

The National Research Platform: Stretched, Multi-Tenant, Scientific Kubernetes Cluster

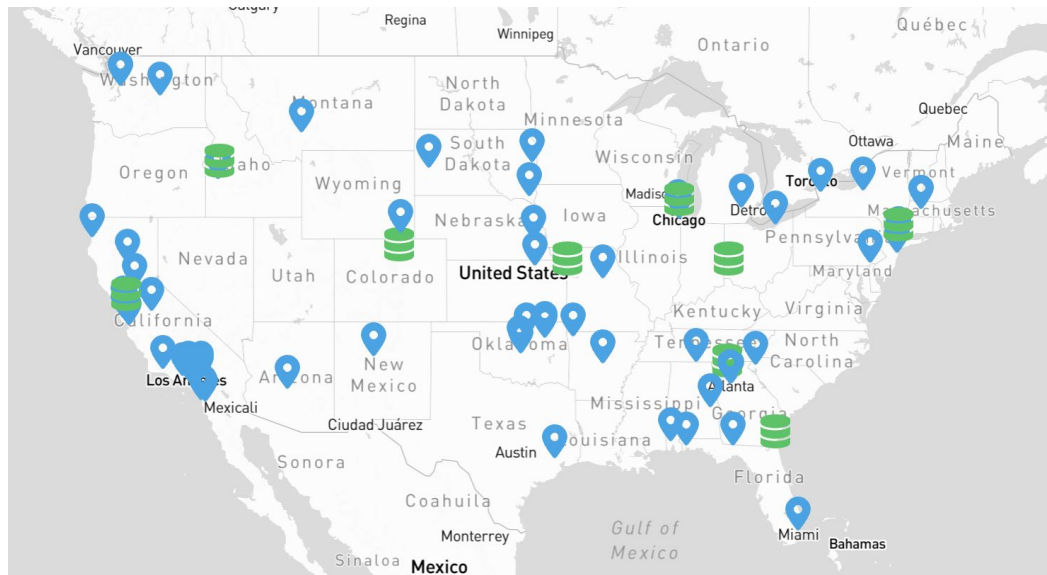
Derek Weitzel
University of Nebraska-Lincoln
On behalf of my many co-authors

This work was supported in part by National Science Foundation (NSF) awards CNS-1730158, ACI-1540112, ACI-1541349, OAC-1826967, OAC-2112167, CNS-2100237, CNS-2120019.

What is the National Research Platform?

Collaboration of universities,
national labs, and non-profit
institutions.

66+ Institutions donate **439**
nodes hosted at **84+** physical
sites. This creates a pool of
~1,500 GPUs and **29,000 CPUs**.

