Author: Remy Hoek Spaans

Pre-requisites

- Having access to a workstation (WS) and being able to connect through ssh: https://earth.bsc.es/wiki/doku.php?id=computing:workstations&s[]=workstation&s[]=remotely
- Install VScode on your laptop: https://code.visualstudio.com/download
- Familiarise yourself with: https://earth.bsc.es/gitlab/ghr/welcome_pack
    - Focus on section "2. Access WS from Visual Studio Code"
    - You can try this out yourself before the meeting if you like (optional)
- Read: https://earth.bsc.es/wiki/doku.php?id=computing:bsceshub

VS Code download

Once you have downloaded the file, open a terminal and type the following (replace filename) (ubuntu).

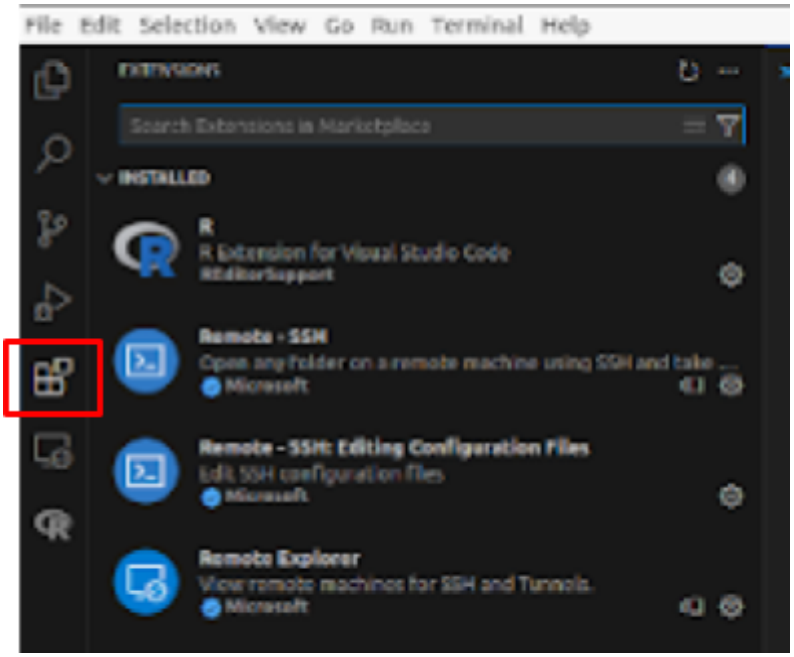
```
sudo dpkg -i filename
```

There are alternative ways to install VS Code too, using snap or through the Software Center. Choose what you feel comfortable with.

VS Code set up

See screenshot below for extensions that I have installed to be able to use and SSH connection to workstation or hub.

- R
- Remote - SSH
- Remote Explorer
- Remote - SSH: Editing Configuration Files


File Edit Selection View Go Run Terminal Help
EXTENSIONS
Search Extensions in Marketplace
INSTALLED
R
R Extension for Visual Studio Code
REditorSupport
Remote - SSH
Open any folder on a remote machine using SSH and take ...
Microsoft
Remote - SSH: Editing Configuration Files
Edit SSH configuration files
Microsoft
Remote Explorer
View remote machines for SSH and Tunnels.
Microsoft

Outline

1. What is the hub and when to use it
2. Get set up
3. Demo accessing the hub and navigating
4. Demo running r script from command line
5. Demo running r script from VScode

VIM cheat sheet
https://devhints.io/vim

GIT cheat sheet
https://education.github.com/git-cheat-sheet-education.pdf
https://about.gitlab.com/images/press/git-cheat-sheet.pdf

1. What is the hub and when to use it

You can view the hub as another workstation, but with a different environment set up. For the GHR group, the most relevant feature of this is that R-INLA is installed. However, one limitation is that you cannot use Rstudio on the hub. This means that you have to save plots and other outputs and cannot view them interactively, unless you are using VScode. If you have a gitlab or github repository set up, you can clone the repo to the hub and SSH into it via VScode. This leaves you with two options: running the R scripts via the command line or using VScode to connect.

