

Scaling JupyterHub Using Kubernetes on Jetstream Cloud: Platform as a Service for Research and Educational Initiatives in the Atmospheric Sciences

Semir Sarajlic
Partnership for an Advanced
Computing Environment (PACE)
Georgia Institute of Technology
Atlanta, Georgia
semir.sarajlic@oit.gatech.edu

Julien Chastang
Unidata, University Corporation for
Atmospheric Sciences (UCAR)
Boulder, Colorado
chastang@ucar.edu

Suresh Marru
Pervasive Technology Institute (PTI)
Indiana University
Bloomington, Indiana
smarru@iu.edu

Jeremy Fischer
Pervasive Technology Institute (PTI)
Indiana University
Bloomington, Indiana
jeremy@iu.edu

Mike Lowe
Pervasive Technology Institute (PTI)
Indiana University
Bloomington, Indiana
jomlowe@iu.edu

ABSTRACT

Unidata, an NSF funded project that started in 1983, is a diverse community of education and research institutions with the common goal of sharing geoscience data and the tools to access and visualize that data. Unidata provides weather observations and other data, software tools, and support to enhance Earth-system education and research, and continuously examines ways of adapting their workflows for new technologies to maximize the reach of their education and research efforts.

In support of Unidata objectives to host workshops for atmospheric data analysis using JupyterHub, we explore a cloud computing approach leveraging Kubernetes coupled with JupyterHub that when combined will provide a solution for researchers and students to pull data from Unidata and burst onto Jetstream cloud by requesting resources dynamically via easy to use JupyterHub. More specifically, on Jetstream, Kubernetes is used for automating deployment and scaling of domain specific containerized applications, and JupyterHub is used for spawning multiple hubs within the same Kubernetes cluster instance that will be used for supporting classroom settings. JupyterHub's modular kernel feature will support dynamic needs of classroom application requirements. The proposed approach will serve as an end-to-end solution for researchers to execute their workflows, with

JupyterHub serving as a powerful tool for user training and next-generation workforce development in atmospheric sciences.

CCS CONCEPTS

• **Applied Computing** → **Interactive learning environments**; • **Computer systems organization** → Architectures;

KEYWORDS

JupyterHub, Kubernetes, Magnum, OpenStack, Unidata, Cloud Computing, Workforce Development

ACM Reference format:

S. Sarajlic, J. Chastang, S. Marru, J. Fischer, M. Lowe. 2018. Scaling JupyterHub Using Kubernetes on Jetstream Cloud: Platform as a Service for Research and Educational Initiatives in the Atmospheric Sciences. *In Proceedings of the Practice and Experience in Advanced Research Computing 2017, Pittsburgh, Pennsylvania USA, July 2018*, 4 pages. <https://doi.org/10.1145/3219104.3229249>

1 INTRODUCTION

Data proliferation across all domains is enabled by the advancement in data sharing and procurement that has led to an explosion in the volume, velocity, veracity, and variety of data [1]. The exponential growth in data has provided opportunities for machine learning algorithms and computational algorithms that benefit from greater availability of data. Hence, rapid growth in the importance of data science is making researchers dependent on advanced research computing infrastructure, which presents both opportunities and challenges.

Unidata's long experience in helping geoscientists incorporate data-centric techniques into their scientific workflows positions Unidata to help the geoscience community take advantage of a

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author.
PEARC '18, July 22–26, 2018, Pittsburgh, PA, USA
© 2018 Copyright is held by the owner/author(s).
ACM ISBN 978-1-4503-6446-1/18/07.
<https://doi.org/10.1145/3219104.3229249>

rapidly dynamic scientific landscape [2]. Part of Unidata’s mission is to reduce “data friction”, lower the barriers for accessing and using data, and “shrink the time to science” [3]. This mission is motivated by the data proliferation in which the data volumes are getting too large to bring all of the data to scientists’ local environment; hence, the move from “bringing the data to a scientist” to “bringing the science to the data” [3]. More specifically, scientists and students may access the data and data services from data centers (e.g., cloud environments), which are leading towards an end-to-end solution that provides an ecosystem of integrated tools that scientist may use to analyze the data [4 - 5]. With growing expectations from the community for interactive and intuitive user interfaces, Unidata recognizes that it’s products and services must evolve to support new approaches to research and education [5]. As a result, Unidata has prototyped a JupyterHub¹ that plays a vital role in empowering data science and abstracting the complexities of advanced cyber-infrastructure from its users [4, 6, 7]. JupyterHub provides an easy to use interactive programmatic interface that is accessible from your browser that alleviates the burden for users to access resources via more cumbersome command line methods [8, 9].

