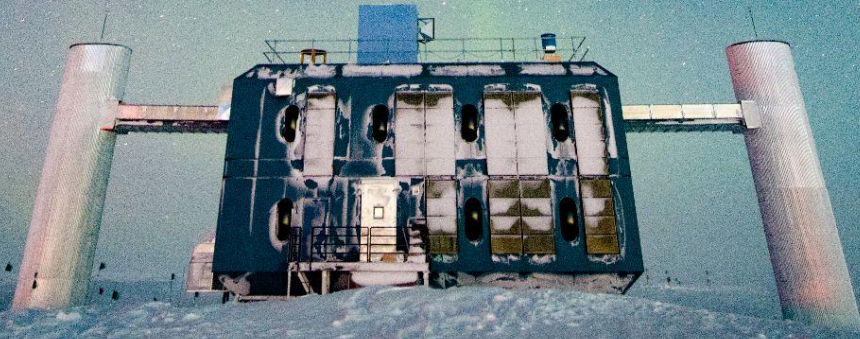


# GitHub, NPX, And the Grid Oh my!

Jessie & Vedant  
IceCube Summer School 2024



*GitHub*



# GitHub setup

- Create a free Github account (<https://github.com/>)
- To join the IceCube organization, you need to include:
  - Your full name in your GitHub account profile (J. Smith is OK)
  - Include your current institution in your account profile
  - Set up 2 factor authentication - REQUIRED for the IceCube org on GitHub
- You can ask in #software or #icecube-it on Slack for an invite (we will do this as a group for Summer School!)
- Take a look at the [IceCube Github Guide](#)

# Set up on your local computer or cobalt!

- You can install github on your computer

Linux: `sudo apt-get install git`

Mac: `sudo port install git`

- Also preinstalled on Cobalt (IceCube machines)
- Set your name and email for your command line client (replace stuff in quotes with your own):

```
git config --global user.name "First Last"
```

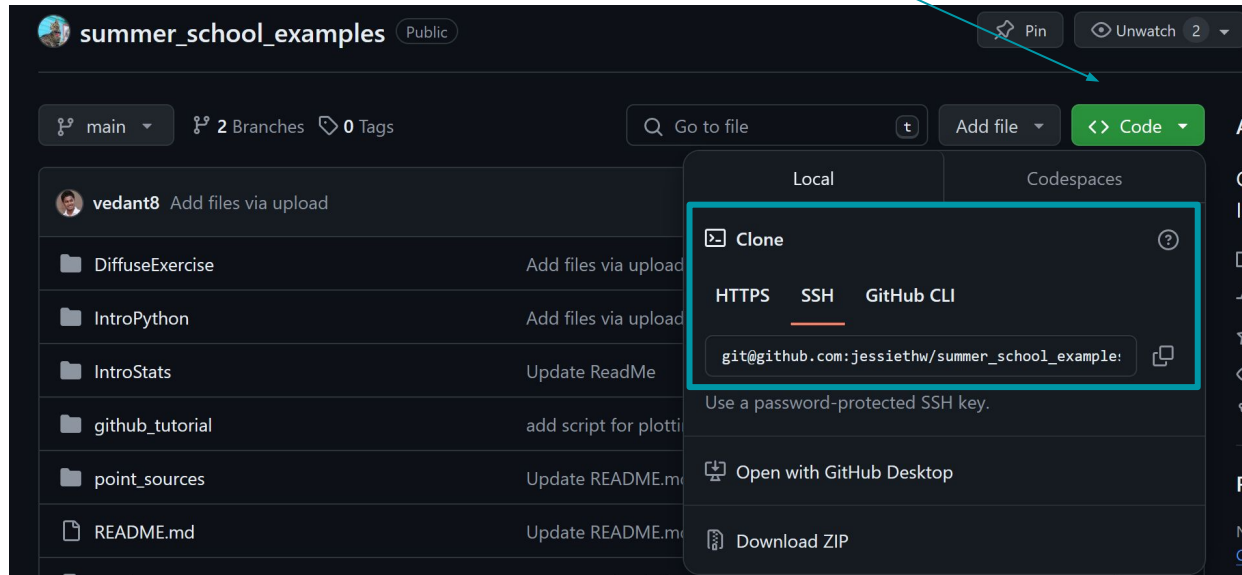
```
git config --global user.email "user@icecube.wisc.edu"
```

