Open OnDemand User Guide

HPC Cluster @ Vogelwarte

Version 1.0 | Last Updated: December 2025 | Thx to Claude Code

Introduction

What is Open OnDemand?

Open OnDemand (OOD) is a web-based portal that provides easy access to the VoWa HPC cluster. Through your web browser, you can:

- Launch interactive applications (JupyterLab, RStudio, VS Code, Remote Desktop)
- Manage files on the cluster
- Access terminal shells
- Submit and monitor computational jobs
- Work with your research data

Accessing Open OnDemand

Portal URL: https://hpc.vogelwarte.ch

Authentication: Single Sign-On (SSO) via your Vogelwarte Microsoft/Azure AD account

Requirements:

- A modern web browser (Chrome, Firefox, Edge, or Safari)
- Vogelwarte network access (VPN if working remotely)
- Active Vogelwarte account with HPC access permissions

Getting Started

First Login

1. Navigate to https://hpc.vogelwarte.ch

2. Click the login button
3. Authenticate using your Vogelwarte credentials (same as email/Office 365)
4. You'll be redirected to the Open OnDemand dashboard

Dashboard Overview

After logging in, you'll see the OOD dashboard with:

- **Pinned Apps**: Quick access to frequently used applications
- **Files**: File browser for managing your data
- **Jobs**: View and manage your running computational jobs
- **Clusters**: Shell access to cluster nodes
- **Interactive Apps**: Launch graphical applications

Storage Access

Your home directory and shared storage are automatically accessible:

Location	Path on hpc	Path on Windows (Mac)	Purpose
Home Directory	`~/` or `/home/vogelwarte.ch/[username]`	`\\pallidus.vogelwarte.ch\[username]`	Personal files and settings
SciData	`~/SciData`	`Z:\SciData`	Shared scientific data storage (CephFS)
Scratch	`~/scratch`	`Z:\SciData\ORG_Vogelwarte\scratch`	High-performance temporary storage
Data	`/mnt/ceph`	Direct access to CephFS shared storage	

Note: The `SciData` and `scratch` directories are symbolic links created automatically in your home directory for convenient access.

Interactive Applications

Open OnDemand provides several interactive applications that run on compute nodes with dedicated resources.

JupyterLab

Description: Modern web-based interface for Jupyter notebooks, code editing, and data visualization.

Pre-installed Packages:

- Python data science stack: NumPy, Pandas, Scikit-learn
- Visualization: Matplotlib, Seaborn
- JupyterLab, IPython kernel

How to Launch:

1. Click **Interactive Apps → JupyterLab**
2. Configure your session:
   - **Account**: Select your Slurm account (usually `sci_it` or `root`)
   - **Partition**: Choose `compute` for general work
   - **Number of cores**: 1-16 (start with 2)
   - **Memory (GB)**: 1-64 (start with 4)
   - **Hours**: Maximum session time (1-72 hours)
3. Click **Launch**
4. Wait for the job to start (status: Queued → Running)
5. Click **Connect to JupyterLab** when ready

Tips:

- Start small (2 cores, 4GB RAM) and increase if needed
- Save your work frequently
- Your notebooks are saved in your home directory
- Use `~/SciData` for accessing shared datasets

RStudio Server

Description: Full RStudio IDE in your browser for R programming and statistical analysis.

Pre-installed Packages:

- **Core:** tidyverse, ggplot2, dplyr, data.table
- **Spatial:** sf, tmap, rnaturalearth, amt
- **Statistics:** randomForest, ranger, Bayesian tools (NIMBLE, JAGS)
- **Data:** RPostgres, DBI, readr, readxl
- **Visualization:** viridis, bayesplot, kableExtra
- **And many more** (see full list in role configuration)

How to Launch:

1. Click **Interactive Apps → RStudio Server**
2. Configure your session:
   - **Account**: Select your Slurm account
   - **Partition**: Choose `compute`

- **Number of cores**: 1-16 (start with 2)
- **Memory (GB)**: 4-64 (R can be memory-intensive, start with 8GB)
- **Hours**: Session duration

3. Click **Launch**
4. Wait for job allocation
5. Copy `password` because for security reason there is a temprary login
6. Click **Connect to RStudio Server**
7. Enter login `[username]` and copied temporary `password`

Tips:

- RStudio sessions use more memory than JupyterLab (request at least 8GB)
- Install additional packages with `install.packages()` (saved in your home directory)
- Use `renv` for reproducible project environments
- Connect to PostgreSQL databases using `RPostgres` package
- Parallel processing available with `foreach` and `doParallel` packages

VS Code Server (Code Server)

Description: Full-featured Visual Studio Code development environment in your browser.

