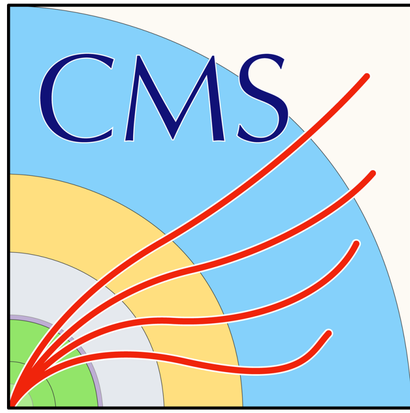
NRP NATIONAL RESEARCH PLATFORM

NRP as a Resource for US-CMS

LPC-HATS MM-dd-2026

UCSD

SAN DIEGO SUPERCOMPUTER CENTER



What is NRP?

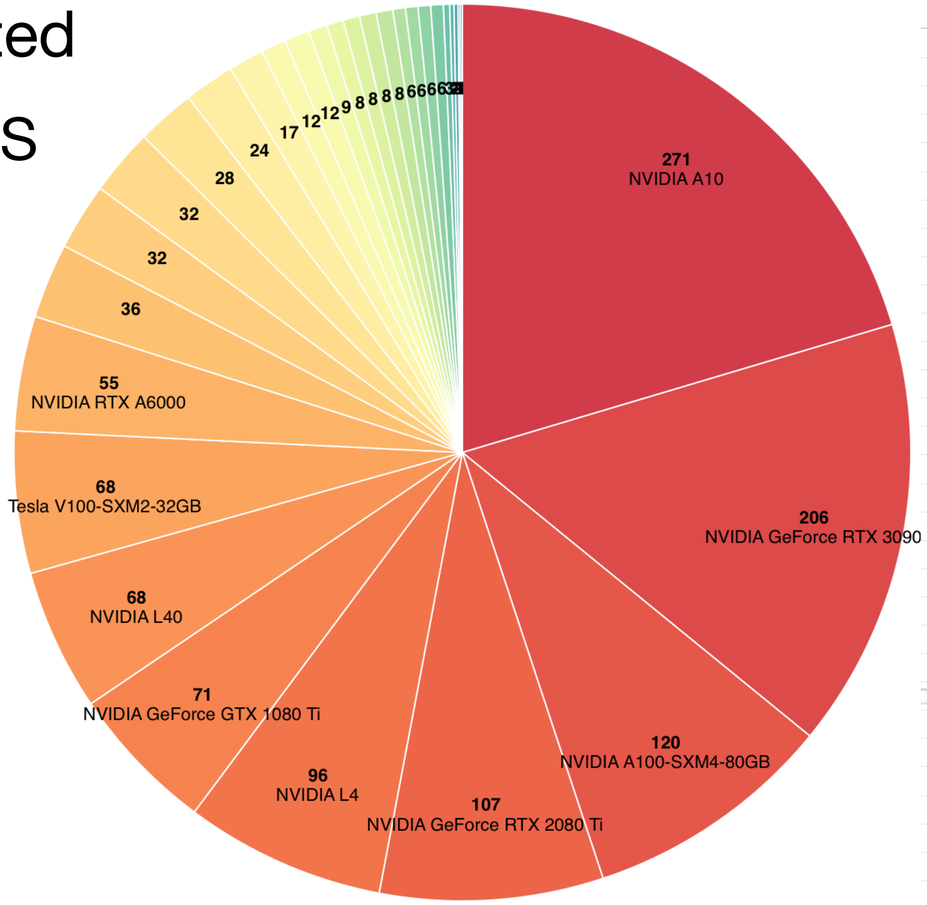
- Internationally distributed
  - Most sites are in the US
- Scale of NRP
  - ~500 compute nodes
  - ~32K CPU cores
  - ~1400 GPUs

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

NRP is a shared national cyberinfrastructure built on the Nautilus Kubernetes cluster. It provides hundreds of GPU nodes, shared storage, and managed services such as JupyterHub, S3, a vector database (Milvus), and an OpenAI-compatible managed LLM endpoint — all free for U.S. academic research, teaching, and outreach.

