Next-Generation DTN Architecture/Kubernetes

Nautilus

John Graham
System Integration Engineer
Calit2/Qualcomm Institute, UCSD
jjgraham@eng.ucsd.edu

Dmitry Mishin, PhD.
Applications Programmer
SDSC, UCSD
dmishin@ucsd.edu

PACIFIC RESEARCH PLATFORM

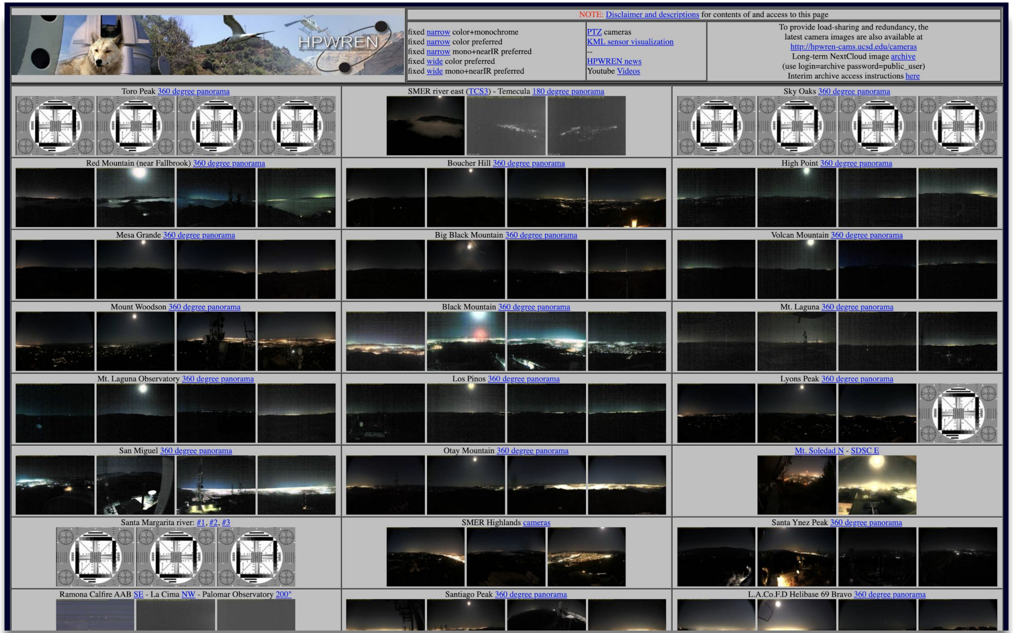
September 18th 2019

Rook: Ceph + EdgeFS

- HPWREN
  - EdgeFS
    - S3X (posix + S3)
    - Nextcloud
- Nautilus
  - Ceph Block / CephFS / S3
    - Nextcloud/Aria2
    - Globus (container)

PACIFIC RESEARCH PLATFORM

HPWREN K8s HA cluster



NOTE: Disclaimer and descriptions for contents of and access to this page

fixed narrow color+monochrome
fixed narrow color preferred
fixed narrow mono+nearIR preferred
fixed wide color preferred
fixed wide mono+nearIR preferred

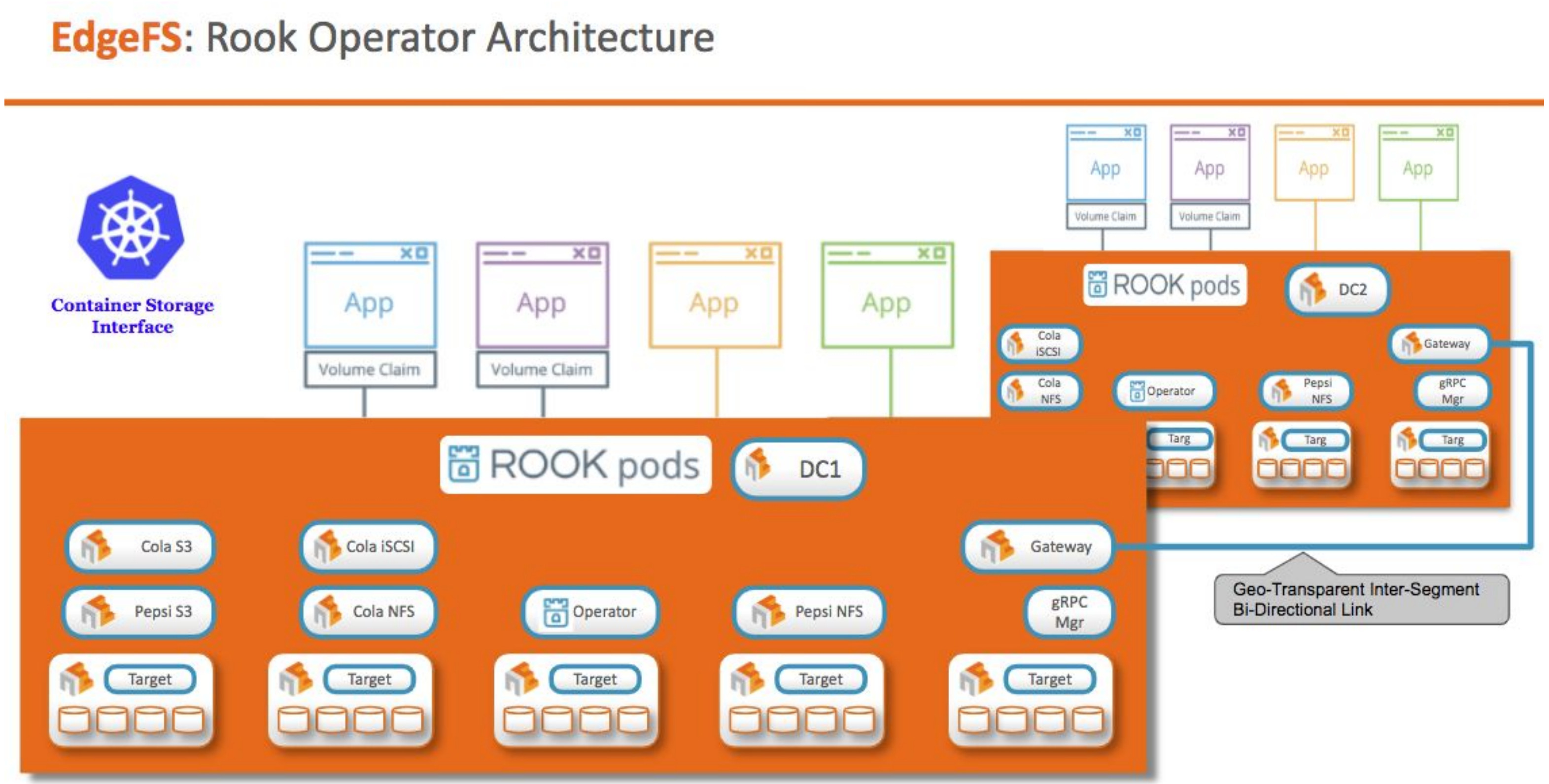
PTZ cameras
KML sensor visualization
--
HPWREN news
Youtube Videos

