

# Jetstream2: Accelerating cloud computing via Jetstream

David Y. Hancock  
Jeremy Fischer  
John Michael Lowe  
Winona Snapp-Childs  
dyhancoc@iu.edu  
jeremy@iu.edu  
jomlowe@iu.edu  
wsnappch@iu.edu  
Indiana University  
UITS Research Technologies  
Bloomington, IN, USA

Nirav Merchant  
Edwin Skidmore  
nirav@email.arizona.edu  
edwins@email.arizona.edu  
University of Arizona  
Tucson, AZ, USA

Marlon Pierce  
Suresh Marru  
J. Eric Coulter  
marpierc@iu.edu  
smarru@iu.edu  
jecoulte@iu.edu  
Indiana University  
Pervasive Technology Institute  
Bloomington, IN, USA

Matthew Vaughn  
Brian Beck  
vaughn@tacc.utexas.edu  
bbeck@tacc.utexas.edu  
University of Texas at Austin  
Texas Advanced Computing Center  
Austin, TX, USA

Gwen Jacobs  
gwenj@hawaii.edu  
University of Hawai'i  
Honolulu, HI, USA

## ABSTRACT

Jetstream2 will be a category I production cloud resource that is part of the National Science Foundation's Innovative HPC Program. The project's aim is to accelerate science and engineering by providing "on-demand" programmable infrastructure built around a core system at Indiana University and four regional sites. Jetstream2 is an evolution of the Jetstream platform, which functions primarily as an Infrastructure-as-a-Service cloud. The lessons learned in cloud architecture, distributed storage, and container orchestration have inspired changes in both hardware and software for Jetstream2. These lessons have wide implications as institutions converge HPC and cloud technology while building on prior work when deploying their own cloud environments. Jetstream2's next-generation hardware, robust open-source software, and enhanced virtualization will provide a significant platform to further cloud adoption within the US research and education communities.

## CCS CONCEPTS

• Computer systems organization → Cloud computing; • Human-centered computing → User interface management systems.

## KEYWORDS

cloud computing, containers, orchestration, computer architecture, computer interfaces



This work is licensed under a Creative Commons Attribution International 4.0 License.

## ACM Reference Format:

David Y. Hancock, Jeremy Fischer, John Michael Lowe, Winona Snapp-Childs, Marlon Pierce, Suresh Marru, J. Eric Coulter, Matthew Vaughn, Brian Beck, Nirav Merchant, Edwin Skidmore, and Gwen Jacobs. 2021. Jetstream2: Accelerating cloud computing via Jetstream. In *Practice and Experience in Advanced Research Computing (PEARC '21), July 18–22, 2021, Boston, MA, USA*. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3437359.3465565>

## 1 INTRODUCTION

Research and education (R&E) institutions across the world face a convergence in High Performance Computing (HPC) and cloud computing. Indiana University (IU) and its collaborators in creating the Jetstream cloud system, and users of this system, have demonstrated that one can adapt the latest methods in data science and artificial intelligence while still supporting traditional computational paradigms. In the midst of this convergence, a global pandemic emerged that has already impacted budgets and shifted expectations with respect to remote work and learning. This is the context in which we proposed and were awarded Jetstream2 by the National Science Foundation (NSF) in the United States.

Here we describe the motivations for Jetstream2, including the challenges and limitations faced in the implementation and operation of the original Jetstream system [20], the hardware and services architecture of the new environment, and open challenges. Much has been learned, and inherited, from the Jetstream environment, a system that has been at the frontier of internet computing as the NSF's first pilot and production cloud computing service for domain science, one positioned primarily as an Infrastructure-as-a-Service (IaaS) platform. The Jetstream2 project aims to expand services and the communities that benefit from those services.

The NSF Office of Advanced Cyberinfrastructure (OAC) outlines areas of HPC investment in advanced cyberinfrastructure (CI) resources and services in three broad areas. These areas include

leadership-class computing (BlueWaters and Frontera), innovative HPC/advanced systems and services (e.g., Stampede2, Jetstream, Expanse, Bridges-2), and support/CI services (e.g., XSEDE [21], XD-MoD). Jetstream2 is a Category I system, awarded as part of OAC's advanced computing systems and services solicitation. Jetstream2 was funded along with two other production and two experimental resources [13], but remains the only native cloud computing platform funded within the overall program. Jetstream2 will be available to the R&E community in 2021 through XSEDE.

The Jetstream2 project aims to provide new utility computing services and extend its predecessor's impact through an enhanced software environment (adding multi-cloud orchestration, push-button virtual clusters, and easy federation of JupyterHubs), and a new architecture that includes next-generation accelerators and processors, high-bandwidth and flexible networking, and tiered high-speed software-defined cloud storage. Importantly, we will provide this access in a user-friendly environment for training, workforce development, and education that enables *AI for Everyone*. The details are discussed in sections 4 and 5. Jetstream2's implementation is led by the IU Pervasive Technology Institute (PTI) [9], with a large group of institutional collaborators: Arizona State University (ASU), University of Arizona (UA), Cornell University, University of Hawai'i (UH), Johns Hopkins University (JHU), Texas Advanced Computing Center (TACC) at the University of Texas at Austin, and the University Corporation for Atmospheric Research (UCAR).

