Intro to WAVE HPC

David C. Anastasiu
HPC Center Director
Santa Clara University

Agenda

1

What is an HPC?

Understanding high-performance computing fundamentals and architecture.

2

System Access

Methods to connect to the HPC (ssh, Open OnDemand).

3

Data Management

Transferring and storing training data, models, and code (scp, rsync, user space, workspaces).

4

Software Environment

Setting up required tools using module system and conda environments.

5

Job Execution

Running, testing, and debugging applications (batch jobs, interactive jobs).

6

Examples

Practical examples showing you how to use the WAVE HPC system.

Intro to WAVE HPC

What is an HPC?

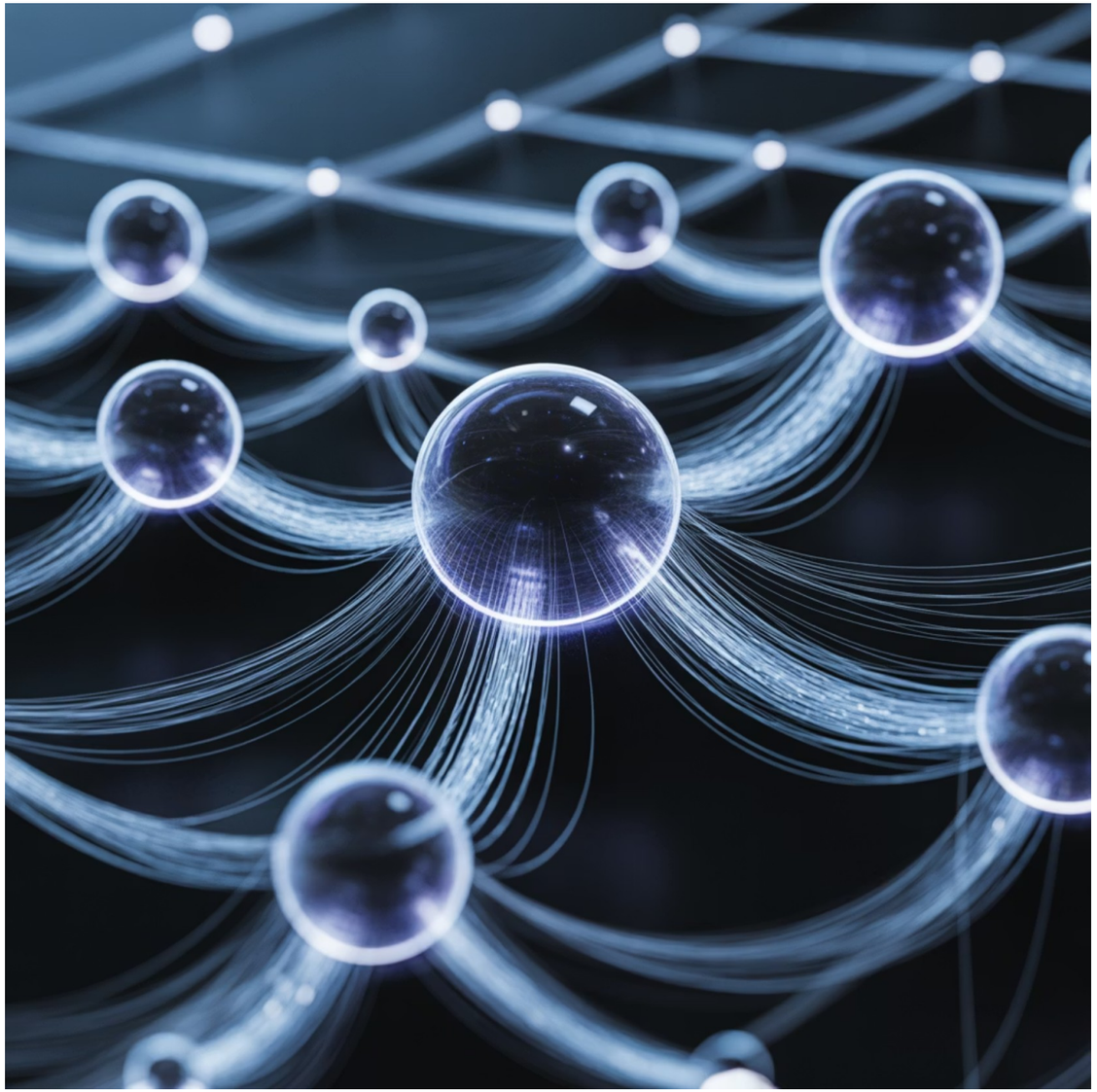
What is High-Performance Computing (HPC)?

A Team of Computers

High-Performance Computing (HPC) harnesses the collective power of multiple high-end computers working in parallel to solve complex computational problems.

Rather than relying on a single powerful machine, HPC creates a "super-team" or cluster of interconnected computers that collaborate on tasks.

This distributed approach enables researchers to tackle problems at scales previously unimaginable, forming the foundation of SCU's WAVE system.



Key Components of an HPC Cluster

Inside the WAVE System

Compute Nodes

The processing workhorses of the cluster. Each node contains multiple CPUs and potentially GPUs, working together to execute complex calculations.

High-Speed Interconnect

Specialized network infrastructure (InfiniBand) that allows nodes to communicate at ultra-fast speeds with minimal latency, critical for parallel computing tasks.