Jupyter is a widely adopted technology in the research computing community that provides users access to campus and national high performance computing (HPC) resources. For example, at Clemson, JupyterHub was deployed to provide access to a campus Palmetto cluster for users to prototype and develop their code before scaling to production [10]. Additionally, Ngo *et al.* integrated the use of JupyterHub to support their teaching of parallel and distributed computing (PDC) at an undergraduate level [11]. Similarly, Georgia State University has deployed a JupyterHub for their instructional cluster, Data Intensive Computing Environment (DICE), which was adopted into Big Data Programming and Scientific Computing courses from Computer Science and Mathematics and Statistics departments, respectively [12]. Additionally, Georgia State used it’s JupyterHub for conducting “The Fundamentals in Data Science” workshop series, which provided students and researchers, mostly representing the *long tail of science*, an opportunity to learn the tools while interacting with a production cluster. On a larger scale, an experimental JupyterHub was deployed at SDSC for spawning single user Jupyter notebooks on Comet for *in-situ* visualization and interactive data analysis [13] that are great features for a large-scale resource to have as it promotes active learning. More recently, Holdgraf *et al.* presented on recent advancements in using cloud computing and containerization for improving the learning experience in boot camp-style workshops and benefits of hosting such workshops on national resources such as Jetstream that provides needed environment and resources that may sometimes be unavailable to local institutions [14, 15].

To support this drive for democratization of quickly and reliably deploying production grade learning environments on cloud computing environments using JupyterHub, Andrea Zonca has provided invaluable documentation for deploying a JupyterHub using Kubernetes on Jetstream [16] along with Yuvi Panda’s [17]

and project Zero to JupyterHub guide [18] on deploying Kubernetes cluster. Kubernetes provides deployment automation, scaling, and management of containerized applications. Basically, Kubernetes may be viewed as an advanced workload manager and orchestrator for the cloud computing infrastructure. Using JupyterHub, Kubernetes offers a level of abstraction for spawning multiple hubs in the same Kubernetes cluster while offering a comprehensive resource control that each namespace may use, which helps better utilize the cluster resources by closely packing the users’ spawned notebooks. Additionally, Kubernetes is interoperable across multiple cloud providers, and supports running a same cluster across multiple cloud providers at the same time [19, 20]. Furthermore, Magnum is a container orchestration engine for deploying and managing containers as first-class resources in OpenStack, which enabled rapid scaling of Kubernetes cluster using the Heat templates [21, 22]. These features are appealing to Unidata as they provide a path to achieving scaling objectives. Additionally, Kubernetes is supported on an NSF sponsored resources such as the Jetstream cloud hosted at Indiana University and TACC (Figure 1) that Unidata is currently using to host containerized software.

2 OBJECTIVE

The objective of this project is to make it straightforward for users to pull data from Unidata and request resources dynamically to burst to Jetstream using JupyterHub as a Platform as a Service. For example, in a classroom use case, one would need to scale up the resources based on the size of the class, and scale down after the class while maintaining and saving user data via long term storage volumes for next session(s). In Figure 2, we provide a high-level illustration of the end-user service in which the users are able to use the Jupyter notebook to start a session on Jetstream cloud resource, access the available data from Unidata repository hosted on Jetstream or get the data from an external source using the APIs imbedded in the workflows, and run analysis and visualization workflows that’s provided and developed by the Unidata team.

In the following section we present our approach in deploying the JupyterHub within a Kubernetes cluster on Jetstream, and we provide a templated approach for rapidly scaling up or down a Kubernetes cluster.

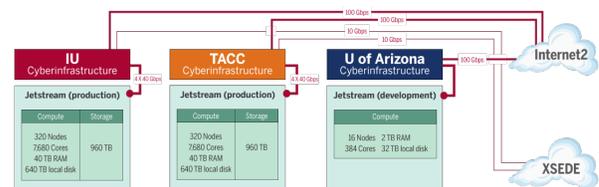


Figure 1: Jetstream architecture diagram (adopted from [23]).

