

rt.iu.edu



RESEARCH TECHNOLOGIES
UNIVERSITY INFORMATION TECHNOLOGY SERVICES



Jetstream Overview: A national research and education cloud

**CSCI B649 – School of Informatics – Indiana University
February 25, 2020 – Bloomington, IN.**

*Jeremy Fischer – Jeremy@iu.edu - Indiana University
Manager, Jetstream Cloud, UITS Research Technologies*

*Fischer, J. (2019). Jetstream Overview: A national research and
education cloud. Moon Township, PA. Retrieved from
<https://jetstream-cloud.org/research/publications.php>*

What is Jetstream and why does it exist?

- *NSF's first production cloud facility*
- *Focus on ease-of-use, broad accessibility*
- *User-selectable library of preconfigured virtual machines*
- *Provides **on-demand** interactive computing and analysis or persistent services such as gateways*
- *Enables configurable environments; **programmable cyberinfrastructure***

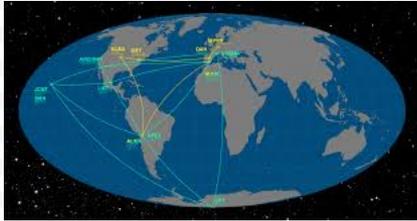


Who uses Jetstream?

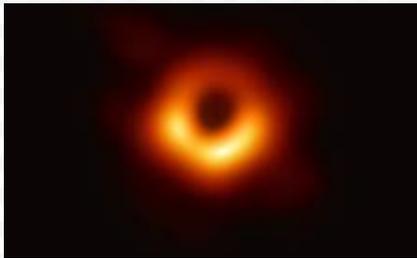
- *The researcher needing a handful of cores (1 to 44/vCPU)*
- *Software creators and researchers needing to create their own VMs and workflows*
- *Science gateway creators using Jetstream as either the frontend or processor for scientific jobs*
- *STEM Educators teaching on a variety of subjects*



M87 black hole: how cloud computing supports astronomy



Event Horizon Telescope

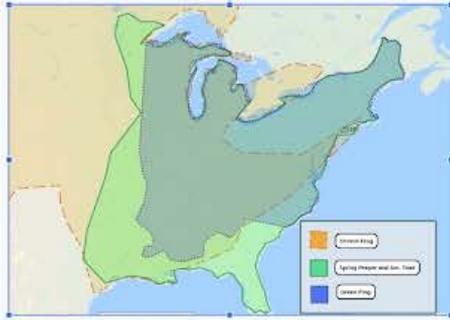


*M87 black hole image
generated by EHT*

- *Event Horizon Telescope (a telescope array consisting of a global network of radio telescopes), a large number of scientists, NASA spacecraft, and a variety of computing resources enabled the first image of a black hole.*
- *For the M87 black hole image, two critical steps were done in the cloud and piloted on Jetstream*
 - *correcting for anomalies, so that further image processing could occur, and*
 - *large survey study of how image reconstruction algorithms affect the final images.*
- *The team is also developing new methods to correlate data from multiple telescopes (to reduce data from petascale to terascale) in the cloud*