## 2 JETSTREAM LESSONS LEARNED

The challenges faced in the original Jetstream project have been technical, organizational, and social. The system was designed in 2014 with a simple architecture and homogeneous equipment across two primary sites that served as both a pilot and production cloud environment, a first-of-a-kind endeavor for IU and the NSF. Primary concerns were stabilizing an entirely open-source software ecosystem, and attracting non-traditional communities as well as traditional HPC projects. The project sacrificed storage capacity (to be relevant in an HPC program), heterogeneous memory sizes (to simplify allocations and instance sizes), and exclusion of both InfiniBand (to support live migration) and GPUs (an emerging technology at the time of the proposal). We also integrated the resource with a set of service providers through XSEDE who were almost exclusively targeting services and systems based on traditional batch-scheduled HPC clusters with shared parallel file systems. This resulted in social and technical challenges in outreach, training, and support, as well as technical challenges in integration, as services assumed gridFTP-based data transfer nodes (with a parallel file system back-end), SSH-based login nodes for single sign-on, and resource usage reporting of batch jobs that ran for a few days at most. The Jetstream team worked extensively with XSEDE and the integration resulted in significant changes and improvements in XSEDE allocation and support processes.

**Organizational and Technical Challenges.** In addition to those challenges integrating into the NSF CI, we faced the organizational and technical challenges of bringing together new partners for software and hardware integration. The goal was to provide native OpenStack API access and integrate the Atmosphere GUI front end

in order to lower the barrier for new communities utilizing IaaS clouds. API access to OpenStack was immediately in demand, but Atmosphere was designed to be the primary interface to the community as leveraged by the iPlant, and subsequently CyVerse, project [11]. In order to speed deployment of both API access and Atmosphere simultaneously, separate domains for managing instances and credentials were used. Federated authentication through Globus for Atmosphere was available, but one could not use those credentials for the OpenStack API to share projects or manage instances on the API side (a necessity for infrastructure services that desired persistent IPs). There were also organizational challenges in tightly integrating independent security infrastructure and policies across two production sites to provide uniform services and guidelines to researchers and educators (networking scanning services, network intrusion detection, firewall defaults). Additionally, organizational technical preferences led to site-specific configuration management and deployment tools with SaltStack, Puppet, and Ansible all being leveraged within the project.

Unexpected technical challenges also forced urgent system changes or inspired Jetstream2 design changes. As a first-of-kind system, and one of the largest known OpenStack deployments at the time, scaling issues were discovered. In configuring early services for telemetry data, we quickly overwhelmed our disk-based Ceph storage system due to the number of IOPS and frequency/type of data collection. NVMe devices were added to create a separate pool for these data to prevent starving user-space (volumes, images, and objects) of IOPS. The network design relied on propagating topology information over a RabbitMQ message bus that could not scale to accommodate the number of tenant virtual networks. Correcting the loss of network topology synchronization required regular reboots of OpenStack Neutron network nodes (one per compute rack at IU), which had to be spread over days to avoid exacerbating the problem. At TACC, the system was deployed without high-availability (HA) for many of the central OpenStack services, so nodes managing virtual networking and the message bus became frequent pain points for users and admins alike as networks rebuilt for hours (or days) after an outage, or RabbitMQ ceasing to respond caused operations to fail. These problems were not fatal to the project's overall success, and were within our targets for unplanned outages, but represent areas in which to improve.

**Administrative Challenges.** The simultaneous pilot and production nature of the Jetstream implementation also proved to be a challenge for NSF processes and review panels. While the following retrospective may seem pedantic, it is our intent that these experiences positively impact the community as well as the Jetstream2 project. Mapping the required operational metrics from the NSF HPC acquisition program to Jetstream led to requirements, through external review processes, that were not analogous to HPC systems. The most problematic was reporting percent CPU utilization. Traditional XSEDE HPC systems report use of available wall clock time, which is also how projects are charged (typically in core or node hours). Reporting CPU utilization for a resource versus wall clock time utilization for another resource can present a false sense of equivalence. Analysis of data from an internal IU HPC resource over a two-year period early in the project showed that average utilization measured via *CPU time* per month was often less than half

of the resource's average utilization measured via wall clock time. Such a metric can provide perverse incentives that avoid targeting use cases such as science gateway hosting or educational usage that may be more interactive and less CPU-intensive. We also found optimizing cloud storage performance within Jetstream through later software upgrades appreciably lowered CPU utilization. Focusing on achieving a significant percent CPU utilization for an on-demand, interactive cloud environment was not something the project could achieve and meet core objectives.

The nature of Jetstream also made it difficult to navigate acceptance processes. While extensive documentation [4] was provided as part of the Jetstream acceptance review, it was deemed necessary to perform MPI-based parallel file system benchmarks on a cloud-optimized storage system (used primarily for block storage for instances and an image repository). This resulted in a supplementary report [7] that diverted resources from providing users with a fully accepted and operational system earlier. These were lessons learned in the acceptance and operations of a unique environment for the NSF and it's clear both IU and the NSF have benefited from these shared experiences.

### 3 PROGRESSION TO JETSTREAM2

As a result of the challenges and learning experiences from Jetstream, IU was able to propose and receive NSF funding for Jetstream2, which is a significant expansion and improvement of Jetstream's capabilities. A new network topology with higher bandwidth and cloud native features was a design priority, as described in Section 4. Leveraging Cumulus Networks software provides multiple benefits, such as the ability to simulate configuration changes in a virtual framework without interrupting production operations and providing distributed NAT on the compute nodes, removing the need for many OpenStack Neutron nodes. Additionally, heterogeneous hardware was a core design point, not just because the community wanted GPUs and larger memory instances, but because providing those resources in novel ways is desirable. We were one of the first institutions to publish data on vGPU performance [12] and will expand on that capability by leveraging NVIDIA's multi-instance GPU (MIG) feature to offer GPUs to more individuals, particularly for education, where the vGPU slice can be sized appropriately. Researcher requests for streaming instrument and IoT data, as well as our own telemetry data experiences, also led us to provide a large pool of NVMe-based storage and an order of magnitude more disk-based storage than Jetstream.