Shared Filesystem

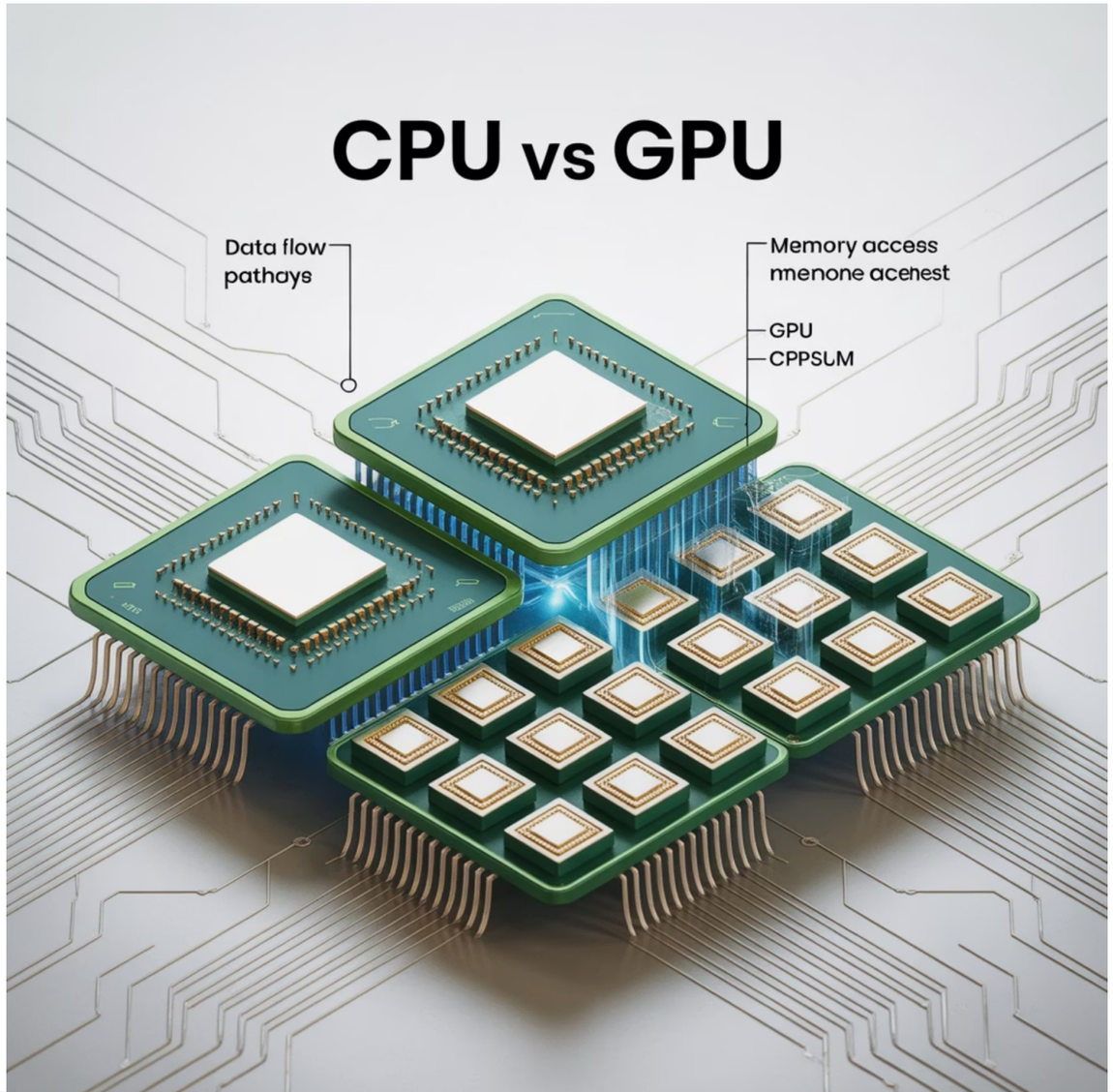
Centralized storage accessible to all nodes, enabling seamless data access and preventing redundant data transfers between computation steps.

Scheduler (Slurm)

Resource management software that efficiently allocates cluster resources, queues jobs, and maximizes utilization of the entire system.

The Power of Parallelism: Multicore CPUs and GPUs

The Workhorse of AI



CPU Architecture

Designed for sequential processing with few powerful cores and complex control logic. Optimized for diverse workloads and general computing.

GPU Architecture

Built for massive parallelism with thousands of simpler cores. Perfect for matrix operations that dominate neural network training.

Performance Gain

GPUs can accelerate deep learning workloads by 10-100x compared to CPUs, dramatically reducing training time from weeks to hours.

The Benefits Summarized

What HPC Gives You

100x

Speed Increase

Accelerate research by reducing computation time from weeks to hours

TB+

Data Scale

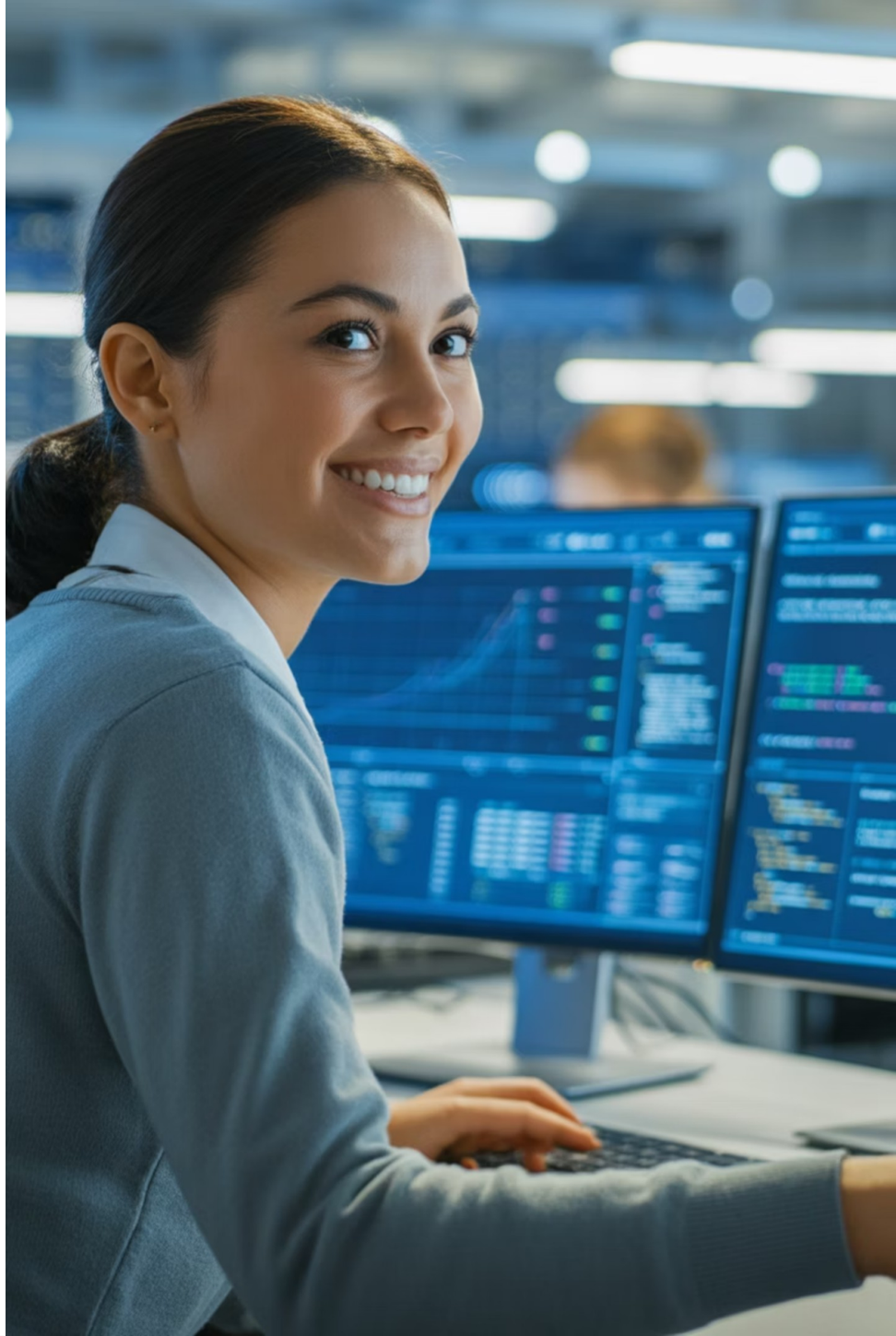
Process terabyte-scale datasets impossible on personal computers

$0

Cost Savings

Access to enterprise-grade hardware without personal investment

SCU's WAVE system empowers researchers to conduct studies that would be impossible with conventional computing resources



Hardware Tour: Compute Nodes

These nodes are the workhorses for general-purpose parallel computing.

1

CPU Nodes (cpu partition)

4 Dell servers, each with 96 Intel Xeon Platinum CPUs and 512 GB of RAM.

2

High-Memory Nodes (mem partition)

2 Dell servers designed for memory-intensive tasks, each with 96 Intel Xeon Platinum CPUs and a massive 2 TB of RAM.

3

Hub Nodes (hub partition)

8 Dell workstations primarily used by our Open OnDemand service for interactive tasks.

Hardware Tour: Accelerator (GPU) Nodes

These nodes provide the massive processing power needed for AI and ML training.