Pre-installed Tools:

Python:

- Development: black, flake8, pylint
- Interactive: IPython, Jupyter
- Data Science: pandas, numpy, matplotlib, seaborn
- Utilities: requests, pytest

JavaScript/TypeScript:

- TypeScript compiler
- ESLint, Prettier
- Node.js and npm

System Tools:

- Git, vim, wget, curl
- Build tools (gcc, make)

How to Launch:

1. Click **Interactive Apps → Code Server**
2. Configure your session:

- **Account**: Select your Slurm account
- **Partition**: Choose `compute`
- **Number of cores**: 1-8 (start with 2)
- **Memory (GB)**: 2-32 (start with 4)
- **Hours**: Session duration

3. Click **Launch**
4. Copy temporary `password`
5. Connect when ready
6. Enter temporary `password`

Tips:

- Install VS Code extensions from the marketplace
- Settings and extensions persist in your home directory
- Use integrated terminal for command-line access
- Great for multi-language projects
- Git integration built-in

Remote Desktop (MATE)

Description: Full Linux desktop environment with graphical applications.

Use Cases:

- Running GUI applications (GIS tools, visualization software)
- Using applications not available in other interfaces
- Traditional desktop workflow

How to Launch:

1. Click **Interactive Apps → Desktop**
2. Configure resources (similar to other apps)
3. Choose **MATE** desktop environment
4. Launch and connect
5. Use the desktop like a regular Linux workstation

Tips:

- Requires more resources (start with 4 cores, 8GB RAM)
- Best for applications that require GUI
- Can run multiple terminal windows
- Copy/paste between your local machine and remote desktop

Resource Selection Guidelines

Choosing the right resources helps you get work done efficiently without wasting cluster capacity:

Application	Typical Use	Cores	Memory	Duration
JupyterLab	Data exploration	2	4 GB	2-4 hours
JupyterLab	Data processing	4-8	8-16 GB	4-8 hours
RStudio	Interactive analysis	2-4	8 GB	2-4 hours
RStudio	Large datasets	8-16	32-64 GB	4-8 hours
Code Server	Development	2	4 GB	4-8 hours
Desktop	GUI applications	4	8 GB	2-4 hours

Remember: You can always launch a new session with more resources if needed. Start small and scale up.

File Management

Files App

The built-in file manager lets you:

- Browse your home directory and shared storage
- Upload/download files
- Create, rename, move, and delete files/folders
- Edit text files directly in the browser
- View file permissions

Accessing the File Manager:

1. Click **Files** in the top menu
2. Choose a location:
   - **Home Directory**: Your personal files
   - **SciData**: Shared scientific data
   - Any custom path

Common Operations:

- **Upload**: Click **Upload** button, select files
- **Download**: Right-click file → **Download**

- **Create Folder**: Click **New Folder**
- **Edit File**: Click on text file to open editor
- **Move/Copy**: Select files → Use toolbar buttons
- **Change Permissions**: Right-click → **Change Permissions**

Data Transfer

Small Files (<100 MB): Use the web file manager upload/download feature.

Large Files (>100 MB): Use command-line tools via shell access:

```
# From your local machine to cluster
scp large_file.tar.gz username@hpc.vogelwarte.ch:/home/username@vogelwarte.ch/

# Using rsync for efficient transfer
rsync -avzP local_directory/ username@hpc.vogelwarte.ch:~/remote_directory/

# From cluster to local machine
scp username@hpc.vogelwarte.ch:~/results.zip ./
```

Shared Data Storage:

- Use `~/SciData` for data that needs to be shared with collaborators
- Use `~/scratch` for temporary high-performance storage
- Regular backups are performed on home directories, not scratch

---

Shell Access

Cluster Shell Access

Open OnDemand provides web-based terminal access to the cluster.

How to Access:

1. Click **Clusters** in the top menu
2. Select **Shell Access** or your cluster name
3. A terminal window opens in your browser

What You Can Do:

- Run command-line tools

- Submit batch jobs with Slurm
- Check job status
- Compile code
- Manage files with CLI tools

Session Timeouts:

- **Inactive timeout**: 5 minutes (default)
- **Maximum duration**: 1 hour (default)
- Sessions close automatically after timeout for security

Tips:

- Use interactive apps for long-running work
- For persistent sessions, use `tmux` or `screen`
- Shell ping-pong can be enabled (contact admin) for keep-alive

Basic Slurm Commands

If you need to submit batch jobs from the shell:

```
# View partition information
sinfo

# Submit a batch job
sbatch job_script.sh

# Check your job queue
squeue -u $USER

# Cancel a job
scancel <job_id>

# View job details
scontrol show job <job_id>

# View cluster usage
squeue
```

Note: Most users will use interactive apps and won't need to submit batch jobs directly.