What is NRP?

- Internationally distributed
  - Most sites are in the US
- Scale of NRP
  - ~500 compute nodes
  - ~32K CPU cores
  - ~1400 GPUs



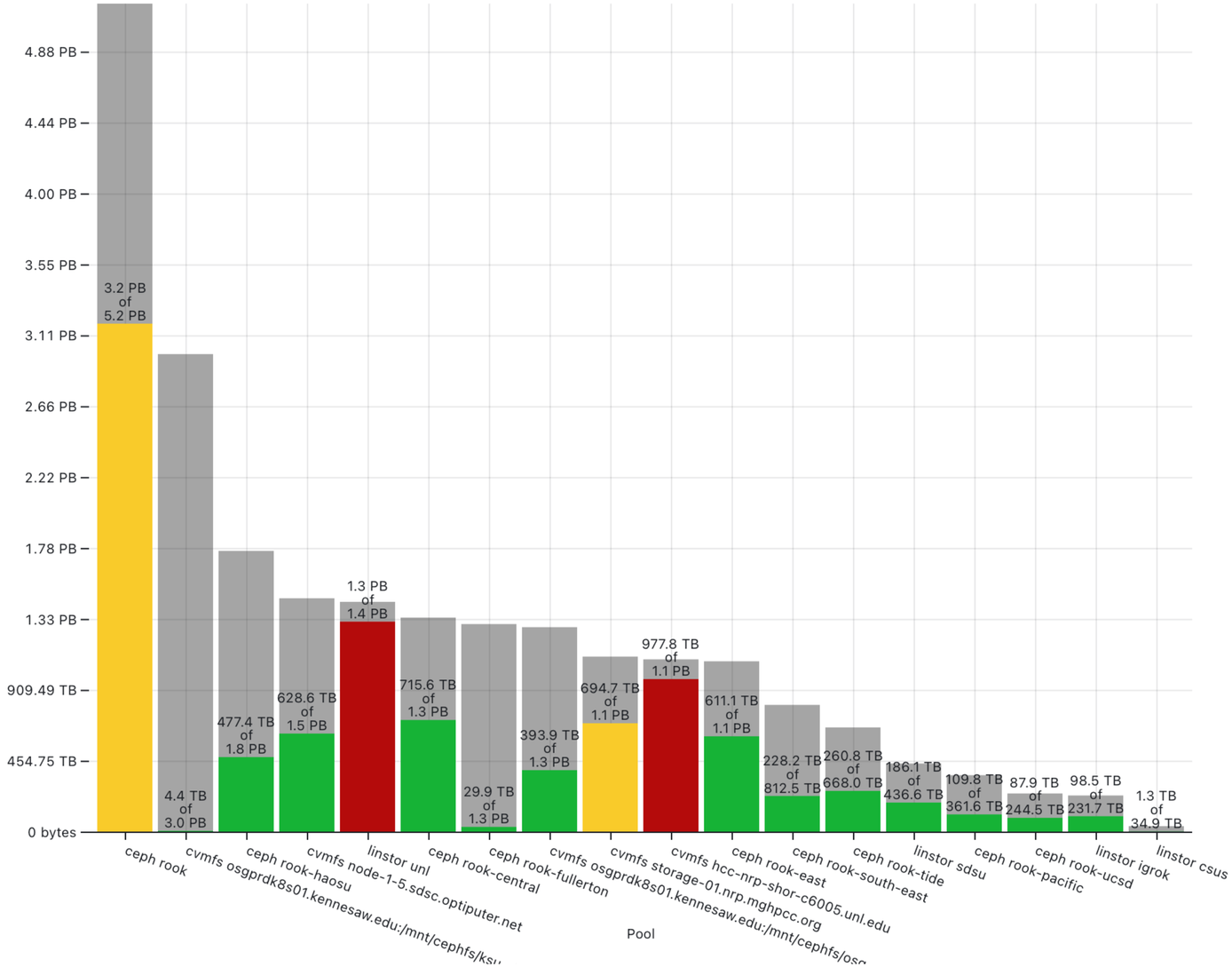
GPU Table

GPU Model	Number
NVIDIA A10	271
NVIDIA GeForce RTX 3090	206
NVIDIA A100-SXM4-80GB	120
NVIDIA GeForce RTX 2080 Ti	107
NVIDIA L4	96
NVIDIA GeForce GTX 1080 Ti	71
NVIDIA L40	68
Tesla V100-SXM2-32GB	68
NVIDIA RTX A6000	55
NVIDIA A100 80GB PCIe	36
Tesla V100-SXM2-16GB	32
NVIDIA RTX A4000	32
NVIDIA A100 80GB PCIe	28
NVIDIA GeForce RTX 4090	24
NVIDIA TITAN Xp	17
NVIDIA L40S	12
NVIDIA A40	12
NVIDIA A100-PCIE-40GB	9
NVIDIA GeForce GTX 1080	8
NVIDIA RTX 5000 Ada Generation	8
NVIDIA RTX PRO 6000 Blackwell Max-Q Workstation Edition	8
NVIDIA H100 80GB HBM3	8
NVIDIA H200 NVL	6
Tesla V100-PCIE-16GB	6
NVIDIA TITAN RTX	6
NVIDIA RTX A5000	6
Tesla T4	3
NVIDIA RTX 4000 Ada Generation	2
Quadro RTX 6000	2
Quadro RTX 8000	1
NVIDIA TITAN X (Pascal)	1

Storage Options

- Ceph File System → **Ceph FS** → RWX
- Ceph Rados Block Device → **Ceph RBD** → RWO
- **Linstor** Block → RWO
- **S3** Compatible storage → Bucket Endpoints
- Open Science Data Federation → **OSDF**

- **No Archival Storage**
  - Stale data will be deleted after 6 months if not used.



Pause for Policy

- More details at https://nrp.ai/documentation/userdocs/start/policies/
- **Golden Rule**: Do not waste resources
- **Bannable offense**: using sleep in `Jobs`
- **Utilization rules** (% of requested amount)
  - GPU: > 40%
  - CPU: 20-200%
  - RAM: 20-150%

Monitoring

https://grafana.nrp-nautilus.io