NVIDIA GPU Nodes (gpu partition)

4 Dell servers, each equipped with 2 NVIDIA V100 (Volta) GPUs, 80 Intel Xeon Gold CPUs, and 384 GB of RAM. These are the standard for most CUDA-based AI workloads.

AMD GPU Node (amd partition)

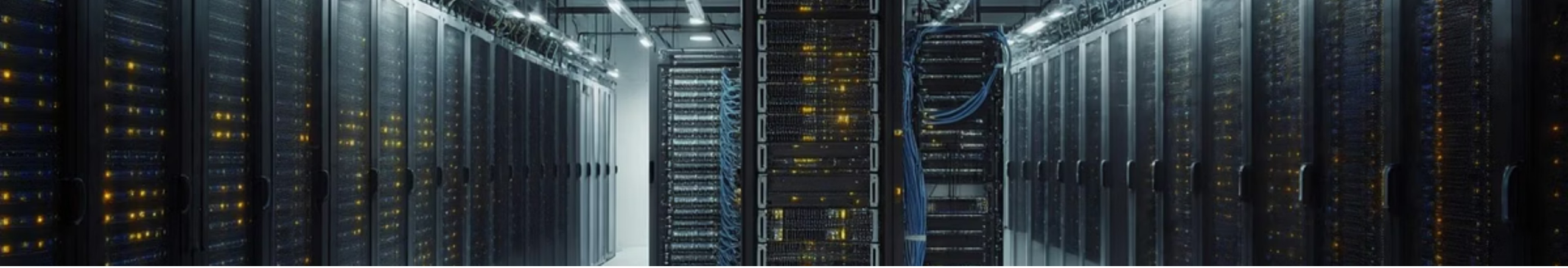
A single, powerful AMAX server featuring 8 AMD Instinct MI100 GPUs and 256 AMD EPYC CPUs, built for massive parallelism.

Hardware Tour: Condominium Nodes

The condo queue contains nodes owned by specific research labs that are available to the general community.

This provides access to a variety of cutting-edge hardware, including powerful NVIDIA A100, A16, and L40S GPUs.

Note: General-user jobs on these nodes can be preempted (stopped and requeued) if the node's owner needs the resources.



Hardware Tour: Storage & Interconnect

The components that store your data and allow the nodes to communicate at high speed.

High-Speed Storage

Flash Tier: A 92 TB Fujitsu all-flash SSD array for fast access to active data.

Capacity Tier: A 456 TB Dell EMC array for larger, long-term project storage.

High-Speed Interconnect

The cluster is linked by a 200Gb/s Mellanox Infiniband network, ensuring that data can move between nodes rapidly for large, multi-node jobs.

The Software Environment

Operating System

The entire cluster runs on Rocky Linux, providing a stable and consistent environment for research.

Job Scheduler

We use Slurm to manage all jobs. Slurm acts as a traffic controller, ensuring fair and efficient use of the WAVE's resources for everyone.

Software Management

Use Environment Modules and Conda to easily load and manage specific versions of applications, libraries, and frameworks you need for your work.

A High-Level Workflow

Step 1

Connect to a Login Node from your computer.

Step 2

Prepare a batch script that tells the scheduler what resources you need and what commands to run.

Step 3

Submit your script to the Slurm scheduler.

Step 4

Slurm finds available Compute or GPU Nodes and runs your job.

Step 5

Your results are written to the shared filesystem, ready for you to analyze.

Intro to WAVE HPC

System Access

Next Step: Log In

Your Checklist Before You Log In

Before you can connect to the WAVE system, there are three essential steps to complete. We'll walk through each one.

1. **Get an HPC Account**
2. **Install required software on your computer**
3. **Brush up on basic Linux commands**

Step 1: Requesting Your Account

1. Eligibility and Access

Faculty and Staff: All SCU faculty and staff are eligible for access.

Students: Students are welcome to use the cluster but must have a faculty sponsor to gain access.

Authentication: Accessing WAVE systems requires valid SCU authentication and DUO multi-factor verification.

2. Creating Your Account

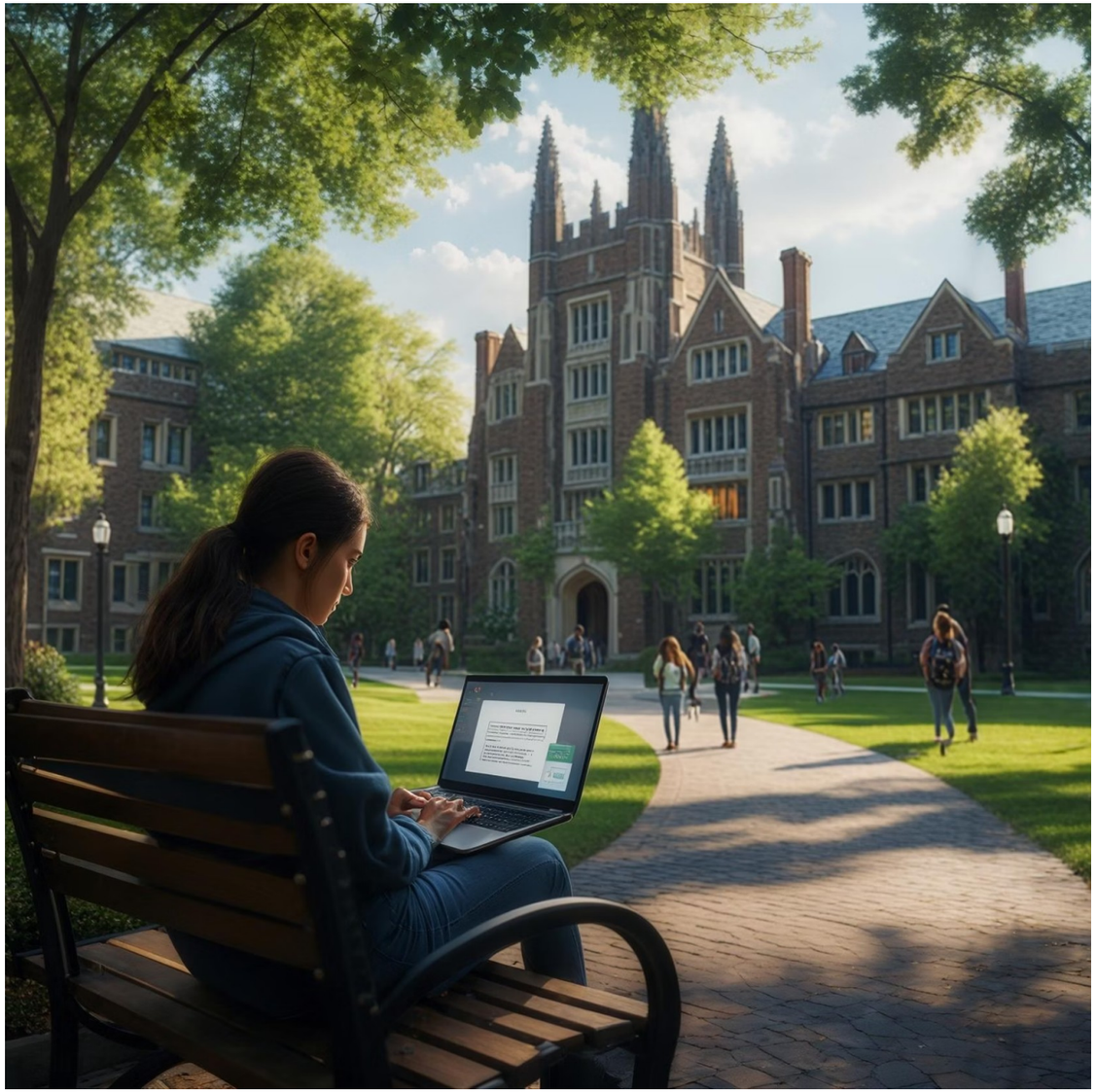
We have transitioned away from email-based requests to a more secure ticketing system. To request a new account:

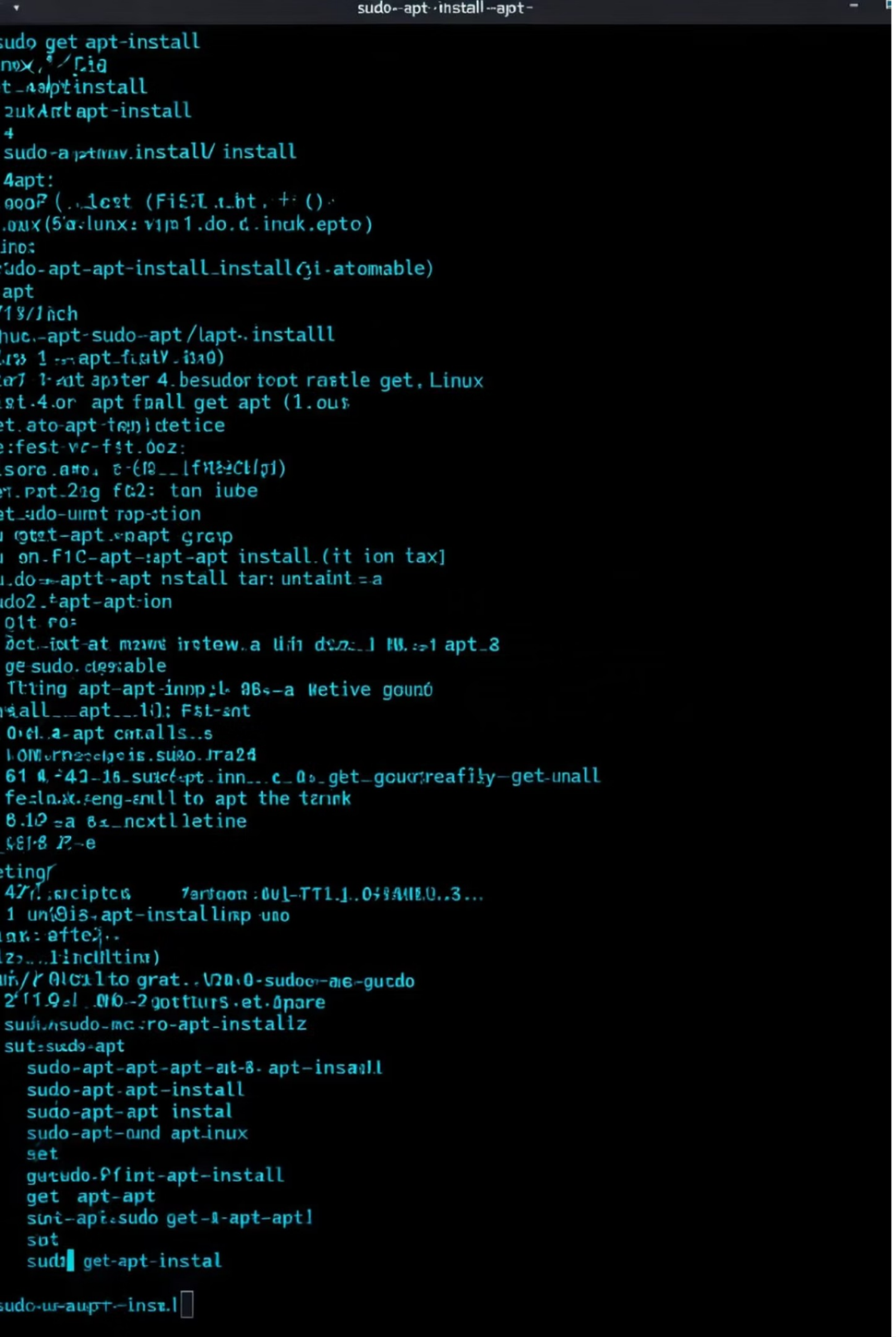
Submit a WAVE Support Request.

Log in using your SCU credentials (DUO verification required).

Select "User Account Creation" from the form options.

Fill out the form fields. If you need access for a research project, note the Faculty Sponsor for that project. If you are a faculty requesting access for a class, note the class details (e.g., CSEN 342 Deep Learning Fall 2026) and attach the class roster you obtain from Workday via the “SCU Course Roster” report.





Step 2 & 3: Software and Linux Skills

Required Software:

SSH Client: Needed for terminal access.

- **Windows: We recommend MobaXterm or PuTTY.**
- **macOS & Linux: The Terminal application is built-in.**

Web Browser: For access to Open OnDemand and other web tools.

Brush Up on Linux (Recommended):

Basic command-line skills are very helpful. We recommend one or more of the following tutorials:

- The Linux Command Line by William Shotts
- Software Carpentry: The Unix Shell
- Ryan's Tutorials

The Golden Rule of HPC

The login node is for logistics, NOT for computation.