AI for Everyone – Recognizing Frog Calls



American Toad



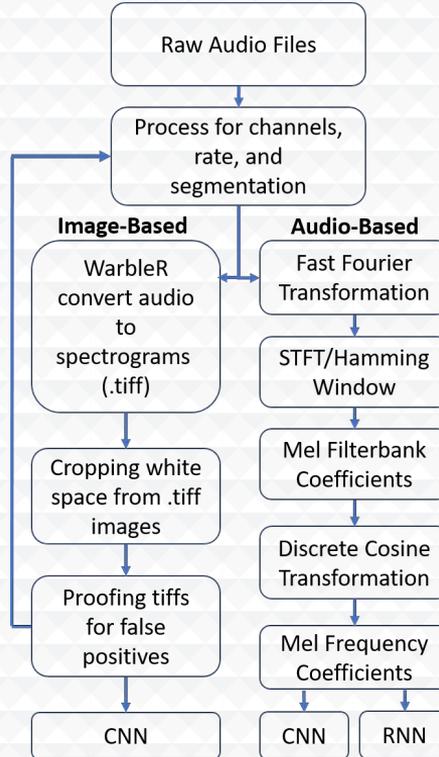
Spring Peeper



Chorus Frog



Green Frog



2019 Jetstream REU participants examined best-practices for supporting AI projects for field biologists



Project TERRIER: modernizing political science research



Journalists, through their words, provide an invaluable service, sharing information about global events to which many of us would not otherwise have access.

They send missives directly from event sites, recording during protests, summits, and speeches. For political scientists, these articles offer a rich mine of data about these events.

Project Temporally Extended, Regular, Reproducible International Event Records ([TERRIER](#)) extracts event data from roughly 300M news articles and puts it into a form usable by researchers.

Jetstream provides the storage and structure to launch the pipeline and process news articles for TERRIER.



What Jetstream isn't...

- *It's not traditional HPC*
- *There's no shared filesystem (think cloudy!)*
- *There's no high-end interconnect fabric (keep thinking cloudy!)*
- *There aren't GPUs widely available*
- *It isn't Amazon, Azure, or GCE (similar, but...)*



HPC vs Cloud

Adapting to a different environment:

- No reservations, no queueing – more interactive usage
- Being your own admin – hey, we have root!**
- You really can have almost any (linux) software you want**

** Here there be dragons...



Jetstream and way of the cloud...

- **Cloudy Technologies:** clouds are more than just virtual machines (VM)
 - **Old way:** robust (expensive) infrastructure, weak (cheap) software
 - You expect the hardware to not fail
 - State is maintained in volatile data structures
 - **Cloudy way:** commodity infrastructure, robust software
 - Expect & plan for infrastructure to fail
 - Put intelligence into the software to handle infrastructure failure
- **And my favorite...**



Thinking about VMs...

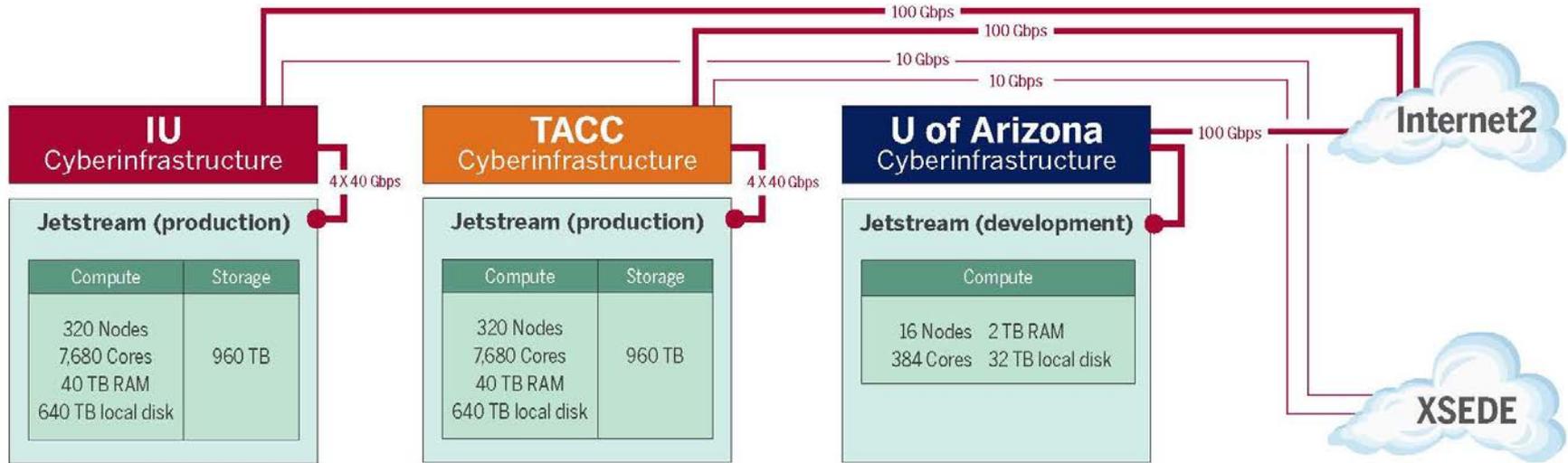
Cows, not pets: pets take great amount of care, feeding, and you name them; cows you intend to have high turnover and you give them numbers.