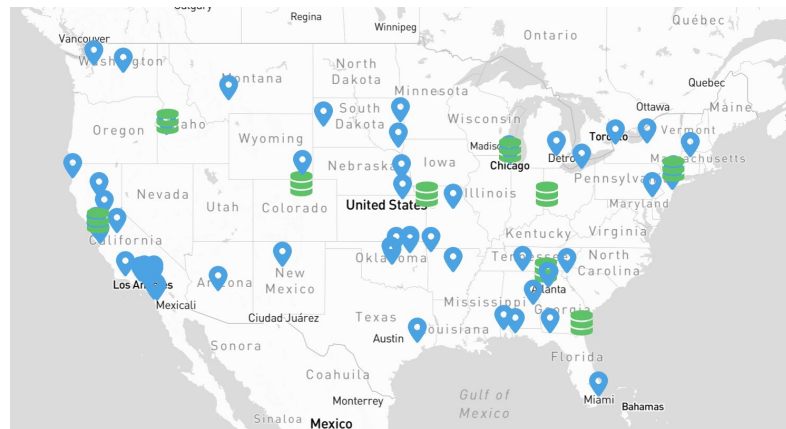


<https://dash.nrp-nautilus.io>

Who uses the NRP

In Calendar year 2024

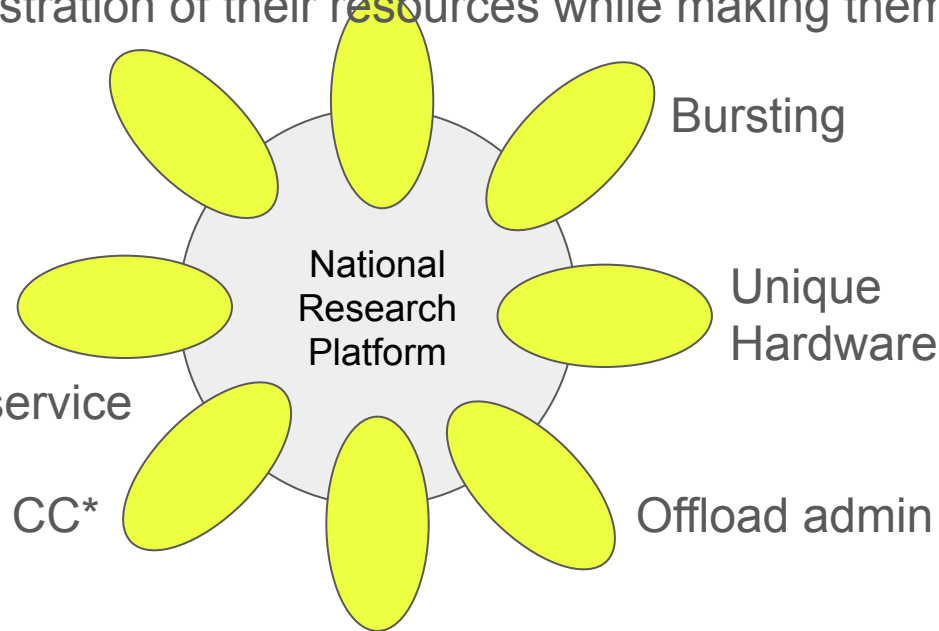
- 540 research groups utilized the NRP
 - Community Software (2)
 - Observatories (13)
 - Individual campus researchers (525)
- 5M GPU Hours - 500 GPUs utilized continuously
- Some research groups run in “opportunistic” mode, and can be preempted as needed by others (such as OSG)
- NRP has many “front doors”: (There is overlap)
 - **Kubernetes**: 267 Institutions
 - **Jupyterhub**: 177 Institutions
 - **Contributing Resources**: 66+ Institutions



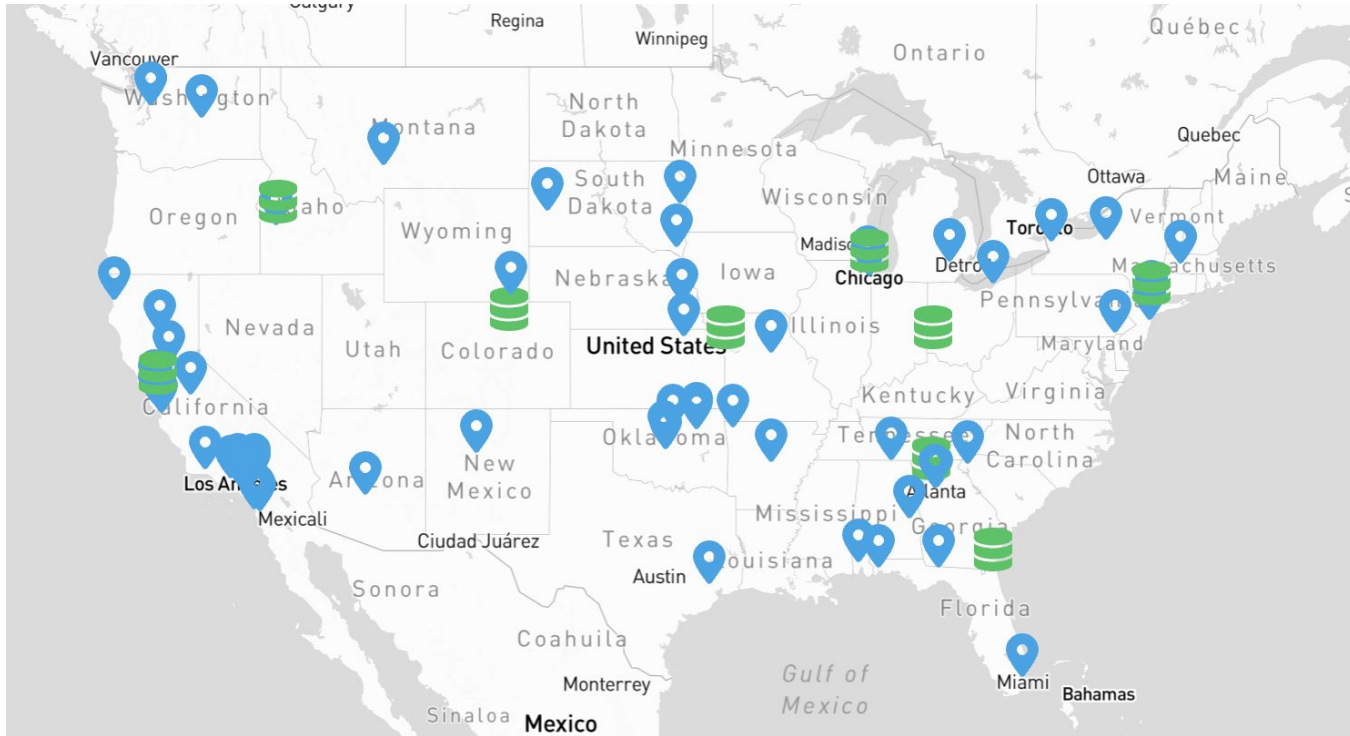
Who contributes to the NRP

Number of resources on the NRP 84 sites contribute for a number of reasons:

