

Code-Server (VS Code) on LUIS Cluster

The [Code-Server](#) app available via [Open OnDemand](#) provides a web-based [Visual Studio Code](#) environment running directly on a LUIS Cluster compute node. Code-Server offers an almost full-featured VS Code experience and allows users to work entirely in a web browser. Visual Studio Code itself is a widely used, extensible source-code editor with support for many programming languages, debugging, and integrated terminals.

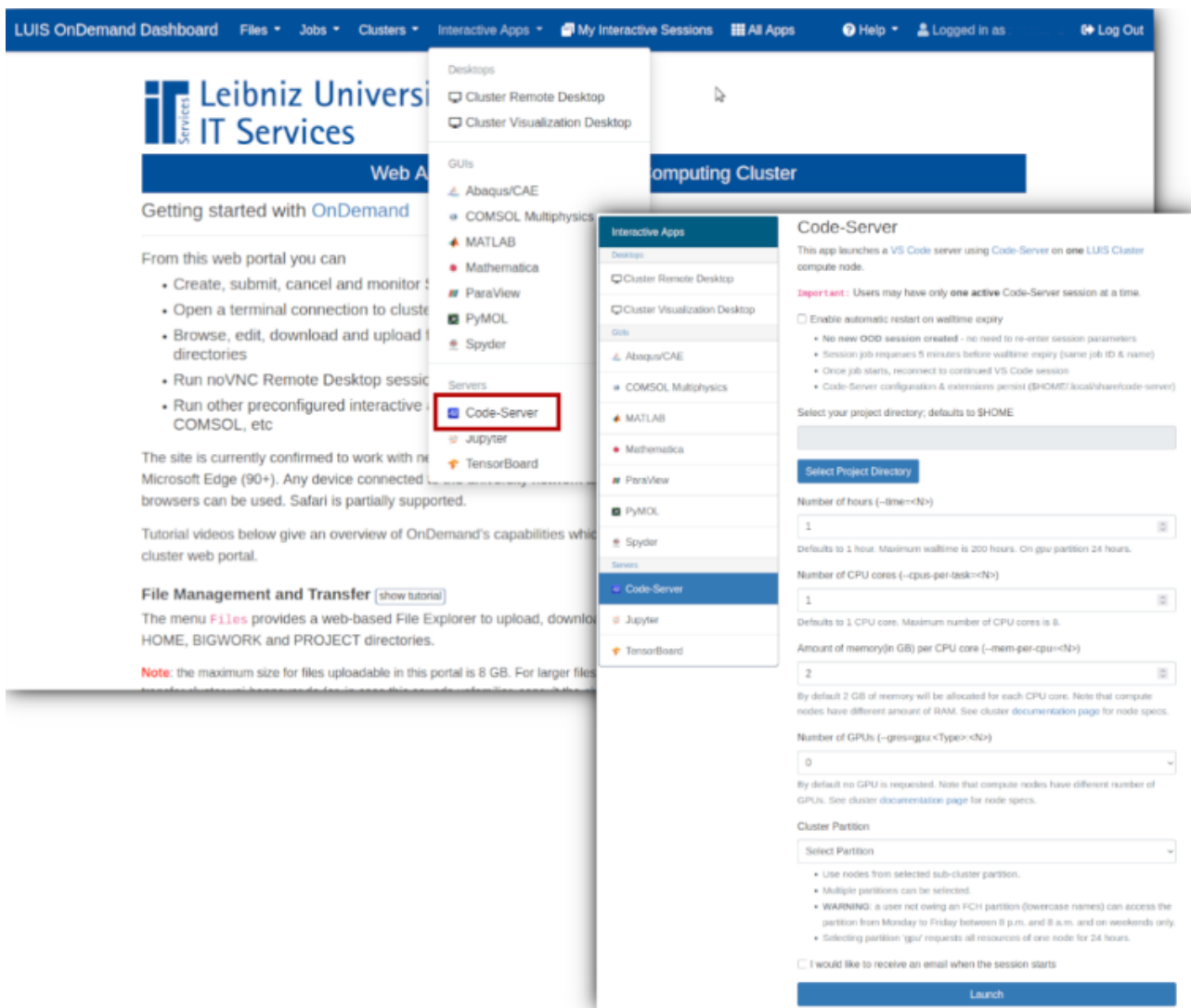


Fig. 1: Code-Server on the cluster

To start a VS Code session, navigate to the [Open OnDemand](#) dashboard and select Interactive Apps → Code-Server from the top menu. Complete the form by specifying the working directory, walltime, memory, CPUs, GPUs, and partition. The default values shown in the form will be used if left unchanged. Click Launch to start the session.