- Make sure this account is associated with your GitHub account (can have many!)
- Add and use your ssh keys
  - Nearly impossible to push commits with git on the command line otherwise
  - Follow instructions here: [generating ssh keys](#)

# Do a first clone

- Go here: [https://github.com/jessiethw/summer\\_school\\_examples](https://github.com/jessiethw/summer_school_examples)
- Under the “Code” tab: copy the SSH link & type this into your terminal

```
git clone git@github.com:jessiethw/summer_school_examples
```



The screenshot shows the GitHub interface for the repository 'summer\_school\_examples'. The 'Code' button is highlighted with a red box, and a red arrow points from the text 'copy the SSH link & type this into your terminal' to the SSH link 'git@github.com:jessiethw/summer\_school\_example:' in the dropdown menu. The dropdown menu also shows options for 'Local' and 'Codespaces', and the 'SSH' option is selected.

# Make your branch

Create a new branch for yourself via the command line

```
git checkout -b <branchname>
```

Now you're on your own branch – make some changes (maybe change the readme? Or try out a notebook there). Then check the status (this will list any changed files)

```
git status
```

Add and commit files you've changed to your local branch

```
git add <filename>
```

```
git commit -m "my first commit"
```

A note about good commits: you should write a descriptive message, and make sure that the changes you've staged can be easily associated with a single commit (don't add too much for one commit!)

# Pushing to a remote

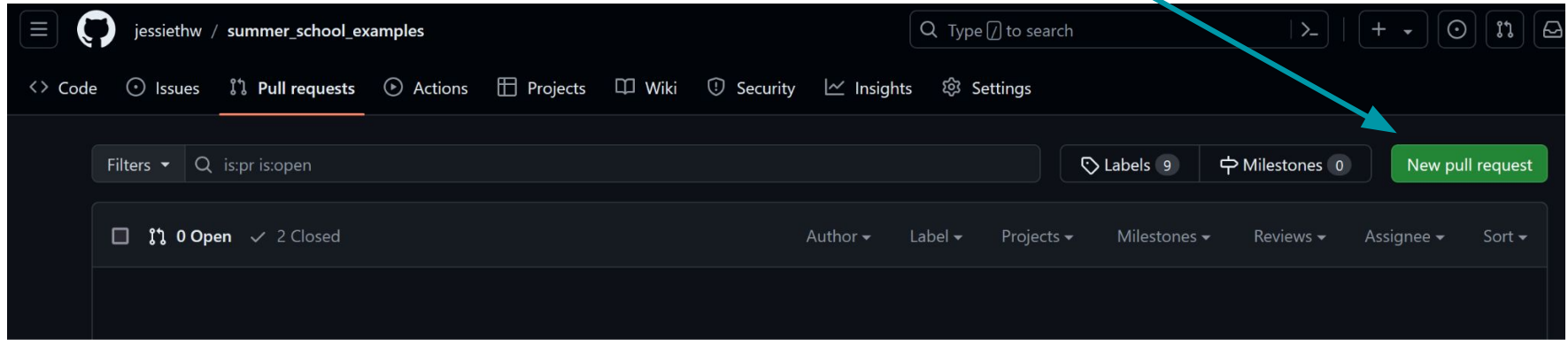
It's good practice to pull before you push (sometimes the remote changes!)

```
git pull
```

Push your changes to the remote

```
git push
```

Once you've pushed all your changes, you can also open a pull request!



