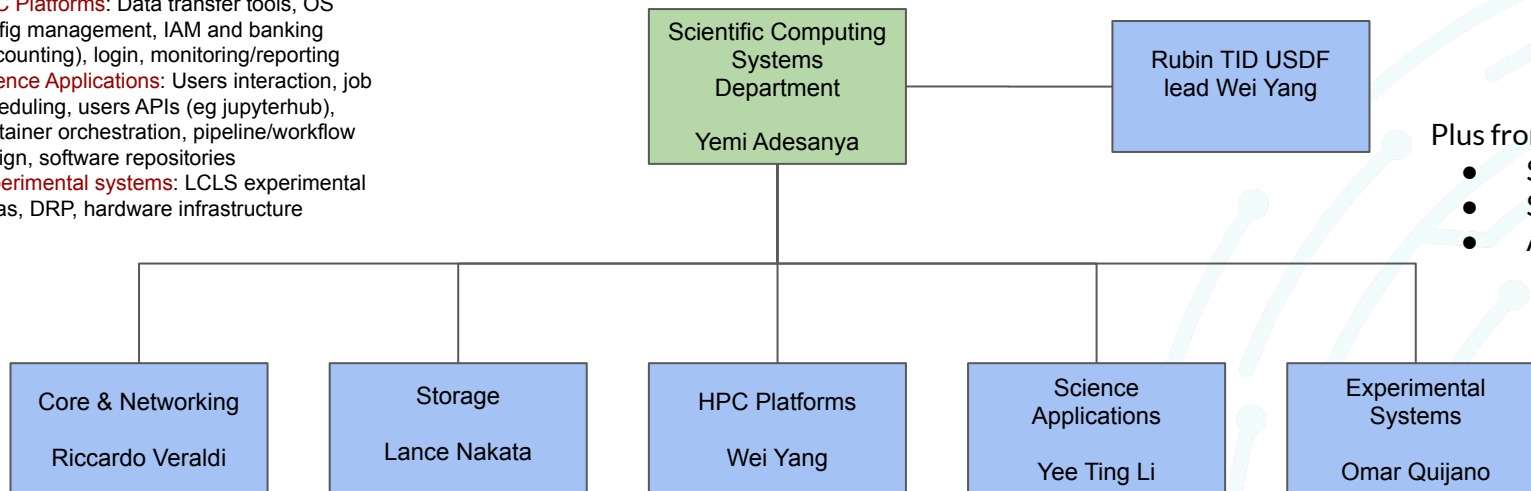
Storage: Posix and Object storage platforms (home, group, science data) tape archive and backup, migrations, policies, quota mgmt

HPC Platforms: Data transfer tools, OS config management, IAM and banking (accounting), login, monitoring/reporting

Science Applications: Users interaction, job scheduling, users APIs (eg jupyterhub), container orchestration, pipeline/workflow design, software repositories

Experimental systems: LCLS experimental areas, DRP, hardware infrastructure

Wei Yang acts as interface between Rubin & TID
Rubin & TID meet weekly to plan architecture

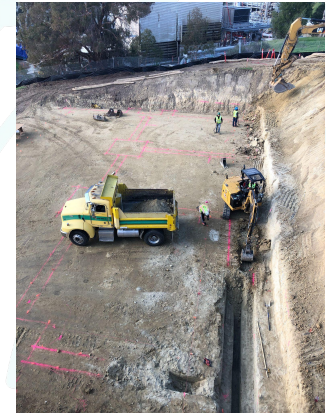


Plus from IT:

- Sysadmin effort
- Site networking
- Accounts/help desk

SRCF: Home of the S3DF

- S3DF - SLAC Shared Science Data Facility
- SRCF-II will double the current data center capabilities of the **Stanford Research Computing Facility** (aka SRCF-I)
 - Construction of the SRCF-II started last November
 - Beneficial occupancy - spring 2023
- Combined facility will offer **6 megawatts** of power and host **300 racks**
- SRCF has a **resilient** but not redundant **power infrastructure**
 - UPS and generator protected, providing significant assurance should there be a regional power outage
 - **Flywheel technology** allows for glitch-free power at all times and for smooth transitions to generator power in case of a power cut.
 - Cooling design is non-traditional and especially energy efficient: ambient air fan systems for 90% of the year (for the hotter days and for equipment needing chilled water, high-efficiency air cooled chillers are available)
- **Large fiber infrastructure** connecting experimental halls, and ESnet, with SRCF