1. Wanting to burst into NRP resources while using their own hardware
2. Wanting to offload sys-administration of their resources while making them available to others
3. Part of a CC* obligation
4. Provide unique hardware that can be difficult to share without kubernetes
5. (Rare) Hosting a nationwide service



All these sites! All these admins!



<https://dash.nrp-nautilus.io>

Working with Sites

Working with this many sites is challenging. We have to adhere to many different sites policies.

- Some institutions want to maintain access to their nodes ✓
- Some want to run scanning or antivirus software on their nodes ~ ✓

There have been multiple instances where a site reported detected malware from their scanning tools, but our investigation revealed it was just normal usage, but sometimes they are also correct.

Most scanning and antivirus tools used by sites don't recognize containers properly, and often flag users running "sudo" or writing to privileged directories within those containers.

Working with Site Networking

Networking has consistently been a challenge for the NRP.

The NRP operates under a “default-open” model, where all traffic can reach the node, and Kubernetes enforces its own firewall policies to isolate namespaces from each other and from external networks. **This approach is unusual in traditional enterprise IT environments.**

As a result, we are often the ones advocating to IT departments for the adoption of a Science DMZ.

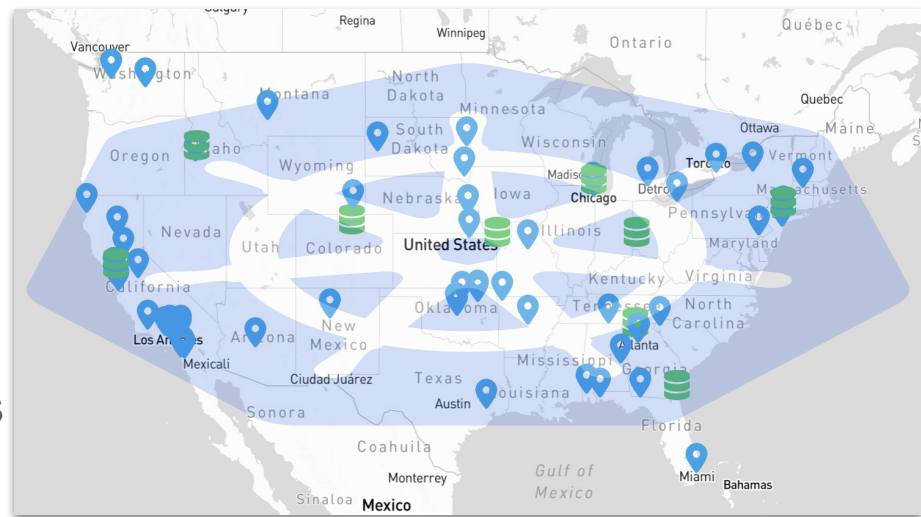
Additionally, the NRP requires a 9000 MTU network, which can take time for both institutions and regional network providers to fully support.

Stretched Kubernetes

A stretched Kubernetes cluster spans multiple administrative domains.

To enable Kubernetes networking, we use VXLAN to create a virtual Layer 2 network over a Layer 3 infrastructure.

This setup allows Kubernetes to perform its standard namespace-based network isolation.