To provide load-sharing and redundancy, the latest camera images are also available at http://hpwren-cams.ucsd.edu/cameras
Long-term NextCloud image archive
(use login=archive password=public_user)
Interim archive access instructions here

HPWREN
Toro Peak 360 degree panorama
SMER river east (TCS3) - Temecula 180 degree panorama
Sky Oaks 360 degree panorama
Red Mountain (near Fallbrook) 360 degree panorama
Boucher Hill 360 degree panorama
High Point 360 degree panorama
Mesa Grande 360 degree panorama
Big Black Mountain 360 degree panorama
Volcan Mountain 360 degree panorama
Mount Woodson 360 degree panorama
Black Mountain 360 degree panorama
Mt. Laguna 360 degree panorama
Mt. Laguna Observatory 360 degree panorama
Los Pinos 360 degree panorama
Lyons Peak 360 degree panorama
San Miguel 360 degree panorama
Otay Mountain 360 degree panorama
Mt. Soledad N - SDSC E
Santa Margarita river: #1, #2, #3
SMER Highlands cameras
Santa Ynez Peak 360 degree panorama
Ramona Calfire AAB SE - La Cima NW - Palomar Observatory 200°
Santiago Peak 360 degree panorama
L.A.Co.F.D Helibase 69 Bravo 360 degree panorama

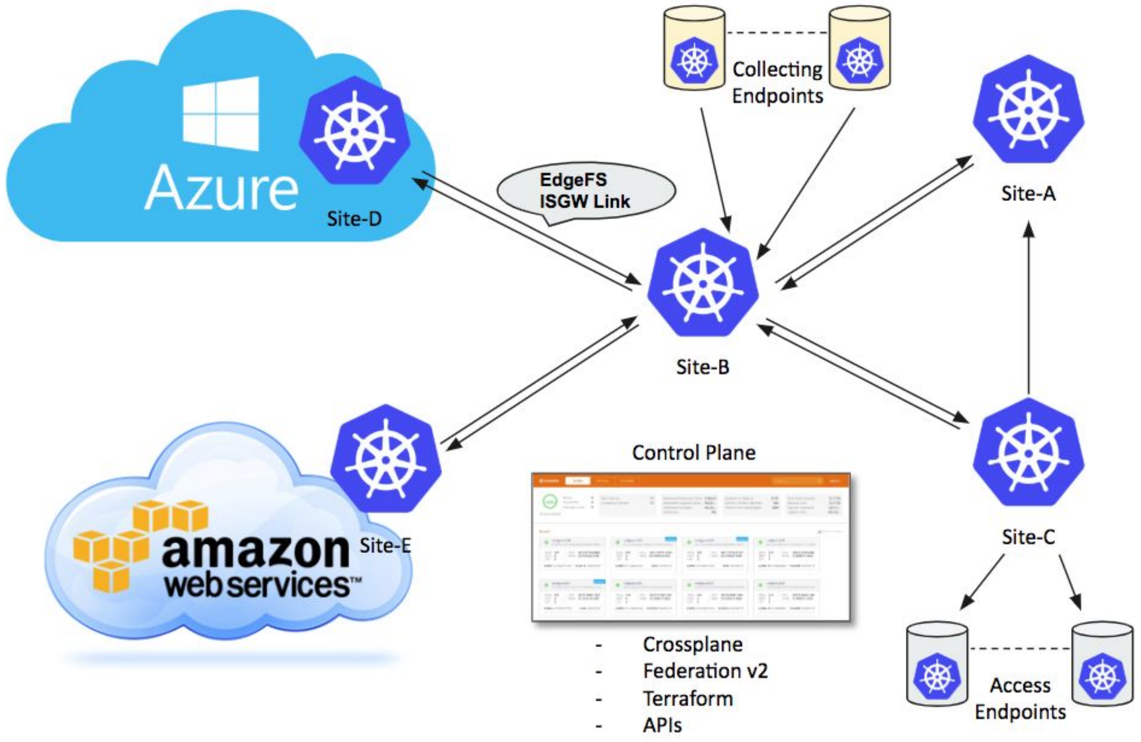
- m1.calit2.optiputer.net
- m2.calit2.optiputer.net
- m3.calit2.optiputer.net
- k8s-hpwren-01.sdsc.optiputer.net
- k8s-hpwren-02.sdsc.optiputer.net
- ps-100g-scidmz-0.tools.ucla.net
- c3.hpwren.calit2.uci.edu
- fiona-100g.ucsc.edu

Rook Operator for EdgeFS



Rook enables EdgeFS storage systems to run on Kubernetes using Kubernetes primitives.

