

# SciServer at MPE

Migration to HPC

SciServer 

Collaborative data-driven science

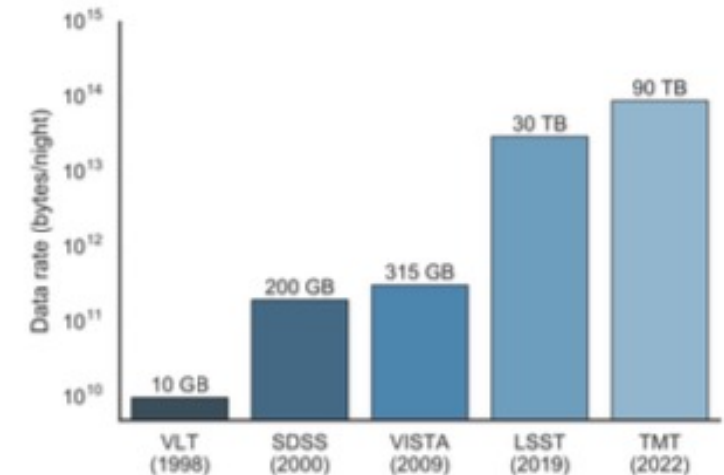
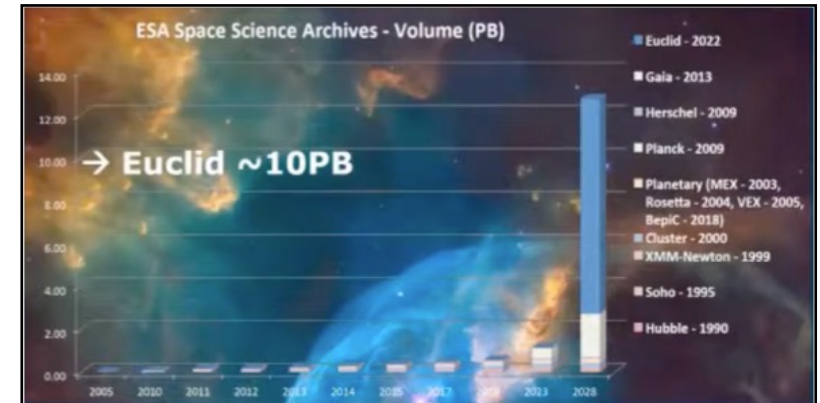


MPCDF



# Background

- SciServer was developed, and continues to be maintained, by the Institute for Data-Intensive Engineering and Science (IDIES) at Johns Hopkins University (JHU).
- Created in response to the need of working with astronomical (literally and figuratively) datasets in an efficient manner.
- MPE entered conversation with their team in 2018.
- Initial support existed for the implementation at MPE, then limited support for every day use cases.