The screenshot shows the GitHub web interface for a repository named 'summer\_school\_examples' by user 'jessiethw'. The 'Pull requests' tab is selected and highlighted in red. A search bar contains the query 'is:pr is:open'. To the right of the search bar are buttons for 'Labels 9' and 'Milestones 0'. A prominent green button labeled 'New pull request' is visible. A red arrow points from the text 'you can also open a pull request!' to this button. Below the search bar, there is a summary of pull requests: '0 Open' and '2 Closed'. At the bottom, there are dropdown menus for 'Author', 'Label', 'Projects', 'Milestones', 'Reviews', 'Assignee', and 'Sort'.

*NPX*



<https://wiki.icecube.wisc.edu/index.php/Condor8>



# NPX setup

- Create an icetray python script to iterate over an i3 data file
  - Or swipe one!  
"/home/vbasu/scripts/MCTreeReader\_2024SummerSchool.py"
- How do we run it? We specify an input, output and geometry GCD file
  - `python /home/vbasu/scripts/MCTreeReader_2024SummerSchool.py`  
`-i/data/ana/MESE/NuGen/SnowStorm/L3/L3_NewIce_022018_016000.i3.zst`  
`-o /$HOME/testoutput`  
`-g/cvmfs/icecube.opensciencegrid.org/data/GCD/GeoCalibDetectorStatus_AVG_55697-57531_PASS2_SPE_withScaledNoise.i3.gz`

Create a shell script

```
#!/bin/bash

#prep inputs

input=$1

output=$2

GCD_file=$3

#setup env

source /home/vbasu/test/testdev_sh.sh

eval `cvmfs/icecube.opensciencegrid.org/py3-v4.1.1/setup.sh`

export HDF5_USE_FILE_LOCKING='FALSE'

#run job

/cvmfs/icecube.opensciencegrid.org/users/vbasu/meta-projects/combo3/build/env-shell.sh python
/home/vbasu/scripts/CorsikaShowerReader.py -i ${input} -o ${output} -g ${GCD_file}
```

## Create a submit script

```
should_transfer_files = YES
transfer_input_files = SummerSchool24.sh
# # but do not try to copy outputs back (see: https://htcondor-wiki.cs.wisc.edu/index.cgi/tktview?tn=3081)
+TransferOutput=""
executable = SummerSchool24.sh

output = /scratch/${USER}/SummerSchool24.out
error = /scratch/${USER}/SummerSchool24.err
log = /scratch/${USER}/SummerSchool24.log

universe = vanilla
notification = never
Requirements = (OpSysMajorVer == 7 || OSGVO_OS_VERSION == 7)
+SingularityImage="/cvmfs/singularity.opensciencegrid.org/opensciencegrid/osgvo-el7:latest"

request_memory = (NumJobStarts is undefined) ? 2 * pow(2, 10) : 2048 * pow(2, NumJobStarts + 1)
periodic_release = (HoldReasonCode =?= 21 && HoldReasonSubCode =?= 1001) || HoldReasonCode =?= 21
periodic_remove = (JobStatus =?= 5 && (HoldReasonCode != 34 && HoldReasonCode != 21)) || (RequestMemory > 13192)

arguments = $(infile) $(outfile) $(gcd)
queue
```

Bonus points: Create a DAG!

```
JOB SummerSchool_020904_000000_reader SummerSchool24.sub
```

```
VARs SummerSchool_020904_000000_reader
```

```
infile="/data/sim/IceCube/2016/filtered/level2/CORSIKA-in-ice/20904/0000000-0000999/Level2_IC86.2016_corsika.020904.000000.i3.zst"
```

```
outfile="/home/${USER}/L2_020904_000000"
```

```
gcd="/cvmfs/icecube.opensciencegrid.org/data/GCD/GeoCalibDetectorStatus_AVG_55697-57531_PASS2_SPE_withScaledNoise.i3.gz"
```

```
filenum="020904_000000"
```

```
Retry SummerSchool_020904_000000_reader 2
```