EdgeFS ISGW (Inter-Segment GateWay) CRD

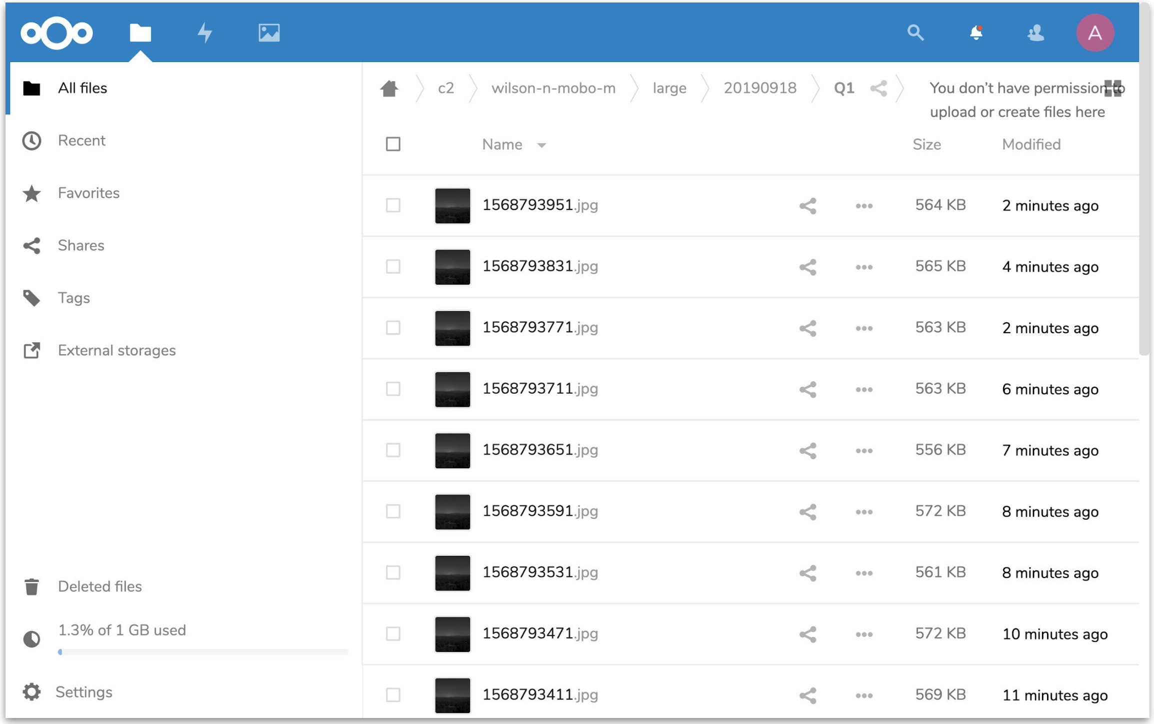


EdgeFS Inter-Segment Gateway link is a building block for EdgeFS cross-site, cross-cloud global namespace synchronization functionality.

It distributes modified chunks of data asynchronously and enables seamless as well as geographically transparent access to files, objects and block devices.

It is important to note that a file or a block device consists of one or more objects, and so, within EdgeFS scope, ultimately everything is an object, globally immutable and self-validated.

HPWREN Public Facing Nextcloud

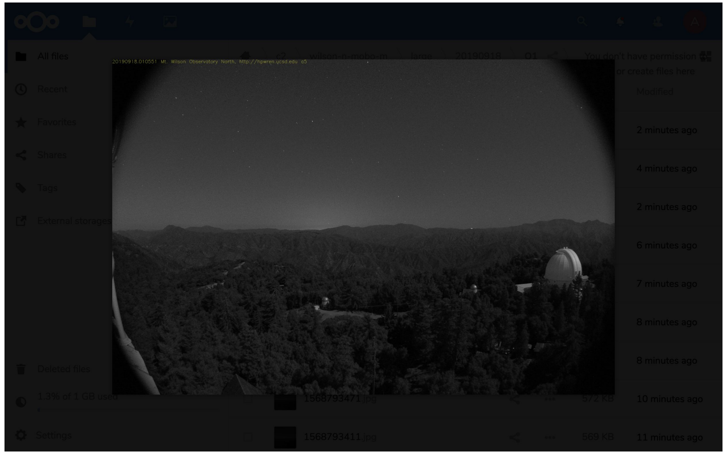


- All files
- Recent
- Favorites
- Shares
- Tags
- External storages
- Deleted files
- 1.3% of 1 GB used
- Settings

c2 〉 wilson-n-mobo-m 〉 large 〉 20190918 〉 Q1

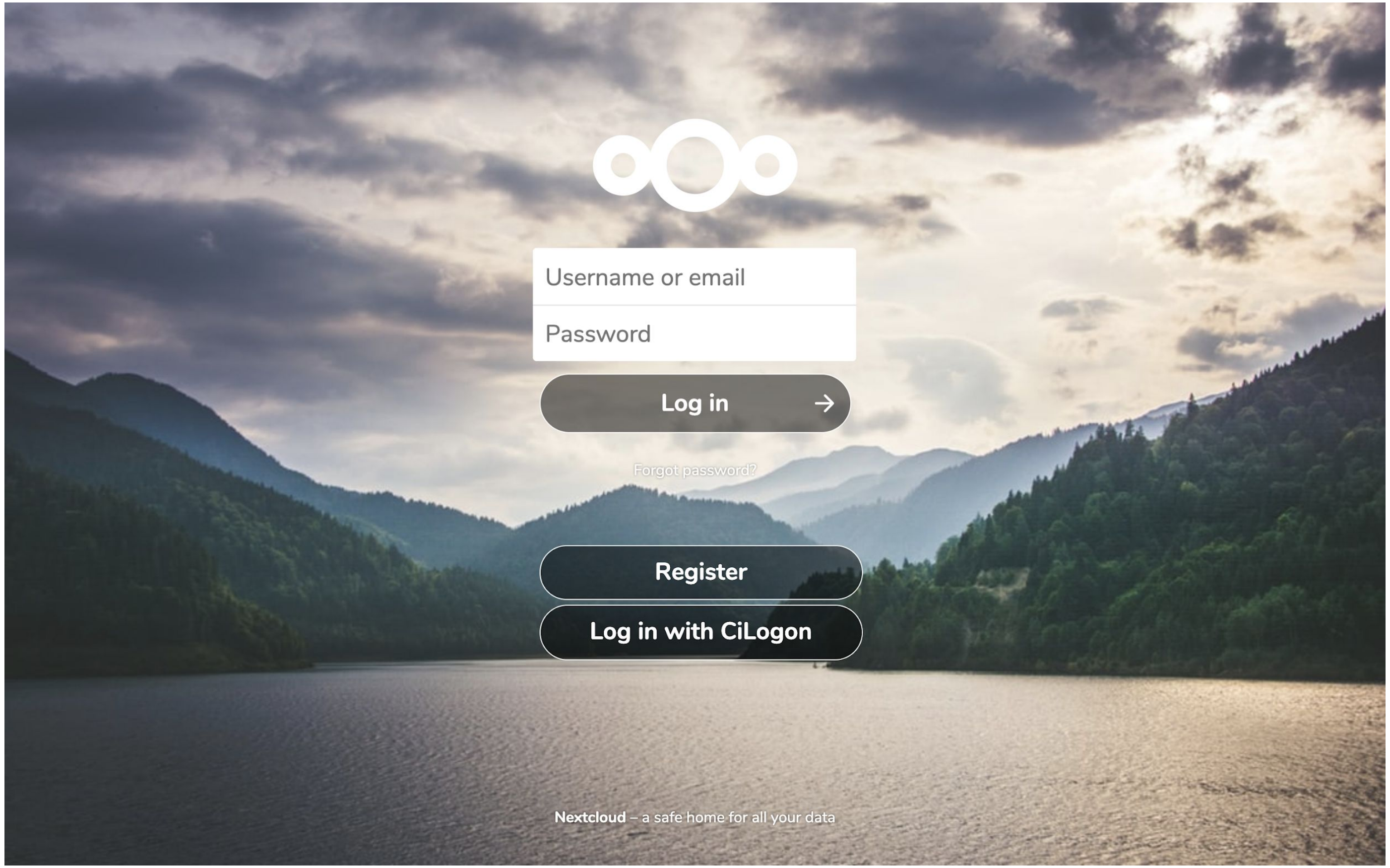
You don't have permission to upload or create files here

