ISARIC Outbreak Data Analysis Platform

This document provides information for users of the ISARIC data analysis platform hosted in EPCC at the University of Edinburgh.

Revision	Author	Changes
2021-12-06	abrooks	Tidied up the sections on R.
2021-11-29	abrooks	Updated screenshot showing menu entries

Contents

Introduction

The data analysis platform consists of several components:

- A database in the National Safe Haven where it is safe to store personally identifiable health data.
- A database and file storage outside the National Safe Haven for the storage of data which is not personally identifiable
- Processing systems which can operate safely on the personally identifiable data within the Safe Haven to link with other datasets, produce aggregated reports or to de-identify the data for further use.
- Access to desktops for approved researchers to work on the de-identified data.
- Access to High-Performance Computing (HPC) systems, Ultra2 and Eddie, for working with large datasets or the data which are not personally identifiable
- Access to desktops for deploying a web application for reporting

Terminology

- ODAP – Outbreak Data Analysis Platform, encompasses all of the above for the purposes of processing ISARIC and related datasets
- FCS – Flexible Compute Space, the systems which lie outside the National Safe Haven
- PDA – Protected Data Access environment, the technical name for the FCS
- Ultra2, SDF-CS1 – both names refer to the High Performance Computer accessed from the FCS
- EIDF – Edinburgh International Data Facility, the organization within EPCC which looks after the HPC and other systems
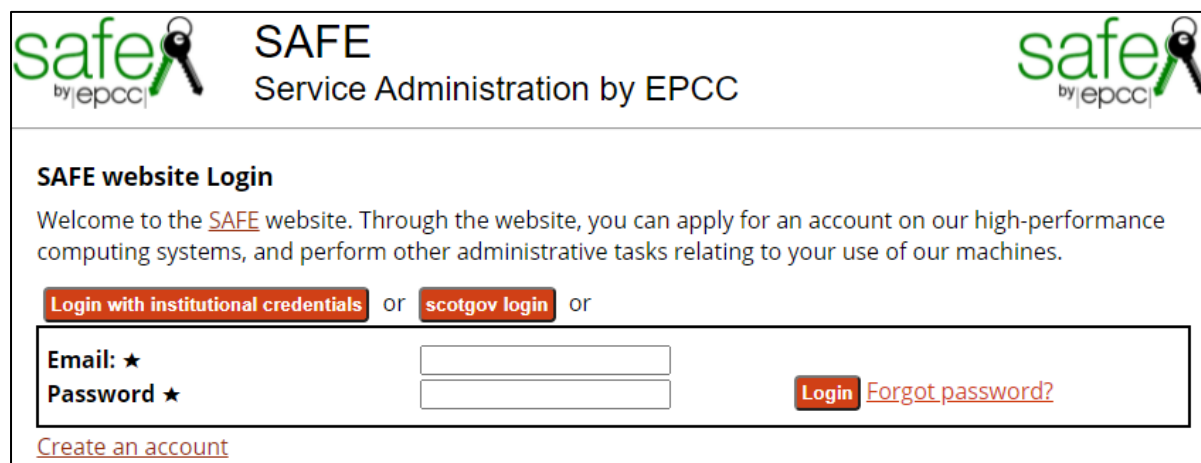
TL;DR

Quick summary:

- Register for an account in SAFE, then apply within SAFE to join project u036 (Ultra PDA).
- Wait for your application to be approved and for your VDI account credentials to be sent to you.
- Login to the Ultra VDI service https://eidf.epcc.ed.ac.uk/eidf01/ using the VDI credentials
- Select the c19-desktop (SSH) option, login using the u036 account, change your password, logout.
- Select the c19-desktop (RDP) option and login using the u036 account with new password.
- Inside this desktop you can SSH to ultra2, and you can use RStudio and PyCharm IDEs.
- Follow the guide to use Anaconda, and to use RStudio or PyCharm in "remote" mode.