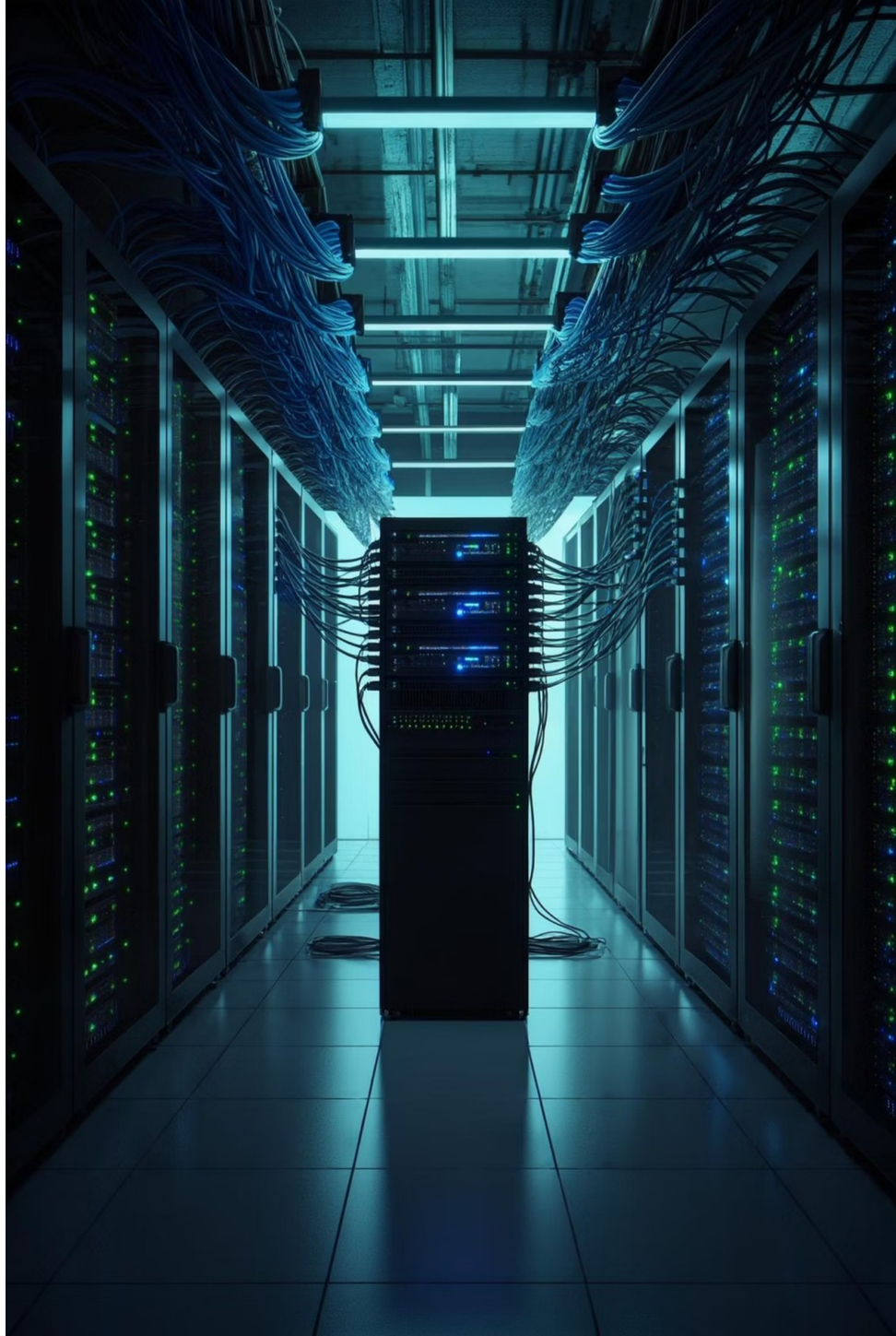
This is the most important rule for using a shared system.

✅ DO use the login node for:

- **Editing and managing files.**
- **Compiling code.**
- **Submitting jobs to the Slurm scheduler.**

❌ DO NOT run computations on the login node.

Running scripts or models here slows the system down for *everyone*, including you, and can make the system unstable. Always use a compute node for your calculations.



Where to Find Help and Information

WAVE System Documentation

For detailed user guides and FAQs, visit our official webpage:

https://www.scu.edu/wave/wave-hpc/

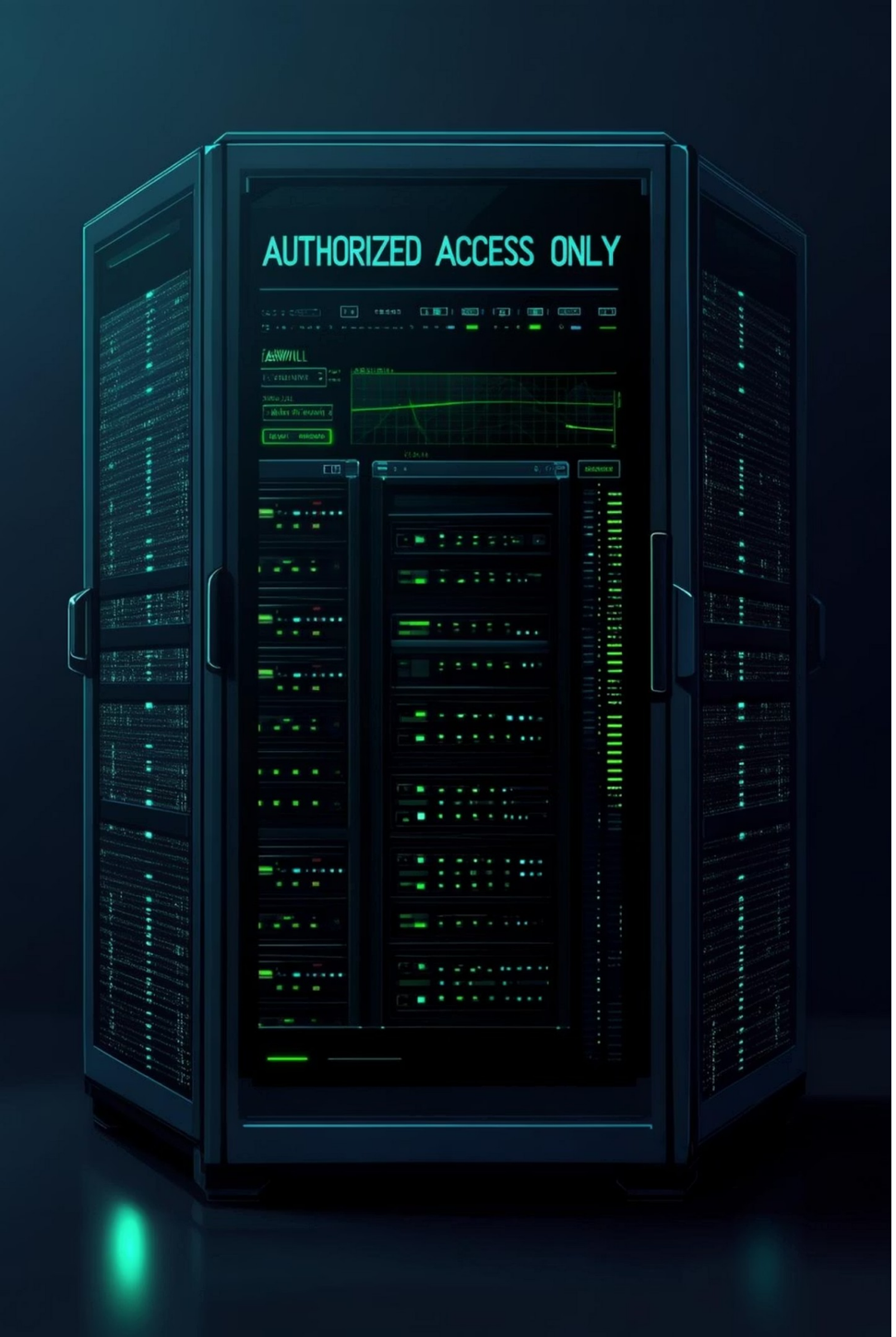
Technical Support

If you're stuck or something is broken, please email us at:

wavesupport@scu.edu

Why We Talk About Security First

- You have your WAVE account and are ready to connect.
- The WAVE system is a powerful, shared resource that houses important research data for many groups across campus.
- Protecting the system—and by extension, your data and your colleagues' data—is a critical, shared responsibility.
- This starts with ensuring secure and authorized access.



The WAVE System's Network Perimeter

Protected Environment

The WAVE cluster is not directly exposed to the public internet.

Secure Location

It lives inside the secure Santa Clara University network.