¹ <https://jupyter-jetstream.unidata.ucar.edu/>

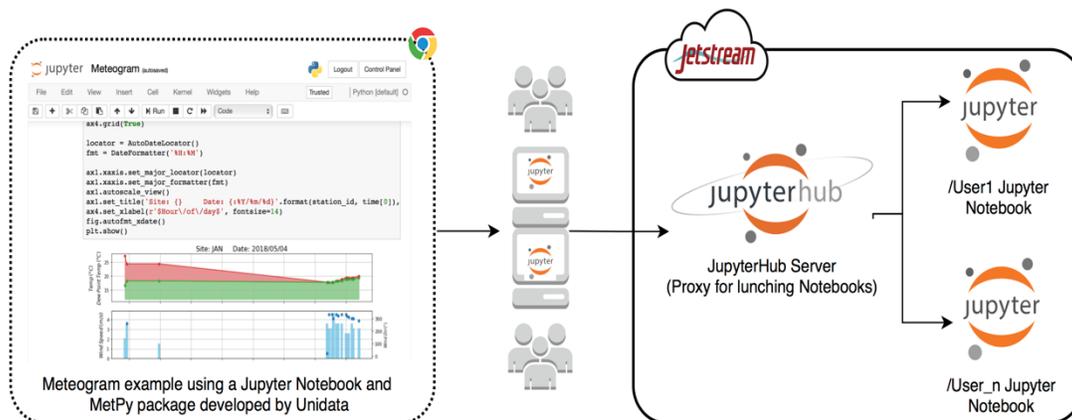


Figure 2: JupyterHub service for user training and next-generation workforce development for the geosciences. This platform will provide users easy access via Globus authentication to interactive Jupyter notebooks. These notebooks are pre-configured with Unidata Python libraries, geoscience APIs and analysis and visualization tools.

3 DESIGN AND APPROACH

3.1 Design

Our design for the JupyterHub platform is detailed in Figure 3. In current design, the Kubernetes cluster resources may be provisioned at either of the Jetstream sites (i.e., IU or TACC). For this project we used IU resources exclusively, but we had ambition to scale across the two sites for the deployment of Kubernetes cluster, which would have provided a more fault tolerant solution. When users connect to JupyterHub using Globus OAuth, a notebook session is initiated for the user with the predefined resource limit and guarantee for memory, CPU, and storage (as illustrated below) that are configurable in the JupyterHub's

```
memory:
  guarantee: 1G
  limit: 2G
cpu:
  guarantee: 1
  limit: 2
storage:
  type: dynamic
  capacity: 5Gi
```

(1)

config.yaml file. This enables us to tightly pack the user sessions and maximize the utilization of the resources. In the design, long term storage (i.e., persistent storage) for user sessions is provisioned using *rook*. This plays a critical role in allowing users to start, stop, and resume their session(s) without losing their data.

3.2 Approach in scaling JupyterHub

We used two methods for scaling Kubernetes cluster. Our initial

approach was manual in which we deployed additional VMs that we provisioned and added to the cluster using *kubeadm join*, and when we were scaling down the cluster we *drain* the nodes using *kubectl drain*. This initial approach proved valuable while exploring Kubernetes technology. For a scalable solution we required an alternative, which was available via templated approach using *magnum*. With *magnum*, one is able to update the number of nodes within the cluster as listed the code example below. This enables one to rapidly scale up or down the number of VMs in the cluster.

```
magnum bay-update kubernetes_cluster replace node_count=3
```

(2)

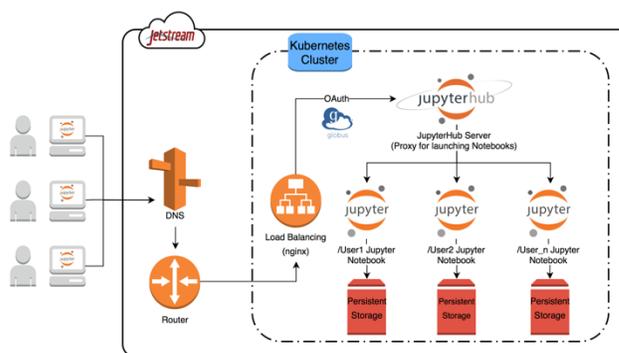


Figure 3: More detailed illustration of the design for JupyterHub deployment using Kubernetes on Jetstream. Users are routed via Jetstream's DNS and router to the JupyterHub service hosted in the Kubernetes cluster deployed on the Jetstream's (IU/TACC) VM instances along with mounted

