

# RStudio Server @Ares

- [Prerequisites](#)
- [How to use on Ares?](#)
  - [Via SSH tunnel](#)
    - [Submit rstudio job to a computing node](#)
    - [Make a tunnel](#)
    - [Start the rstudio web gui](#)
    - [Stop job](#)
    - [Status of jobs and finished jobs data](#)

## Prerequisites

RStudio-Server is usually started on a localhost, which results in starting local webserver and using web browser to interact with the web gui.

On Ares we cannot easily expose the web socket to external world, as calculations are done internally on a computing node, not visible from Internet.

## How to use on Ares?

### Via SSH tunnel

The trick is to start RStudio server via a job submitted to a computing node and creating a SSH tunnel to access it on a local PC.

### Submit rstudio job to a computing node

Create the following file:

#### **rstudio-notebook.slurm**

```
#!/bin/bash
#SBATCH --partition plgrid
#SBATCH --nodes 1
#SBATCH --ntasks-per-node 6
#SBATCH --time 0:30:00
#SBATCH --job-name rstudio-server-tunnel
#SBATCH --output rstudio-log-%J.txt

## get tunneling info
XDG_RUNTIME_DIR=""
ipnport=$(shuf -i8000-9999 -n1)
ipnip=$(hostname -i)
user=$USER

## print tunneling instructions to rstudio-log-{jobid}.txt
echo -e "
Copy/Paste this in your local terminal to ssh tunnel with remote
-----
ssh -o ServerAliveInterval=300 -N -L $ipnport:$ipnip:$ipnport ${user}@ares.cyfronet.pl
-----

Then open a browser on your local machine to the following address
-----
localhost:$ipnport  (prefix w/ https:// if using password)
-----
"

module load rstudio-server/2022.12.0+353-foss-2021b-java-11-r-4.2.0

## start an rserver instance
rserver-start --www-port $ipnport
```

Save it as `rserver-run.slurm`.

Send job to queue using `sbatch` command on login node of Ares

```
sbatch rserver-run.slurm
```

Wait until your job enters running state.

To check status of submitted job using `squeue` command

```
squeue -j <JobID>
```

or all jobs of user

```
squeue -u $USER
```

which lists all current user jobs submitted to queue (\$USER - is enviromental).

Common states of jobs:

- PD - PENDING - Job is awaiting resource allocation.
- R - RUNNING - Job currently has an allocation and is running.
- CF - CONFIGURING - Job has been allocated resources, but are waiting for them to become ready for use (e.g. booting). On Ares CF state could last for up to 8 minutes in case when nodes that have been in power save mode.
- CG - COMPLETING - Job is in the process of completing. Some processes on some nodes may still be active.

## Make a tunnel

In your directory cat `rstudio-log` file:

```
cat rstudio-log-XXXXXXX.txt
```

where `'XXXXXXX'` is your sbatch job id which is displayed after you run it f.e. `'cat rstudio-log-7123485.txt'`

It will show you something like this:

```
Copy/Paste this in your local terminal to ssh tunnel with remote
-----
ssh -o ServerAliveInterval=300 -N -L 8511:172.20.68.193:8511 plgusername@ares.cyfronet.pl
-----
Then open a browser on your local machine to the following address
-----
localhost:8511 (prefix w/ https:// if using password)
-----
## You exec in another shell given command:
> ssh -o ServerAliveInterval=300 -N -L 8511:172.20.68.193:8511 plgusername@ares.cyfronet.pl
## And you are set, you can open in browser:
> `localhost:8511`
```

Exec in another shell at your local computer given command to make a tunnel:

```
ssh -o ServerAliveInterval=300 -N -L 8511:172.20.68.193:8511 plgusername@ares.cyfronet.pl
```

## Start the rstudio web gui

Open in browser: `'localhost:8511'`

## Stop job

if you wish to to end your sbatch, use `scancel <JOBID>` command, where JOBID is your tunnel JOBID you can look it up with `hpc-jobs` or `gsqueue -u $USER` commands.

```
scancel <JOBID>
```

## Status of jobs and finished jobs data

To check submitted and running jobs use `hpc-jobs` or `qsueue -u $USER` commands.

To check information about finished and historic jobs use `hpc-jobs-history` command. For example with option `"-d 30"` that command shows all user's jobs from last 30 days. More info in `hpc-jobs-history -h`.

```
hpc-jobs-history -d 30
```