Name	Size	Modified
1568793951.jpg	564 KB	2 minutes ago
1568793831.jpg	565 KB	4 minutes ago
1568793771.jpg	563 KB	2 minutes ago
1568793711.jpg	563 KB	6 minutes ago
1568793651.jpg	556 KB	7 minutes ago
1568793591.jpg	572 KB	8 minutes ago
1568793531.jpg	561 KB	8 minutes ago
1568793471.jpg	572 KB	10 minutes ago
1568793411.jpg	569 KB	11 minutes ago



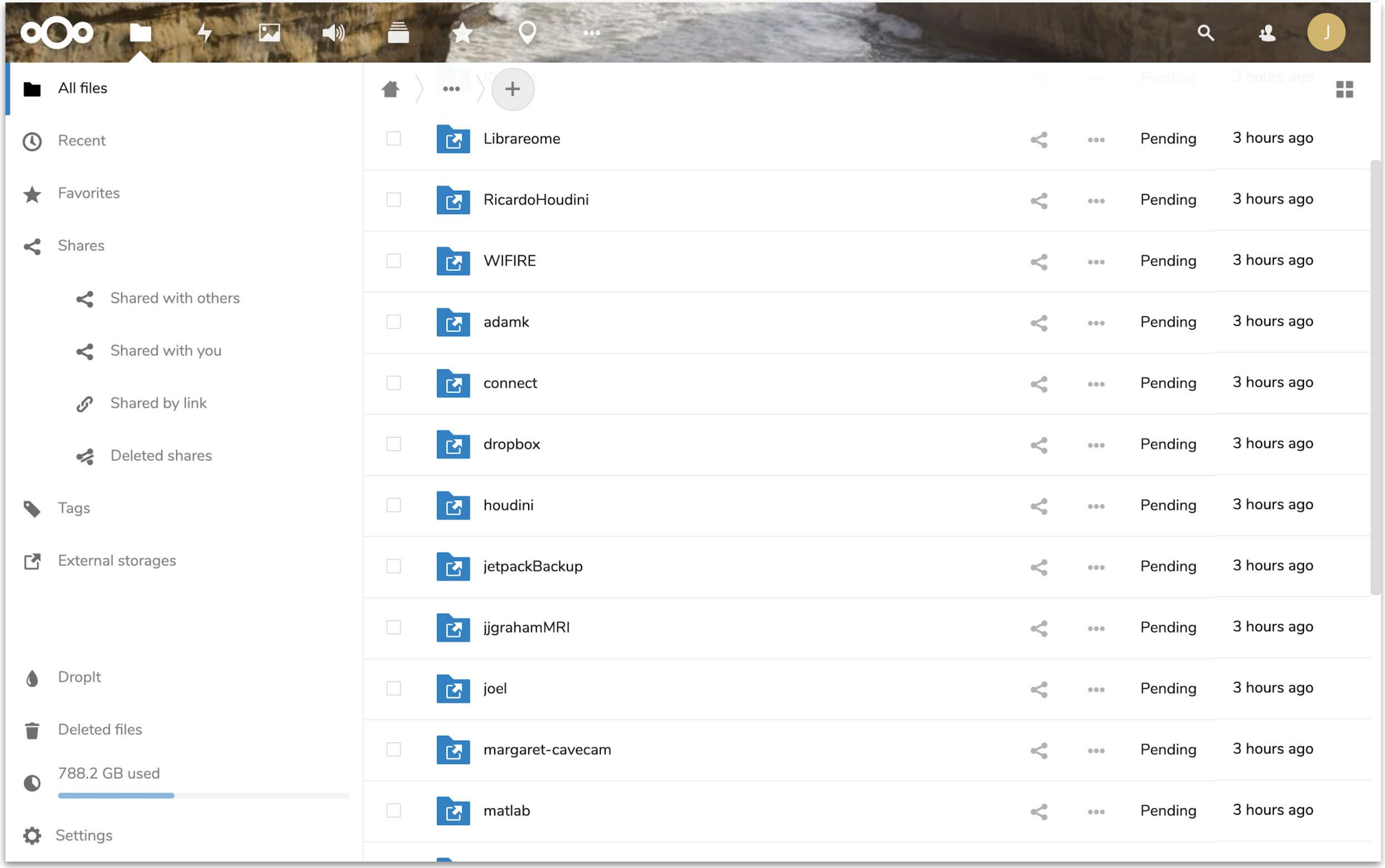
September 18th 2019

Nautilus Nextcloud



PACIFIC RESEARCH PLATFORM

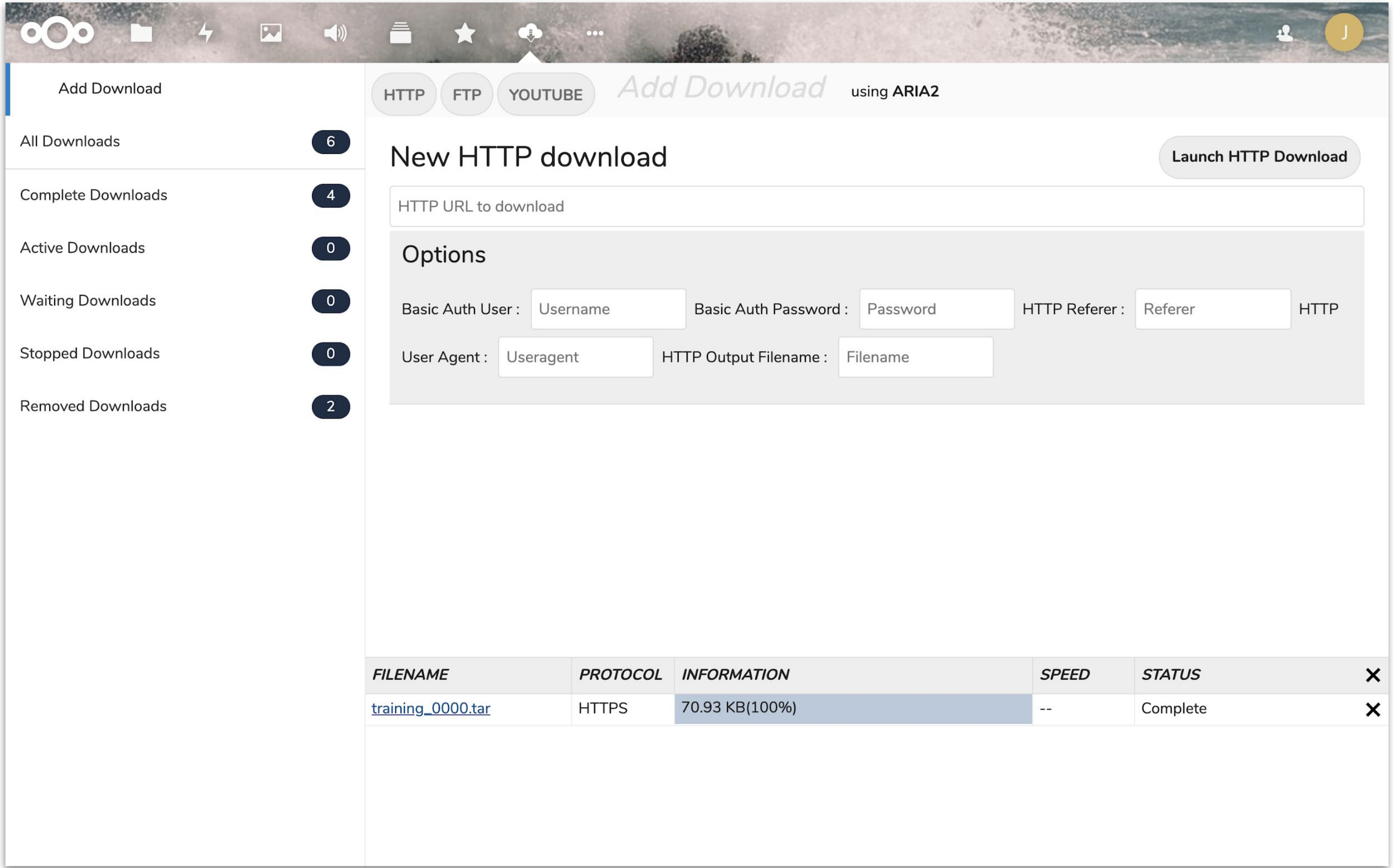
Nautilus NextCloud S3



Librareome	Pending	3 hours ago
RicardoHoudini	Pending	3 hours ago
WIFIRE	Pending	3 hours ago
adamk	Pending	3 hours ago
connect	Pending	3 hours ago
dropbox	Pending	3 hours ago
houdini	Pending	3 hours ago
jetpackBackup	Pending	3 hours ago
jjgrahamMRI	Pending	3 hours ago
joel	Pending	3 hours ago
margaret-cavecam	Pending	3 hours ago
matlab	Pending	3 hours ago

PACIFIC RESEARCH PLATFORM

Nextcloud OCDownloader (Aria2)



Add Download	
All Downloads	6
Complete Downloads	4
Active Downloads	0
Waiting Downloads	0
Stopped Downloads	0
Removed Downloads	2