RUN!

```
submitter ~ $ condor_submit SummerSchool24.sub
```

```
Submitting job(s).
```

```
1 job(s) submitted to cluster 12898721.
```

```
submitter ~ $ condor_submit_dag SummerSchool24.dag
```

```
Renaming rescue DAGs newer than number 0
```

```
-----  
File for submitting this DAG to HTCondor           : SummerSchool24.dag.condor.sub  
Log of DAGMan debugging messages                   : SummerSchool24.dag.dagman.out  
Log of HTCondor library output                     : SummerSchool24.dag.lib.out  
Log of HTCondor library error messages             : SummerSchool24.dag.lib.err  
Log of the life of condor_dagman itself            : SummerSchool24.dag.dagman.log
```

```
Submitting job(s).
```

```
1 job(s) submitted to cluster 210840367.
```

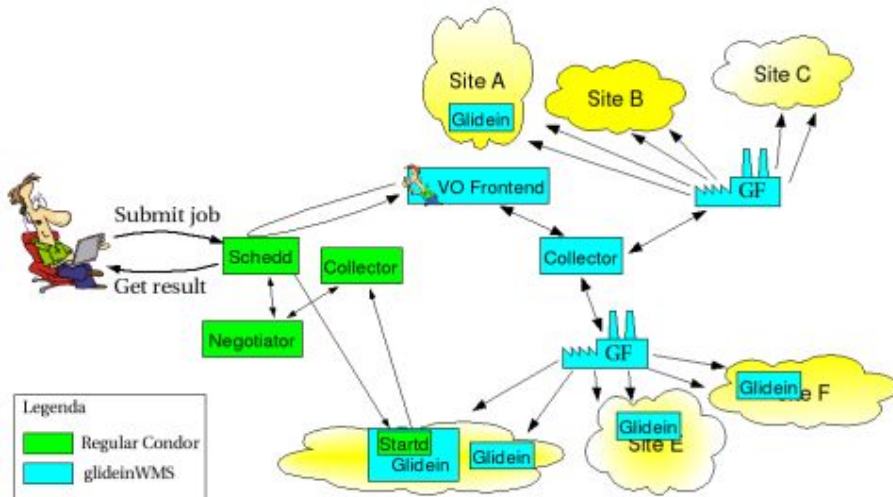
## Monitor!

```
vbasu@submit-1 ~/scripts/NNMFit/Scripts/Stage2PreUnblinding $ condor_q
```

```
-- Schedd: submit-1.icecube.wisc.edu : <10.128.11.86:25462?... @ 05/30/24 22:16:27
```

OWNER	BATCH_NAME		SUBMITTED	DONE	RUN	IDLE	TOTAL	JOB_IDS
vbasu	SummerSchool24.dag+210830953	5/30 22:12	_	_	_	_	210840366.0	

# Grid



# Grid Setup

- Set up a [proxy certificate](#)
- Remote sites do not have access to software in /home or /data/user
- Need to add software to user CVMFS, a remote filesystem which is accessible to Grid sites
- Data files can be transferred using GridFTP or HTTP



# User CVMFS

- For config files and scripts: From Cobalt, anything written to `/net/cvmfs_users/$USER` is synced to CVMFS every hour, so should appear in CVMFS within 2 hours. Each user has 10GB of space, so this typically isn't enough for all your data.
- For Software:
  - Make the software portable, by building in sub-2's scratch.
  - Use a singularity container to stage the installation i.e. Make the scripts think they are in the `/cvmfs/icecube.opensciencegrid.org/users/USERNAME` folder, and not in `/net/cvmfs_users/USERNAME`:

# Accessing IceTray on the Grid

- The following example builds a release candidate of icetray on local scratch against the py3-v4.1.1 toolset and installs the built metaproject to USERNAME's CVMFS staging area.

```
eval $(/cvmfs/icecube.opensciencegrid.org/py3-v4.1.1/setup.sh)
cd /scratch/USERNAME
mkdir icetray; cd icetray
git clone -b v1.3.0-beta.0 git@github.com:icecube/icetray.git src
mkdir build; cd build
cmake ../src -DCMAKE_INSTALL_PREFIX=/net/cvmfs_users/USERNAME/icetray/v1.3.0-beta.0
-DCMAKE_BUILD_TYPE=Release
nice make -kj16
make install
```