Procedure for Gaining Access

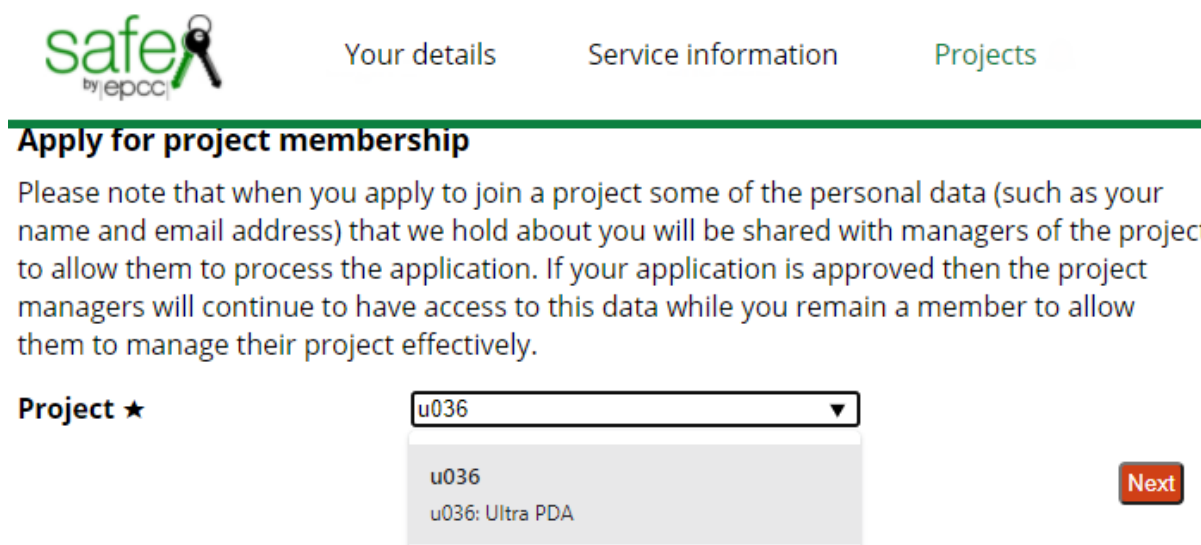Potential users first need to register in the EPCC "SAFE" which is a user registration and account management system.

https://safe.epcc.ed.ac.uk/

Click on the link to Create an account. Once your account has been created you can Login.

(You can use your University of Edinburgh credentials (via EASE) to login, but only after you have created a SAFE account and registered your EASE credentials within SAFE).

Use the Projects menu to Request access:

Type the project code u036 which is a PDA (Protected Data Access) account on Ultra2.

Your project membership request will be sent to a Project Manager for review. The project manager may need to check with an approvals board so access may not be granted immediately.

The next step is to apply for a machine account. The SAFE system has only one option at this point, which is labelled "sdf-cs1".

New account policies

If a check-box does not appear beside a machine then the project you selected is allowed to use the machine but one of the policies that apply to the machine is preventing you from applying.

A cross will be marked against the policy that is preventing you from applying.

You would also be able to enable access to this machine by updating your account to meet any policy marked with an arrow.

Select a machine for the login account

Select	Machine	Type	Description	Policies
◉	sdf-cs1			Users must have a public key registered to use the machine ✓

Next

When you click Next you can choose an account username. This is restricted to 8 letters. Please choose a username in the format: first initial plus surname, eg. "jsmith", if possible. The username must be unique across other machines in the SAFE so you may want to append some code or letter to indicate this is your ISARIC account.

SAFE Login account Request

This form is for requesting new login accounts. To request additional access for an existing account, select it from the navigation menu at the top of the page

Your username will be visible to other users on the system

This machine support ssh key authentication. You can upload a public key to use here.

A SSH public key is required to use this machine.

Requested username ★

SSH public key ★   ssh-rsa AAXXYZ...

Choose file   No file chosen

Request

The system requires a SSH public key be supplied. This will not be used but unfortunately is a requirement that we cannot change, so at this stage it does not matter what you supply, as long as it looks like a valid key. A key can be generated on the website: https://8gwifi.org/sshfunctions.jsp Click **Generate SSH Keys** and then copy and paste the **Public Key** text into the SAFE SSH public key field. You can save the Private and Public keys to files if you wish. If you get the error "Corrupt key" then check you are pasting a single line of text which begins with ssh-rsa.

