

HPC Cluster Onboarding

Research Computing

Medical College of Wisconsin

Updated Feb 2026

Overview

- Research Computing
- Accounts & Access
- Cluster Resources
- Storage & Data Transfer
- Using Software in the Cluster
- Submit, Monitor and Troubleshoot Jobs
- Getting Help

Research Computing

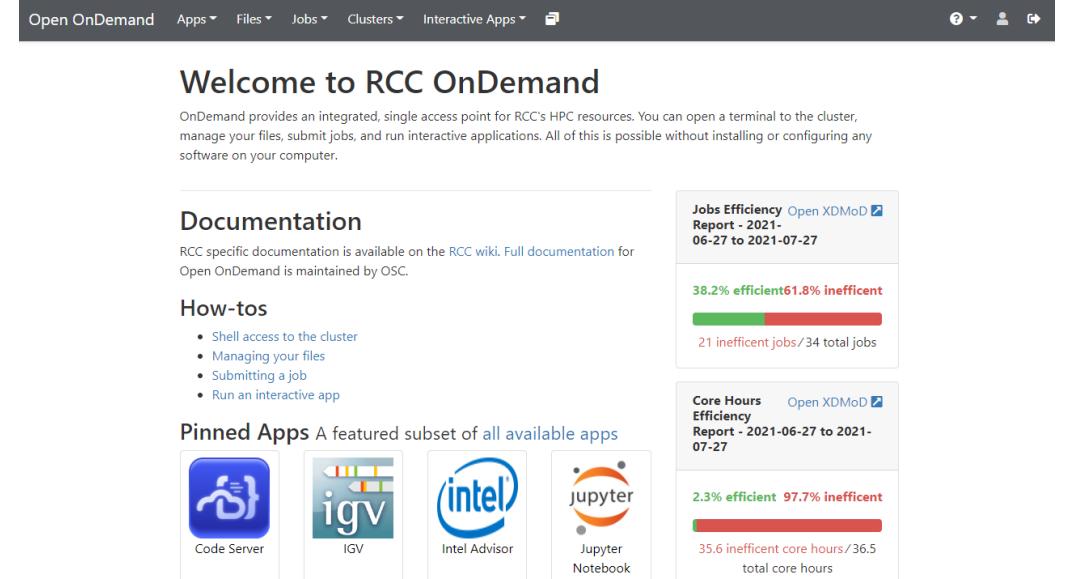
- What is Research Computing?
 - Research Computing is a division of MCW-IS that **provides campus-wide access to high performance computing (HPC)** resources that are designed for computational biomedical research.
- What is HPC?
 - HPC enables users to **tackle problems too large for a typical desktop or laptop** by leveraging high-speed compute, storage, and network resources to perform complex calculations.
- How much does it cost?
 - All MCW faculty are eligible for an allocation of free storage and unlimited computing time for their lab. [Additional storage](#) is available for fee.

Request an RCC account

- RCC user accounts are available to all MCW researchers
- Visit <https://docs.rcc.mcw.edu/user-guide/accounts/>
- Most requests are handled same day
- *Please note - to obtain RCC access, you must be a PI or sponsored by a PI and have an active MCW account. A PI may sponsor students, postdoctoral fellows, staff, or colleagues with whom they are collaborating on research. PI sponsors are required to have an RCC account.*

Access HPC Cluster

- Open OnDemand – web-based portal
 - Access cluster CLI
 - Manage files
 - Manage, write and submit jobs
 - Run pre-configured interactive apps (i.e. Rstudio, Jupyter, Remote Desktop)
 - All of these are possible without leaving your browser!
 - **This is the recommended login method for most users.** For more info, see <https://docs.rcc.mcw.edu/user-guide/access/ondemand/>
 - To connect remotely you must have VPN or use Citrix.