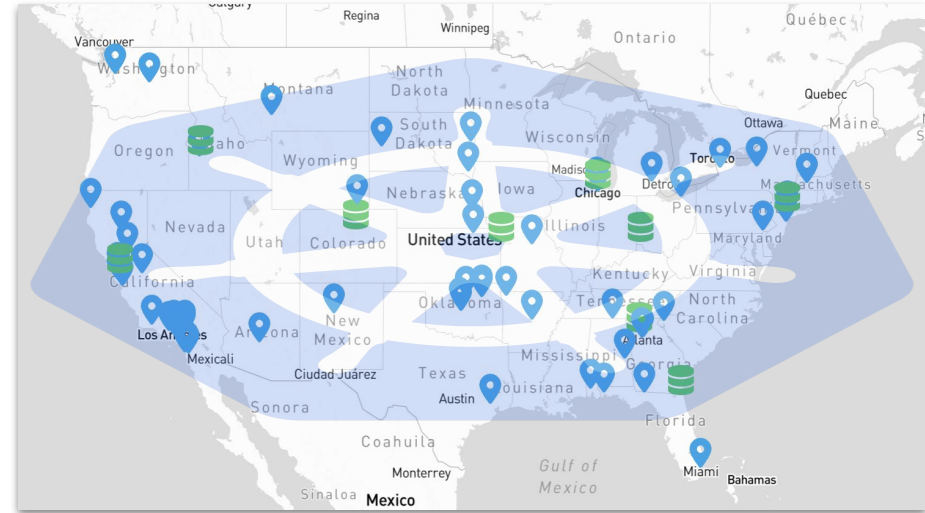


Stretched Kubernetes - Consequences

Because of the distance between nodes, we separate certain services by region.

For storage, we use regional pools since **CEPH** (the underlying storage technology) is sensitive to latency.

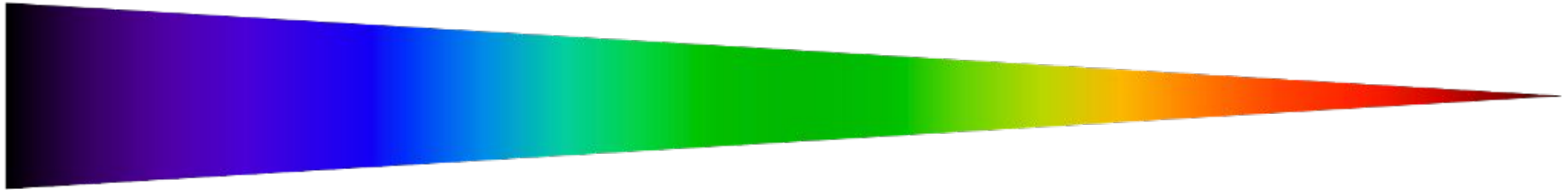
As a result, we maintain a Western Pool (West Coast), a Central Pool (Great Plains), an Eastern Pool, and a few additional custom pools.



Storage resources on the NRP

Easy to Use

Scalability



CephFS Volumes

- Shared between multiple Pods
- Easy to create and attach
- File access can be slowed by metadata

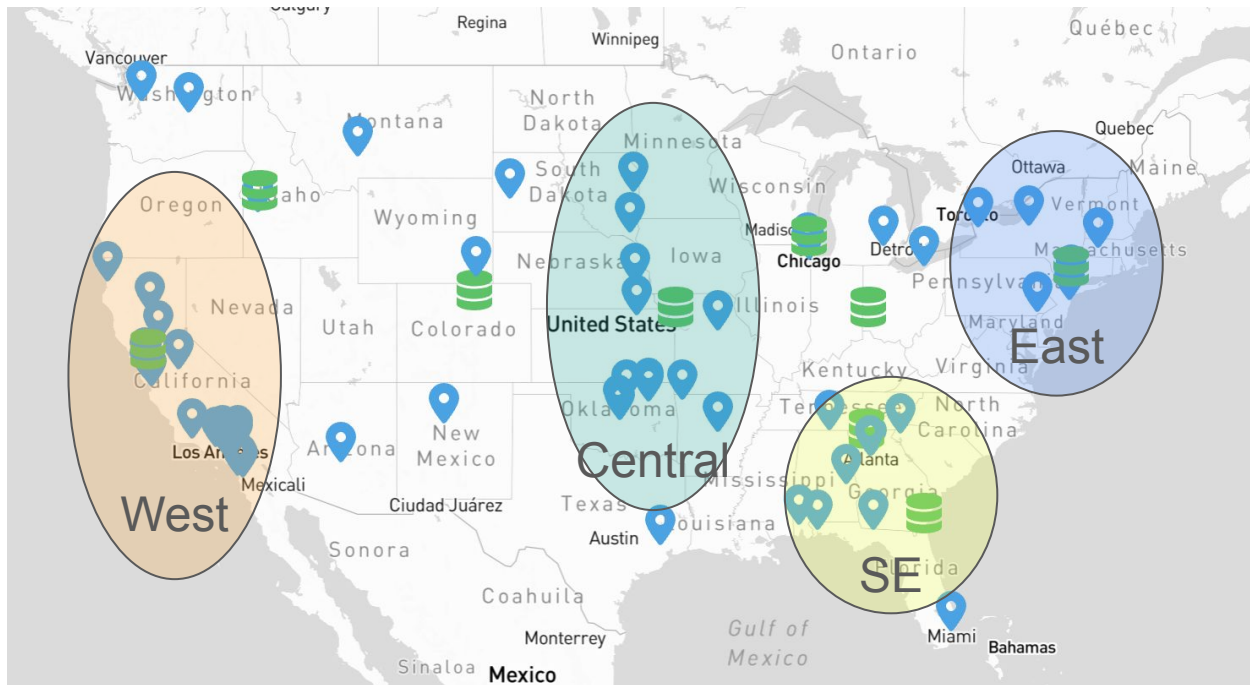
CEPH RBD Volumes

- Cannot be shared
- Easy to create and attach
- File access is fast

S3 Interface

- Special tool to download data
- Multiple download points for scale

Regional Storage



Installation (Ansible and IPMI)

Given the large number of nodes, we use Ansible to fully automate machine installation and configuration.