- **Prometheus**: Scrapes metrics from nodes and cluster services.
- **Exporters**: node-exporter, ceph-exporter, kube-state-metrics, and custom exporters for FPGA/DPU telemetry.
- **Thanos**: Federates Prometheus data for global querying, compacts metrics, and stores them in object storage (Ceph/S3).
- **Grafana**: Dashboards for real-time and historical analysis.

Prometheus, Thanos & Grafana: How It Works

Collect → Store → Federate → Visualize

1. METRICS SOURCES & EXPORTERS

Applications, nodes, services expose metrics.

- Linux / Windows / MacOS (node metrics) — node-exporter
- Kubernetes (object state metrics) — kube-state-metrics
- Ceph (storage metrics) — ceph-exporter
- FPGA / DPU / Custom (telemetry) — custom exporter

...and more

EXPORTERS

Exporters expose metrics in Prometheus format at an HTTP endpoint (e.g. /metrics).

/metric

They do not push data. Prometheus pulls (scrapes) the metrics.

2. PROMETHEUS

Scrapes, stores, and queries metrics (time-series DB).

Prometheus

- Scrapes metrics from exporters on a schedule (e.g., every 15s)
- Stores time-series data locally
- Provides a powerful query language (PromQL)
- Can be queried directly by Grafana

3. THANOS

Federates, stores long-term, compacts, and provides global querying.

Thanos

- Aggregates data from multiple Prometheus instances
- Stores long-term data in object storage (Ceph/S3)
- Compacts and downsamples data for efficiency
- Provides a single global query endpoint

4. GRAFANA

Queries Prometheus or Thanos and turns data into dashboards.