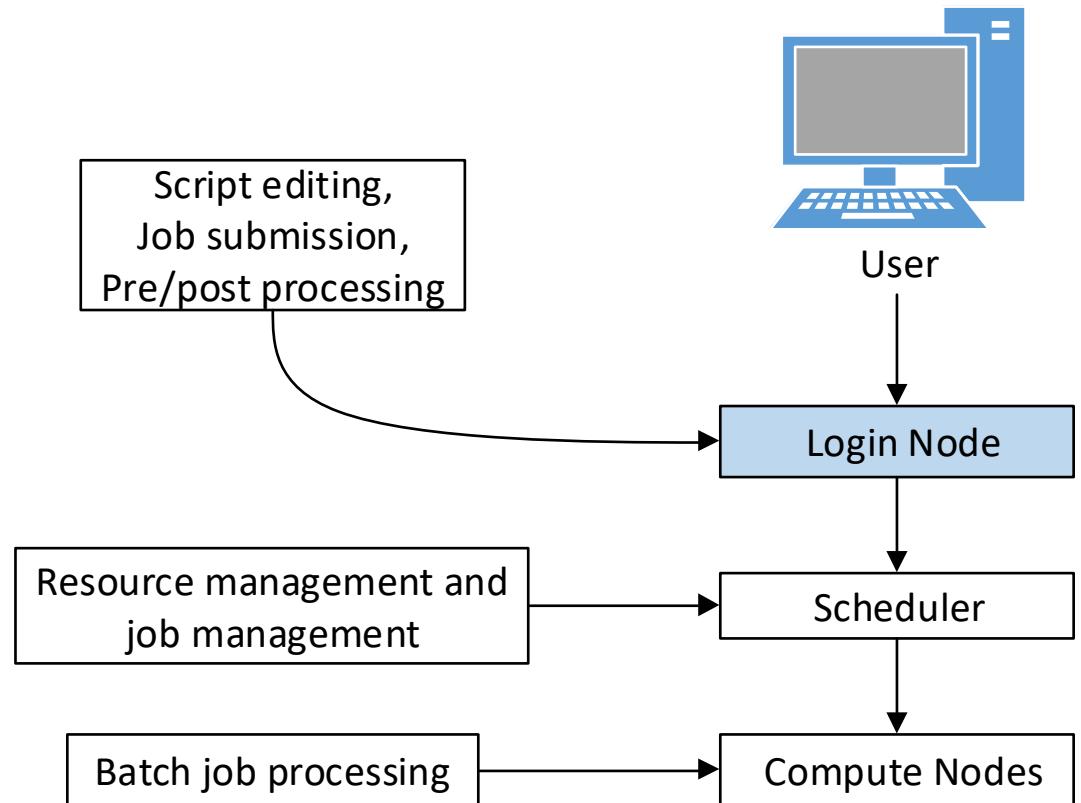
The screenshot shows the "Welcome to RCC OnDemand" page. At the top, there's a navigation bar with links for "Open OnDemand", "Apps", "Files", "Jobs", "Clusters", "Interactive Apps", and a user icon. Below the navigation, a main heading says "Welcome to RCC OnDemand" with a subtext: "OnDemand provides an integrated, single access point for RCC's HPC resources. You can open a terminal to the cluster, manage your files, submit jobs, and run interactive applications. All of this is possible without installing or configuring any software on your computer." To the right, there are two sections: "Jobs Efficiency" (report from 06-27 to 07-27) showing 38.2% efficient and 61.8% inefficient, and 21 inefficient jobs out of 34 total; and "Core Hours Efficiency" (report from 06-27 to 07-27) showing 2.3% efficient and 97.7% inefficient, with 35.6 inefficient core hours out of 36.5 total. In the center, there's a "Documentation" section with a link to the RCC wiki and a "How-to" section with links to shell access, managing files, submitting jobs, and running interactive apps. At the bottom, there's a "Pinned Apps" section featuring icons for Code Server, igv, Intel Advisor, and Jupyter Notebook.