Security Practice

This is a fundamental security practice that shields the system from automated attacks and unauthorized access attempts.

Network Requirement

To reach WAVE, you must first be on the SCU network.

Your Two Paths to the Network

To access WAVE, your computer must be on the SCU network.

Option 1: On-Campus Network

If you are physically on campus, connect your computer to the eduroam wireless network or a wired Ethernet port. This places you directly on the campus network.

Option 2: SCU VPN (Off-Campus)

When working from off-campus, you must use the SCU Virtual Private Network (VPN).

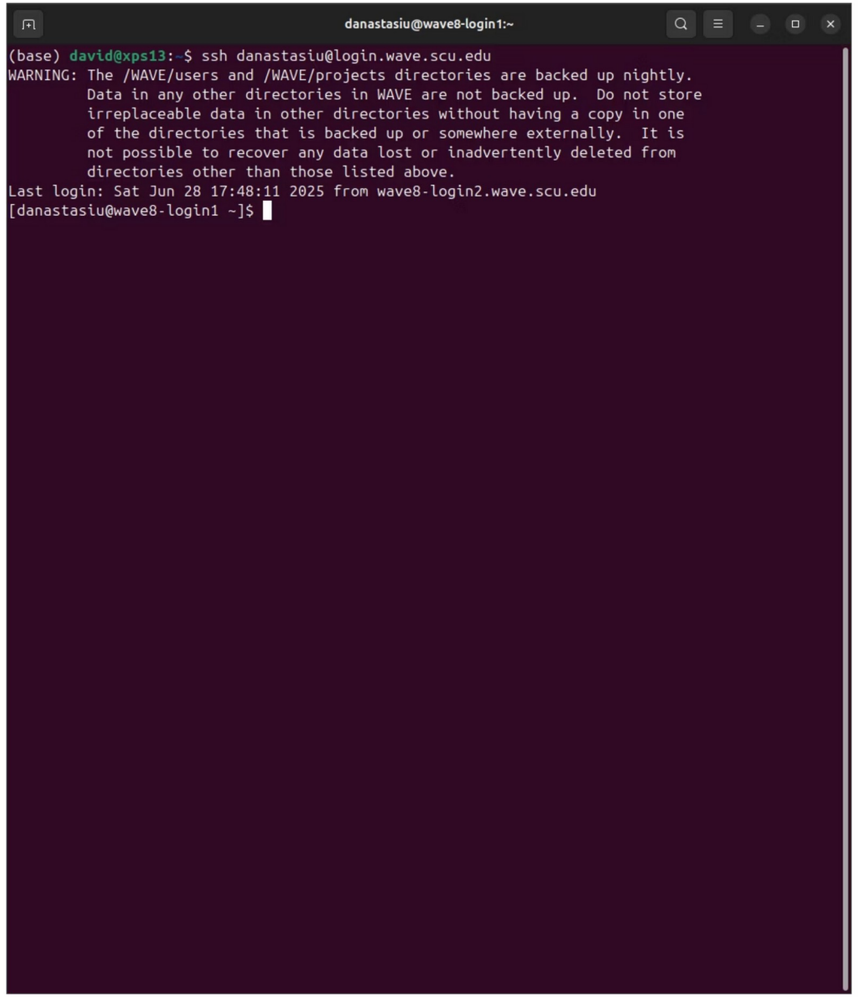
Choosing Your Door to WAVE

You are on the SCU network (physically or via VPN) and ready to log in.

WAVE offers two primary methods for connecting, each suited for different tasks.

We will cover:

- **SSH: The command-line interface.**
- **Open OnDemand: Web-based interactive notebooks and remote OS access.**



Method 1: SSH (Secure Shell)

What it is:

The standard, most direct way to get a secure command-line terminal on a remote Linux system like WAVE.

Best for:

- **Managing files and directories.**
- **Editing scripts with terminal-based editors (like vim or nano).**
- **Compiling code.**
- **Submitting and monitoring jobs with Slurm.**

How to Connect:

Open your terminal (or MobaXterm) and type:

```
ssh your-username@login.wave.scu.edu
```

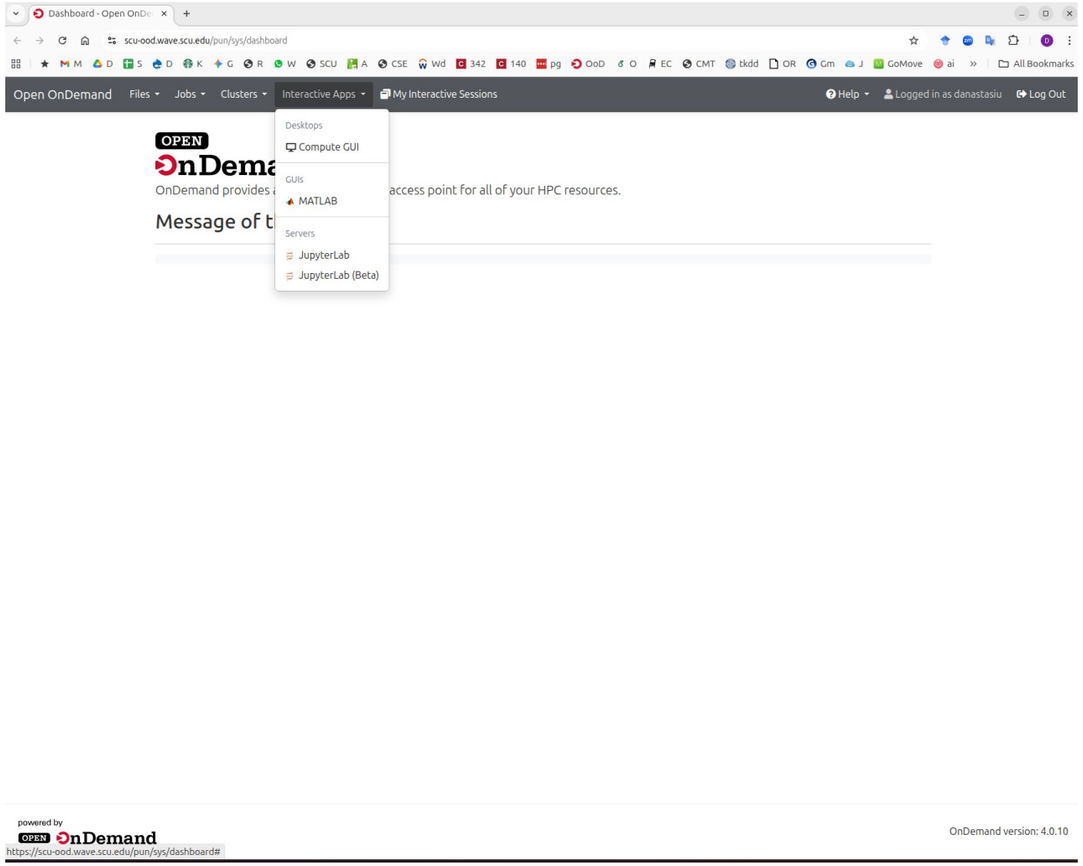
Method 2: Open OnDemand

What it is:

A web-based service that gives you a familiar Jupyter Notebook, MATLAB, or OS GUI environment running directly on the WAVE system's hardware.