Grafana

- Connects to Prometheus (or Thanos)
- Builds dashboards and visualizations
- Supports real-time and historical analysis
- Alerts and annotations

OBJECT STORAGE (Ceph / S3)

Long-term metric store

DATA FLOW

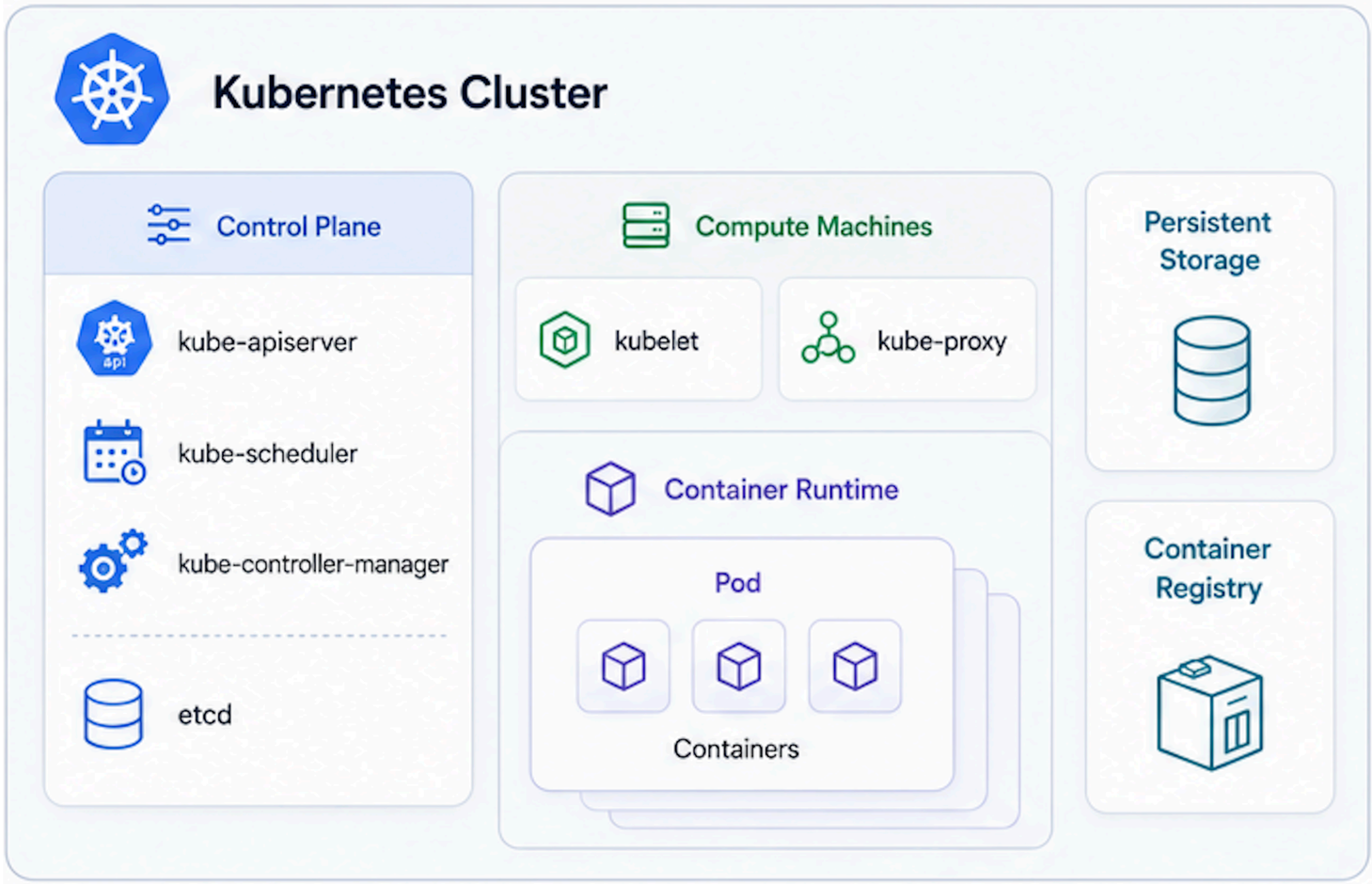
Exporters expose metrics /metrics → Prometheus scrapes (pulls) metrics → Prometheus stores and serves data → Thanos federates, stores long-term, serves global data → Grafana queries and visualizes

- Use Prometheus for fast, local metrics and alerting.
- Use Thanos for scale, long-term storage, and cross-cluster/global queries.
- Use Grafana to visualize, explore, and monitor what matters.