Access HPC Cluster

- SSH client
 - Secure method of connecting to an RCC server
 - Command-line interface
 - Windows clients:
 - Secure Shell Client – [download](#)
 - Putty – <http://www.putty.org/>
 - Mac clients:
 - Built-in Terminal App
 - Iterm2 – <https://www.iterm2.com/>
 - Linux
 - Built-in terminal
- Info available on <https://docs.rcc.mcw.edu/user-guide/access/login/>

HPC Cluster Resources

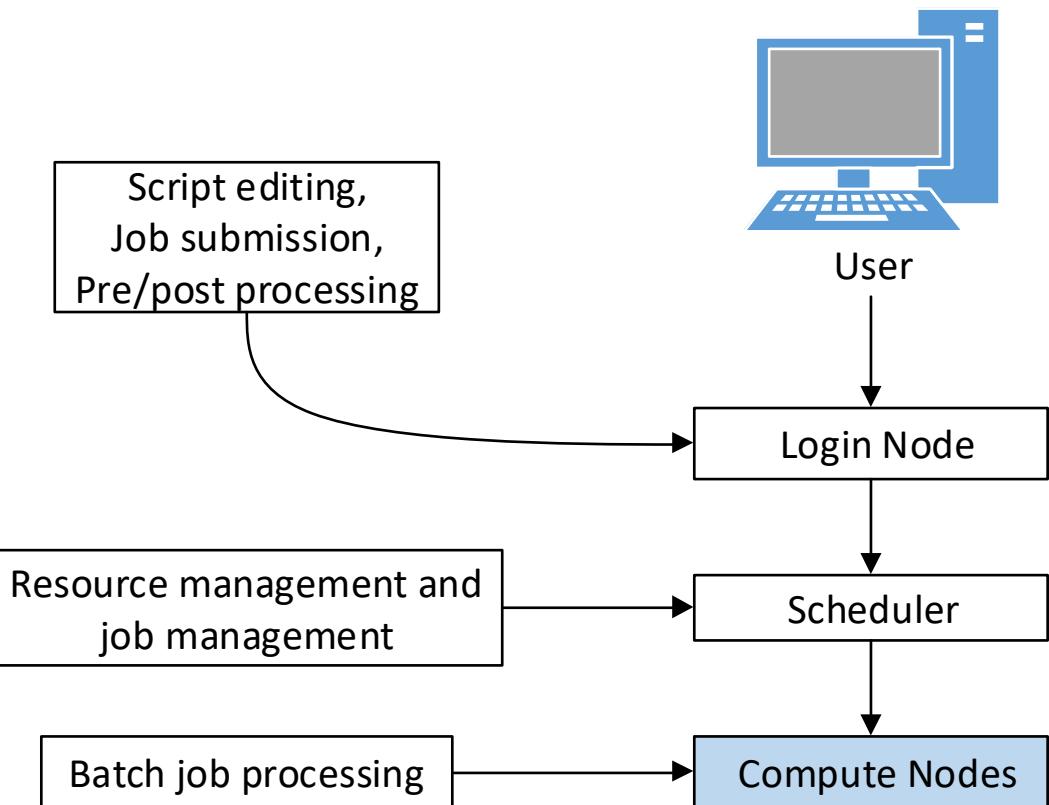
- Login Nodes
 - 4 login nodes
 - 24 cores/node
 - 128GB memory/node
- Use:
 - **Not** for computationally intensive work
 - Editing scripts
 - Submitting jobs
 - Checking the status of jobs
 - Troubleshooting jobs
 - Interactive tasks



HPC Cluster Resources

- Compute Nodes
 - Running batch jobs
 - Supports interactive and batch workload

Standard Nodes	Large memory nodes	GPU nodes
60 nodes	2 nodes	9 GPU nodes
48 cores/node	48 cores/node	48 cores/node
7.5 GB memory/core	32 GB memory/core	7.5 GB memory/core
480GB SSD	480GB SSD	480GB SSD 4 V100 NVIDIA GPUs



Available Storage

- Free to all users:
 - `/home/NetID` – 100Gb hard limit
 - `/group/PI_NetID` – 1TB free limit
 - `/scratch/g/PI_NetID` – Group scratch directory
 - 25TB temporary storage for runtime files
 - NVMe storage
 - `/tmp` – local scratch storage for active jobs
 - Local disk on compute nodes
 - Do not persist across nodes or jobs, not backed up
 - 480GB on all compute nodes
- For fee:
 - Research Group Storage - `/group/PI_NetID`
 - First 1TB is free
 - \$80/TB/year for additional storage