When to use the hub? The only use case GHR currently has for using the hub is when we want to run multiple or complicated INLA models that are expected to have a long runtime. The only alternative we have at the moment is running things locally, which could interfere with other work. For scripts that do not include INLA models, your workstation is probably faster and has the benefit that you can use RStudio (though without interactive plotting). Nord3v2 can also be used to run heavier computations that do not include INLA models. In the future (March 2024?), we could potentially use Nordstream 4 to run large numbers of INLA models.

2. Get set up and try it yourself

Instructions here are
https://earth.bsc.es/wiki/doku.php?id=computing:bsceshub

There are two main things you need to get set up before being able to use the hub effectively.

1. Make sure your ssh config file is saved and set up
2. Make sure you are loading the R module

SSH config
If you used a workstation before using the hub, and you have an SSH key, it is important to import the SSH folder to your local home to use the machines with SSH with the same key.

```
# Connect to hub (replace account name with your own)
ssh -X account@bsceshub02.bsc.es

# check what files and hidden files you have
ls -a
```

Import SSH key (replace account by your bsc user name, eg rhoekspa):

```
# import ssh key
cd # make sure you are in home
ln -s /home/shared/Earth/account/.ssh .

# check what files and hidden files you have
ls -a
```

Now you have your workstation's SSH key in the hub.

Managing modules
The hub has some Earth modules installed, but not all. This is a work in progress. You can either load modules manually each time, or specify which modules you want to load automatically in a .bashrc file.

```
# to show all available module
module avail

# to show all loaded modules
module list

# show available modules that match * pattern
module av *
module av R/

module load R/4.2.1-foss-2021b
```

More information on this and advice on how to avoid conflicts can be found here:
https://earth.bsc.es/wiki/doku.php?id=tools:rtools&s[]=tools

You can specify in the .bashrc file which modules you automatically want to load.

```
# open the .bashrc file
vim ~/.bashrc

# below the user-specific environment you have to include a bit of code to load R
```

```
# or any other modules you wish to use.

if [ -t 1 ] ;
then
   module load R/4.2.1-foss-2021b;
   module load ...
fi

# you have to adapt the above code to include the modules you want and then save and exit
# the file by typing
:wq
```

Example of what Remy´s .bashrc file looks like:

```
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

# User specific environment
if ! [[ "$PATH" =~ "$HOME/.local/bin:$HOME/bin:" ]]
then
    PATH="$HOME/.local/bin:$HOME/bin:$PATH"
fi
export PATH

# Uncomment the following line if you don't like systemctl's auto-paging feature:
if [ -t 1 ] ;
then
   module load Rstudio/1.1.85-foss-2015a
   module load R/4.1.2-foss-2015a-bare
   module load GDAL/2.2.1-foss-2015a
   module load PROJ/4.8.0-foss-2015a
   module load GEOS/3.7.2-foss-2015a-Python-2.7.9;
fi
# export SYSTEMD_PAGER=

# User specific aliases and functions
```

To avoid conflicts the R-tools team recommends loading modules manually.

Choose your method and try loading an R module.

Test if you can open R on the hub

```
# launch R
R
# your location will now show as [account@bsceshub02 R]

# quit R
q()

# do not save workspace
n
```

3. Demo accessing the hub and navigating

Open the Terminal on your laptop and use the following code to connect to the hub.
If you are working from home, make sure you are connected via **VPN**.

```
# Connecting to hub (replace account name with your own)
ssh -X account@bsceshub02.bsc.es

# check what your home is
pwd
# /home/shared/Earth/account
# quota limit of 50GB per user. If you need to save data, you need to use this directory.

# check what files you have
ls -a

# You can also access other partitions such as:
/home # This folder is only for the system to save configurations. It is your local home.
/esarchive # department main data archive
/shared #intended to be accessed by specific teams or working groups

# Exercise: access one of the above and then go back to your home folder
```

If you have not used the hub before - go to the "Get set up and try it yourself" section before attempting the rest of the code.