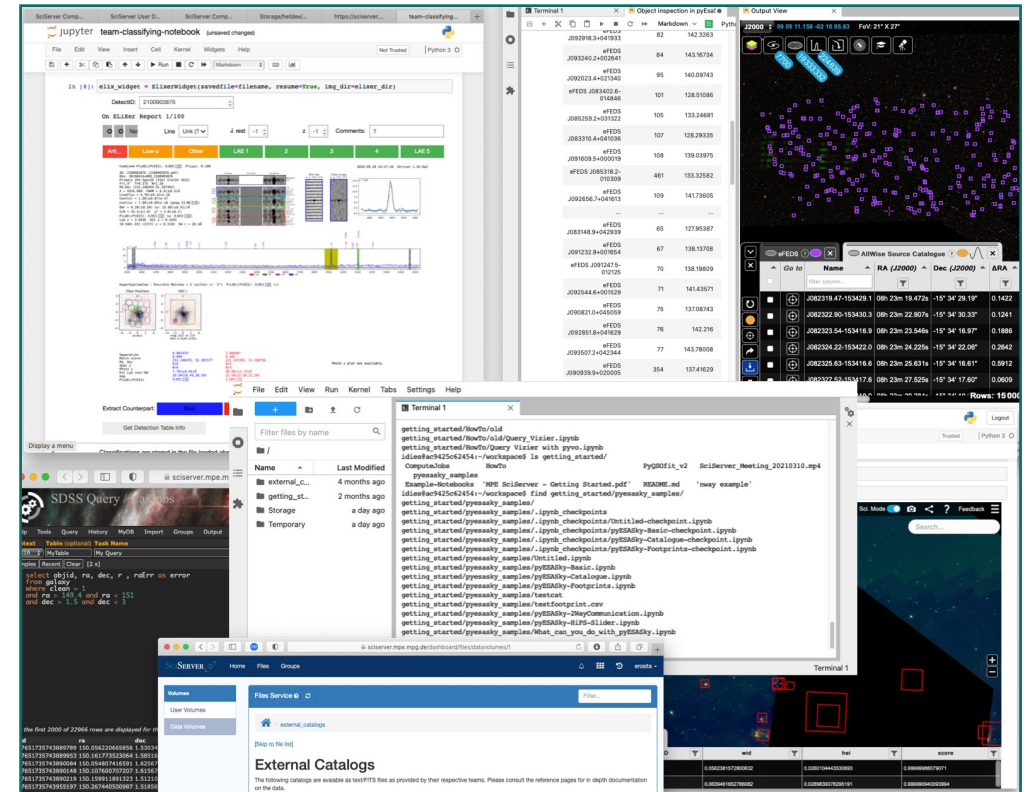


Radio archives  
today:  
ALMA 1PB  
LOFAR 50PB



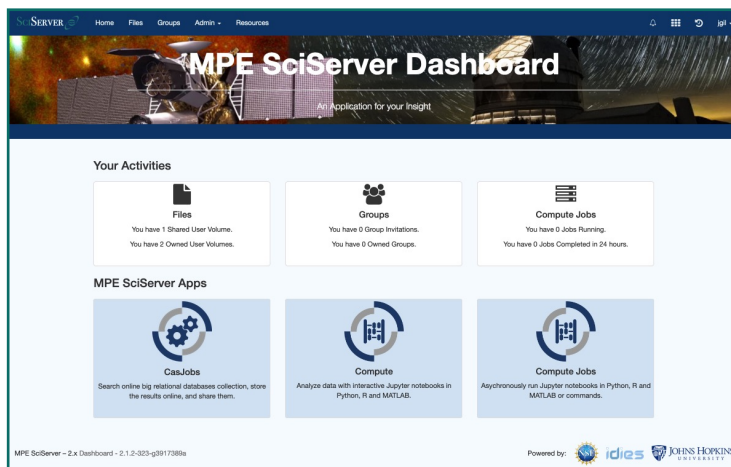
# What is a SciServer and what is it good for?

- Core concept behind the SciServer is to allow different teams to work where the data is.
- SciServer combines mass storage, a database server, and a work environment into one unified UI.
- Scientists can work on their data in Jupyter Notebooks or a terminal and share their work with colleagues.
- Flexibility to create working groups around dataset subsets.
- The MPE SciServer is set up specifically to cater for the needs of the eROSITA and HETDEX collaborations.