IPMI Handoff	SSH with sudo handoff	Joined Cluster	Inventory
<p>Install the Operating system and firmware</p> <p>Advise the site to open IPMI access to only the specific gateway nodes.</p>	<p>Run ansible on the host to install the NRP related hardware.</p> <p>Uses a one-time key to bootstrap the node to the cluster.</p>	<p>Once the node is installed and joined, Kubernetes will start and evaluate the node</p> <p>Test jobs will be run to ensure the hardware functionality</p>	<p>Our inventory system will scan the node</p> <p>The inventory of the system includes the IPMI address, any other public addresses, any hardware attached to the node and will advertise it for user jobs.</p>



User Interface - Kubernetes

Kubernetes interface is through the command line or a kubernetes UI that you might prefer, such as Lens or k9s.

No special requirements to our kubernetes.

Uses oauth tokens for authentication.

```
pod1.yaml

apiVersion: v1
kind: Pod
metadata:
  name: test-pod
spec:
  containers:
  - name: mypod
    image: ubuntu
    resources:
      limits:
        memory: 100Mi
        cpu: 100m
      requests:
        memory: 100Mi
        cpu: 100m
    command: ["sh", "-c", "echo 'Im a new pod' && sleep infinity"]
```




Hosted Jupyterhub

NRP hosts a general purpose Jupyterhub

Comes pre-built with many AI and ML images

Professionally maintained and monitored for uptime and issues

Configurable number of GPUs, and types of GPUs

The screenshot shows a Jupyter Notebook interface. On the left is a file explorer sidebar showing a directory structure with files like 'data', 'ne_110m_admin...', and several 'highlighted_stat...' files, all modified '4d ago'. The main area displays a code cell with the command `!pip install uv && uv pip install geopandas matplotlib`. The output shows the installation of `uv` (0.7.21) and the requested packages (`geopandas==1.1.1`, `pyogrio==0.11.0`, `pyproj==3.7.1`, and `shapely==2.1.1`) in a Python 3.12.8 environment.

```
[3]: !pip install uv && uv pip install geopandas matplotlib

Collecting uv
  Downloading uv-0.7.21-py3-none-manylinux_2_17_x86_64.musllinux_1_1_x86_64.whl (18.6/18.6 MB)
    Installing collected packages: uv
    Successfully installed uv-0.7.21
    Using Python 3.12.8 environment at: /opt/conda
    Resolved 19 packages in 422ms
    Prepared 4 packages in 1.15s
    [0/4] Installing wheels...
    full copy. This may lead to degraded performance.
      If the cache and target directories are on d:
      If this is intentional, set `export UV_LINK_I
    Installed 4 packages in 78ms
    + geopandas==1.1.1
    + pyogrio==0.11.0
    + pyproj==3.7.1
    + shapely==2.1.1
```



Private Jupyter - 113 In CY 2024

Same Jupyter software as the hosted version, but:

- The instructor controls the images and can “bake” in software / lessons
- The instructor controls access
- Can configure larger (or smaller) default storage as required for the course
- Instructor can control the requested resources

But

- Not professionally monitored or maintained
- It's on the instructor to configure and maintain the Jupyterhub (though, we can help)