Unifying the Atmosphere and OpenStack domains is a change in Jetstream2, allowing individuals to see the same instances and volumes regardless of their interface to the system. While federated authentication to a web front-end is still straight-forward through use of OAuth2 technologies, we aim to leverage application-specific credentials underneath those web services in a manner similar to commercial clouds. That single pane of glass management through Atmosphere will be key, allowing individuals to leverage their favorite third-party tool(s) such as Exosphere [16], in addition to a native interface like Openstack Horizon. Atmosphere is evolving into a new tool called Containerized Atmosphere for Continuous Analysis Orchestration (CACAO), which is built on the principles of Atmosphere (abstracting complicated functions such as firewalls

and virtual networking). This interface aims to improve the quality of an individual's experience using cloud computing while exposing cloud native, continuous analysis [3], continuous integration (CI), and continuous deployment (CD) processes and technologies. Underlying CACAO will be domain-specific languages (DSLs), including HashiCorp's Terraform [8] templates, and Kubernetes with a RESTful API and UI that will allow individuals new to cloud computing to coexist with those with advanced cloud-native applications, accommodating oft-requested features such as multi-instance launches, reserving floating IPs, along with fewer base images the project team must maintain.

Finally, multiple service providers now recognize the value of their own cloud environments or cloud-like features in their HPC systems, often in tandem with commercial cloud usage. By supporting tools such as Terraform from the project's outset, multi-cloud launches will be made more accessible to individuals. XSEDE has begun modernizing resource reporting interfaces, and changing data collection to go beyond batch-oriented resource management logs. These changes should lower barriers when integrating non-traditional resource types, a recognition of the ubiquity of utility and cloud computing where "jobs" of indefinite length are commonplace. The expansion to additional regional clouds will result in more consistency of configuration management and deployment while adopting lessons from XSEDE. The XSEDE Cyberinfrastructure and Resource Integration (XCR) team has worked with three other outside institutions (Dartmouth, Columbia, and George Mason University) on Openstack deployments, and is working with the Jetstream2 team to share lessons learned in helping local administrators through the process of building and providing new cloud resources to their campuses. XCR will continue to share those lessons from supporting system administrators at under-resourced institutions in providing computing resources [6].

**Regional clouds.** One of the unique design choices for Jetstream2 is the deployment of modest regional cloud systems (see Figure 1). They will serve multiple purposes within the project: implementing services regionally through a Core Services model, providing additional redundancy for science gateway environments, and serving as feedback mechanisms to other institutions interested in composable campus clouds. The concept of composable campus clouds is based on the concept of campus bridging outlined in the Taskforce of the NSF Advisory Committee for Cyberinfrastructure report [14]. The regional environments will provide data locality and integration with local resources that are necessary in data-intensive computation. Proximity to local instrumentation is often central to modest-scale research that has not generally occurred on XSEDE-supported environments. Incorporating these local resources will also allow further integration with other computational resources to demonstrate federation and dynamic workflows where there are related projects (e.g., Frontera and Stampede2 at TACC, and Aristotle at Cornell).

IU has provided technical leadership during the Jetstream project by testing and deploying services and upgrades first, consulting during software upgrades, and acting as a second tier of technical support for TACC and UA. Early architectural and site policies provided barriers to effectiveness at times when different network topologies or configuration management systems were utilized as

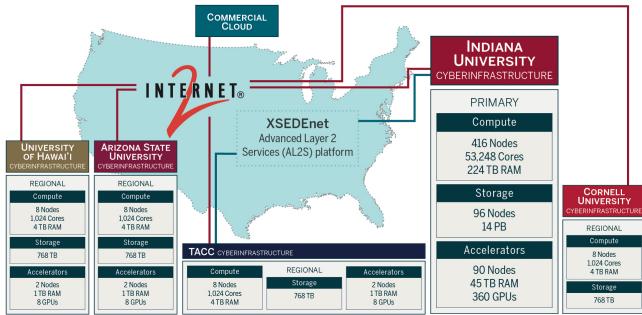


Figure 1: Schematic of connections to Internet2 and commercial cloud providers for Jetstream2.

previously described. Jetstream2 includes a revised approach to such partnerships through the inclusion of four regional clouds at ASU, Cornell, UH, and TACC with the common goal of creating an extensible, capable, and integrated national ecosystem. We will offer the implementation processes through a single configuration management and deployment process, centralized services for authentication/authorization, and technical leadership by supporting those Core Services and providing training on new technologies. This model is similar to the successful Core Services model that Monash University has championed in the federated Nectar Cloud project [1]. IU will provide quality assurance; testing software, management, and reporting scripts; and software upgrades to ensure they are ready for adoption by regional clouds and future partners. This is certainly an open challenge for the project but one partners are eager to undertake together.

## 4 HARDWARE ARCHITECTURE

The main production facility for Jetstream2 is being installed at IU (>90% of the system) with modest facilities housed at ASU, Cornell, UH, and TACC for availability and resilience (see Figure 1). This is a significant change from the distribution of production resources in Jetstream for two reasons: first, the original Jetstream system demonstrated that we can provide >99% uptime for an OpenStack-based cloud located within the IU Data Center, and second, there is significant storage overhead associated with a uniform distribution of resources (storage was a constraining factor for many large-scale research projects). In addition to the uses described previously for regional clouds, we believe such a geographically distributed architecture will encourage further education and training in implementing HA systems, which is not a topic in traditional curricula, but a challenge routinely faced in real world cloud deployment. The hardware architecture is based on best practices for cloud computing and the best available hardware at the time of purchase in five different node-types: CPU nodes (448) including 32 GPU-ready large memory nodes, GPU nodes (96), storage nodes (112), and management nodes (13). All compute node types are Dell PowerEdge servers with dual, next-generation AMD EPYC (64-core) processors, dual-port 100 Gbps-capable Ethernet adapters, and local solid state disks (SSDs). The storage nodes and management nodes will contain 128-256 GiB of RAM; CPU and GPU nodes will contain 512 GiB of RAM; and the large memory nodes 1 TiB

of RAM. The storage nodes, also Dell PowerEdge but with AMD EPYC (16-core) processors, will have dual SSDs and contain either (16) 6.4 TB non-volatile memory express (NVMe) devices or (12) 16 TB (or larger) SAS drives. The management nodes will include AMD EPYC processors (32-core), dual SSDs, and three 1.6 TB NVMe devices. Each of the GPU nodes will contain four NVIDIA A100 NVLink-capable accelerators (384 total) and the GPU-ready large memory nodes will have the capability for PCIe-based GPUs to be added later.