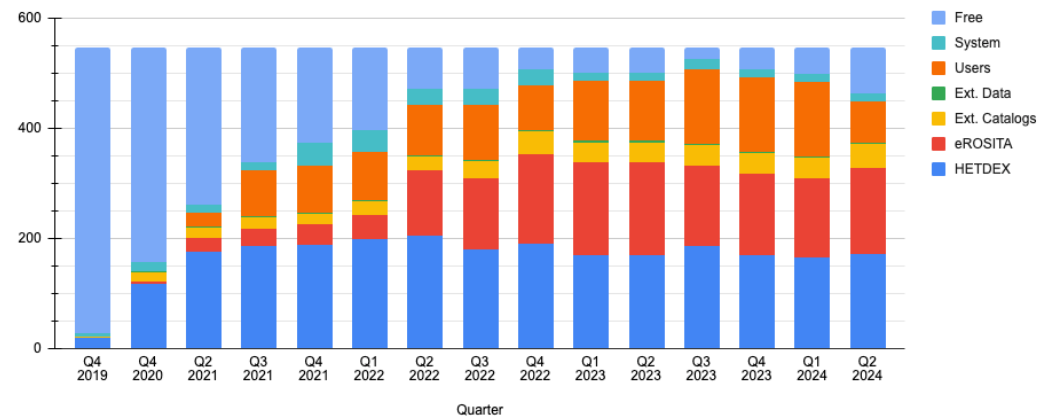


# Sciserver @ MPE

- Implementation started in 2019, currently ~200 active users.
- ~550 TB of storage
- Purpose of managing large datasets, including MPE's own [eROSITA telescope](#) data.
- Reaching its one-man team maintenance limits on its own hardware
- Need to manage EOL hardware warranties
- E-rosita [youtube](#) presentation



SciServer Mass Storage (547 TB total)



# What does it look like? Homepage

SciSERVER Home Files Groups Admin Resources

## MPE SciServer Dashboard

An Application for your Insight

### Your Activities

- Files**  
You have 1 Shared User Volume.  
You have 2 Owned User Volumes.
- Groups**  
You have 0 Group Invitations.  
You have 0 Owned Groups.
- Compute Jobs**  
You have 0 Jobs Running.  
You have 0 Jobs Completed in 24 hours.

### MPE SciServer Apps

- CasJobs**  
Search online big relational databases collection, store the results online, and share them.
- Compute**  
Analyze data with interactive Jupyter notebooks in Python, R and MATLAB.
- Compute Jobs**  
Asynchronously run Jupyter notebooks in Python, R and MATLAB or commands.

MPE SciServer – 2.x Dashboard - 2.1.2-323-g3917389a

Powered by:





# What does it look like? Adding jobs

mpute Interactive Notebooks Jobs

Compute Jobs

Compute Jobs a

Existing Notebook

## New Job

Compute Domain Compute Image Data Volumes User Volumes Command

**Command:**

```
#!/bin/sh  
echo "We do jobs and cronjobs too"
```

**Working Directory:**

Select a location to store standard input/output logs, and act as the current working directory for this job. Enable other writable user volumes on the Files tab to be able to use them here. **Do not use relative paths in the command.**