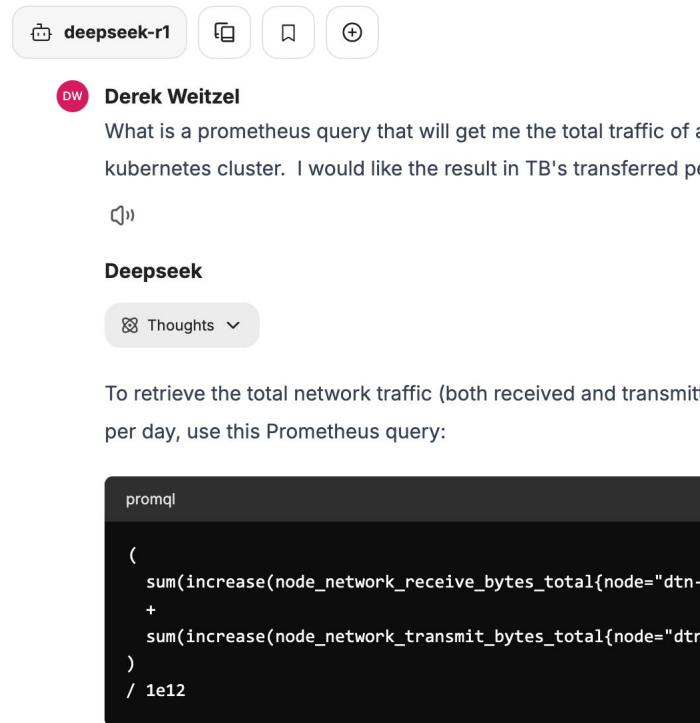
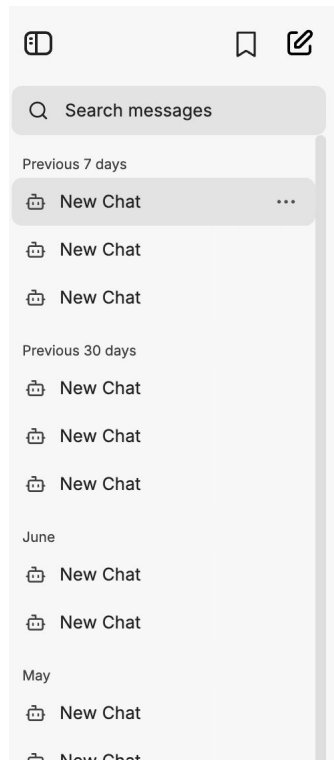


Hosted LLM Service

The NRP hosts some open-model LLMs available

Query through our hosted chat interfaces.

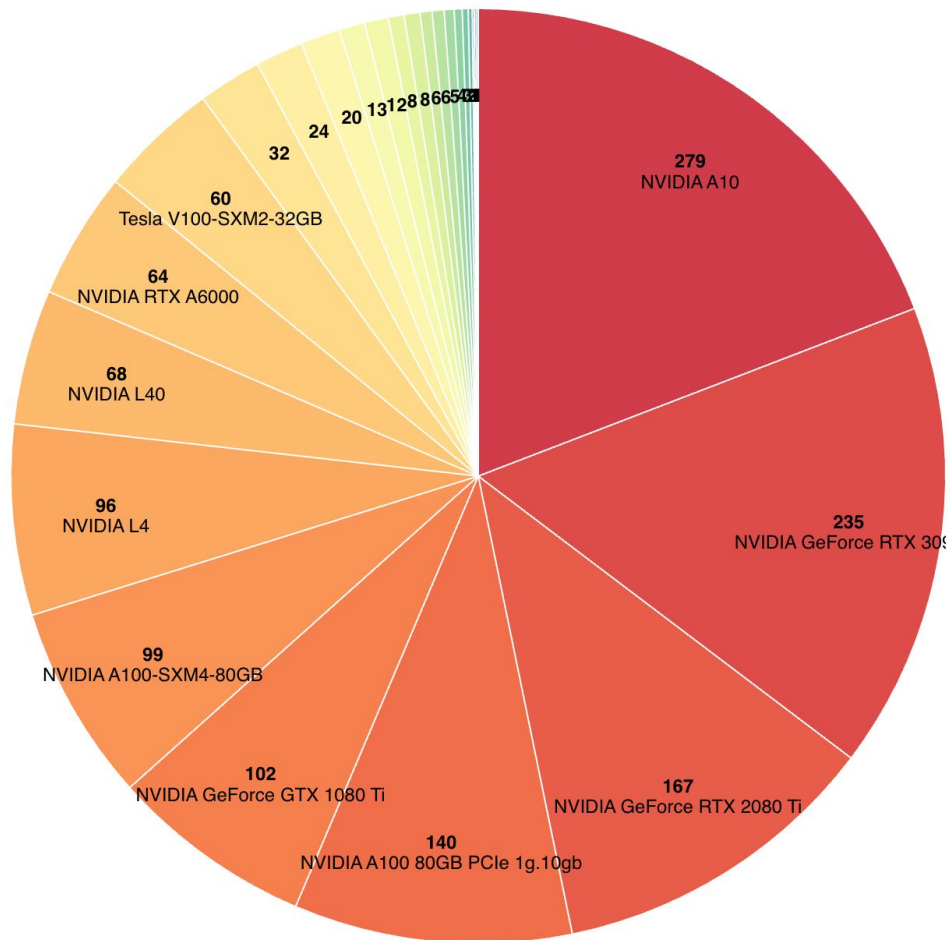
Or... use an API to perform many queries.