4. Demo: running r script from the command line

For this demo we will clone a gitlab repository.
Go to: https://earth.bsc.es/gitlab/ghr/test-inla-bsc

To see the repository and use the clone button to get the link.

```
# Optional: make a folder to hold your repositories
mkdir gitlab_repos

# Clone repo (have to use https method)
git clone https://earth.bsc.es/gitlab/ghr/test-inla-bsc.git

# Navigate to the R script we want to run
cd gitlab_repos/test-inla-bsc/R

# Have a look at the script
ls
vim simple_inla_model.R

# exit vim (without saving) by typing
:q!

# Run the script
Rscript simple_inla_model.R
```

You will see the output printed in the command line, if you want to save the model or the results, you will have to specify this in the script and save the model as an .RDS file or extract the information you need and save as .csv. In this case we just closed the script without saving. It is not part of this tutorial, but of course after changing a script or creating output, you can follow your normal git workflow to add, commit and push changes.

We can also use R directly in the command line

```
# launch R
R
# your location will now show as [account@bsceshub02 R]

# the cars package is installed, so we can use it as an example
head(mtcars)

# quit R
q()

# do not save the workspace when prompted
n
```

Or set up a new R script from the command line

```
# type in command line
vim test.r
# this creates a new file and opens up immediately for you to edit

# start insert mode/editing in vim
i

# write some test code
print ("hello world")

data(mtcars)

print(head(mtcars))

plot(x = mtcars$wt, y = mtcars$mpg,
     pch = 16, frame = FALSE,
     xlab = "wt", ylab = "mpg", col = "#2E9FDF")

# once you are done editing
Esc

# save and exit
:wq

# run your code
Rscript test.r

# you can see that the plot won´t show because we have not saved it

# edit the R script so that it saves your plot and update the script to include:
png(filename="test_plot.png")
plot(x = mtcars$wt, y = mtcars$mpg,
     pch = 16, frame = FALSE,
     xlab = "wt", ylab = "mpg", col = "#2E9FDF")
dev.off()

# rerun the script and view plot
display test_plot.png
```

Other ways to view your output.

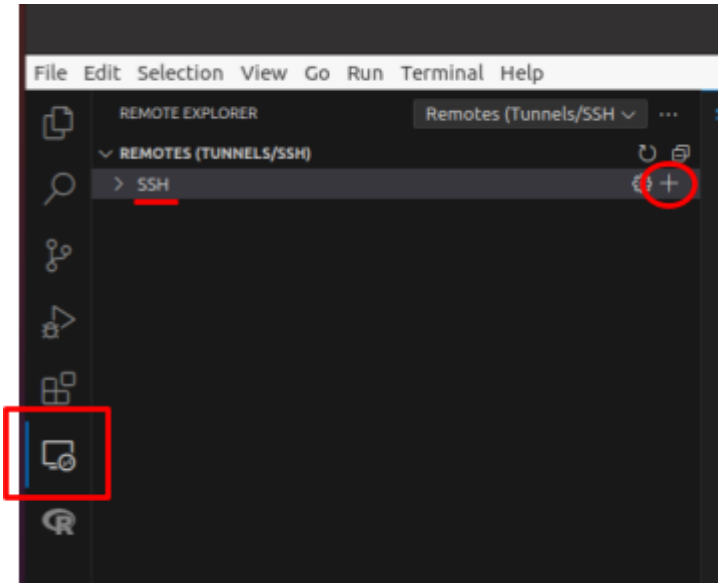
- If you are working within a gitlab, you can save, commit, push your output and then look at it in your browser. From there you can either download output from the browser, or pull results to a local clone of the repo.
- You could copy the output from the hub to your local laptop using the scp command. https://www.geeksforgeeks.org/scp-command-in-linux-with-examples/

5. Demo: running r script from VS Code

Make sure that you meet the pre-meeting requirements (start of doc):

- Installed VScode on your laptop
- Installed the required VScode extensions

Open VS Code and select the Remote Explorer tab. You will see the option to create an SSH connection, click on the + sign.



A text box appears at the top of the screen. Follow the instructions and type the ssh command to connect to the hub.

```
ssh -X account@bsceshub02.bsc.es
```

You should be prompted for a password, enter the password at the top center of the screen. Once you are connected, you can see that a blue square in the bottom left of your screen shows your SSH connection.

Further work

There is also an option to access the hub graphically, for which you have to follow instructions on here:
https://earth.bsc.es/wiki/doku.php?id=computing:machinesgraphicallyremotely

FAQ