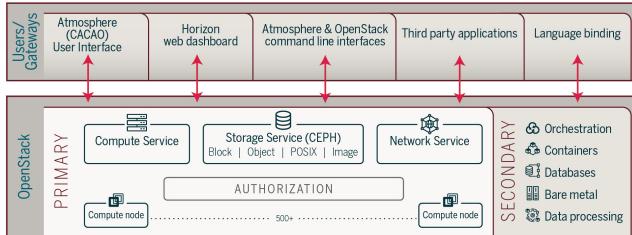
The high-bandwidth, low-latency Ethernet will enable diverse workloads without the additional complexity of requiring SR-IOV when using InfiniBand; live migration capabilities were not available in the libvirt Nova virt-driver when the system was designed. Mellanox ConnectX-6 Ethernet adapters will support RDMA over Converged Ethernet v2 (RoCE) allowing lower latency communications, virtualized network switching offload, overlay network offload, and live migration capabilities. The 100 Gbps spine and leaf network topology from Mellanox will utilize Spectrum-3 and Spectrum-1 switches, respectively, with Cumulus Networks software, which provides a highly-scalable layer 3 fabric utilizing Border Gateway Protocol (BGP) unnumbered. Layer 3 fabrics reduce broadcast traffic while equal cost multipathing (ECMP) provides even utilization of aggregate link members as compared to traditional Ethernet bonding methods like Multichassis Link Aggregation (MLAG) used in the prior system. This divergence from the Jetstream system will allow changes to be tested in a simulated environment through Cumulus VX instead of pushing untested configurations to the production system. In cloud systems, where a significant portion of the network traffic is encapsulated to emulate private layer 2 networks owned by the end users, the topology is relatively static and does not need to be updated with broadcast protocols like Address Resolution Protocol (ARP). Even balancing hides the implementation details presenting researchers with the full aggregate bandwidth for use without requiring multiple streams to utilize multiple links. The topology does not exceed 4:1 oversubscription; GPU and storage nodes will be provisioned at 1.5:1. Inter-node bandwidth will be 100 Gbps from two 50 Gbps Ethernet links for CPU nodes, a single 100 Gbps link for large memory nodes, and dual 100 Gbps links from GPU and storage nodes. Typical latency between adjacent nodes is expected to be less than 1.3  $\mu$ s while each additional hop adds 400 ns.

## 5 SERVICES ARCHITECTURE

A conceptualized service architecture diagram is shown in Figure 2. The essential, or primary, services are the components that are core to the system, and that other advanced (secondary) services rely on. The designations do not represent importance to the users but the relationship between services.

### 5.1 Essential (Primary) Services

Jetstream2 relies on two open source projects, OpenStack and Ceph, to provide foundational compute, network, identity, and storage services as shown in the primary area of Figure 2. Advanced services and additional interfaces combine for the full set of Jetstream2 capabilities as discussed in 5.2 and 5.3.



**Figure 2: Conceptual and architectural diagram of the Jetstream2 software environment.**

Ceph is a replicated object storage system, reliable autonomic distributed object store (RADOS) providing all data storage for Jetstream2. Ceph is one of the few software-defined storage options that provides block, object, and file system access from the same pool of storage and has been used within Jetstream and previously in the XSEDE Gateway Hosting environment operated by IU. Objects are distributed deterministically over the entire set of storage servers and data/metadata are stored with checksums ensuring data integrity. IU and TACC have experienced a total of five major unplanned power outages of Jetstream (one and four per site, respectively); copy-on-write semantics coupled with the checksums allowed for the online detection and correction of any corruption caused by these power outages. Storage functionality was resumed within minutes of power restoration without the need for any offline repairs. If data integrity of a replica is compromised, or an object rendered unavailable, data are replicated in a self-healing manner with no interruption to data access, maintaining uptime and facilitating system maintenance operations. For example, a batch of disks delivered with the TACC system contained a fatal firmware bug drastically shortening the life of the disk. Failure rates upwards of 60% were observed without data loss due to the self-healing replication properties of Ceph.

RADOS Block Devices (RBD) are a collection of objects, accessed with offsets into the object, that are used to emulate block devices. The Ceph File System (CephFS) is used for POSIX semantics, storing files and metadata as objects. CephFS was not a stable feature when Jetstream was implemented but will be used in Jetstream2 with services such as OpenStack Manilla to ease data sharing among instances. A RESTful API, provided by Ceph's RADOS Gateway (RGW), emulates the popular AWS S3 interface, as well as OpenStack Swift. Object storage has not been a high-demand feature on Jetstream, but has been noted as an important feature in responses to our recent annual survey [10]. We expect more projects will recognize the utility in the future.

OpenStack originated as a collaboration between NASA and Rackspace, Inc. in 2010, combining their efforts to create a platform for the delivery of virtualized computing services that consists of distinct but inter-operating projects. Identity services are provided by OpenStack Keystone. Keystone issues and validates limited life tokens with role assignments on projects to users after having authenticated them. These tokens are consumed by other OpenStack services delegating identity management to Keystone as part of a microservice architecture. Application credentials, a subordinate

set of credentials entailed to a particular role assignment, are utilized by Exosphere, CACAO, and OpenStack CLI tools to assume the identity of the user without having to store or obtain the user's actual credentials. Life-cycle functions of virtual machines, networks, images, and block storage are provided by the OpenStack Nova, Neutron, Glance, and Cinder services respectively. This collection of services is the minimum necessary to operate a virtualized environment for computation.