NRP hosted LibreChat utilizing the deepseek-r1 model

GPUs - Kubernetes

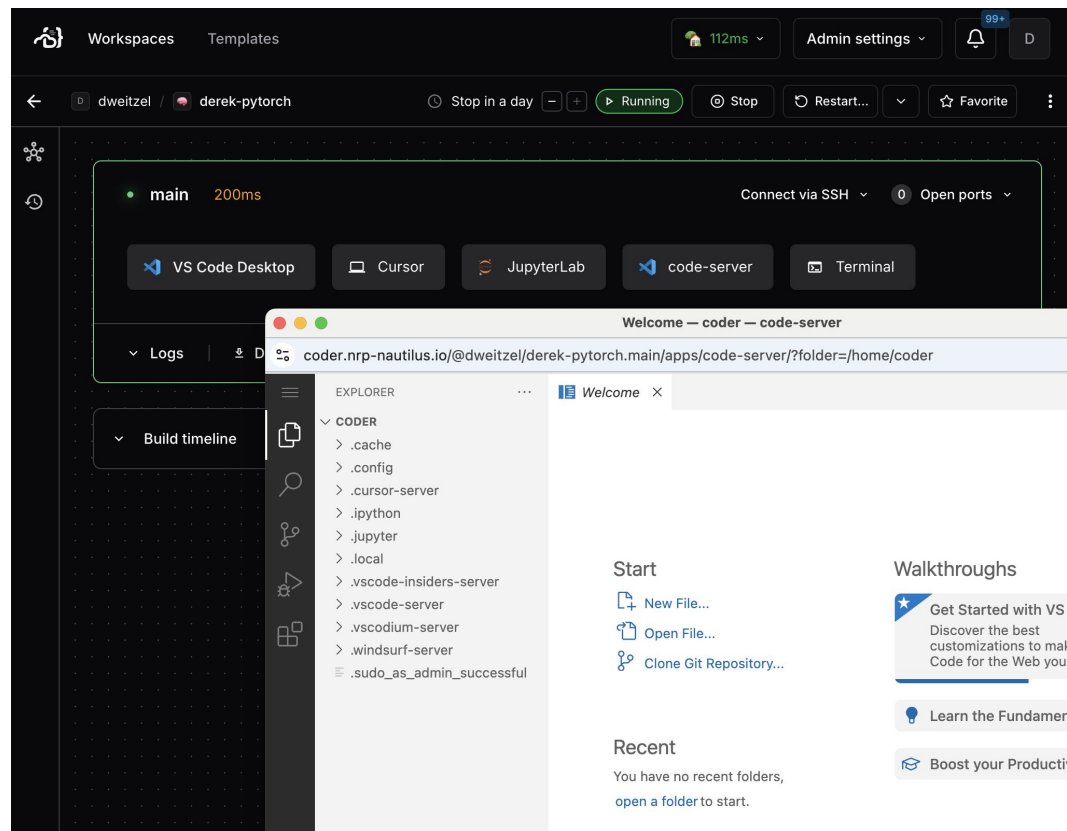
GPU Type	Quantity
NVIDIA A10	279
NVIDIA A100 80GB PCIe	266
NVIDIA GeForce RTX 3090	235
NVIDIA GeForce RTX 2080 Ti	163
NVIDIA L4	96
NVIDIA GeForce GTX 1080 Ti	95
NVIDIA L40	68
NVIDIA RTX A6000	64
Tesla V100-SXM2-32GB	60
NVIDIA RTX A4000	32
NVIDIA GeForce RTX 4090	24
NVIDIA TITAN Xp	13
...	
Total	1455



Coder

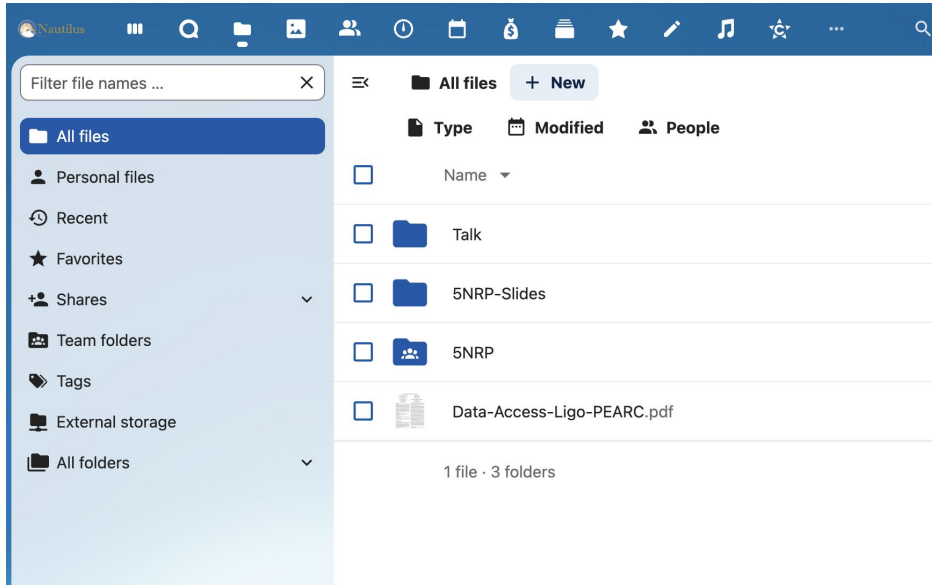
Provides an on-demand jupyter/VS Code environment.