Best for:

- **Interactive data exploration and prototyping.**
- **Developing code in a notebook interface.**
- **Creating visualizations.**
- **Teaching and workshop environments.**

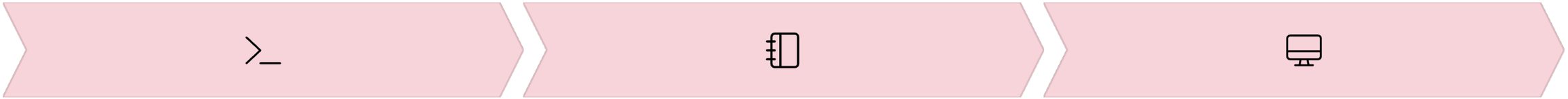


How to Connect:

In your web browser, navigate to:

https://scu-ood.wave.scu.edu/

Which Tool Should I Use?



Use SSH

When you need a terminal for submitting jobs or managing files. This is the workhorse for most traditional HPC tasks.

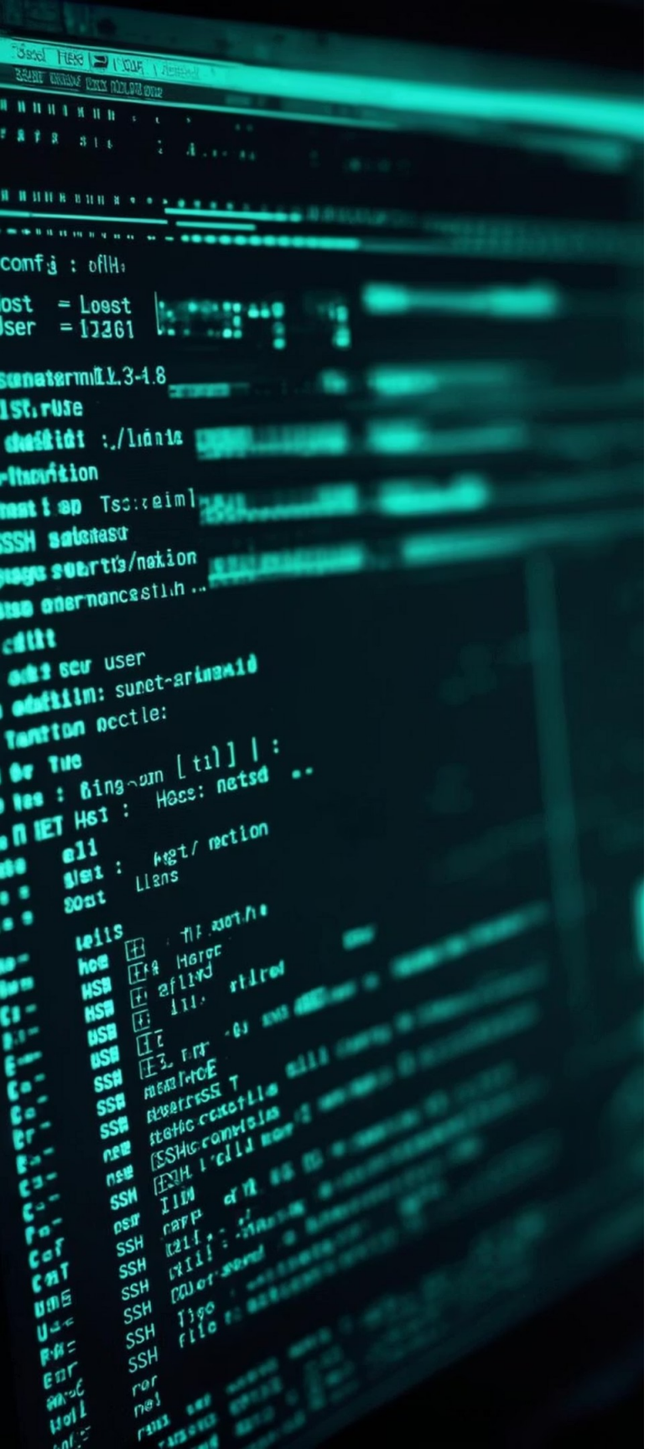
Use Open OnDemand

When you want to work interactively in a web-based notebook.

Use Open OnDemand

When you need to run a graphical program (GUI).

You will likely use a combination of these tools in your research.



The SSH Config File: Your Connection Shortcut Book 📖

1

Create Your Config File

- You can create a file on your local computer at ~/.ssh/config to store shortcuts for your connections. This saves you from typing the long address every time.

2

Add Your Template

- Add this template to the file (creating the file if it doesn't exist):

```
Host wave hpc
    HostName login.wave.scu.edu
    User your-scu-username
    GSSAPIAuthentication=no
```

3

Enjoy The Benefits

- The Benefit: Now, instead of the long command, you can simply type ssh wave. All the details (HostName, User) are read automatically from your config file.

SSH Keys: A Secure "Secret Handshake" 🤝

- An SSH key pair is a set of two cryptographic keys that are far more secure than a password.

Private Key

Lives on your local laptop. It's your secret identity and must never be shared.

Public Key

This is the "lock" that you install on a remote server like WAVE. Only your unique private key can open this lock.

- Advantage: This method is nearly impossible to brute-force, protecting your account from automated attacks.

Step 1: Generating Your Keys (On Your Local Machine)

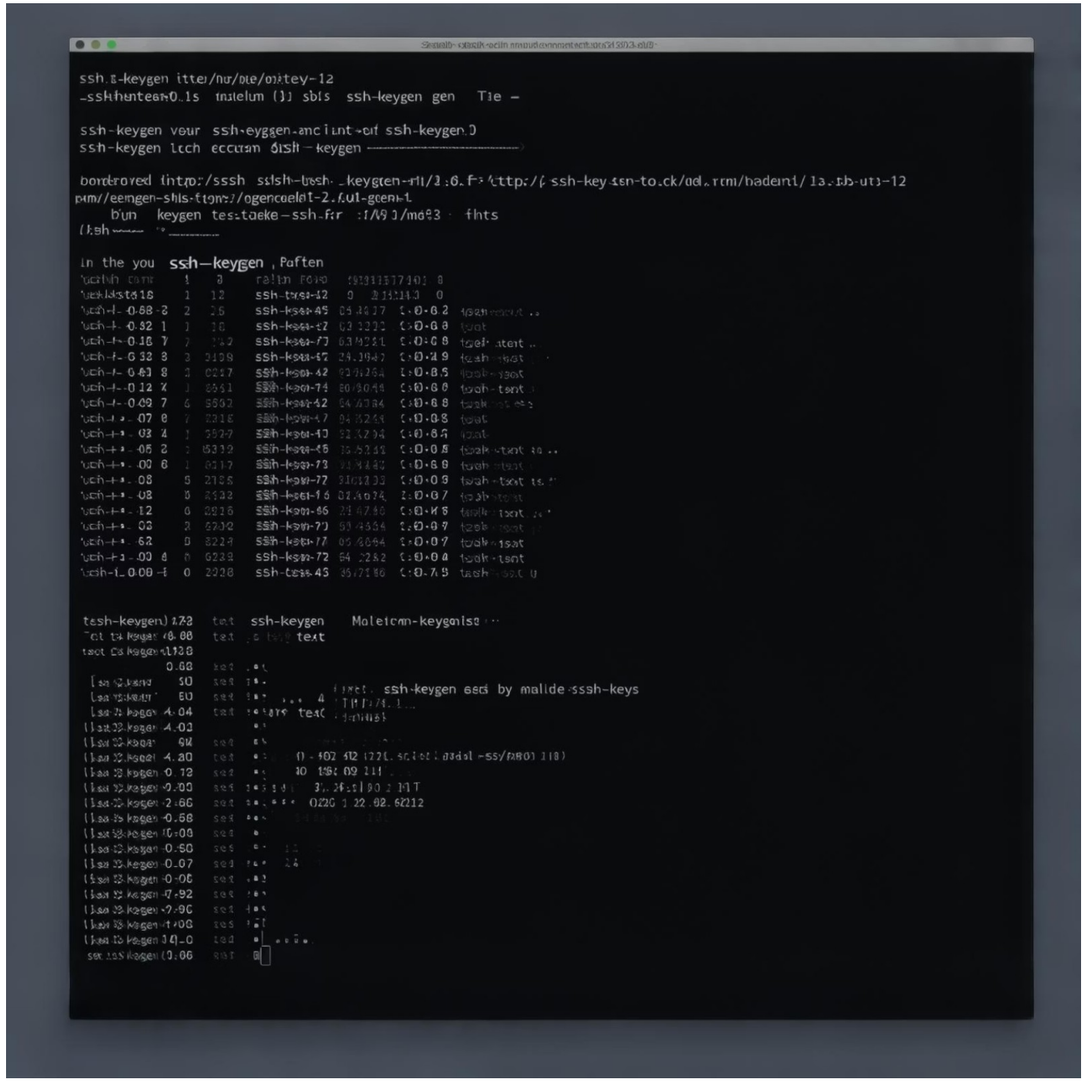
- Run the following command in the terminal on your *own* computer (Mac, Linux, or a Windows environment like WSL or MobaXterm).

```
ssh-keygen
```