## 5.2 Advanced (Secondary) Services

The advanced services within Jetstream2 build on the services described in Section 5.1. Jetstream2 incorporates many of these services for individuals to leverage directly or through other interfaces and tools described in Section 5.3. Documentation for using these flexible services and expanding into custom use cases are key to providing an ecosystem that appeals to many segments of the R&E community.

Linux containers are not a new functionality in research computing. Over the last few years, there has been growth in using local container technologies like Docker and technologies intended to allow containers on shared HPC systems like Singularity. While Jetstream2 will support direct usage of those tools, it will also support managing and scaling container-based workloads via the cloud-native functionality of OpenStack Magnum as has been demonstrated with Jetstream. Users will be able to deploy Docker Swarm, Apache Mesos, or Kubernetes container orchestration engines to manage and run their container-based research workloads. In addition, the features of CACAO, a follow-on to Atmosphere developed by the University of Arizona, will provide similar functionality to individuals who have no desire to access the OpenStack API directly. Both approaches will allow researchers and educators to scale their workloads dynamically according to their needs.

OpenStack Heat is a service that allows individuals to instantiate complex resources with dependencies via a declarative YAML-based language. Similar to Magnum, other OpenStack services such as Trove and Sahara also leverage Heat to provide relational and non-relational databases and to provision data-intensive application clusters. These capabilities build on one of the fundamental aspects of cloud computing that was demonstrated in abundance with Jetstream: the ability of users to create, manage, and orchestrate use of tools autonomously, based on need, without involving sysadmins to install or enable new software. OpenStack Ironic provides the ability to provision instances running on bare metal as one provisions VMs running on hypervisors. Some features of virtualization, such as live migration and high-availability, afforded by redundant virtual infrastructure, are not available to bare metal instances. For needs that are not well-served by virtual infrastructure, bare metal instances can be provided on Jetstream2, something not previously supported.

## 5.3 Community and Science Gateway Interfaces

As with most cloud platforms, there are multiple interfaces available to individuals, each with different levels of sophistication, complexity, and ease-of-use. The top of Figure 2 shows some of those interfaces described below.

**User Interface.** As the primary entry-level web interface for Jetstream2, CACAO (formerly Atmosphere) focuses on making a set of common operations more usable. Specifically, life cycle management of resources in a multi-cloud environment, including metering resources, sharing, and collaboration. CACAO provides a simple point-and-click interface, hiding the more technically demanding operations, like networking and security groups, while deferring access to more sophisticated OpenStack services to other native interfaces within Jetstream2. CACAO allows individuals to obtain secure shell and desktop access to their cloud instances through an integrated web interface built from Apache Guacamole. As their computational and data movement requirements increase in complexity, they can access CACAO’s multi-cloud functions through a RESTful API and a command line interface. Using CACAO, one can also orchestrate and configure instances into virtual clusters by launching multiple instances simultaneously and defining custom scripts as they launch.

CACAO further enhances the utility within instances by pre-installing “ez” command-line scripts that allow individuals to deploy common software stacks, like JupyterLab (“ezj”) and Singularity (“ezs”). These scripts are small, locally-deployed Ansible codes that may be a single tool or a set of software executed as a workflow. Jetstream2 will further expose the “ez” facility through the CACAO web interface while providing one-click access to AI, ML, DL, and GPU optimized software, such as NVIDIA GPU cloud and PySpark. Coupled with vGPUs, which should make GPU access less scarce, it supports Jetstream2’s *AI for Everyone* strategy.

CACAO on Jetstream2 will simplify the access, utility, and management of containers and container orchestration software within a multi-cloud environment. It can already dynamically deploy single-host Docker or Singularity software stacks. Using DSLs and template-based provisioning frameworks CACAO will also enable individuals to provision and configure multiple instances into Kubernetes clusters as well as provide an interface to existing APIs to manage their Kubernetes clusters. These features provide enhanced usability over the base functions in OpenStack Magnum. A feature that will be unique to CACAO will be its ability to update a user’s infrastructure automatically from a Git repository whenever the [Terraform] template changes.

**Native Interfaces.** OpenStack provides multiple interfaces to access and manage virtual machines and accompanying infrastructure: the Horizon graphical UI, a Python command-line client, and a Python software development kit. Using native OpenStack interfaces allows more granular control over the available resources and allows individuals to use advanced features of the Jetstream2 cloud either interactively or programmatically. Tools such as Horizon are not well-suited for large OpenStack clouds as they are not as responsive to common operations where many projects, networks, and instances exist. This has been a limitation of Jetstream as individuals would often leverage Horizon when operations could not be performed within the legacy Atmosphere UI, a merging of the OpenStack and Atmosphere domains, and an increased availability of third-party tools, such as Exosphere, should provide sufficient alternatives for those cases.

**Programmable Infrastructure and Virtual Clusters.** Programmable cyberinfrastructure is a core tenet of both Jetstream projects and implies that the infrastructure can change programmatically, in addition to having diverse interfaces and tools available. A desire to develop and version infrastructure, managing infrastructure as code, has spread from DevOps to researchers and educators. Jetstream2 capabilities will better meet the desire to deploy resources on a variety of platforms, including not only Jetstream2, but commercial cloud services, enabled by tools such as Terraform and already piloted by Jetstream users. Orchestration and use of containers will allow individuals to have a more consistent environment across platforms and facilitate transitions as desired.