 You can easily find your available storage directories and current utilization on the cluster with the `mydisks` command.

```
$ mydisks
=====My Lab=====
Size  Used  Avail  Use%  File
47G   29G   19G   61%  /home/user
932G  158G  774G  17%  /group/pi
4.6T   0     4.6T   0%   /scratch/u/user
4.6T   0     4.6T   0%   /scratch/g/pi
```

Disk space will show as less than quota limit. This is due to base-2 vs. base-10 math. Rest assured, the actual limit is being enforced despite `mydisks` output

File Transfer – Command line SCP

Copy a file to the HPC Cluster:

```
scp local_file user@login-hpc.rcc.mcw.edu:/path/to/remote/target-directory
```

Copy a directory to the HPC Cluster:

```
scp -r local_directory user@login-hpc.rcc.mcw.edu:/path/to/remote/target-directory
```

Copy a file from the HPC Cluster:

```
scp user@login-hpc.rcc.mcw.edu:/path/to/remote_file /path/to/local/target-directory
```

Copy a directory from the HPC Cluster:

```
scp -r user@login-hpc.rcc.mcw.edu:/path/to/remote_directory /path/to/local/target-directory
```

File Transfer – Command line RSYNC

Copy a file to the HPC Cluster:

```
rsync -avz local_file user@login-hpc.rcc.mcw.edu:/path/to/target-directory
```

Copy a directory to the HPC Cluster:

```
rsync -avz local_directory user@login-hpc.rcc.mcw.edu:/path/to/target-directory
```

Copy a file from the HPC Cluster:

```
rsync -avz user@login-hpc.rcc.mcw.edu:/path/to/remote_file /path/to/local/target-directory
```

Copy a directory from the HPC Cluster:

```
rsync -avz user@login-hpc.rcc.mcw.edu:/path/to/remote_directory /path/to/local/target-directory
```

File Transfer – Command line RCLONE

Rclone is a command-line tool for efficiently syncing, copying and transferring files between the cluster (or local) storage and many different cloud services (like OneDrive, Google Drive, S3, Dropbox), making it ideal for moving large datasets to or from the HPC cluster through the Terminal.

It is a more advanced tool, for more details on how to configure and use it please visit [our documentation](#).

HPC Software Environment

- Most packages installed as modules
- Modules dynamically load software package environments
- Commands:
 - *module avail* – list all installed software modules
 - *module avail gcc* – list installed versions of gcc compiler
 - *module load gcc (ml gcc)* – load default module for gcc compiler
 - *module load gcc/9.3.0 (ml gcc/9.3.0)* – load specific version module of gcc compiler
 - *module list (ml)* – display your currently loaded modules
 - *module unload gcc/9.3.0* – unload module
 - *module help gcc/9.3.0* – display help information
- How do I find my software?
 - *module avail*
- Request software install – email help-rcc@mcw.edu