New HTTP download

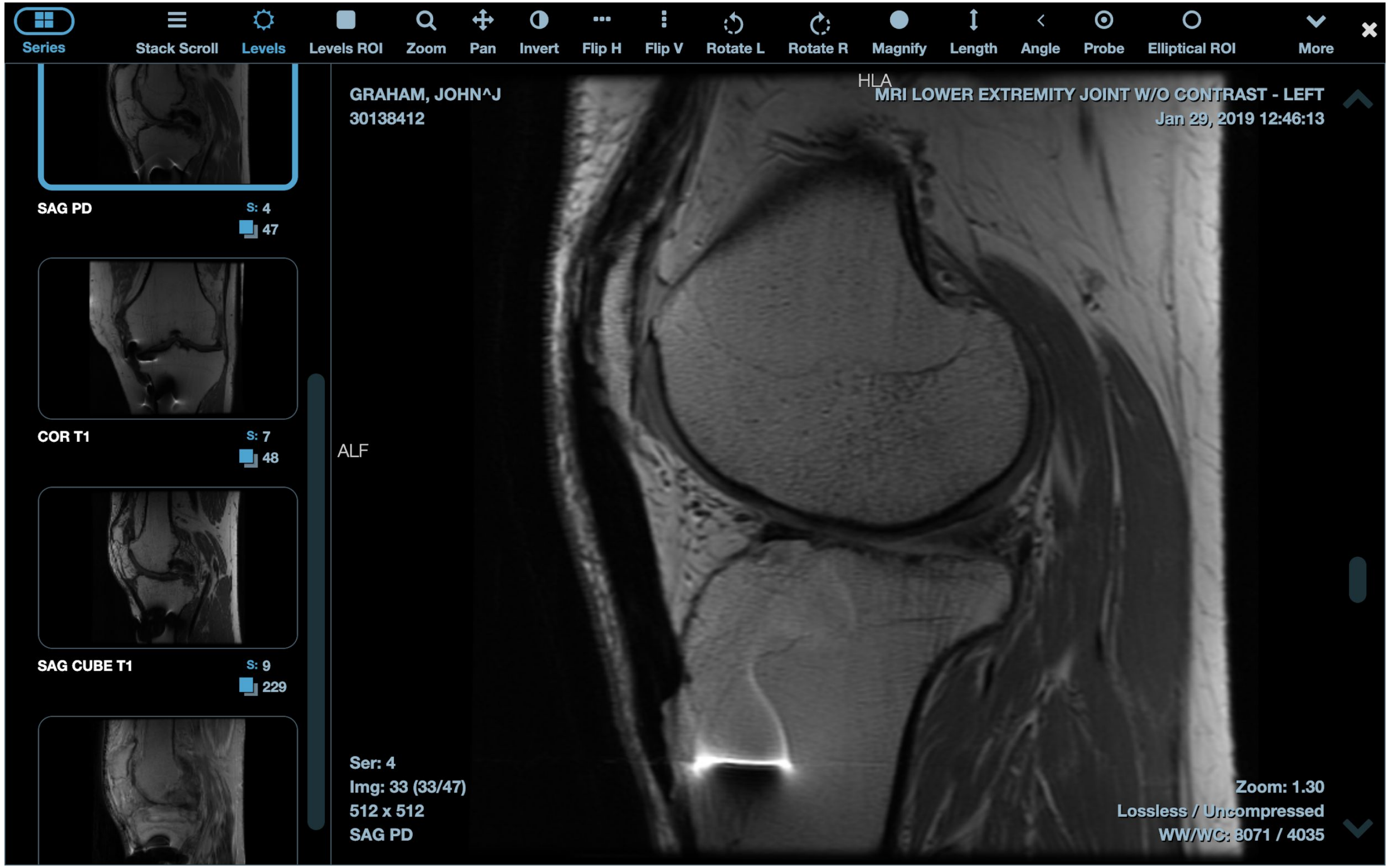
HTTP URL to download

Options

Basic Auth User : Username
Basic Auth Password : Password
HTTP Referer : Referer
HTTP User Agent : Useragent
HTTP Output Filename : Filename

FILENAME	PROTOCOL	INFORMATION	SPEED	STATUS	✕
training_0000.tar	HTTPS	70.93 KB(100%)	--	Complete	✕

Nextcloud DICOM Viewer



SAG PD S: 4 47

COR T1 S: 7 48

SAG CUBE T1 S: 9 229

GRAHAM, JOHN^J
30138412

HLA
MRI LOWER EXTREMITY JOINT W/O CONTRAST - LEFT
Jan 29, 2019 12:46:13

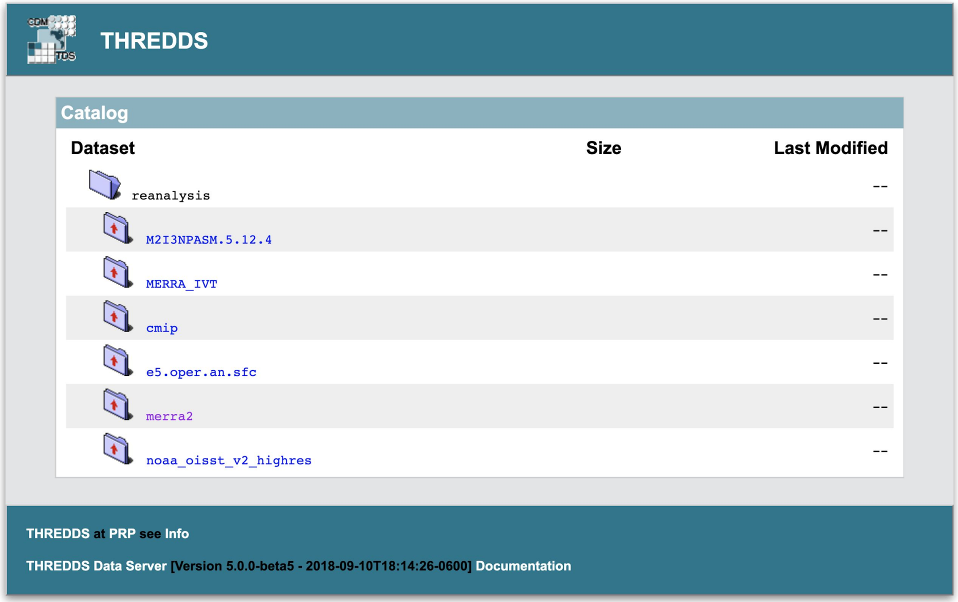
ALF

Ser: 4
Img: 33 (33/47)
512 x 512
SAG PD

Zoom: 1.30
Lossless / Uncompressed
WW/WC: 8071 / 4035

PACIFIC RESEARCH PLATFORM

Globus transfer container for THREDDS



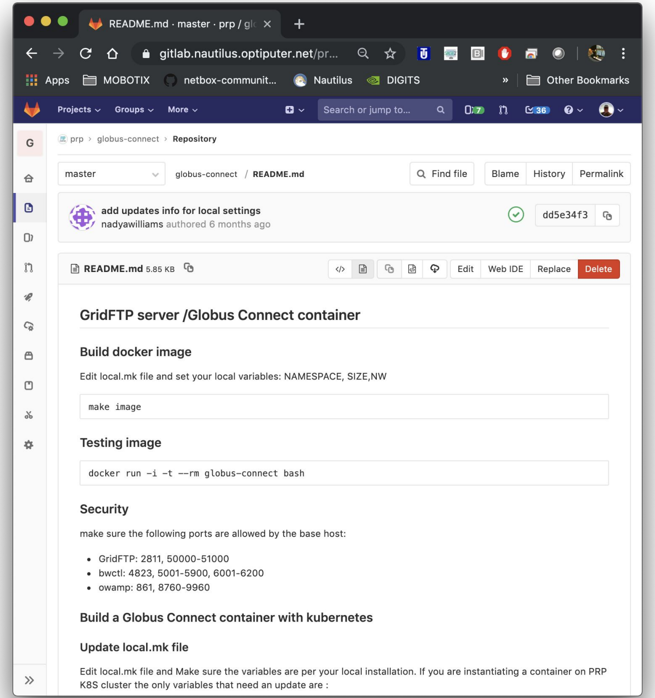
THREDDS

Catalog

Dataset	Size	Last Modified
reanalysis		--
M2I3NPASM.5.12.4		--
MERRA_IVT		--
cmip		--
e5.oper.an.sfc		--
merra2		--
noaa_oisst_v2_highres		--