- Once the staging area is published, you can enter the metaproject environment with

```
/cvmfs/icecube.opensciencegrid.org/users/USERNAME/icetray/v1.3.0-beta.0
```

# Accessing IceTray on the Grid

- Building IceTray in singularity

```
singularity exec -B /tmp:/tmp -B /cvmfs:/cvmfs -B  
/net/cvmfs_users/$USER:/cvmfs/icecube.opensciencegrid.org/users/$USER -c  
/cvmfs/singularity.opensciencegrid.org/opensciencegrid/osgvo-el7:latest bash
```

```
cd /cvmfs/icecube.opensciencegrid.org/users/\$USER  
mkdir metaproject
```

```
cd metaproject  
cp -r /data/user/$USER/icetray-src/ src  
mkdir build
```

```
cd build
```

```
eval `cvmfs/icecube.opensciencegrid.org/PYTHON-VERSION/setup.sh`  
cmake -DCMAKE_BUILD_TYPE=Release  
/cvmfs/icecube.opensciencegrid.org/users/USERNAME/metaproject/src
```

```
make -j 8
```

# Grid Submit files

```
# # The executable itself will always be staged if the node is on a different filesystem.
should_transfer_files = YES
transfer_input_files =script.py
# # but do not try to copy outputs back
(see:https://htcondor-wiki.cs.wisc.edu/index.cgi/tktview?tn=3081)
+TransferOutput=""
executable =script.py
output = /scratch/$USER/script.out
error = /scratch/$USER/script.err
log = /scratch/$USER/script.log
universe = vanilla
notification = never
+OriginalTime = 21600
Requirements = HAS_CVMFS_icecube_opensciencegrid_org && ((OpSysMajorVer =?= 7) || (OSG_OS_VERSION ==
"7")) && (GLIDEIN_Site != "Nebraska")
+SingularityImage="/cvmfs/singularity.opensciencegrid.org/opensciencegrid/osgvo-el7:latest"
request_memory = 6GB
arguments = -i$(infile) -o$(outfile) -g$(gcd) -f$(grid)
use_x509userproxy=true
```

# Grid Submit files

```
# # The executable itself will always be staged if the node is on a different filesystem.
should_transfer_files = YES
transfer_input_files =script.py
# # but do not try to copy outputs back
(see:https://htcondor-wiki.cs.wisc.edu/index.cgi/tktview?tn=3081)
+TransferOutput=""
executable =script.py
output = /scratch/$USER/script.out
error = /scratch/$USER/script.err
log = /scratch/$USER/script.log
universe = vanilla
notification = never
+OriginalTime = 21600
Requirements = HAS_CVMFS_icecube_opensciencegrid_org && ((OpSysMajorVer =?= 7) || (OSG_OS_VERSION ==
"7")) && (GLIDEIN_Site != "Nebraska")
+SingularityImage="/cvmfs/singularity.opensciencegrid.org/opensciencegrid/osgvo-el7:latest"
request_memory = 6GB
arguments = -i$(infile) -o$(outfile) -g$(gcd) -f$(grid)
use_x509userproxy=true
queue
```

# Grid Computing - Data Transfer

GridFTP file transfer - the old way of handling large data

```
#!/bin/bash
set -e
eval $(/cvmfs/icecube.opensciencegrid.org/py3-v4.2.1/setup.sh)
infile = $1
shift;
outfile = $1
shift;
echo "transferring input $infile"
globus-url-copy gsiftp://gridftp.icecube.wisc.edu/$infile file:/$PWD/infile

echo "running program"
$SROOT/metaprojects/icetray/v1.5.1/env-shell.sh python physics.py
infile outfile $@

echo "transferring output $outfile"
globus-url-copy file:/$PWD/outfile
gsiftp://gridftp.icecube.wisc.edu/$outfile

echo "Job complete!"
```

```
transfer_input_files = physics.py, x509
transfer_output_files =
arguments = /data/user/me/my_file.i3
/data/user/me/out_file.i3 --prog args
```

```
dschultz@sub-2 $ grid-proxy-init -valid 24:0 -out x509
Your identity: /DC=org/DC=cilogon/C=US/O=University of
Wisconsin-Madison/CN=David Schultz B47305562
Enter GRID pass phrase for this identity:
Creating proxy
.....
..... Done
Your proxy is valid until: Thu Jun  8 21:22:03 2023

dschultz@sub-2 $ condor_submit job.sub
```