In addition to the ability for individuals to control their infrastructure programmatically, Jetstream2 will provide the capability to spin up elastic HPC virtual clusters (VCs) at the push of a button. These have been tested extensively on Jetstream, with about thirty VCs running in production at different times [5]. These Slurm-powered virtual clusters allow individuals to transition easily between cloud and HPC resources, acting as both a test-bed environment for custom software, and a highly-available production resource for projects with modest computational needs. The deployment process for these resources in Jetstream2 will be streamlined, allowing individuals to deploy an instance, acting as a head node, that is ready to accept jobs. Once jobs are submitted, worker instances will be automatically created and destroyed as needed. The Singularity container runtime environment will be built into these VCs, allowing individuals to use containerized scientific software without lengthy installation processes.

**Science Gateways and Scientific Notebooks.** Supporting science gateways is a key use case for Jetstream2 as they are critical infrastructure for democratizing access to scientific software and data, and for supporting reproducible scientific research. The current Jetstream system supports over fifty science gateways, with nine that use virtual clusters. Jetstream2 will provide flexible, reliable IaaS that will support science gateway web servers and middleware. As discussed above, virtual clusters and native container support on Jetstream2 will allow science gateways to provide scheduled and managed computing resources under their control.

In addition, Jetstream2 will support improved science gateway software engineering practices by providing continuous integration and deployment services using software like Apache Jenkins to help build, test, and deploy scientific applications to gateways on a constant basis as they become available. By expanding existing partnerships with the Science Gateways Community Institute [22] and XSEDE, we will help gateway providers identify opportunities to use such services and improve the service(s) they provide.

Many gateways rely on Jupyter Notebooks or JupyterLabs, which have grown rapidly in popularity as interactive scientific computing tools; JupyterHubs provide similar functionality to groups of users. Execution of these environments is best suited for on-demand systems. These may be coupled to container technologies to provide elastic resources. While Jetstream2’s capabilities can provide the needed IaaS, the third-party operators of JupyterHub instances need additional services for authenticating and authorizing their communities. To address this need, Jetstream2 will offer expanded

support of JupyterLabs in general, specifically supporting JupyterHub deployment based on well-tested workflows [23]. This will be further enhanced by addressing authentication and access control based on the NSF-funded Custos service [18] operated by IU PTI's Cyberinfrastructure Integration Research Center and collaborators in the Galaxy science gateway and CILogon [2] federated identity projects. Instead of just providing access to a Notebook with a one-time login URL as the "ezj" tool provides, individuals will be able to use an existing identity provider, particularly useful for courses and workshops. JupyterHub instances deployed on Jetstream2 will also have the ability to use the Custos service to manage third-party OAuth2-based authorization and access credentials of commercial clouds. User-deployed services on Jetstream2, including such JupyterHubs, will be able to use this credential management capability to federate data access and computations to accessible cloud resources. Finally, Custos integration will enable those JupyterHub instances to support collaboration using group-based authorization. Individuals will be able to share Notebooks under their control.

## 6 BROADER ECOSYSTEM AND OPEN CHALLENGES

Jetstream2 is situated in a broader suite of systems and services funded by the NSF, and is thus useful to identify the direct links with other projects. In addition to leveraging work from the CyVerse project (via Co-PIs Merchant and Vaughn), there are strong ties to other NSF projects that include the Science Gateways Community Institute (via Co-PI Pierce and Senior Personnel Marru), the direct participation of the Galaxy and Unidata [17] science gateway teams (via JHU and UCAR), and links to the Aristotle project (via Cornell). The project also aims to leverage efforts by the CloudBank project to ease access to commercial cloud computing resources while providing a native cloud environment to transition to and from.

A system such as Jetstream2 would not be innovative if there were no open challenges within the project; we will expand here on challenges not previously detailed. Federated authentication and authorization across user-provisioned services is a key challenge, as discussed in the prior section. Development of a standard deployment and adoption of a Core Services model for regional clouds is another challenge. Inter-operating with multiple commercial cloud providers in meaningful ways for science gateways and other advanced users is a challenge. Deploying GPUs within the platform comes with potential limitations in live migration capabilities that will need to be solved to leverage GPUs effectively.

There are also new open challenges in R&E support, training, and diversity. The inclusion of GPUs in Jetstream2 introduces an additional complexity for system-level options and software images. Jetstream has a small number of instance types due to its homogeneity; this is not the case for Jetstream2. There are an order of magnitude more community-contributed images on Jetstream than featured images maintained by the project; heterogeneous hardware is expected to increase those contributions. Furthermore, introducing vGPUs adds another dimension of complexity. In [12], the authors show that running DL benchmark workloads on half of an NVIDIA V100 GPU has little to modest impact on performance. If GPU memory, and not memory bandwidth, is the limiting factor, up to seven vGPUs can be created from a single NVIDIA A100

GPU using MIG. By surveying GPU workloads being executed, we will provide the most useful instance sizes so as not to overwhelm individuals with options.

While mentioned earlier as a great benefit, there are potentially problematic consequences with the amount of control users have over their own operating environments (ability to gain full root access), which can complicate support and training. Normal users on an HPC cluster do not have root credentials, whereas they do on Jetstream, and will on Jetstream2. This has not been a fatal flaw and provides usage modes not possible on other systems. Proactive network scanning and intrusion detection systems have detected 72 vulnerable instances per year on average, with only 14 per year being reported after a suspected compromise. Of these compromised instances, the vast majority (i.e. 13 per year) have been instances hijacked for cryptocoin mining operations and only 1 instance per year engaged in offensive (i.e. network scanning) operations.

The Jetstream project set goals to increase the diversity of domains and non-traditional researchers and educators that leveraged the platform with demonstrated success. This success is illustrated through analysis of XDMoD report data [15, 19]. These data reveal that, although Jetstream represented less than 10% of the NSF investment in HPC systems in 2017, the environment served 24% of institutions, 23% of active Principal Investigators, and 32% of individuals that used CI resources supported by XSEDE. Importantly, the project has been successful at filling a gap in the CI ecosystem: 80.4% of individuals using Jetstream have never run a job on any other XSEDE-supported system (as of November 2019). The impact extends to educational usage; more than six times the service units were allocated on Jetstream than the nearest XSEDE resource in calendar year 2019 in that category.