Status of S3DF

- Launched a new core infrastructure - last week!
<https://s3df.slac.stanford.edu/public/doc/#/>
- Shiny new tape silo in production
- Home and group files will be on flash, on the Weka filesystem - all backed up
- Science data for Rubin is object store: Weka+ceph (flash tier with ceph in behind; POSIX and S3 access)
- deploy most services via kubernetes
- Slurm batch system - have started with a single roma partition

Rubin is the first group onto S3DF

Some Details

- With launch of S3DF, we're no longer requiring 2 accounts
 - Only the unix account matters
 - Password changes every 6 months currently; will likely change once DUO use expands^(*)
- No VPN needed
- DUO only needed for access to S3DF web portals; not for direct ssh or RSP
- Default home directory quotas are 25 GB. We're told they can be expanded upon request (presumably within reason)
- Providing separate 1 TB per user directories in Rubin space (/sdf/group/rubin/u)
- SDF and s3df filesystems are separate! Specifically, if you have already been on SDF, your home directories and /sdf/group/rubin are NOT the same on both systems. SDF accessed by /fs/ddn/sdf/

(*) Zero Trust Architecture is coming; White House mandate

How It Looks

- Dev guide doc:
 - <https://developer.lsst.io/usdf/lsst-login.html>
- Goal is to make it look much like NCSA (storage layout): /sdf/group/rubin/ et seq
- Login to bastion nodes: s3dflogin.slac.stanford.edu
- Jump to Rubin nodes rubin-dev1 (load balancer)
- Weeklies and releases provided via cvmfs
- Daily builds in shared filespace; also to deploy Jenkins linux workers on k8s
- Staff RSP
- “bps-batch”: plan on 3 options:
 - PanDA - same as production workflow system
 - Parsl - in heavy use by DESC
 - HTCondor

Lay of the Storage Land

```
[richard@sdflogin003 ~]$ ssh rubin-devl
```

```
┌───┬───┬───┬───┬───┬───┬───┬───┬───┬───┐  
│   │   │   │   │   │   │   │   │   │   │  
└───┴───┴───┴───┴───┴───┴───┴───┴───┴───┘
```

```
Last login: Fri Aug 5 20:51:06 2022 from 134.79.23.21
```

```
[richard@sdfrome001 ~]$ ls /sdf/group/rubin
```

```
datasets g repo sw u
```

Directory per user - 1 TB quota

Shared stack of daily builds

Butler repos

Shared storage O(1 PB)
replaces /project

Precursor data

File system organization (**flash**):

- **`/sdf/group/rubin/`** top level organization
 - Links to various places
- “Real” **home dirs** are `/sdf/home/<first_letter>/account_name`
- **Large Rubin usage in `/sdf/data/rubin/` (weka tiered)**

Will add links to `ncsa_home/` and `ncsa_jhome/` when ready

Lay of the Storage Land - Today

```
[richard@sdflogin003 ~]$ ssh rubin-dev1
```

```

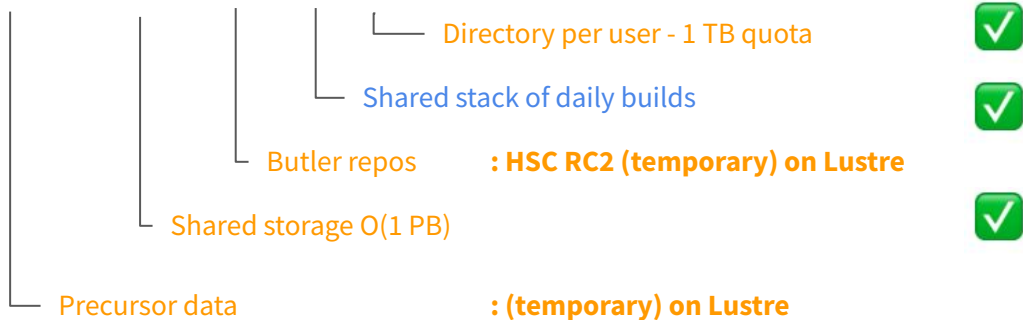
  ____|____|____|____|
 /  _  |  _  |  _  |  _  |
 \  _  |  _  |  _  |  _  |
 |  _  |  _  |  _  |  _  |