Useful for GPU/FPGA development where you need the GPU/FPGA locally to run tests.

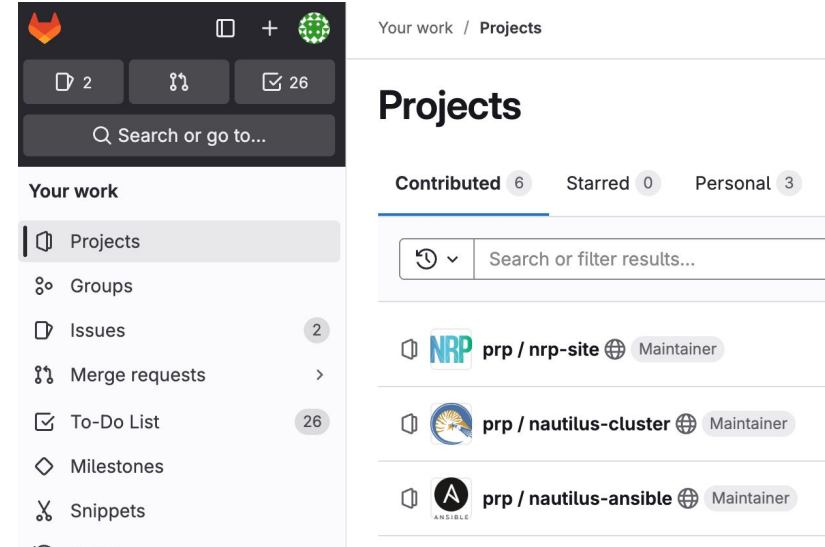




User Interface - Other Services



NextCloud



Gitlab for Code and
Container Hosting

Security

Malicious users exploited our resources to run cryptocurrency miners and unauthorized proxies, first flagged by anomalous spikes in CPU/GPU consumption.

To counter this, we now deploy Falco, a Kubernetes-native runtime security engine, which uses eBPF technology to monitor system calls and container behavior.

Falco correlates Kubernetes and container metadata (e.g., namespaces, image names) with this runtime data to significantly improve threat detection accuracy.

Falco - Performance

Falco System Resources

Type	CPU	Memory
Allocated	0.2 CPU	1.5 GB
Used	0.041 CPU	96.8 MB


Based on our informal testing, Falco's performance impact ranges from negligible to a decrease of approximately 7%, depending on the workload.

Because Falco only operates during system calls, we expect minimal impact on pure CPU/GPU-bound workloads. Furthermore, since our system's I/O is typically remote, any overhead introduced by Falco is unlikely to significantly affect workload performance.

Falco Alerts Example

Our alerting will get the message “malicious proxy software detected” if it detects a suite of proxy software. Falco automatically kills the pod instantly after detection.

Operators then investigate the pod and notify the namespace admins for remediation.

Timestamp ↑	Source	Hostname	Priority	Rule
2025/07/08 13:32:47:156	syscall		Critical	Malicious proxy software detected

Accounting and Measurements

Most of our accounting is in long time series databases such as Thanos. The user interface is largely through Observable notebooks.

Nautilus Allocated Resources

Data gathered since 09-15-23

Missing Input: Start Date

Start Date	<input type="text" value="mm/dd/yyyy"/>	<input type="button" value="📅"/>
End Date	<input type="text" value="mm/dd/yyyy"/>	<input type="button" value="📅"/>

Namespace Filter

Name:	<input type="text"/>
Institution:	<input type="text"/>
User Institution:	<input type="text"/>

Future Directions

As our user base expands, we're continuously enhancing our services.

This includes implementing a new scheduler based on Yunikorn to provide **fair-sharing and resource guarantees**.

- Resource owners will always have guaranteed access to the resources they contribute to the cluster.

Targeting additional institutions for NRP in education, such as community colleges