Best Practices

Resource Management

1. **Request Appropriate Resources**
   - Don't over-request cores/memory you won't use
   - Start small and scale up if needed
   - Consider other users sharing the cluster
2. **Session Duration**
   - Choose realistic time limits
   - Terminate sessions when done (don't leave them running)
   - Save your work frequently
3. **Data Storage**
   - Home directory: Personal files, code, small datasets
   - SciData: Shared datasets, collaborative projects
   - Scratch: Temporary high-I/O work (files may be deleted)

Security

1. **Authentication**
   - Never share your credentials
   - Log out when finished
   - Use VPN when accessing remotely
2. **Data Handling**
   - Don't store sensitive data without proper permissions
   - Check file permissions for shared data
   - Follow institutional data policies

Performance

1. **Efficient Computing**
   - Close unused applications to free resources
   - Use appropriate partitions for your work
   - Optimize code before requesting large resources
2. **File Operations**
   - Use `rsync` for large transfers
   - Avoid many small file operations
   - Clean up old files and data regularly

Troubleshooting

Common Issues

Issue: Cannot log in

- **Solution**: Verify VPN connection, check credentials, contact IT

Issue: Interactive app won't start (stays in "Queued" state)

- **Possible causes:**
  - Cluster is busy (wait a bit)
  - Requested resources exceed limits
  - Requested partition doesn't exist
- **Solution**: Try reducing resources or contact support

Issue: Session disconnected unexpectedly

- **Possible causes:**
  - Network interruption
  - Session timeout
  - Cluster maintenance
- **Solution**: Reconnect; your work may be saved depending on the application

Issue: Application runs out of memory

- **Solution**: Terminate and relaunch with more memory

Issue: Can't access shared data

- **Possible causes:**
  - Permissions issue
  - Mount point not available
- **Solution**: Check file permissions, contact admin if storage mount is down

Issue: Files don't appear in file manager

- **Solution**: Refresh browser, check path, verify permissions

Getting Help

Before Contacting Support:

1. Note the exact error message

2. Record what you were trying to do
3. Check this guide and FAQs
4. Try basic troubleshooting steps

Session Information: When reporting issues with interactive apps, provide:

- Application name (JupyterLab, RStudio, etc.)
- Session ID (visible in "My Interactive Sessions")
- Time of issue
- Error messages

Support

Documentation

- **This Guide**: Comprehensive user documentation
- **Open OnDemand Official Docs**: https://osc.github.io/ood-documentation/
- **Slurm Documentation**: https://slurm.schedmd.com/documentation.html

Contact

VoWa HPC Support Team

- **Email**: scientific.it@vogelwarte.ch

What to Include in Support Requests:

- Your username
- Description of the issue
- Steps to reproduce
- Error messages (screenshots helpful)
- Application and session information

System Status

Check Cluster Status:

- Dashboard shows current cluster availability
- Maintenance windows announced via email
- Emergency maintenance posted on login page

Appendix

Slurm Accounts

Your jobs run under Slurm accounts for resource tracking:

Account	Description	Typical Use
`sci_it`	IT Science Account	General scientific computing
`root`	Root Account	Administrative or special projects

Check your accounts:

```
sacctmgr show user $USER
```

Partitions

Compute resources are divided into partitions:

Partition	Description	Typical Resources
`normal`	General computing	Standard CPU nodes

Software Environment

Containerized Applications: All interactive apps run in Apptainer (formerly Singularity) containers, providing:

- Consistent software environments
- Pre-configured tool stacks
- Isolation and security
- Reproducibility

Custom Software: Contact support if you need:

- Additional Python/R packages
- Specialized scientific software
- Custom container images
- System-wide installations

Keyboard Shortcuts

In Web Shell:

- `Ctrl+C`: Cancel current command
- `Ctrl+D`: Exit shell
- `Ctrl+L`: Clear screen
- `Tab`: Auto-complete

In File Manager:

- `Ctrl+A`: Select all
- `Delete`: Delete selected
- `F2`: Rename

In Interactive Apps: Depends on the application (JupyterLab, RStudio, VS Code each have their own shortcuts)

Changelog

Version 1.0 (December 2025)

- Initial release
- Covers JupyterLab, RStudio, Code Server, and Desktop apps
- Basic file management and shell access
- Resource management guidelines

Quick Reference Card

URLs

- **Portal**: https://hpc.vogelwarte.ch
- **File Manager**: Click "Files" → "Home Directory"
- **Shell**: Click "Clusters" → "Shell Access"

Getting Help

- Check this guide first
- Contact HPC support via [email/portal]

- Include error messages and session details

Resource Recommendations

- **Light work**: 2 cores, 4 GB, 2-4 hours
- **Medium work**: 4-8 cores, 8-16 GB, 4-8 hours
- **Heavy work**: 8-16 cores, 32-64 GB, 8-24 hours

Storage Paths

- **Home**: `~/` or `/home/vogelwarte.ch/[username]`
- **Shared Data**: `~/SciData` or `/mnt/ceph`
- **Scratch**: `~/scratch`

End of User Guide

This guide is maintained by the SciIT-Team. Suggestions and corrections welcome!

Revision #3
Created 2025-12-03 06:04:21 UTC by Stefan Hofstetter
Updated 2025-12-03 12:41:40 UTC by Stefan Hofstetter