Overall, for the deployment, we used the Kubernetes package manager, *helm*, for installing *rook* for persistent storage. Ingress controller was provided using *nginx* that was installed via *helm*, and *kube-lego* was installed via *helm* that is for setting up https using *letsencrypt*. Lastly, we installed JupyterHub from *helm* while exploring Kubernetes, which is detailed at our Git repository, which also provides Docker files for the JupyterHub that are used for the deployment (second approach) of JupyterHub from a Docker image [24].

4 RESULT AND DISCUSSION

Our design for the Unidata's JupyterHub platform provides users easy access via Globus authentication to interactive Jupyter notebooks. These notebooks are pre-configured with Unidata Python libraries, geoscience APIs and analysis and visualization tools. Users data is saved after they log out of their running sessions allowing them to resume from where they left off previously, which is ideal for a classroom use case. We did not explore deploying the Kubernetes cluster across the two Jetstream sites (TACC and IU) as it was outside of the scope of this project, but it warrants an interesting discussion as this would provide a greater reliability to the end users in cases of single site experiencing an event. Plus, this would provide an attractive alternative for researchers who want to use academic cloud resources for their research but require greater reliability that currently only commercial cloud is able to provide.

5 CONCLUSIONS

We presented a brief overview of the JupyterHub efforts with various deployments within the research computing community in which implementations ranged from campus clusters and national resources using batch spawner method to cloud computing using containerization technology such as Kubernetes that we discussed. In this manuscript, we described our approach in exploring the deployment of scalable JupyterHub using Kubernetes on Jetstream cloud using both manual and templated approach that leverages magnum.

ACKNOWLEDGMENTS

We would like to thank Dr. Mohan Ramamurthy, George Turner, Chris Hanna, Nancy Wilkins-Diehr, Marisa Brazil, Jay Alameda, Dimithu Wannipurage, and XSEDE's Extended Collaborative Support Service (ECSS) program for their support. Also, we would like to thank Andrea Zonca and Yuvi Panda for the documentation on Kubernetes that was an invaluable resource. This work was supported by XSEDE [25] Allocation Grants: GEO150002, ATM170032, ATM160027. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by NSF grant number ACI-1053575.

REFERENCES

- [1] H. Howe, m. Coostanzo, P. Fey, T. Gojorbori, L. Hannick, W. hide, D.P. hill, R. Kania, M. Schaeffer, S.S. Pierre, S. Twigger, O. White, S.Y. Rhee. Big Data: The future of biocuration. *Nature*, 2008. 455(7209): p. 47-50
- [2] M. K. Ramamurthy. Unidata 2018: Transforming geoscience through innovative data services. Technical report, Unidata Program Center, UCAR