Generate SSH Keys Online

SSH-Keygen Online Algorithm ◉RSA ○DSA ○ECDSA
RSA Key Size ○1024 ○2048 ○4096
Passphrase

for generating encrypted keys

Generate-SSH-Keys   Email-SSH-Keys

SSH-keygen for [RSA-2048] (PrivateKey/PublicKey)
FingerPrint [6c:79:e3:a6:5a:3c:dc:29:eb:22:8a:b9:14:be:59:d6]
Private Key

-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAqZYD+bJtYJqICE9LRSTV08JmEoz5LzRdvO9t6q3pXwB7o7L2
CVfAKqNbhywzO6qRJWMLINfL79QQUIcePdnDG+TMEw/S92jyD
wmyDvVsMirbB6bXFCn6S7e18Y/UyrdwJRnvS2UuU8My/GIPEM
1eSsxi2iqMzZfdVPn7b
vd1rDdDCEsc/97wMroSv0rLi3Jv73IiCiAVLDuDRy3kCTWtsSBj

Copy the Public Key

Public Key

ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAAABAQCpIgP5sm1gmogIT0tFJNXTwmYSjPkvNF28723pDelfAHujs
vYJV8Aqo1uHLDM7qpElYwsg18vv1BBQhx492cMb5MwTD9L3aPIPBU1QTS44rGy9y1rCbIO9WwyKtsH
ptcUKfpLt7XxJ9TKt3AIGe9LZS5TwzL8aI8QyNSZiKNPp1KIuWifV5KzGLaKozNl91U+juFSRQY/Y/Jg4TG/
xqF2BMVf6RKi3td/R1EuXZdNIZpYzRxDQ1PcwCD/DC7AcqV5QWNEWk4iY65+dmnx7QihZ+b/IIfbw5v
xFJR33HPRTBcJXm8jTc28Yv5fZ6KuM+Xarz13QCbsMsJuyZP1IBZK79

Once your application has been approved you should login to SAFE and use the option to view your password from the Login Accounts menu. The machine name is "sdf-cs1" but the account may be listed as "*username*@eidf". Note: This is a one-time password; you will be required to change it when you first login.
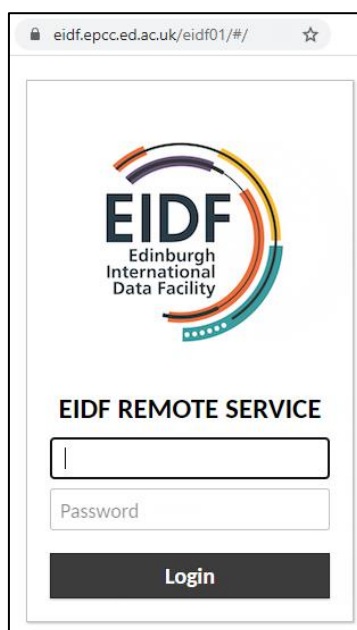
Your machine account will give you a login to two computers, the "sdf-cs1" (which we will call "Ultra2" from now on) and a Linux desktop inside the ISARIC system. However, the only access to these systems, for security reasons, is via a *virtual* desktop. Access to the virtual desktop is through a VDI (Virtual Desktop Infrastructure)[1]. Again, for security reasons, the VDI requires a separate username and password, and these will be sent to you by email.

From now on you only need to login to the VDI, not into SAFE, to access ISARIC.

Logging Into ISARIC at EPCC

The Virtual Desktop Interface gives access to a virtual Linux desktop inside the secure archive area.

https://eidf.epcc.ed.ac.uk/eidf01/

Use your VDI account username and password to login here. The VDI account is not the same as your SAFE account, and is not the same as the "sdf-cs1" machine account you requested within SAFE.

Please change your password by clicking your name in the top right, click Settings, and change your VDI password from the Preferences tab.

---

[1] Some people refer to this as *guacamole* because that is the name of the software which implements the VDI.