-- Mike Lowe (Jetstream architect)

***some caveats for gateways...*



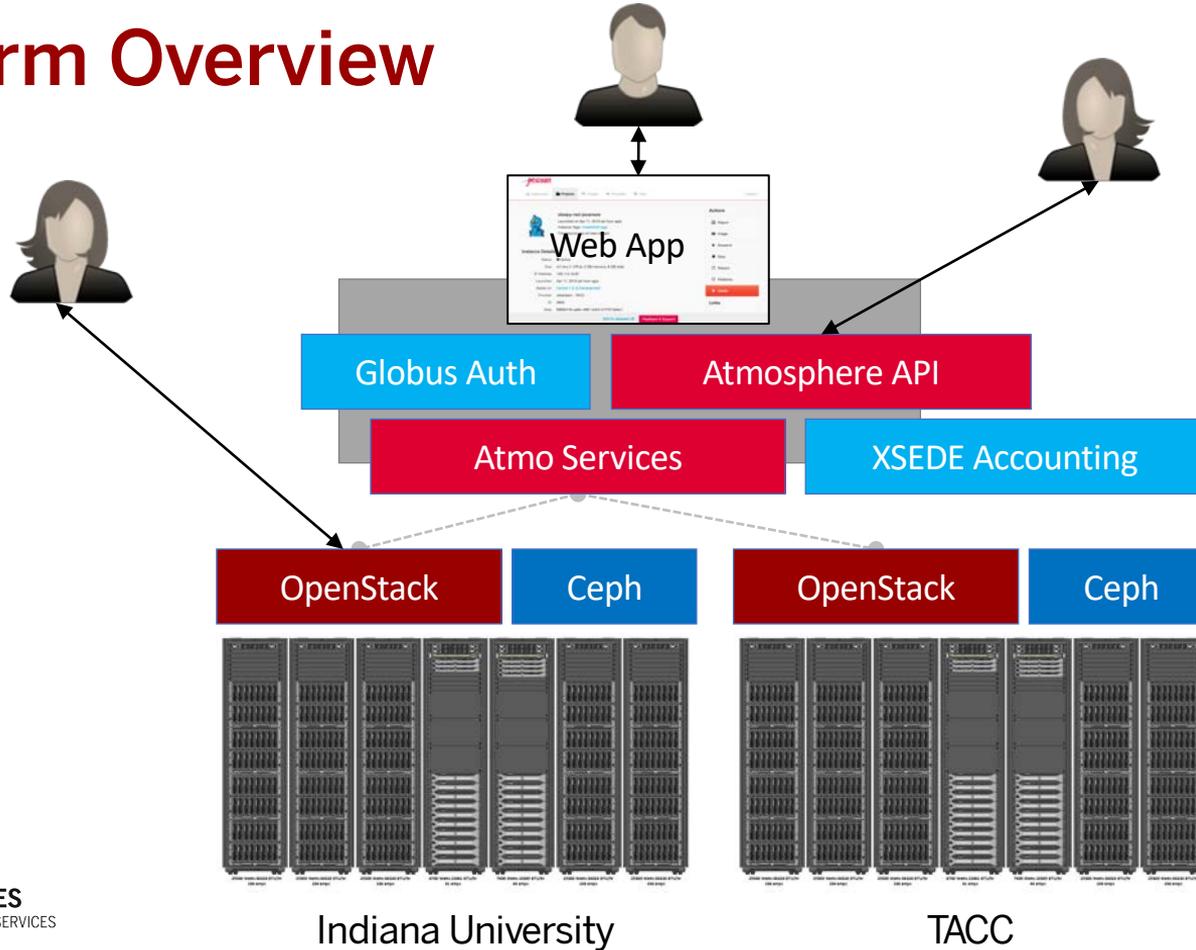
Jetstream System Overview



<http://wiki.jetstream-cloud.org/Network+configuration+and+policies>



Platform Overview



Cloud Computing Terms...simplified

Image: a file on a disk. It will be booted to create an...