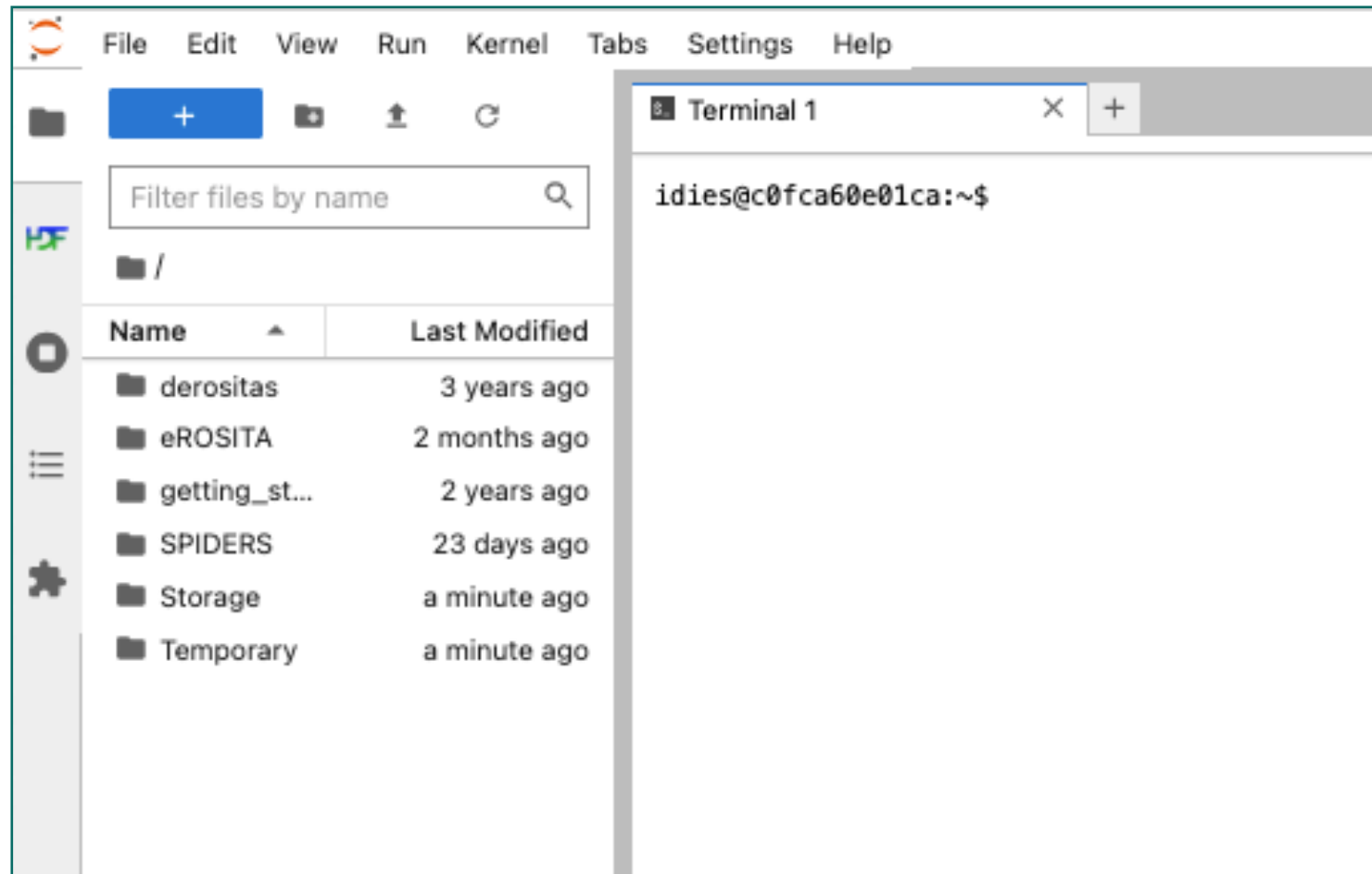
Use a new folder in the "jobs" Temporary volume (will be created)

- A copy of this command will be placed in a unique, nested subfolder of `/home/idies/workspace/Temporary/jg1l/jobs/`.
- Relative paths will be resolved from this location.