The VDI home page will give a list of machines you can log into:
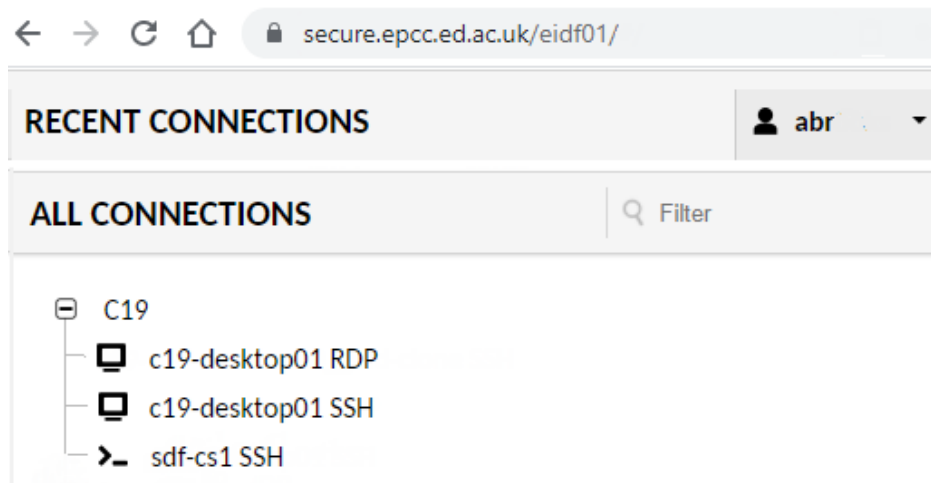
IMPORTANT NOTE: Please click on the "c19-desktop SSH" session first and login. This is the sdf-cs1 machine account you created within SAFE and the password which can be found in the accounts section of SAFE. You will be prompted to change your password. This procedure must be done in the SSH session as this will set your password and create your home directory.
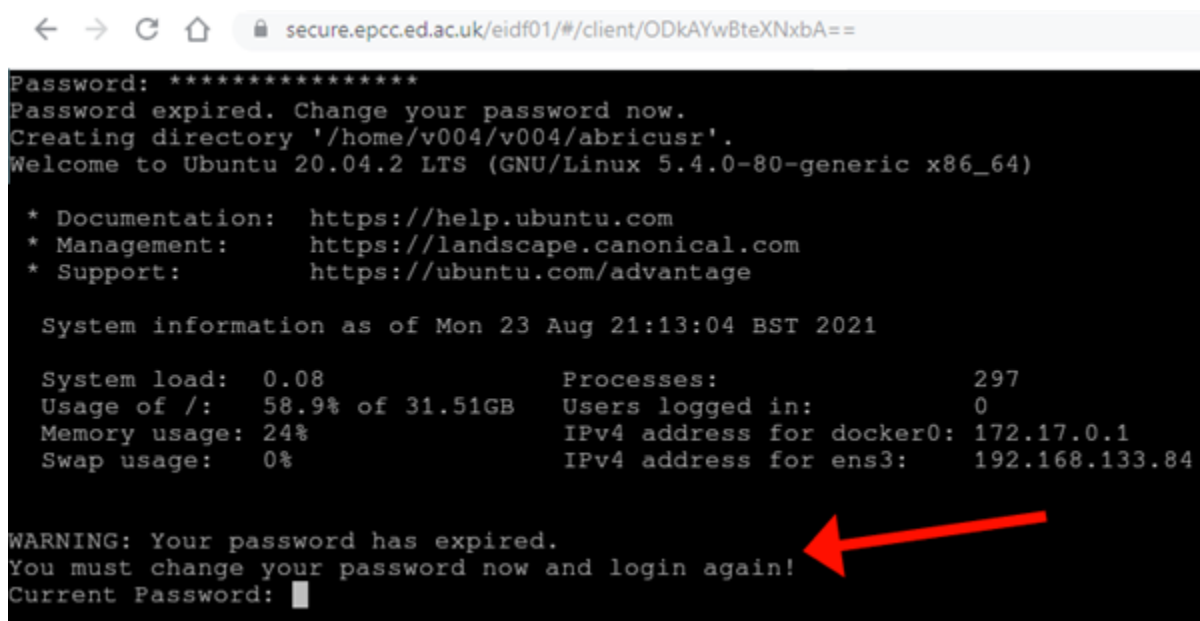
USEFUL TIP: Press the Shift + Ctrl + Alt keys together to get the settings menu. You can paste your password from SAFE into here to avoid mistakes typing it. You can also change the colour scheme if you find white-on-black difficult to read. Press the three keys together again to hide the menu.
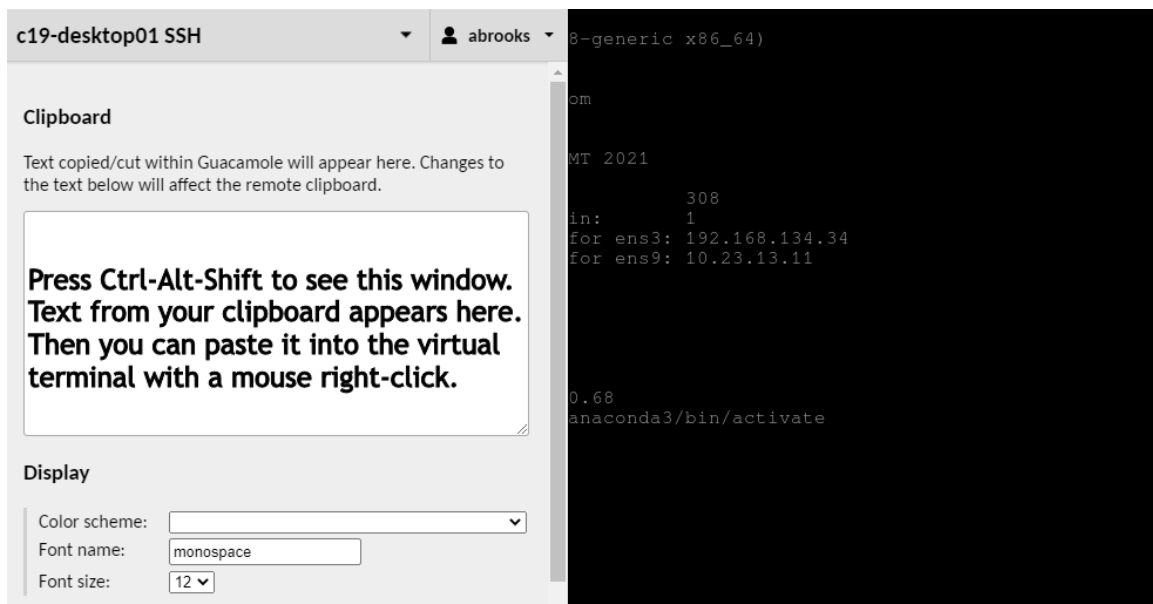
After logging in you will be prompted to choose a new password.