- Community Programs, University Corporation for Atmospheric Research, 3300 Mitchell Ln, Boulder CO 80301, USA, October 2013. https://www.unidata.ucar.edu/publications/Unidata_2018.pdf
- [3] Mohan Ramamurthy, Julien Chastang, Ryan May, and Michael James. 2017. Unidata and data-proximate analysis and visualization in the cloud. DOI: <https://doi.org/10.6084/m9.figshare.5249839.v1>
- [4] J. Chastang, R. Signell, J. Fischer. Reducing Time to Science: Unidata and JupyterHub Technology using the XSEDE Jetstream Cloud. In 2017 AGU Fall Meeting, New Orleans, Louisiana, USA, Dec. 11-15 2017. <https://doi.org/10.6084/m9.figshare.5660128.v1>
- [5] M. Ramamurthy, Chapter 4 - Data-Driven Atmospheric Sciences Using Cloud-Based Cyberinfrastructure: Plans, Opportunities, and Challenges for a Real-Time Weather Data Facility, In *Cloud Computing in Ocean and Atmospheric Sciences*, edited by Tiffany C. Vance, Nazila Merati, Chaowei Yang and May Yuan, Academic Press, 2016, Pages 43-58, ISBN 9780128031926, <https://doi.org/10.1016/B978-0-12-803192-6.00004-9>.
- [6] Unidata Science Gateways on the XSEDE Jetstream Cloud. Retrieved April 27, 2018. <http://science-gateway.unidata.ucar.edu>
- [7] Unidata JupyterHub. Retrieved May 6, 2018. <https://jupyter-jetstream.unidata.ucar.edu/>
- [8] Jupyter. Retrieved May 6, 2018. <http://jupyter.org>
- [9] JupyterHub. Retrieved May 6, 2018. <http://jupyterhub.readthedocs.io/en/latest/>
- [10] JupyterHub on the Palmetto Cluster. Retrieved May 5, 2018. Clemson University. https://www.palmetto.clemson.edu/palmetto/jupyterhub_index.html
- [11] Linh B. Ngo, Ashwin Trikuta Srinath, Jeffrey Denton, Marcin Ziolkowski, Unifying computing resources and access interface to support parallel and distributed computing education. *Journal of Parallel and Distributed Computing*, 2018, ISSN 0743-7315, <https://doi.org/10.1016/j.jpdc.2018.02.020>.
- [12] Semir Sarajlic, Naranjan Edirisinghe, Yubao Wu, Yi Jiang, and Gregori Faroux. 2017. Training-based Workforce Development in Advanced Computing for Research and Education (ACoRE). In *Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact (PEARC17)*. ACM, New York, NY, USA, Article 71, 4 pages. DOI: <https://doi.org/10.1145/3093338.3104178>
- [13] Andrea Zonca. Sample deployment of JupyterHub in HPC on SDSC Comet. Published February 26, 2017. Retrieved May 6, 2018. <https://zonca.github.io/2017/02/sample-deployment-jupyterhub-hpc.html>
- [14] Chris Holdgraf, Aaron Culich, Ariel Rokem, Fatma Deniz, Maryana Alegro, and Dani Ushizima. 2017. Portable Learning Environments for Hands-On Computational Instruction: Using Container- and Cloud-Based Technology to Teach Data Science. In *Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact (PEARC17)*. ACM, New York, NY, USA, Article 32, 9 pages. DOI: <https://doi.org/10.1145/3093338.3093370>
- [15] 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 (XSEDE '15)*. ACM, New York, NY, USA, Article 29, 8 pages. DOI: <http://dx.doi.org/10.1145/2792745.2792774>
- [16] Andrea Zonca. Deploy Scalable JupyterHub with Kubernetes on Jetstream. Published December 5, 2017. Retrieved December 19, 2017. <https://zonca.github.io/2017/12/scalable-jupyterhub-kubernetes-jetstream.html>
- [17] Kubeadm Bootstrapper. Git. <https://github.com/data-8/kubeadm-bootstrap>
- [18] Zero to JupyterHub. <https://zero-to-jupyterhub.readthedocs.io/en/latest/>
- [19] Kubernetes Documentation. Retrieved on May 6, 2018. <https://kubernetes.io/docs/home/>
- [20] Kubespawner (jupyterhub-kubernetes-spawner). Retrieved May 6, 2018. <https://github.com/jupyterhub/kubespawner>
- [21] Magnum user guide. Retrieved May 6, 2018. <https://docs.openstack.org/magnum/latest/user/>
- [22] Magnum. Retrieved on May 6, 2018. <https://wiki.openstack.org/wiki/Magnum>
- [23] Jetstream - Tech Overview. 2017. <https://jetstream-cloud.org/tech-specs/index.php>
- [24] Unidata on the Jetstream Cloud. Git. Retrieved on May 6, 2018. <https://github.com/Unidata/xsede-jetstream>
- [25] Towns, J., Cockerill, T., Dahan, M., Foster, I., Gathier, K., Grimshaw, A., Hazlewood, V., Lathrop, S., Liika, D., Peterson, G. D., Roskies, R., Scott J.R., Wilkins-Diehr, N., "XSEDE: Accelerating Scientific Discovery", *Computing in Science & Engineering*, vol.16, no. 5, pp. 62--74, Sept.-Oct. 2014, doi:10.1109/MCSE.2014.80