Qualitative analysis also underscores the importance of such resources. During the most recent annual user survey [10], the number of individuals who rate Jetstream "very important" or "essential" to their educational activities is on the rise (78%, up from 73% in 2019) as well as for their research activities (84%, up from 74% in 2019). The project has not been as successful in diversity of individuals in other dimensions, although we have not compared these data to other Service Providers. The most typical Jetstream user is male, Caucasian, and of non-Hispanic and non-Latino ethnicity (70% male, 49% Caucasian, 80% non-Hispanic and non-Latino), working at a doctoral-granting and/or research institution. Jetstream2 needs to have an increased focus on outreach to minority-serving institutions and to under-served groups through Diversity, Equity, and Inclusion events at relevant conferences, continued research experiences for undergraduates, and increasing the profile of women on the project team and in communities supported by the project.

## 7 DISCUSSION AND CONCLUSION

Jetstream2 will continue the most utilized and high-impact features of its predecessor while increasing the hardware capability and performance. The platform will be built using layers of open source software and designed to meet a variety of complex use cases from reliable IaaS to transparent "serverless" functions. The shift toward hardware heterogeneity with increased storage capability and performance should serve to expand the community of individuals using the environment while bringing new challenges with it. We

plan to have at least a three month overlap for migration to the new system and will permit projects to supplement and/or transfer allocations. For science gateways and other projects leveraging OpenStack APIs the transition should be straight-forward (create an instance snapshot, convert to an image, import image on the new system). Expert consulting will be available to help with any issues encountered during the migration.

The vision for the Jetstream2 project is to achieve societal benefits from the research it enables while also contributing as a platform to increase workforce diversity (both in terms of individuals and projects). The desire is for Jetstream2 to serve more students than any other CI resource funded by the NSF while simultaneously enabling new research, discovery, and innovation across many disciplines. Achieving these goals will only be possible by leveraging other NSF investments through partners, each with their own broader impact plans and expertise.

The project has already begun to encounter challenges during the implementation phase and operational challenges are sure to arise. Some of those challenges will be brought about through the inclusion of new software and features (e.g., virtual GPUs, commercial cloud integration, heterogeneous equipment) and some through support and training of new researcher and educators leveraging such features (e.g., increased object storage usage, promotion of multi-cloud orchestration, new advanced interfaces). Ultimately, our goal is to provide a resource to the community that has made Jetstream a success, in 2021, and also increase the utility of Jetstream2's programmable cyberinfrastructure capabilities over the project's lifetime.

## ACKNOWLEDGMENTS

The IU Pervasive Technology Institute was established with funding from the Lilly Endowment, Inc. This material is based upon work supported by the National Science Foundation under Grant No. 1445604 and 2005506. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF. We thank the following individuals for their contributions as well as their tireless support for the R&E community: C.A. Stewart, H. Jankowski, S. Michael, S. Sudarshan, G. Turner, T. Miller, S. Bird, J. Wernert, L.M. Weakley, and S. Cleveland.