IMPORTANT NOTE: you could also use this SSH session to prepare your SSH key for Ultra2:

```
ssh-keygen
touch .ssh/authorized_keys
cat .ssh/id_rsa.pub >> .ssh/authorized_keys
```
(note the dot before each ssh) Press Enter for all three questions from the first command.

Type exit to close this session. Return to the VDI session page and select the RDP (Remote Desktop) option "c19-desktop RDP". This will present a login screen to the Linux desktop. Again, use your "sdf-cs1" machine username and the password you have just chosen.

Summary

You will have three accounts:

1. Your SAFE website login (only needed during account creation)
2. Your VDI website login
3. Your machine account login (for the desktop and for the sdf-cs1/ultra2 computer)

These actions only need to be completed once:

1. Create an account in SAFE
2. Join the ISARIC project u036 and create a machine account
3. Await approval and your VDI account credentials
4. Log into the VDI eidf01 using your VDI account
5. Change your VDI password
6. Choose the SSH session option and login with your machine account
7. Change the password for your machine account
8. Logout

These actions need to be done every time:

1. Log into the VDI eidf01 using your VDI account
2. Choose the RDP session option
3. Log into the desktop using your machine account

Troubleshooting

Help using SAFE

Please see the documents https://epcced.github.io/safe-docs/ and contact the helpdesk if you have any questions.