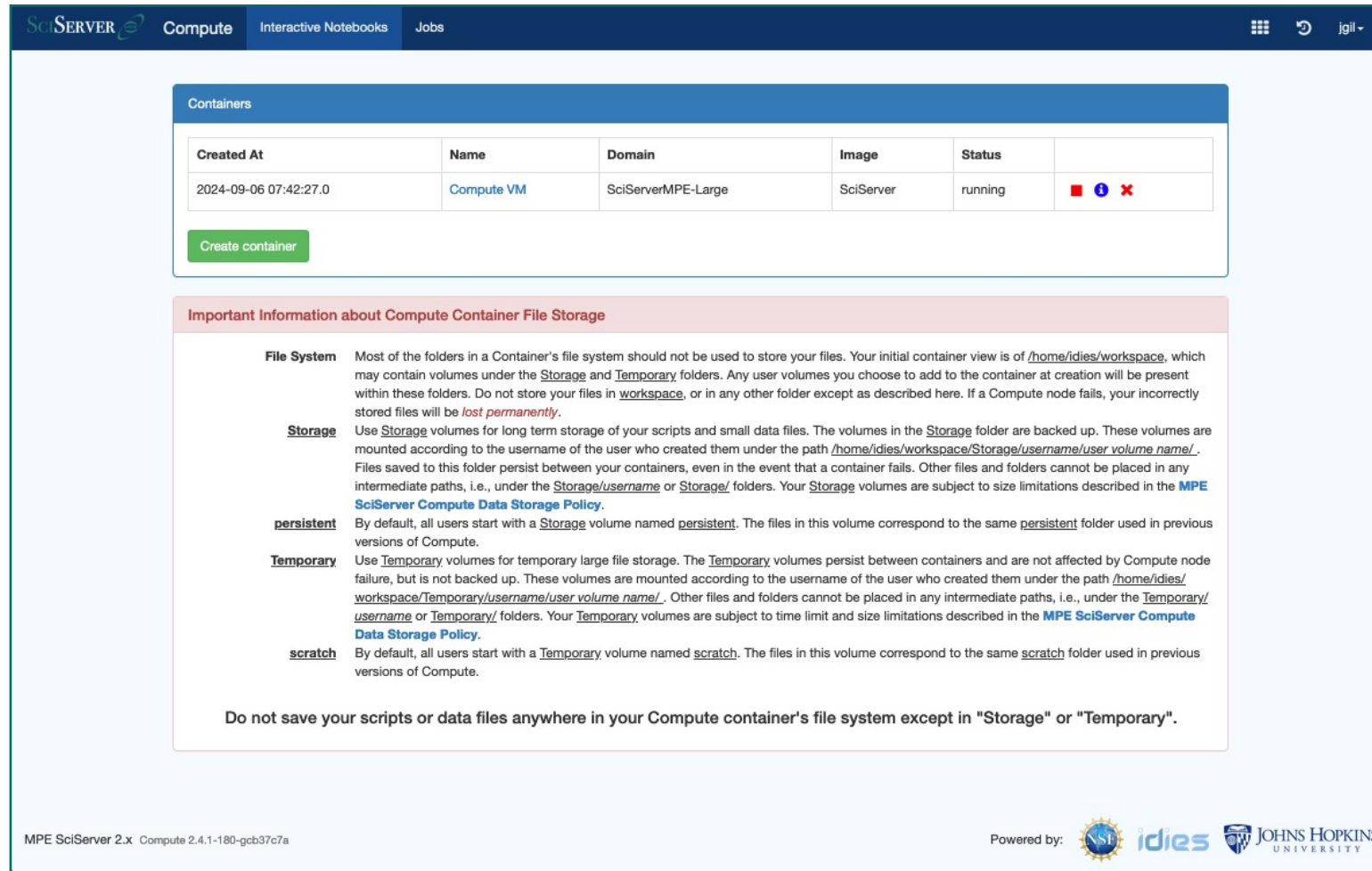
< > Create Job



# What does it look like? Use a terminal



# What does it look like? Container mgmt



The screenshot displays the SciServer Compute interface. At the top, there are navigation tabs for 'Compute', 'Interactive Notebooks', and 'Jobs'. The 'Compute' tab is active. Below the navigation, there is a 'Containers' section with a table listing containers. The table has columns for 'Created At', 'Name', 'Domain', 'Image', 'Status', and actions. One container is listed: 'Compute VM' created on '2024-09-06 07:42:27.0', with domain 'SciServerMPE-Large', image 'SciServer', and status 'running'. Below the table is a 'Create container' button. Underneath is a section titled 'Important Information about Compute Container File Storage' with detailed instructions on file system usage, storage volumes, and temporary storage.

Created At	Name	Domain	Image	Status	
2024-09-06 07:42:27.0	Compute VM	SciServerMPE-Large	SciServer	running	■ ⓘ ✕

**Important Information about Compute Container File Storage**

**File System** Most of the folders in a Container's file system should not be used to store your files. Your initial container view is of `/home/uides/workspace`, which may contain volumes under the `Storage` and `Temporary` folders. Any user volumes you choose to add to the container at creation will be present within these folders. Do not store your files in `workspace`, or in any other folder except as described here. If a Compute node fails, your incorrectly stored files will be *lost permanently*.

**Storage** Use `Storage` volumes for long term storage of your scripts and small data files. The volumes in the `Storage` folder are backed up. These volumes are mounted according to the username of the user who created them under the path `/home/uides/workspace/Storage/username/user volume name/`. Files saved to this folder persist between your containers, even in the event that a container fails. Other files and folders cannot be placed in any intermediate paths, i.e., under the `Storage/username` or `Storage/` folders. Your `Storage` volumes are subject to size limitations described in the [MPE SciServer Compute Data Storage Policy](#).

