

# Easier JupyterLab Instances for HPC Users



**Dylan Wallace**

**Mentor: Giovanni Cone**

August 13, 2020



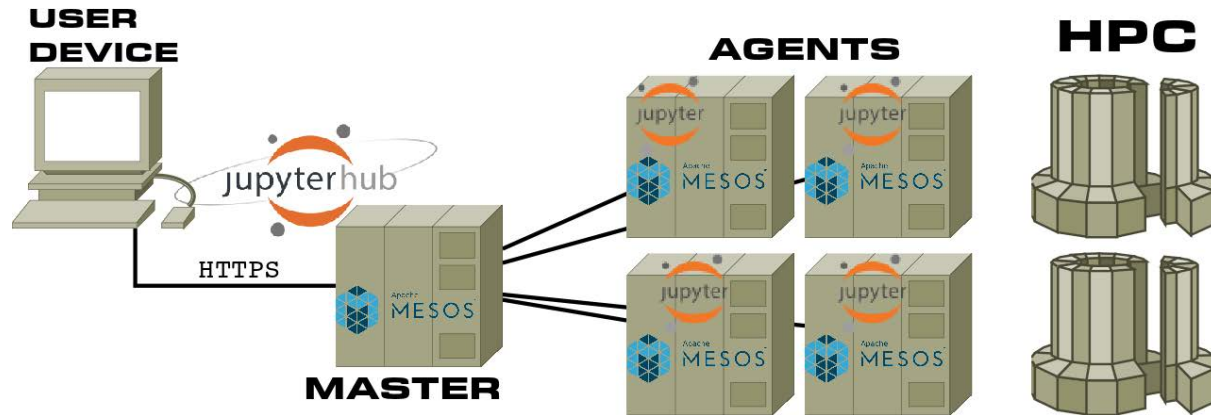
Managed by Triad National Security, LLC for the U.S. Department of Energy's NNSA

# Motivation

- Currently, HPC users wanting to use a JupyterLab instance have one of two options, and both involve tunneling the JupyterLab session over an SSH connection
- Option 1 (easiest method)
  - Launch a JupyterLab instance on a front-end node (bad)
- Option 2
  - Tunnel the JupyterLab instance from an interactive SLURM allocation

# First approach - Apache Mesos

- Apache Mesos is a cluster manager
- Send JupyterLab instances to the worker nodes
- Send computationally expensive jobs to HPC clusters



# The problem with Mesos

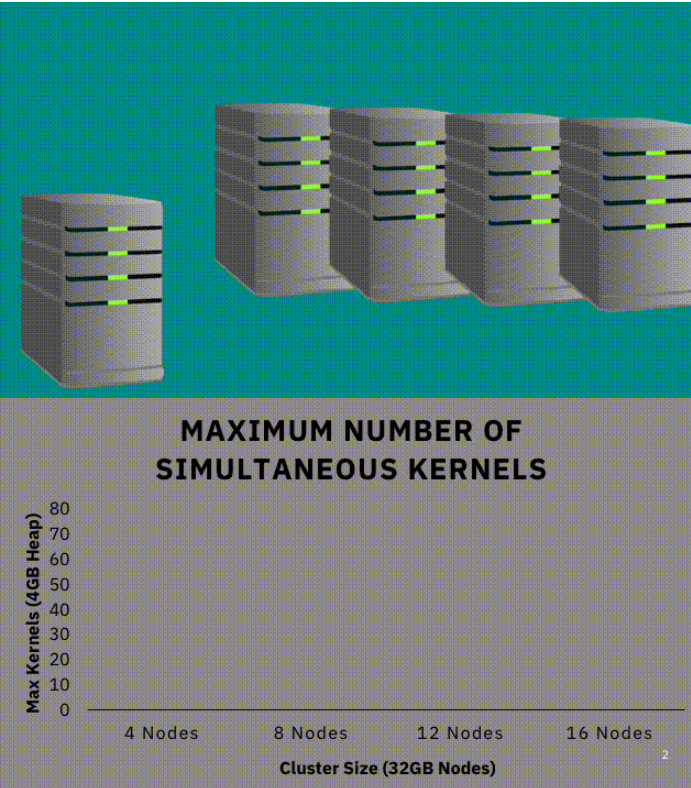
- Lack of support
- Bad documentation

# New approach – Jupyter Enterprise Gateway on Kubernetes

- Well-supported by a large community

## Limitations of Jupyter Notebook Stack

- Scalability
  - Jupyter kernels running as local processes
    - Resources are limited by what is available on the **single node** that runs **all kernels** and associated Spark drivers
- Security
  - **Single user** sharing the same privileges
  - Users can see and control each other's process using kernel macros or system commands



Source: <https://jupyter-enterprise-gateway.readthedocs.io/en/latest/>

# New approach – Jupyter Enterprise Gateway on Kubernetes

## Jupyter Enterprise Gateway Features

### Optimized Resource Allocation

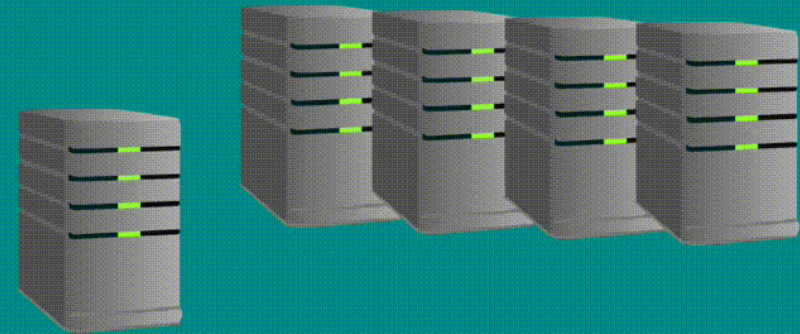
- Utilize resources on all cluster nodes by running kernels as managed resources leveraging underlying resource manager (e.g., Hadoop YARN, Kubernetes, IBM Spectrum Conductor)
- Pluggable architecture to enable support for additional Resource Managers

### Enhanced Security

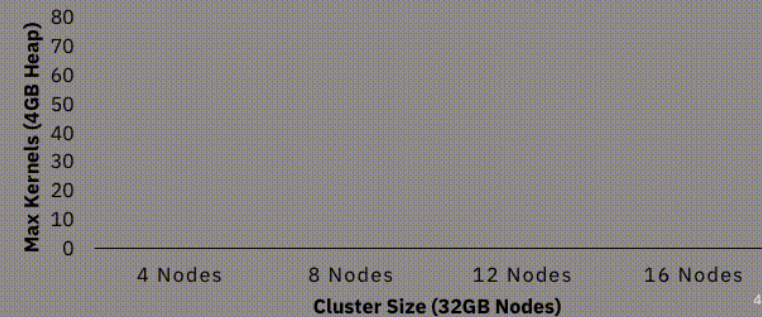
- End-to-End secure communications
  - Secure socket communications
  - Encrypted HTTP communication using SSL

### Multiuser support with user impersonation

- Enhance security and sandboxing by enabling user impersonation when running kernels (using Kerberos).



## MAXIMUM NUMBER OF SIMULTANEOUS KERNELS



Source: <https://jupyter-enterprise-gateway.readthedocs.io/en/latest/>

# Next Steps

- Finish setting up the Kubernetes cluster in a test environment
- Launch kernels on the cluster

# Questions?