Cannot login

If you cannot contact the SAFE website or the VDI website then please try connecting to your institution's VPN.

If you are not sure about your username and/or password:

- SAFE website – use the Forgot Password? button on the SAFE website. If you have problems with this please contact the helpdesk, contact details on the SAFE website.
- VDI website – please contact the helpdesk and ask for your ticket to be assigned to Andrew Brooks.
- c19-isaric desktop – Use the password reset procedure provided on the SAFE website. You will need to request a password reset for the specific machine account, in this case on the "sdf-cs1" as part of the "u036" project. After a reset you will be able to log into SAFE and view the new password by selecting your username@eidf from the Login Accounts menu and clicking the View Login Account Password button. If you have problems with this please contact the helpdesk and ask for your ticket to be assigned to Andrew Brooks.

Virtual desktop problems

- **Color scheme**; cannot read text in the SSH window – Press the Shift-Ctrl-Alt keys together to get the Guacamole settings and scroll down to change the colour scheme. Press the same keys again to hide the settings.
- The **CAPS-LOCK** key seems to be stuck on. Even if you press it again, the CAPS state remains on. Press the Shift-Ctrl-Alt keys together to get the Guacamole settings and press CAPS LOCK. Press the same 3 keys again to hide the settings. Now CAPS LOCK is off in the virtual desktop and you can press CAPS LOCK again to turn it off on your local desktop.
- Unstable network connection: If your network connection drops then it is possible to log back into the desktop and continue where you left off. However, do not be tempted to rely on this and leave programs running overnight, as there are various reasons why you might come back and find the desktop has been restarted. (Technically, it might still be running but you can no longer access it). Please save your work and log off before disconnecting whenever possible.

How to use the c19-desktop

Logging in

Login to the VDI at https://secure.epcc.ed.ac.uk/eidf01/ using your VDI account credentials.

Select the c19-desktop (RDP) option and login using your sdf-cs1 (account@eidf) credentials.
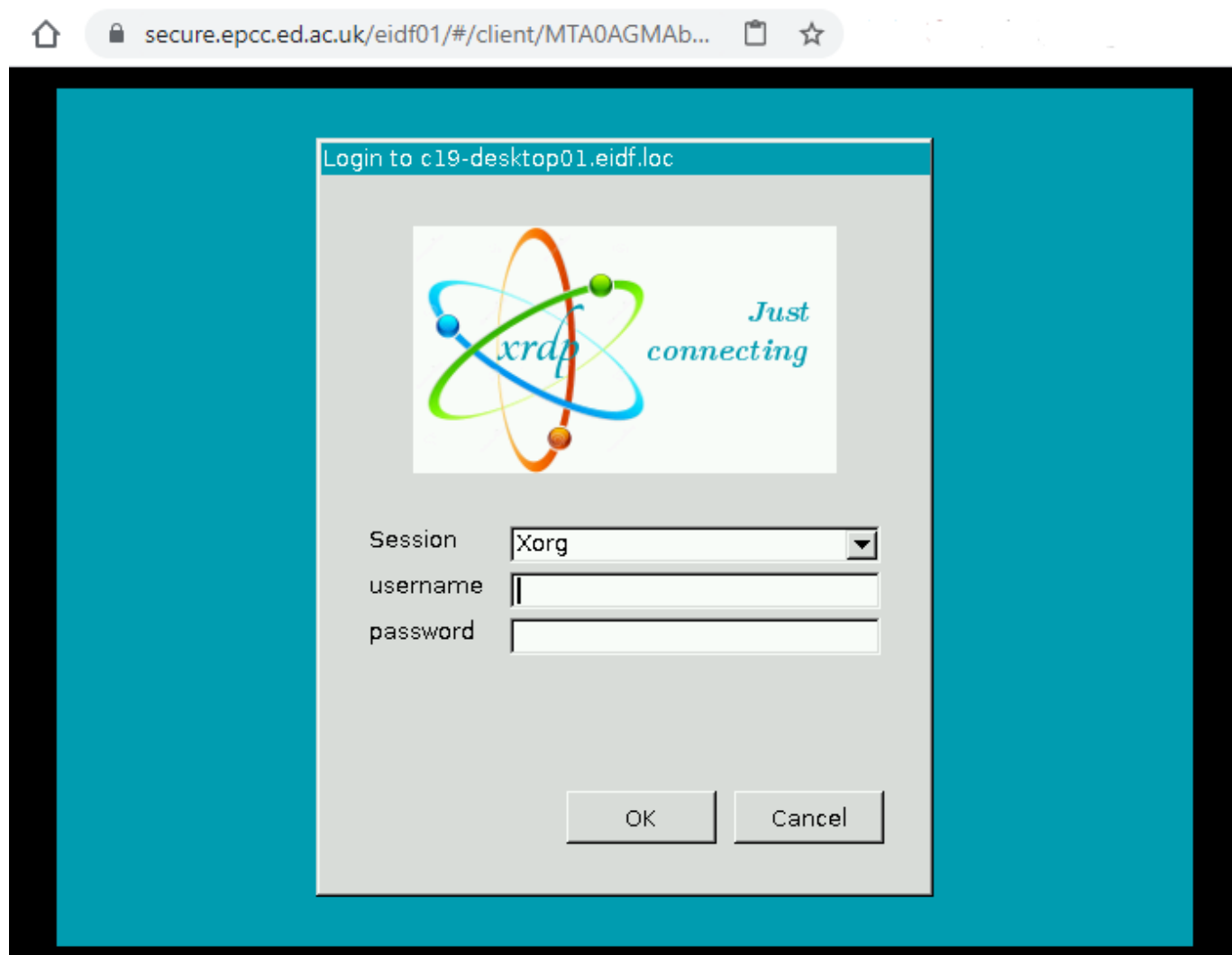
Logging out