SLURM Job Script Format

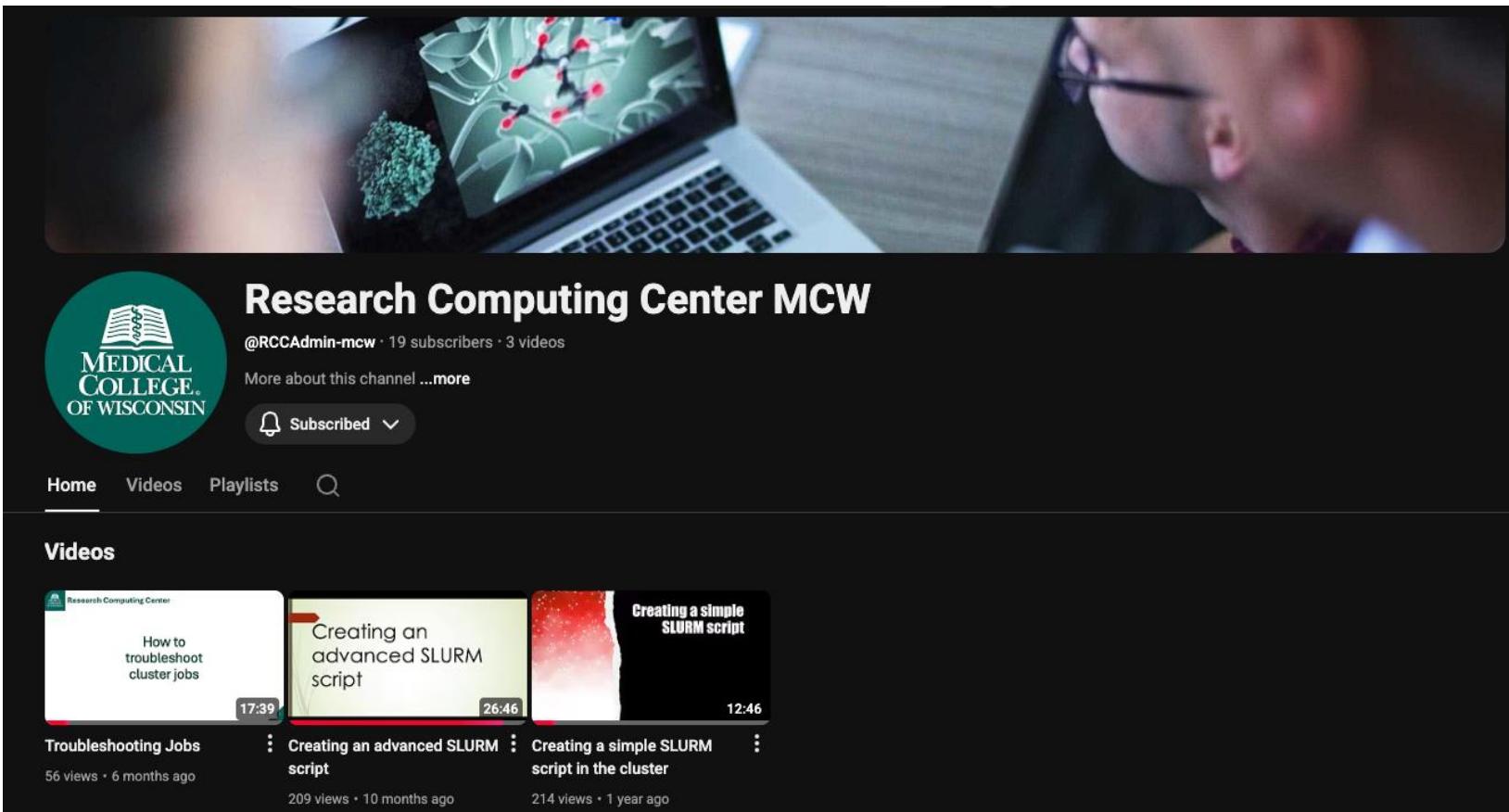
test-job.slurm

```
#!/bin/bash
#SBATCH --job-name=test-job
#SBATCH --ntasks=1
#SBATCH --mem-per-cpu=1gb
#SBATCH --time=00:01:00
#SBATCH --account=PI_NetID
#SBATCH --partition=partition
#SBATCH --output=%x-%j.out
#SBATCH --mail-type=ALL
#SBATCH --mail-user=NetID@mcw.edu

echo "Starting at $(date)"
echo "Job name: ${SLURM_JOB_NAME}, Job ID: ${SLURM_JOB_ID}"
echo "I have ${SLURM_CPUS_ON_NODE} CPUs on compute node $(hostname -s)"
```

SLURM Job Script Format

In [our YouTube channel](#) you will be able to find a video explaining how to create a simple and an advanced SLURM job:



The screenshot shows the YouTube channel page for "Research Computing Center MCW". The channel has 19 subscribers and 3 videos. The user is subscribed to the channel. The channel's logo is the Medical College of Wisconsin seal. The video thumbnails are:

- "How to troubleshoot cluster jobs" (17:39 views)
- "Creating an advanced SLURM script" (26:46 views)
- "Creating a simple SLURM script in the cluster" (12:46 views)

The channel also lists other videos:

- Troubleshooting Jobs (56 views)
- Creating an advanced SLURM script (209 views)
- Creating a simple SLURM script in the cluster (214 views)

SLURM Workload Manager

- Job scheduling system – SLURM
 - Manages available cluster resources
 - Manages job submission and runtime
- Commands
 - *sbatch* – submit a job
 - *squeue* – show status of jobs
 - *scancel* – kill a job
- [In our documentation](#) you can find links to useful SLURM guides.

Submit a Job

- Most jobs that run on HPC are batch jobs
 - Submitted with the *sbatch* command and requires a job script
 - Best method for production job as it allows you to submit many jobs and let SLURM do the work
 - No requirement that you sit and watch the command-line
- Submit the job
 - *sbatch hpc-run.slurm*
- Check the job status
 - *squeue -u NetID*
- Additional info: <https://docs.rcc.mcw.edu/user-guide/jobs/running-jobs/>
- The videos in [our YouTube channel](#) also explain how to submit jobs
- We have installed a new app in OnDemand called Open Composer that can be used to easily write and submit jobs.

Monitoring and Troubleshooting Jobs

- A little time spent here dramatically increases success rate
- Tools
 - *squeue*
 - *sacct -j JobID*
 - *scontrol show job JobID (only works with running job)*
 - Output/error files
 - SSH to compute node and run “top” command
- Make sure your job is doing what you intended
- In our YouTube video [How to troubleshoot cluster jobs](#) you will find information on how to use the tools above and investigate failing jobs.

Getting Help

- Research Computing docs – <https://docs.rcc.mcw.edu/>
 - Help documentation specific to our systems
 - Subjects include accessing resources, running jobs, troubleshooting jobs, and software package
 - FAQ – <https://docs.rcc.mcw.edu/faq/>
- Man pages
 - Manual pages are included within the Linux operating system for many commands
 - *man ls*
- Forums
 - Stack Overflow
 - Stack Exchange
 - Many others
- Send a help request to help-rcc@mcw.edu

Getting Help

- Sending a help request to RCC
 - Contact help-rcc@mcw.edu
 - Include the following when applicable:
 - Your MCW NetID (username)
 - Machine name where the problem occurred (usually at the top of output file)
 - Job number of problem job (if applicable)
 - Name of problem software package (if applicable)
 - Brief explanation of what happened
 - Steps to reproduce the issue (if applicable)
 - Any fixes you've tried (if applicable)
 - RCC may request further information
 - Please be timely in your response

Recommended

- Explore the docs – <https://docs.rcc.mcw.edu/>
- Work on being comfortable in the command line
 - <https://www.codecademy.com/learn/learn-the-command-line>
 - https://monicagiraldochica.github.io/bash_tutorial/
- Man pages
 - Linux includes manual pages for most common commands
 - Information about flags and functions of each command is presented in a uniform way
- Linux Forums – Google is your friend
- Visit [our YouTube Channel](#)

Supporting RCC

- Publications and Acknowledgement
 - For projects that have received support from the RCC, we appreciate an acknowledgement in your publication
 - Suggested:

"This research was completed in part with computational resources and technical support provided by the Research Computing Center at the Medical College of Wisconsin."
 - We're excited when our work helps result in publications. Please send us an email to let us know!

Getting Started

- Docs – <https://docs.rcc.mcw.edu/>
- New Users – HPC Quick Start
 - Request an account
 - <https://docs.rcc.mcw.edu/user-guide/quickstart/>
 - <https://docs.rcc.mcw.edu/user-guide/access/remote-access/>
 - <https://docs.rcc.mcw.edu/user-guide/jobs/running-jobs/>
- Experienced Users – SLURM Guide
 - <https://docs.rcc.mcw.edu/user-guide/jobs/running-jobs/>
- Send a help request to help-rcc@mcw.edu