Instance: a running virtual server; i.e. something you can log into.

State: something worth remembering; i.e. the state of the system



Cloud Computing Terms...simplified

Flavor: the size of a running instance; i.e. #core, RAM, disk

Hypervisor: the thing the instance runs on; something akin to a software defined hardware compute server.

Snapshot: the process of taking an instance and turning it to an image.



Cloud Computing Terms...simplified

States:

Running: the instance is up & running

Suspended: the instance is memory resident on the hypervisor but not running

Stopped: the instance is shutdown akin to powering down

Shelved: the instance is shutdown, written to disk, and stored

XSEDE/Jetstream specific information:

<http://wiki.jetstream-cloud.org/XSEDE+Service+Units+and+Jetstream>



Cloud Computing Terms...simplified (Cont.)

Object store: a blob of bits; it has a starting address & a size. There may be metadata associated with the object. The data is consumed in a streaming manner.

Block store: a software defined entity akin to an unformatted hardware disk drive, data is stored in blocks and has no metadata associated at the lowest levels

Filesystem: hierarchical in nature, directories & files, ability to open, seek, read, write.

Persistent storage: If you pull the plug, it will still exist when power is restored. Safe to store data or state here.

Ephemeral storage: If you pull the plug, it no longer exists. (Don't leave your data here!!!)



What is Jetstream – a closer look

- **Software layers**

- **Atmosphere** web interface
 - library of images, generic, domain specific
 - simplify VM administration
- **OpenStack**: software tools for building and managing cloud computing platforms for public and private clouds.
- **KVM** hypervisor: what the VMs run on
- **Ceph**: storage platform that stores data on a single distributed computer cluster, and provides interfaces for **object**-, **block**- and *file-level* storage.
- **Operating systems**: CentOS, Ubuntu, Windows(?)
- **Applications**; e.g. software developed by the domain specialist, gateways, etc.



API Access to Jetstream

- *What was unexpected*
 - **Demand** for **programmable cyberinfrastructure**
 - Great platform for learning **system administration skills**
 - Great platform for **teaching & learning cloudy technologies**
- *Command line clients*
- *Horizon dashboard very popular; but, incomplete*
- *Programmatic control; python is popular*
(<https://docs.openstack.org/openstacksdk/latest/>)
- *Slack channel for collaboration API users of Jetstream*



Using the OpenStack CLI on Jetstream

What an openrc file looks like:

```
export OS_AUTH_URL=https://iu.jetstream-cloud.org:35357/v3
export OS_PROJECT_NAME="TG-ABC190028"
export OS_USER_DOMAIN_NAME="tacc"
export OS_USERNAME="taccusername"
export OS_IDENTITY_API_VERSION=3
# export OS_PASSWORD='string'
read -sr OS_PASSWORD_INPUT
export OS_PASSWORD=$OS_PASSWORD_INPUT
```

- *Please do not publish the AUTH URLs anywhere*
- *CLI is python based – reads this information from the environment.*
- *Horizon can generate an openrc file for you (see the Wiki docs)*
- **Common pitfall** – *make sure you specify the correct Project (allocation) if you have more than one!*