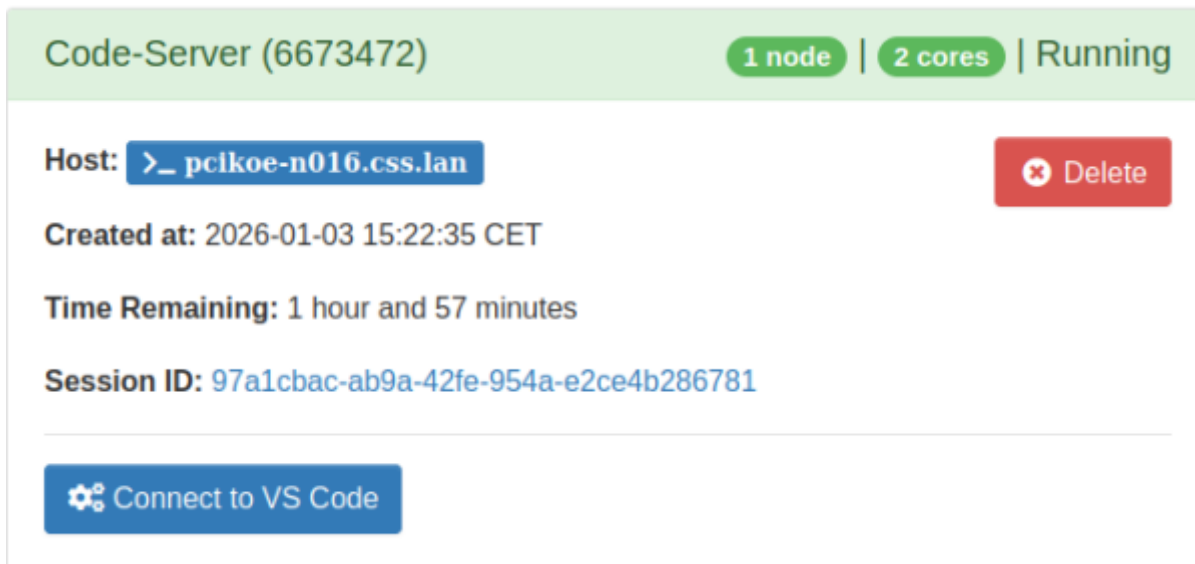


Fig. 2: Launching Code-Server

Once the session job is running, click **Connect to VS Code** to access the VS Code environment in your browser.

Why not use login nodes?

Running VS Code or Code-Server on login nodes is strongly discouraged. Login nodes are shared by many users and are intended only for lightweight tasks such as editing small files, submitting jobs, and basic monitoring. VS Code can consume significant CPU and memory resources, and LUIS Cluster login nodes enforce a strict 30-minute CPU time limit. Processes exceeding this limit are automatically terminated.

To avoid disruption and to ensure fair resource usage, always start Code-Server via Open OnDemand. This ensures that the VS Code backend runs inside a scheduled SLURM job on a compute node, while you interact with the editor through your web browser.

Single active session per user

Each user may run only one active Code-Server session at a time. Any additional Code-Server sessions launched via Open OnDemand will remain in the SLURM queue in a pending state and will not start until the currently active session has finished.

If you need to change resources (for example, GPUs, walltime, or memory), stop the running Code-Server session in Open OnDemand first and then launch a new session with the desired configuration.

Automatic restart near walltime

This deployment provides an option to automatically restart the Code-Server job shortly before the configured walltime expires. When enabled, the SLURM job associated with the active Code-Server session is requeued with the same configuration a few minutes before walltime ends. No new Code-

Server session is created.

After the job is requeued and starts running again, you may need to reconnect to the session via Open OnDemand.