IN SHORT: Exporters expose metrics → Prometheus scrapes and stores them → Thanos makes them global and long-term → Grafana turns them into insights.

What is Kubernetes (K8s)?

- Kubernetes is a platform for running **containerized applications** across many machines.
- The **control plane** manages the cluster and decides where workloads run.
- **Worker nodes** run the actual applications inside pods.
- Kubernetes automatically handles scheduling, restarts, scaling, networking, and storage.
- Users request resources such as CPU, memory, GPUs, and storage instead of managing individual machines.
- **Nautilus** uses Kubernetes to support large-scale research computing, services, and specialized hardware.



K8s = K u b e r n e t e **s**

1 2 3 4 5 6 7 8

Docker/Containers

- **Docker** is a platform for building, packaging, and running applications in **containers**.
  - **Containers** are lightweight, portable, and consistent environments.
  - They include everything your app needs: code, libraries, and dependencies.
  - Docker makes it easy to run the same app on your laptop, a server, or in the cloud.
- Why Docker matters for Kubernetes:
  - Kubernetes doesn’t build containers, it runs them.
  - Docker (or other container runtimes) provides the container images that Kubernetes manages.
- NRP hosts its own **container registry** with gitlab

Kubernetes Resources

- **Pod** - smallest deployable unit in K8s. Consists of one of more containers. Ephemeral
- **Job** - Process that runs until a desired condition is met. Can launch one or multiple pods
- **Deployment** - Persistent process; complex orchestrations e.g., JupyterHub
- Persistent Volume Claim (PVC) - Represents a request for storage. PVC is a namespace scoped object, storage is cluster scoped.
  - Keep in mind that pods are ephemeral workspaces. PVCs allow you to save your work

PEOPLE & ACCESS

Cluster Admins
Manage the entire cluster and all namespaces

Namespace Admins
Manage resources within assigned namespaces

Users / Developers
Create and manage resources in their namespace

View Only
Can view resources based on permissions

Kubernetes Cluster

Cluster-scoped resources shared across all namespaces

- Nodes
- StorageClasses
- PersistentVolumes
- ClusterRoles
- CustomResourceDefinitions

Namespace: research
Namespace Admin

NAMESPACE-SCOPED RESOURCES
- Deployments
- Services
- ConfigMaps
- Secrets
- Roles & RoleBindings
- Pods

Pods

Namespace: data
Namespace Admin

NAMESPACE-SCOPED RESOURCES
- Deployments
- Services
- ConfigMaps
- Secrets
- Roles & RoleBindings
- Pods

Pods

Namespace: ml
Namespace Admin

NAMESPACE-SCOPED RESOURCES
- Deployments
- Services
- ConfigMaps
- Secrets
- Roles & RoleBindings
- Pods

Pods

RESOURCE SCOPE

Namespace-scoped
Exist within a specific namespace. Isolated between teams.
- Deployments
- Services
- ConfigMaps
- Secrets
- Roles & RoleBindings
- Pods

Cluster-scoped
Exist across the entire cluster. Shared globally.
- Nodes
- PersistentVolumes
- StorageClasses
- ClusterRoles
- CustomResourceDefinitions

Access Methods

Kubernetes

JupyterHub

Coder

- Kubernetes: Write yaml files and use `kubectl` command line tool.
- Jupyterhub*: https://jupyterhub-west.nrp-nautilus.io
- Coder*: https://coder.nrp-nautilus.io

***Note:** These are the NRP provided JupyterHub and coder, there are others. Share the url you use to access these services when asking for help