```

```
Last login: Fri Aug 5 20:51:06 2022 from 134.79.23.21
```

```
[richard@sdfrome001 ~]$ ls /sdf/group/rubin
```

```
datasets  g  repo  sw  u
```



- Can use cvmfs or shared dailies for the stack

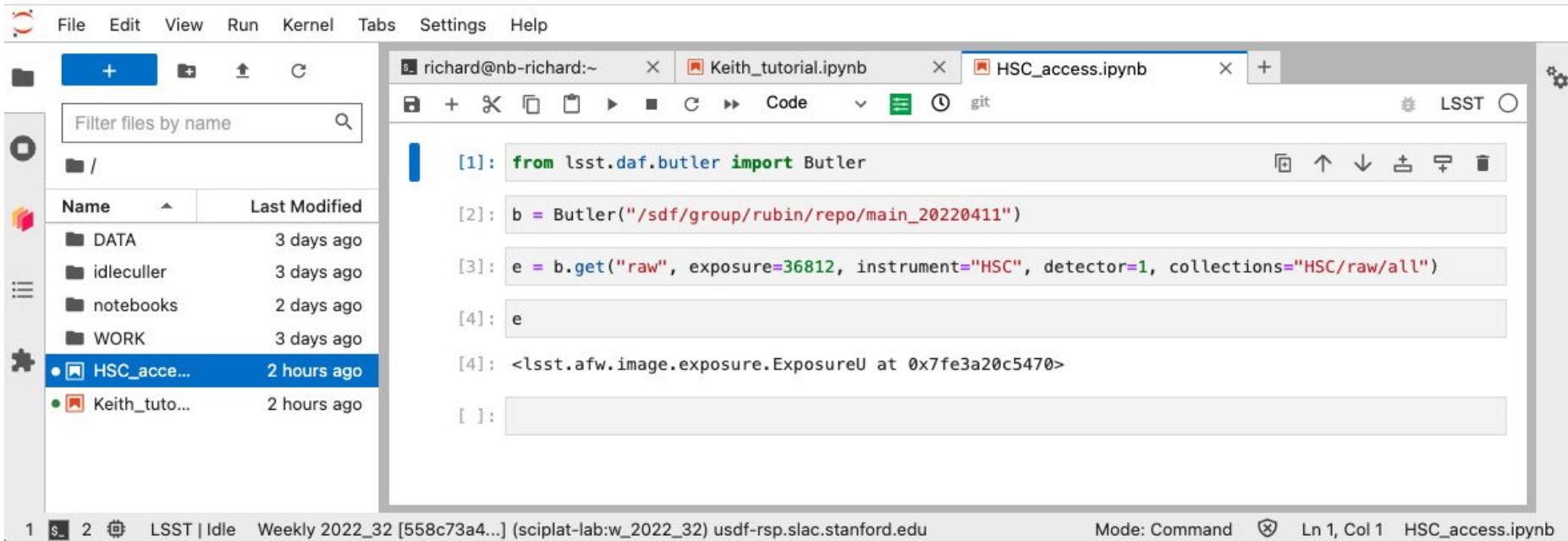
```
[richard@sdfrome001 richard]$ source /cvmfs/sw.lsst.eu/linux-x86_64/lsst_distrib/w_2022_32/loadLSST.bash
(lsst-scipipe-4.1.0) [richard@sdfrome001 richard]$ setup lsst_distrib
(lsst-scipipe-4.1.0) [richard@sdfrome001 richard]$ python
Python 3.10.5 | packaged by conda-forge | (main, Jun 14 2022, 07:04:59) [GCC 10.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from lsst.daf.butler import Butler
>>> b = Butler("/sdf/group/rubin/repo/main_20220411")
>>> e = b.get("raw", exposure=36812, instrument="HSC", detector=1, collections="HSC/raw/all")
>>> e
<lsst_afw_image_exposure.ExposureU object at 0x7f15b3597370>
```

Or, dailies:

```
[richard@sdfrome001 richard]$ source /sdf/group/rubin/sw/loadLSST.sh
(lsst-scipipe-4.1.0) [richard@sdfrome001 richard]$ setup lsst_distrib
(lsst-scipipe-4.1.0) [richard@sdfrome001 richard]$ python
```

Must put protected db-auth.yaml file in ~/.lsst
AND update db name from "butler" to "lsstdb1" if you
already have one.