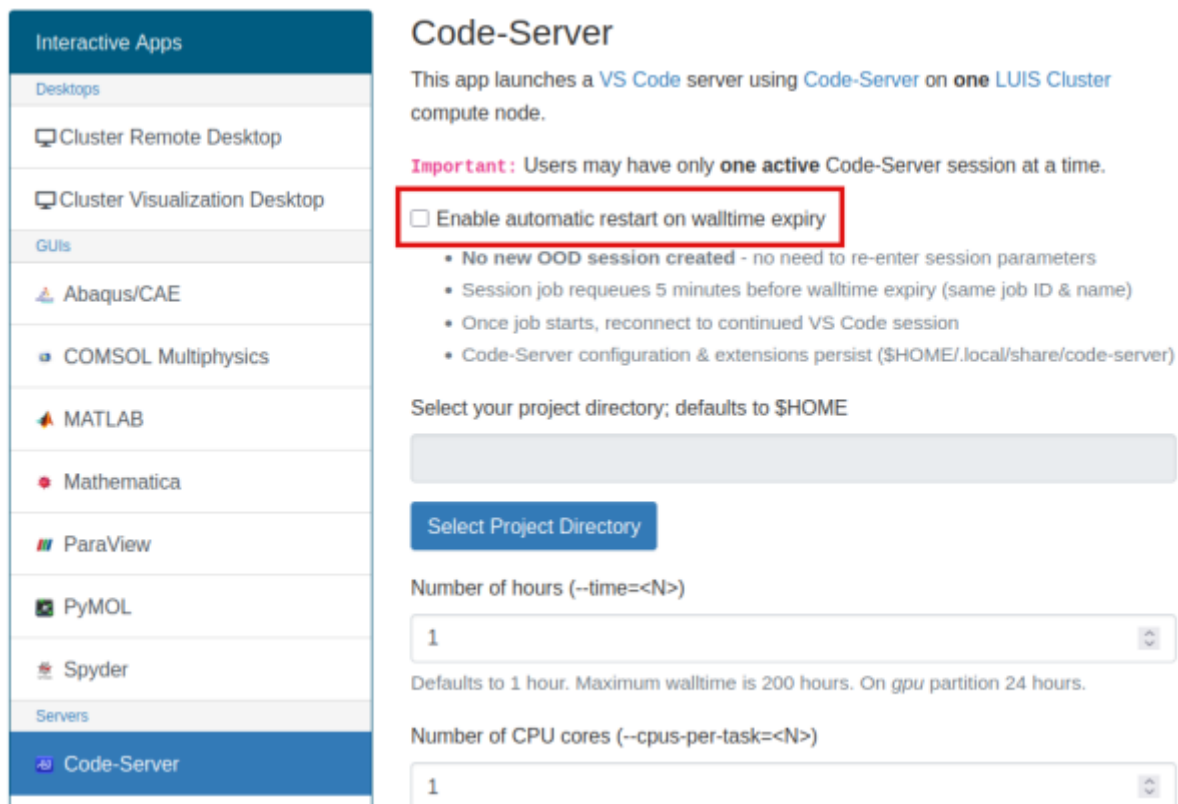


Fig. 3: Code-Server automatic restart

User data, project files, and the Code-Server configuration (including settings and extensions) are stored in the home directory. Therefore, they persist across job restarts and across new Code-Server sessions.

Best practices and tips

Choose resources carefully: Request only the CPUs, memory, GPUs, and walltime you actually need. This helps reduce queue waiting times and ensures fair usage of cluster resources.

Stop sessions when not needed: Terminate Code-Server sessions that are no longer required to free resources for other users.

Avoid running long computations inside Code-Server: Code-Server is suitable for development, testing, and interactive work. For long or resource-intensive computations, it is recommended to [submit batch jobs via SLURM](#) instead.

Monitor walltime: Be aware of the remaining walltime of your session, especially if automatic restart is disabled.

Save work regularly: Although your files persist across Code-Server restarts, it is recommended to save your work frequently.

Reconnect if your browser disconnects: The Code-Server job continues running on the compute node. You can reconnect to the session later via Open OnDemand as long as the job is still active.

From:

<https://docs.cluster.uni-hannover.de/> - **Cluster Docs**

Permanent link:

<https://docs.cluster.uni-hannover.de/doku.php/guide/soft/code-server>

Last update: **2026/01/05 11:55**