**persistent** By default, all users start with a `Storage` volume named `persistent`. The files in this volume correspond to the same `persistent` folder used in previous versions of Compute.

**Temporary** Use `Temporary` volumes for temporary large file storage. The `Temporary` volumes persist between containers and are not affected by Compute node failure, but is not backed up. These volumes are mounted according to the username of the user who created them under the path `/home/uides/workspace/Temporary/username/user volume name/`. Other files and folders cannot be placed in any intermediate paths, i.e., under the `Temporary/username` or `Temporary/` folders. Your `Temporary` volumes are subject to time limit and size limitations described in the [MPE SciServer Compute Data Storage Policy](#).

**scratch** By default, all users start with a `Temporary` volume named `scratch`. The files in this volume correspond to the same `scratch` folder used in previous versions of Compute.

Do not save your scripts or data files anywhere in your Compute container's file system except in "Storage" or "Temporary".

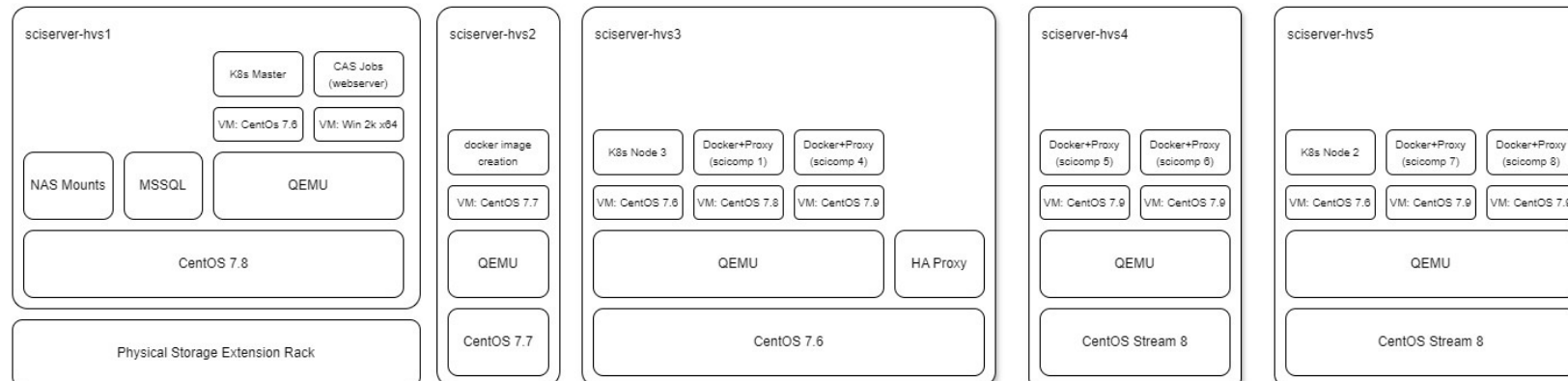


# Under the hood

- Infra bootstrapped manually (and heroically)
- Large storage rig and servers placed at MCPDF.
- Mostly CentOS (looking at Alma/Rocky)
- Kubernetes application hosts main app UI / server / user management DB / NFS mounts management
- App launches docker VMs across different nodes
- App mostly implemented in Python
- MSSQL DB for large datasets
- Images hosted at MPCDF's gitlab