Installing the client

- *Simple on most Mac OS X and Linux hosts (a single pip command)*
- *Less simple, but still do-able on Windows*
 - *Once you have a python installed, becomes a simple pip install*
- *Latest python-openstackclient (> 4.0.0) works with Python 3*
- *Best practice – use a virtual environment like venv for your install*
- *Docs on the wiki for this!*
- *Other CLI clients are available – e.g. python-swiftclient (Swift and S3), python-heatclient (Heat templates), etc*
 - *These are optional and not necessary for basic operations!*



Horizon GUI interface

- *Allows most things you can do from the CLI*
- *Nice for some tasks*
 - *Network visualizer is something we tend to use as a troubleshooting tool*
 - *Easier to look at security groups on Horizon (IMHO)*
- *Downsides:*
 - *considerably slower than using CLI*
 - *not all features are present that are in CLI*
 - *can't do things programmatically*



The screenshot shows the OpenStack Dashboard login page. At the top, there is the OpenStack logo (a red cube) and the text "openstack" in a sans-serif font, with "DASHBOARD" in a smaller font below it. Below the logo is the heading "Log In". There are three input fields: "Domain" with the value "TACC", "User Name" (empty), and "Password" (empty). Each input field has a small icon on the right side. At the bottom right of the form is a blue button labeled "Connect".



Getting started with the API

Things you'll set up once (hopefully):

- *SSH keys*
- *Security groups (though you'll build on the basics as you do more advanced things)*
- *Create a network*
- *Create a subnet*
- *Create a router*

Things you'll likely do many times:

- *Create and launch instances*
- *Screw up and delete instances*
- *Launch more instances*
- *Expand security groups*

API CLI Tutorial walkthrough: <https://github.com/jlf599/JetstreamAPITutorial>

API Horizon walkthrough: <http://wiki.jetstream-cloud.org/Using+the+OpenStack+Horizon+GUI+Interface>



API General Best Practices

- *Jetstream-specific – don't use Atmosphere images on the API side (start with JS-API-Featured-* images)*
- *Think about your security groups and only open what you REALLY need to open.*
- *Give objects unique and descriptive names*
- *When in doubt, use the universally unique identifier (UUID)*
- *When deleting items, use the universally unique identifier (UUID)*
- *Before deleting anything, though, “measure twice, cut once”*
- *Understand that an allocation/tenant lets you see everyone else's things. Be aware and be ware of deleting things – do unto others...*
- *Put your toys away if you're done with them*