- You will be asked a few questions:

"Enter file...": Press Enter to accept the default location.

"Enter passphrase...": Strongly Recommended! Enter a secure passphrase. This encrypts your private key, so even if your laptop is stolen, your WAVE account remains safe.



Step 2: Copying Your Public Key to WAVE

Use ssh-copy-id

- The easiest and most reliable way to install your public key on the WAVE system is with the ssh-copy-id command.

Execute The Command

- Because we set up the shortcut in our config file, the command is very simple:

```
ssh-copy-id wave
```

Final Authentication

- This command will prompt you for your SCU password one last time. It then automatically logs in, finds your public key, and appends it to the correct file (~/.ssh/authorized_keys) on WAVE.

The Payoff: Fast & Secure Logins

- You're all set! Now, when you connect to WAVE, it will be instant and passwordless.

- Simply type:

```
ssh wave
ssh hpc
```

- If you set a passphrase for your key, your local computer will ask for that passphrase once per session. This is more secure and often less frequent than password authentication.

Intro to WAVE HPC

Data Management

Data is the Foundation

- We now know how to run computations, but those computations are meaningless without data.

The Goal

To understand the "layout of the land" on the WAVE storage system and learn the best ways to move your data to and from the cluster.

Key Concept

Where you store your data dramatically impacts performance, determines if it's backed up, and is governed by important system policies.

WAVE's Filesystems: A Tour

WAVE provides three main storage areas, each with a different purpose. Using the correct one is critical.

$HOME directory

Your personal space for configurations and source code.

/WAVE/projects/

Your lab or class's primary space for shared datasets and persistent results.

/WAVE/scratch/

A temporary, high-performance space for intermediate files from massive computations.

Your $HOME Directory

Properties

Path: Available via the $HOME environment variable.

Quota: 50 GB

Backups: Yes, backed up regularly.

Purpose

This space is for personal configuration files (.bashrc), SSH keys, and small source code repositories.

Golden Rule: DO NOT store datasets, Conda environments, or run computational jobs from your home directory. Its performance is not suited for heavy I/O.

Your Project Directory: /WAVE/projects/

Properties

Path: /WAVE/projects/<your_project_name>

Quota: Typically 200 GB (can vary by project).

Backups: Yes, backed up regularly.

Purpose

This is the primary location for your research.

- **Store your shared lab datasets here.**
- **Save your important, final results here.**
- **This is the recommended location for your Conda environments.**

The Scratch Directory: /WAVE/scratch/

Properties

Path: /WAVE/scratch/<your_project_name> (optional, created upon request).

Quota: Varies.

Backups: NO, this filesystem is NOT backed up.

Purge Policy: Files older than 30 days are automatically deleted nightly.

Purpose: For temporary, intermediate files that can be easily regenerated. This is a high-performance workspace, not a place for permanent storage.

Pro Tip

To prevent a file from being deleted, you can update its access timestamp with the touch command:

```
touch my_important_temp_file
```

The Datasets Directory: /WAVE/datasets/

Path

/WAVE/datasets/<your_project_name> (optional, created upon request).

Quota

Varies.

Backups

NO, this filesystem is NOT backed up.

Purpose

For large datasets used by more than one member of your group or for more than one project.

Data Transfer: The Core Concept

- Data transfer commands like `scp` and `rsync` must be run from the terminal on your local computer, not after you SSH into WAVE.

- **Why?**

  Think of it like a phone call. Your laptop knows WAVE's public address (`login.wave.scu.edu`), so it can "call" the server. However, WAVE does not know your laptop's private, dynamic IP address on the campus or your home network. The server cannot "call" you back.

- Therefore, all connections must be initiated from your machine to the server.

Data Transfer Tool 1: `scp` (Secure Copy)

`scp` is great for copying single files or small directories. It uses the `source destination` syntax. Remember to use your hpc SSH shortcut.

Upload a file

```
scp my_data.csv hpc:/WAVE/projects/my_lab/datasets/
```

Download a file

(The `.` means "the current local directory")

```
scp hpc:/WAVE/projects/my_lab/results.txt .
```

Upload a whole directory

(The `-r` means "recursive")

```
scp -r my_local_folder/ hpc:/WAVE/projects/my_lab/
```

Data Transfer Tool 2: `rsync` (Remote Sync)

`rsync` is the preferred tool for large directories or repeated transfers. It's "smarter" and only copies the differences, making it much faster for updates.

Upload a directory

```
rsync -avh --progress my_project_data/
hpc:/WAVE/projects/my_lab/data/
```

`-a`: archive mode (preserves permissions, etc.)
`-v`: verbose (shows what's happening)
`-h`: human-readable numbers (e.g., MB, GB)
`-z`: compress data with zlib before transfering and uncompress at destination.
`--progress`: shows transfer progress per file.

The Trailing Slash / is CRITICAL

`source_dir/` (with slash) copies the *contents* of the directory.

`source_dir` (no slash) copies the *directory itself*.

Graphical Data Transfer Clients

If you prefer a graphical interface, you can use a free SFTP (SSH File Transfer Protocol) client. Popular options include:

FileZilla

Cyberduck

macOS, Windows

WinSCP

Windows

In the connection settings for these tools, simply use `hpc` as the host (or `login.wave.scu.edu`), your username, and your password or SSH key to connect. You can then drag and drop files.

Intro to WAVE HPC

Software Environment

Next time…