THREDDS at PRP see Info

THREDDS Data Server [Version 5.0.0-beta5 - 2018-09-10T18:14:26-0600] Documentation



add updates info for local settings
nadyawilliams authored 6 months ago
dd5e34f3

README.md 5.85 KB

GridFTP server /Globus Connect container

Build docker image

Edit local.mk file and set your local variables: NAMESPACE, SIZE,NW

```
make image
```

Testing image

```
docker run -i -t --rm globus-connect bash
```

Security

make sure the following ports are allowed by the base host:

- GridFTP: 2811, 50000-51000
- bwctl: 4823, 5001-5900, 6001-6200
- owamp: 861, 8760-9960

Build a Globus Connect container with kubernetes

Update local.mk file

Edit local.mk file and Make sure the variables are per your local installation. If you are instantiating a container on PRP K8S cluster the only variables that need an update are :

https://gitlab.nautilus.optiputer.net/prp/globus-connect/blob/master/README.md

Braingeneers

Name	Institution	Software	PI
braingeneers	UCSC	Tensorflow	David Haussler

Description

The Braingeneers are developing the infrastructure to grow cortical organoids at scale and interface with them in order to record and stimulate neurons. This will enable the application of modern AI approaches to uncover how genetic changes enhanced human brain architecture and computing capacity during primate evolution as well as to better understand how neurons function towards back porting this into in-silico machine learning models.

Admins

Rob Currie
David Freeman Parks

Users

Ash Seymour Robbins
Kateryna Voitiuk
Alexander Thomas Spaeth
Aviv Elor
Craig Patrick Hunter
Nick Justin Rezaee
Yatish G Turakhia
Nico Hawthorne
Lucas Jean-Louis, Francis Seninge
Stephen Jen-Der Hwang
Erich Weiler

PACIFIC RESEARCH PLATFORM

David Parks

dfparks@ucsc.edu

I'm currently using **S3/PRP to train deep learning models** on neural recordings of freely behaving mice. A 12 hour recording on 256 electrodes is over ***1TB of data that is housed on S3.*** I utilize the PRP GPUs for training on tensorflow and perform random-access reads of **3.6GB data files** at nearly **100MB/sec data read rates on S3 from within the PRP cluster.** The most basic of these models is trained to detect the animals sleep state (wake, non-rem, rem sleep) from raw neural recordings, and can be seen here: https://www.youtube.com/watch?v=bUdxS29UIvw&feature=youtu.be.

The Braingeneers at UCSC and Genomics Institute at large here is ramping up to work with large datasets like this on a regular basis. Using S3 for storage of those datasets presents a very manageable and scalable platform on which we can run basic data processing, storage, ML workflows, and many other data analysis tasks across a team of people, institutions (UCSC, USF, WUSTL), and devices.

I'm **currently working on a more complex set of models** which require training on **10M image files.** I've uploaded the 10M files to our S3 bucket and am using the project as a litmus test for using S3 as a larger scale datastore for future operations.

Dimitry has been instrumental in helping make S3 capable of scaling to both massive datasets as well as massive numbers of objects. He's solved two significant performance issues which caused our first attempts to upload the 10M file dataset to fail. With those issues out of the way **we now have the files uploaded and I'm able to access them at scale.**

PACIFIC RESEARCH PLATFORM

David Parks

dfparks@ucsc.edu

Key takeaways in terms of S3 performance:

- I've been able to **random-access read large (3.6GB) files on S3 in 5MB chunks at up to 1.5 GB per second in a single python process** (requiring about 50GB of RAM to service a large queue of as many as 1000 parallel IO operations), S3 didn't blink at these tests, which Dimitry was monitoring as I ran them.

- I have had troubles achieving the same with random HDF5 file reads because HDF5 limits you to a less efficient multiprocessing approach rather than the more efficient asyncio. **Tests on large HDF5 files using random-access reads have achieved about 200MB/sec read rates.** I may yet solve that problem and get parallelization of HDF5 reads significantly higher, but it's a work in progress.

- When uploading large numbers of files (10M in my testing) the S3 upload speeds are now very good, **I was getting 80 files per sec upload and that appeared to be scaling nearly linearly with the number of `aws s3 sync` processes I ran** (I got up to 4, and did not test higher).

David Parks

dfparks@ucsc.edu

- **Once we got 10M files uploaded, access to the bucket has remained performant and scalable**.

- The only drawback we currently have is that a **bucket will become nearly inaccessible while a large scale upload like this is going on. Operations like `aws s3 ls` typically timeout,** and the upload process slows for periods of 5-10 minutes before getting back to full speed. Since this only occurs during massive data load operations it's not a show-stopper, but it does leave room for performance improvement.