## REFERENCES

- [1] Australian Research Data Commons. 2019. *The team behind the software powering our Nectar Research Cloud*. Retrieved March 6, 2021 from <https://ardc.edu.au/news/the-team-behind-the-software-powering-our-nectar-research-cloud/>
- [2] Jim Basney, Terry Fleury, and Jeff Gaynor. 2014. CILogon: A federated X. 509 certification authority for cyberinfrastructure logon. *Concurrency and Computation: Practice and Experience* 26, 13 (2014), 2225–2239.
- [3] Brett K Beaulieu-Jones and Casey S Greene. 2017. Reproducibility of computational workflows is automated using continuous analysis. *Nat. Biotechnol.* 35, 4 (Apr 2017), 342–346. <https://doi.org/10.1038/nbt.3780>
- [4] C.A. Stewart, D.Y. Hancock, M. Vaughn, N.C. Merchant, J.M. Lowe, J. Fischer, L. Liming, J. Taylor, E. Afgan, G. Turner, C.B. Hammond, E. Skidmore, M. Packard, and I. Foster. 2016. *System Acceptance Report for NSF award 1445604 "High Performance Computing System Acquisition: Jetstream - A Self-Provisioned, Scalable Science and Engineering Cloud Environment"*. Technical Report. Indiana University. <http://hdl.handle.net/2022/20908>
- [5] J. Eric Coulter, Eromaa Abeyasinghe, Sudhakar Pamidighantam, and Marlon Pierce. 2019. Virtual Clusters in the Jetstream Cloud: A Story of Elasticized HPC. In *Proceedings of the Humans in the Loop: Enabling and Facilitating Research on Cloud Computing* (Chicago, IL, USA) (HARC '19). Association for Computing Machinery, New York, NY, USA, Article 8, 6 pages. <https://doi.org/10.1145/3355738.3355752>
- [6] J. Eric Coulter, Rich Knepper, Resa Reynolds, Jodie Sprouse, and Stephen Bird. 2019. Beyond Campus Bridging: A Retrospective of Cyberinfrastructure Integration Efforts. In *Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (Learning)* (Chicago, IL, USA) (PEARC '19). Association for Computing Machinery, New York, NY, USA, Article 93, 4 pages. <https://doi.org/10.1145/3332186.3333151>
- [7] D.Y. Hancock, M. Packard, G. Turner, and C.A. Stewart. 2016. *System Acceptance Report for NSF award 1445604 "High Performance Computing System Acquisition: Jetstream - A Self-Provisioned, Scalable Science and Engineering Cloud Environment"*. Technical Report. Indiana University. <http://hdl.handle.net/2022/20958>
- [8] HashiCorp Inc. 2020. *Terraform Infrastructure Management Software*. HashiCorp, Inc. Retrieved August 19, 2020 from <https://www.terraform.io/>
- [9] Indiana University Pervasive Technology Institute. [n.d.]. *IU PTI home page*. Retrieved March 6, 2021 from <https://pti.iu.edu/>
- [10] Harmony Jankowski, Julie Wernert, David Y Hancock, and Winona Snapp-Childs. 2020. *Jetstream Annual User Survey - 2020 Summary Report*. Technical Report. Indiana University Pervasive Technology Institute. <https://scholarworks.iu.edu/dspace/handle/2022/25954>
- [11] Nirav Merchant, Eric Lyons, Stephen Goff, Matthew Vaughn, Doreen Ware, David Micklos, and Parker Antin. 2016. The iPlant Collaborative: Cyberinfrastructure for Enabling Data to Discovery for the Life Sciences. *PLoS Biol.* 14, 1 (Jan. 2016), e1002342.
- [12] S. Michael, S. Teige, J. Li, J. Lowe, G. Turner, and R. Henschel. 2020. Performance Characteristics of Virtualized GPUs for Deep Learning. In *2020 IEEE/ACM International Workshop on Interoperability of Supercomputing and Cloud Technologies (SuperCompCloud)*. IEEE Computer Society, Los Alamitos, CA, USA, 14–20. <https://doi.org/10.1109/SuperCompCloud51944.2020.00008>
- [13] National Science Foundation. 2020. *Announcement: Preparing for the future today: Expanding U.S. researchers' access to advanced computing systems*. Retrieved March 6, 2021 from [https://www.nsf.gov/news/special\\_reports/announcements/062220.jsp](https://www.nsf.gov/news/special_reports/announcements/062220.jsp)
- [14] NSF. 2011. *NSF Advisory Committee for Cyberinfrastructure Task Force on Campus Bridging. Final Report*. Technical Report. National Science Foundation. [https://www.nsf.gov/cise/oac/taskforces/TaskForceReport\\_CampusBridging.pdf](https://www.nsf.gov/cise/oac/taskforces/TaskForceReport_CampusBridging.pdf)
- [15] University of Buffalo. [n.d.]. *XD Metrics on Demand*. Retrieved March 6, 2021 from <https://xdmod.ccr.buffalo.edu/>
- [16] J. Pistorius, C. Martin, S. Sudarshan, and D.S. LeBauer. 2020. Exosphere - Bringing The Cloud Closer. In *2020 IEEE/ACM International Workshop on Interoperability of Supercomputing and Cloud Technologies (SuperCompCloud)*. IEEE Computer Society, Los Alamitos, CA, USA, 1–6. <https://doi.org/10.1109/SuperCompCloud51944.2020.00006>
- [17] M Ramamurthy. 2020. Unidata Science Gateway for Enabling Science as a Service to Facilitate Open Science and Reproducible Research. *IOP Conference Series: Earth and Environmental Science* 509 (jul 2020), 012043. <https://doi.org/10.1088/1755-1315/509/1/012043>
- [18] Isuru Ranawaka, Suresh Marru, Juleen Graham, Aarushi Bisht, Jim Basney, Terry Fleury, Jeff Gaynor, Dimuthu Wannipurage, Marcus Christie, Alexandru Mahmoud, Enis Afgan, and Marlon Pierce. 2020. Custos: Security Middleware for Science Gateways. In *Practice and Experience in Advanced Research Computing* (Portland, OR, USA) (PEARC '20). Association for Computing Machinery, New York, NY, USA, 278–284. <https://doi.org/10.1145/3311790.3396635>
- [19] Nikolay A. Simakov, Joseph P. White, Robert L. DeLeon, Steven M. Gallo, Matthew D. Jones, Jeffrey T. Palmer, Benjamin D. Plessinger, and Thomas R. Furlani. 2018. A Workload Analysis of NSF's Innovative HPC Resources Using XDMoD. *CoRR* abs/1801.04306 (2018). arXiv:1801.04306 <http://arxiv.org/abs/1801.04306>
- [20] Craig A. Stewart, Timothy M. Cockerill, Ian Foster, David Hancock, Nirav Merchant, Edwin Skidmore, Daniel Stanzione, James Taylor, Steven Tuecke, George Turner, Matthew Vaughn, and Niall I. Gaffney. 2015. Jetstream: A Self-Provisioned, Scalable Science and Engineering Cloud Environment. In *Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure* (St. Louis, Missouri) (XSEDE '15). Association for Computing Machinery, New York, NY, USA, Article 29, 8 pages. <https://doi.org/10.1145/2792745.2792774>
- [21] J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G. D. Peterson, R. Roskies, J. R. Scott, and N. Wilkins-Diehr. 2014. XSEDE: Accelerating Scientific Discovery. *Computing in Science & Engineering* 16, 5 (2014), 62–74. <https://doi.org/10.1109/MCSE.2014.80>
- [22] Nancy Wilkins-Diehr, Michael Zentner, Marlon Pierce, Maytal Dahan, Katherine Lawrence, Linda Hayden, and Nayiri Mullinx. 2018. The Science Gateways Community Institute at Two Years. In *Proceedings of the Practice and Experience on Advanced Research Computing* (Pittsburgh, PA, USA) (PEARC '18). Association for Computing Machinery, New York, NY, USA, Article 53, 8 pages. <https://doi.org/10.1145/3219104.3219142>
- [23] Andrea Zonca and Robert S. Sankovits. 2018. Deploying Jupyter Notebooks at Scale on XSEDE Resources for Science Gateways and Workshops. In *Proceedings of the Practice and Experience on Advanced Research Computing* (Pittsburgh, PA, USA) (PEARC '18). Association for Computing Machinery, New York, NY, USA, Article 8, 7 pages. <https://doi.org/10.1145/3219104.3219122>