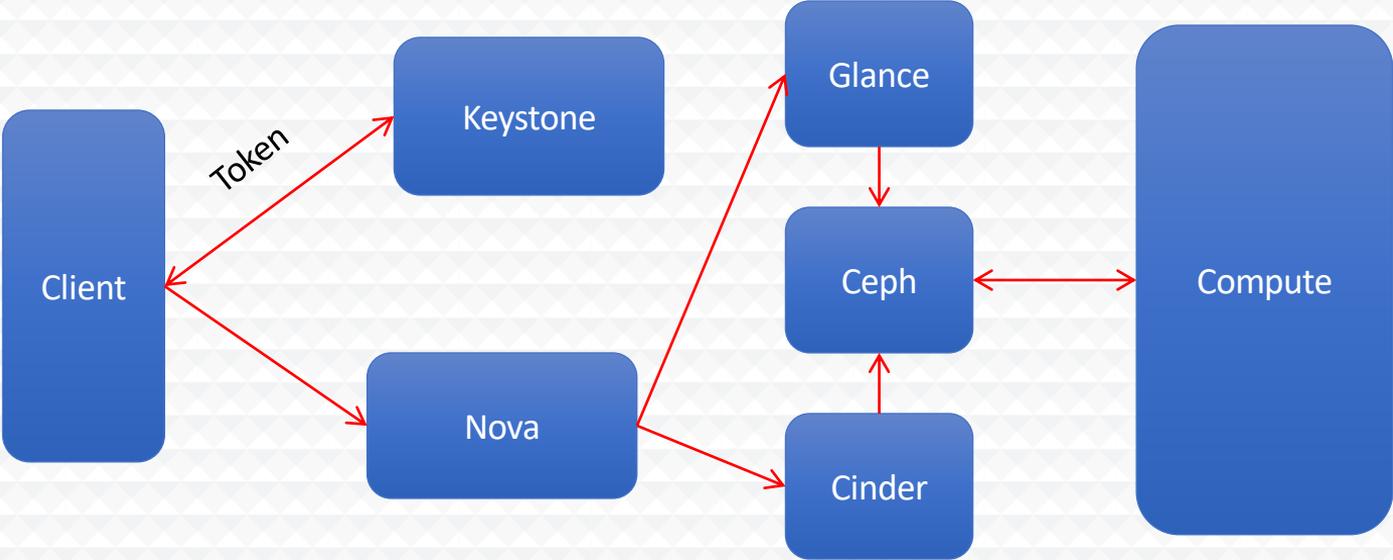


Security Best Practices

- *Think about your security groups and only open what you REALLY need to open. (yes, it's in the slides twice...on purpose...)*
- *In a production system, you'd likely want to also run a host-based firewall in addition to security groups (defense in depth!)*
- *Update often! Unattended security upgrades should be turned on in JS-API-Featured-* images...but still...*
- *Turn off any services/listeners you do not need*
- *For any service you run on a host, limit the access as much as possible – if it's world accessible, make sure permissions and privileges are as limited as possible*
- *Limit the number of people that interactively login – and create accounts for them instead of using shared accounts (e.g. centos or ubuntu account)*
- *Monitor the logs – lots of tools out there to help with this!*



OpenStack Overview



Jetstream usage highlights – 1 Feb 2020

- 385 XSEDE projects covering 77 fields of science and over 2400 active users representing 204 institutions
- **80%** of Jetstream users have **not used any other XSEDE system**
- >283M CPU hours allocated to XSEDE projects since June 2016
- 31 active science gateways
- 52 education/teaching allocations serving around 900 students
- 1189 mean active VMs in previous qtr, 1632 peak active VM count
- **Highest** user satisfaction in most recent XSEDE survey

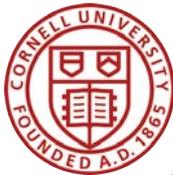


Where can I get help?

- *Wiki / Documentation:* <http://wiki.jetstream-cloud.org>
- *API CLI Tutorial:* <https://github.com/jlf599/JetstreamAPITutorial>
- *User guides:* <https://portal.xsede.org/user-guides>
- *XSEDE KB:* <https://portal.xsede.org/knowledge-base>
- *Email:* help@xsede.org



Jetstream Partners



funded by the National Science Foundation
Award #ACI-1445604



Questions?

- Project website: <http://jetstream-cloud.org/>
- Project email: help@jetstream-cloud.org Direct email: jeremy@iu.edu

License Terms

- Fischer, Jeremy. February 25, 2019. Jetstream Overview – CSCI B649 – School of Informatics – Indiana University. Also available at: <http://Jetstream-cloud.org/research/publications.php>
- Jetstream is supported by NSF award 1445604 (David Y. Hancock, IU, PI)
- XSEDE is supported by NSF award 1053575 (John Towns, UIUC, PI)
- This research was supported in part by the Indiana University Pervasive Technology Institute, which was established with the assistance of a major award from the Lilly Endowment, Inc. Opinions presented here are those of the author(s) and do not necessarily represent the views of the NSF, IUPTI, IU, or the Lilly Endowment, Inc.
- Items indicated with a © are under copyright and used here with permission. Such items may not be reused without permission from the holder of copyright except where license terms noted on a slide permit reuse.
- Except where otherwise noted, contents of this presentation are copyright 2015 by the Trustees of Indiana University.
- This document is released under the Creative Commons Attribution 3.0 Unported license (<http://creativecommons.org/licenses/by/3.0/>). This license includes the following terms: You are free to share – to copy, distribute and transmit the work and to remix – to adapt the work under the following conditions: attribution – you must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work). For any reuse or distribution, you must make clear to others the license terms of this work.