# Grid Computing - Data Transfer

GridFTP file transfer - the old way of handling large data

```
gridftp=bool(options.GridFTP)
if gridftp==True:
    grid = 'gsift://gridftp-users.icecube.wisc.edu'
    outfile = grid+options.OUTPUT
    gcdFile=options.GCD
    infiles=[options.GCD, grid+options.INPUT]

if gridftp==True:
    tray.context['I3FileStager'] = dataio.get_stagers()

    tray.AddModule('I3Reader', 'reader', FilenameList=infiles)
    print ("Reading input file...", grid+options.INPUT)
else:
    tray.AddModule('I3Reader', 'reader', FilenameList=[options.GCD,
infiles])
    print ("Reading input file...", options.INPUT)
```

```
transfer_input_files = physics.py, x509
transfer_output_files =
arguments = /data/user/me/my_file.i3
/data/user/me/out_file.i3 --prog args
```

```
dschultz@sub-2 $ grid-proxy-init -valid 24:0 -out x509
Your identity: /DC=org/DC=cilogon/C=US/O=University of
Wisconsin-Madison/CN=David Schultz B47305562
Enter GRID pass phrase for this identity:
Creating proxy
.....
..... Done
Your proxy is valid until: Thu Jun  8 21:22:03 2023

dschultz@sub-2 $ condor_submit job.sub
```

# Grid Computing - Data Transfer

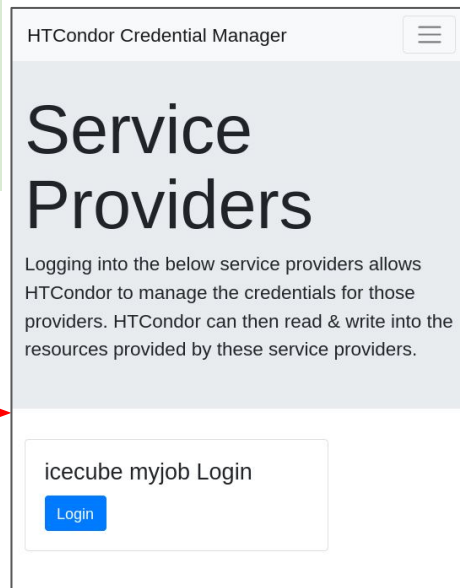
## 3. HTTP file transfer - the new way of handling large data

```
# required lines for http transfers
use_oauth_services = icecube
icecube_oauth_permissions_myjobs = offline_access

transfer_input_files =
physics.py,icecube.myjobs+https://data.icecube.aq/data/user/me/my_input.i3
transfer_output_files = out.hdf5
transfer_output_remaps = "out.hdf5 =
icecube.myjobs+https://data.icecube.aq/data/user/me/out.%(ClusterId).%(ProcId).hdf5"
```

```
dschultz@sub-2:~$ condor_submit job.sub
Submitting job(s)
Hello, dschultz.
Please visit:
http://localhost:22280/key/5b2dfca80ec4b5ebce55c40b114c40ab57290
3171e37efba065de29a2789999e
```

After logging in, `condor_submit` will work



The screenshot shows the HTCondor Credential Manager interface. At the top, it says "HTCondor Credential Manager" with a menu icon. Below that, the heading "Service Providers" is displayed in large, bold letters. Underneath, there is a paragraph of text: "Logging into the below service providers allows HTCondor to manage the credentials for those providers. HTCondor can then read & write into the resources provided by these service providers." At the bottom of the interface, there is a box for "icecube myjob Login" with a blue "Login" button.