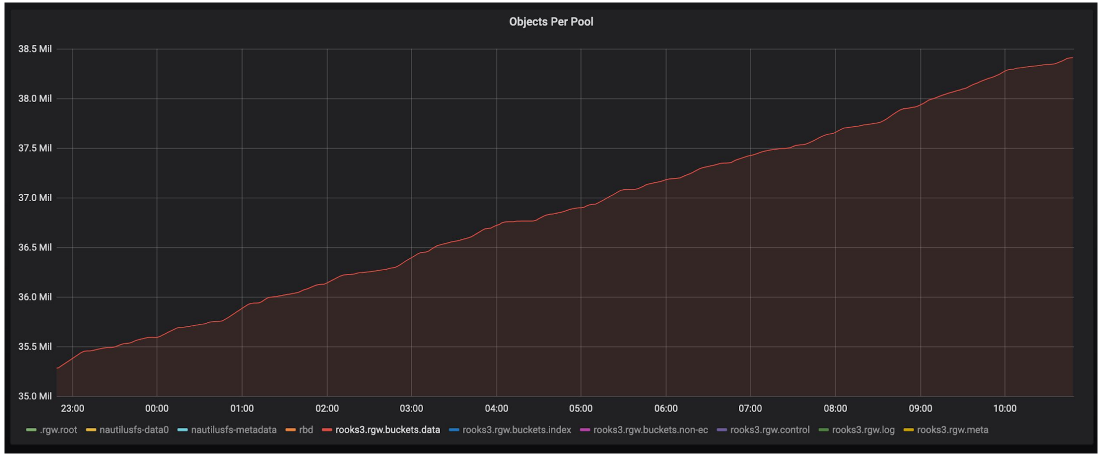
- **My last takeaway is that I wouldn't actually store 10M individual files on S3 again.** A better approach is to **use an smaller set of uncompressed zip files** to store the many individual files, then perform random-access reads against S3 to pull out the file that's needed. However this approach requires a non-trivial implementation of boto3, one I'm still working to perfect in python.

David Parks

dfparks@ucsc.edu

Rob and I were talking about **cost difference** between running what I've done to date on **PRP/S3 vs. AWS/S3**. While we didn't figure out any specific number, it's quite substantial. I've performed operations like this on AWS S3 in the past and come away with **10k+ monthly** billing statements.

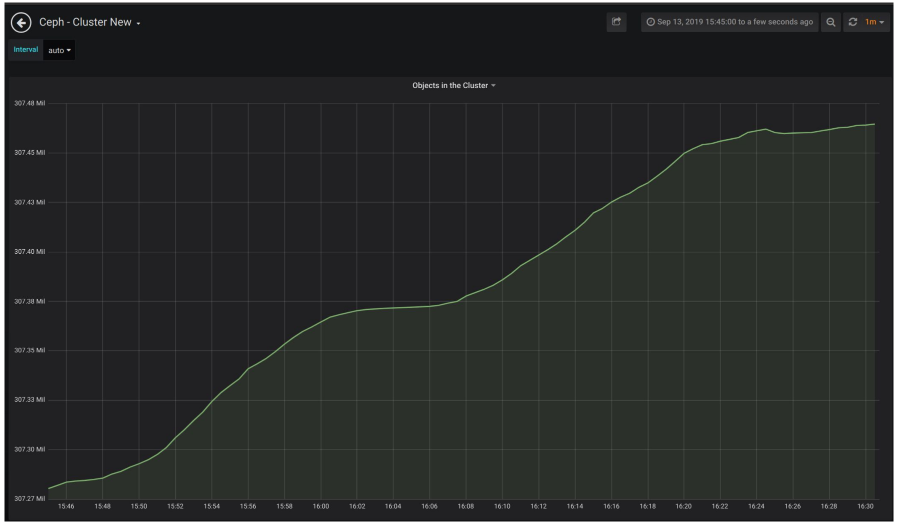
Below are a few Grafana snapshots we took over time to illustrate some of the points above. This first one shows consistent **80 files per second being uploaded to S3**, though as noted I expect the performance can go higher.



David Parks

dfparks@ucsc.edu

This Grafana screenshot is a zoom of the one above, in it we can see that the upload rate charges ahead at periods, and then tapers off in a cyclic fashion. **During the periods where it tapers off the bucket is less accessible for operations like `aws s3 ls`.** On average we still achieve high upload rates, but **the upload does seem to take its toll on the metadata store on S3.** We are still looking into where a solution to this is possible. If so it would really blow the top off performance of S3.



PACIFIC RESEARCH PLATFORM

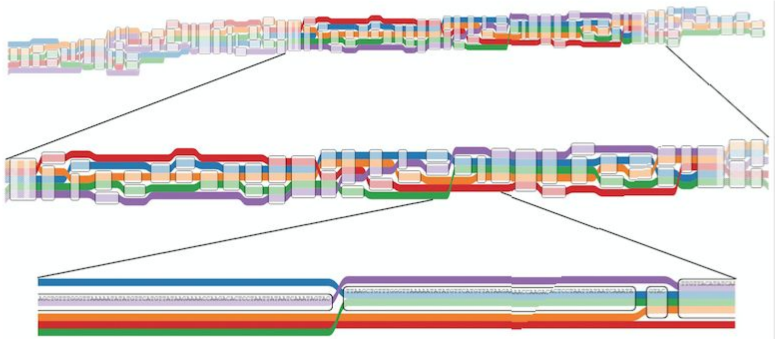
Rob Currie

rcurrie@ucsc.edu

Mike Perez (community manager @ Redhat for CEPH) and Carlos Maltzahn (CROSS @ UCSC) have pinged me about doing a CEPH day @ UCSC.

In parallel I'd like to arrange a working group between Mike (and maybe an engineer he designates), Dmitry, David, our Cluster Admins and possibly a grad student or two.

In addition to the Braingeneers that David is working on we're about to start an NIH funded project called the '**pan genome**' (will replace the current linear reference which UCSC assembled version of 1 of in 2000). It will involve sequencing **350+ individuals where each file will be 3-9TB H5 files** (the first 10 people will be 100TB H5 files). **All data will be stored in an on-prem CEPH cluster as well as Google Cloud.** Computation requires about 48 hours per genome using 8 2080's plus a ton of other things (one step required a 2TB memory machine...) and we'll likely re-process all data many many times. **Ultimately it will generate the pan genome ie a graph of humanity:**



Thanks to Our Support:

- US National Science Foundation (NSF) awards
  - ➢ CNS 0821155, CNS-1338192, CNS-1456638, CNS-1730158, ACI-1540112, & ACI-1541349
- University of California Office of the President CIO
- UCSD Chancellor’s Integrated Digital Infrastructure Program
- UCSD Next Generation Networking initiative
- Calit2 and Calit2 Qualcomm Institute
- CENIC, PacificWave and StarLight
- DOE ESnet

PRP
PACIFIC RESEARCH PLATFORM