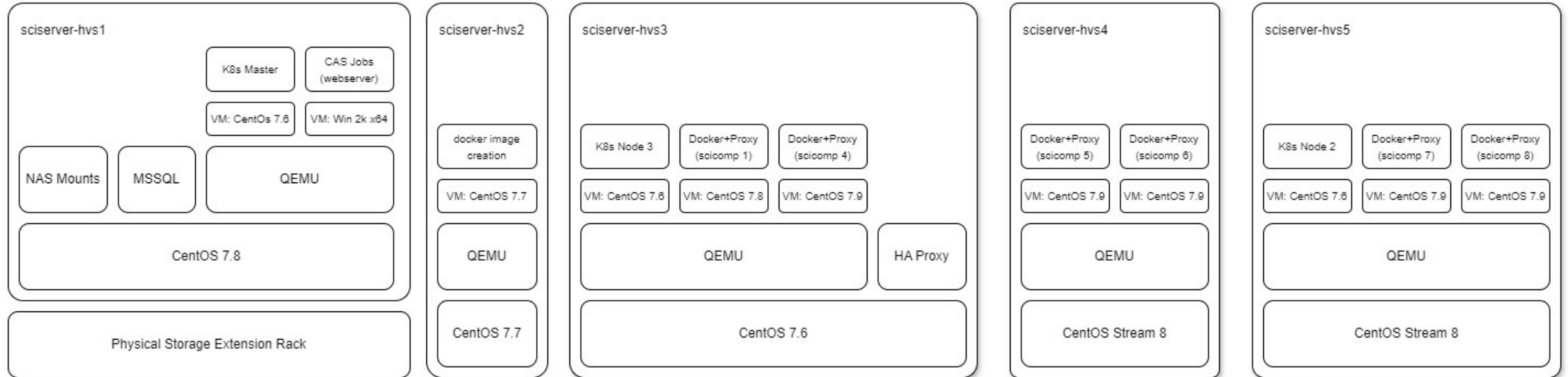


SciServer MPE



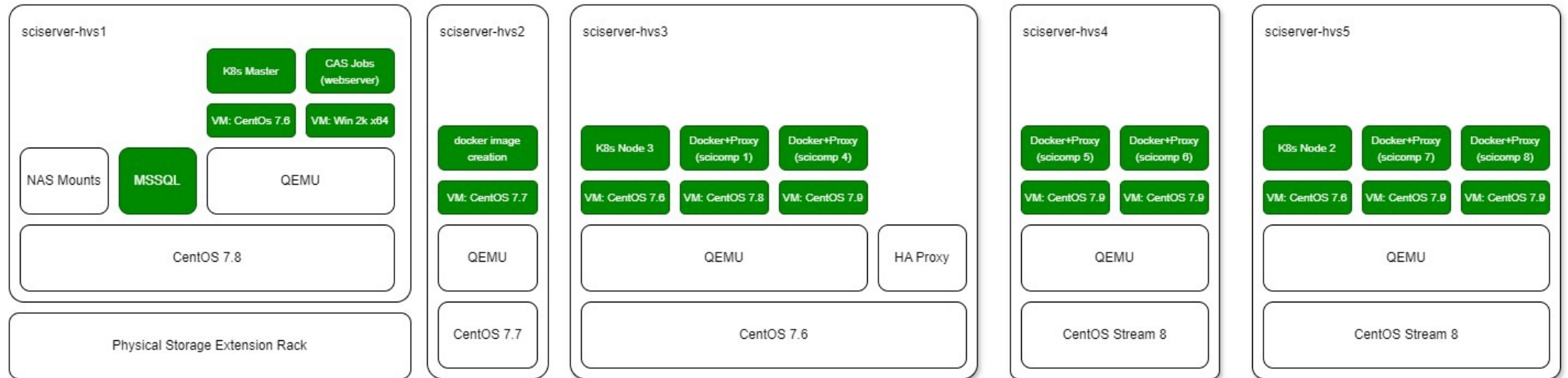
# Hardware Diagram

## SciServer MPE



# Hardware Diagram

## SciServer MPE

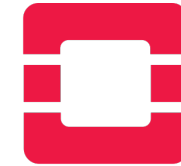


# Migration to HPC Cloud

- MPE doesn't need to worry about hardware anymore.
- Scaling is handed over to a team with larger & specialized resources.
- Use of open source tooling for infrastructure means no licensing headaches.
- Shift to Infrastructure as Code
- Transparent pricing structure.



GitLab



openstack®



# Thanks!

Jonas Haase [jhaase@mpe.mpg.de](mailto:jhaase@mpe.mpg.de)

Joel Gil [joel.gil@mpe.mpg.de](mailto:joel.gil@mpe.mpg.de)