Butler Access via usdf-rsp



The screenshot shows a JupyterLab environment. On the left is a file browser with a search bar and a table of files. The main area is a code editor with three tabs: 'richard@nb-richard:~', 'Keith_tutorial.ipynb', and 'HSC_access.ipynb'. The code in the active tab is as follows:

```
[1]: from lsst.daf.butler import Butler
[2]: b = Butler("/sdf/group/rubin/repo/main_20220411")
[3]: e = b.get("raw", exposure=36812, instrument="HSC", detector=1, collections="HSC/raw/all")
[4]: e
[4]: <lsst.afw.image.exposure.ExposureU at 0x7fe3a20c5470>
[ ]:
```

The bottom status bar shows the current environment is 'LSST | Idle', the working directory is 'Weekly 2022_32 [558c73a4...]', and the file path is '(sciplat-lab:w_2022_32) usdf-rsp.slac.stanford.edu'. The mode is 'Command' and the cursor is at 'Ln 1, Col 1' in the 'HSC_access.ipynb' file.

Known Issues and Open Policies

Known issues:

- ~~RSP seems to see some startup delays (~1 min due to nfs locking issue)~~
- Working on providing butler database authentication via existing unix account
 - In the meantime, workaround is shared secret
- NCSA data still being unpacked at SLAC
- Set up EFD
- Routine transfer of summit data
- Will need some setup on s3df for PanDA + sort out network proxies etc

Open Policies to Settle

- How to manage group/shared space? Free-for-all or quotaed...
- Backup policies for u/ and g/ (well, all the backup policies really)

Timeline

- S3DF core infrastructure released last week
 - 166 people have authorized access now - will send out an announcement shortly
- NCSA goes dark on Aug 15!!
 - Confirming files at SLAC vs NCSA - grabbing files up to Aug 1
 - Did we really get it all? Last chance to verify.
 - Placing files now - hard to predict how long to complete. Another week?
 - Will also need to rationalize uid/gid between sites
 - Long Haul Network overlay to SLAC set up and under test: image data, EFD, etc
 - Complete summit transfers
 - Get EFD populated
- Status of 4 PB storage array
 - Did not have time to get weka flash+ceph set up on it.
 - Installed zfs on 2 PB to get going as a proxy - will need to flip flop it when weka is ready on the other 2 PB
 - It may be we need to leave datasets/ on Lustre until our 11 PB comes in
 - In the meantime, /sdf/group/rubin is links to a hodge-podge of weka, zfs, nfs and Lustre...

- Resources in hand for ComCam era support
 - 4 PB for object store
 - 2000 cores - batch - includes loan from SLAC of 1024 cores.
 - 2 devl node (128 core/512 GB RAM each)
 - 500 k8s cores
- On order for LSSTCam
 - 11 PB disk & tape
 - 5k cores
 - 1k k8s cores
 - 3 data transfer nodes
 - 15 Qserv nodes

- A LOT of effort has gone into this transition, but I'd like to highlight a few individuals for going beyond the call of duty
- Many, many thanks go to our NCSA colleagues, especially Michelle Butler and Steve Pietrowicz
- SLAC TID accelerated their release schedule for us and put a lot of effort into getting the pieces together in time.
- Brandon White has lived the data transfer since April - not only "just" transferring 4 PB of small files from a complicated tree, but also juggling the receiving end where we've had to juggle file systems until S3DF really goes production
- Wei Yang and Yee Ting Li have been on the front lines for the infrastructure, and providing the links into TID
- K-T Lim provided a lot of needed adult supervision, coupled with his encyclopedic knowledge of Rubin
- (think Spock in the Scalosian Waters episode)

Questions?

questions go to **#ops-usdf**; announcements in **#ops-usdf-announce**