To logout from the desktop use the menu at the top right. It should show your username but in some circumstances may show "not populated".
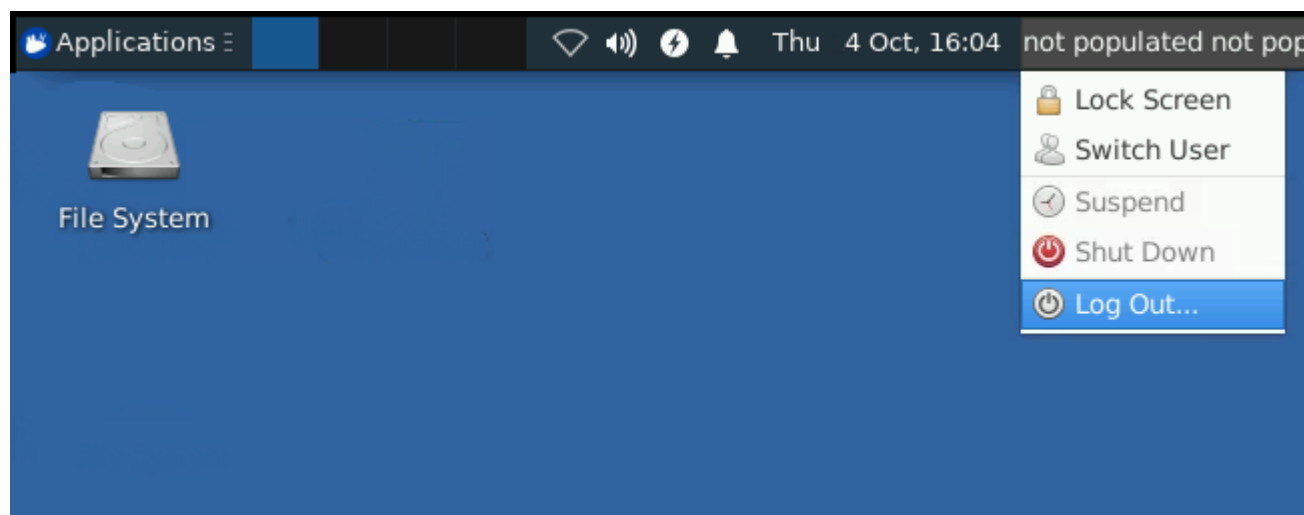
Using desktop software

You can access RStudio and PyCharmPro from the Applications | Development menu:
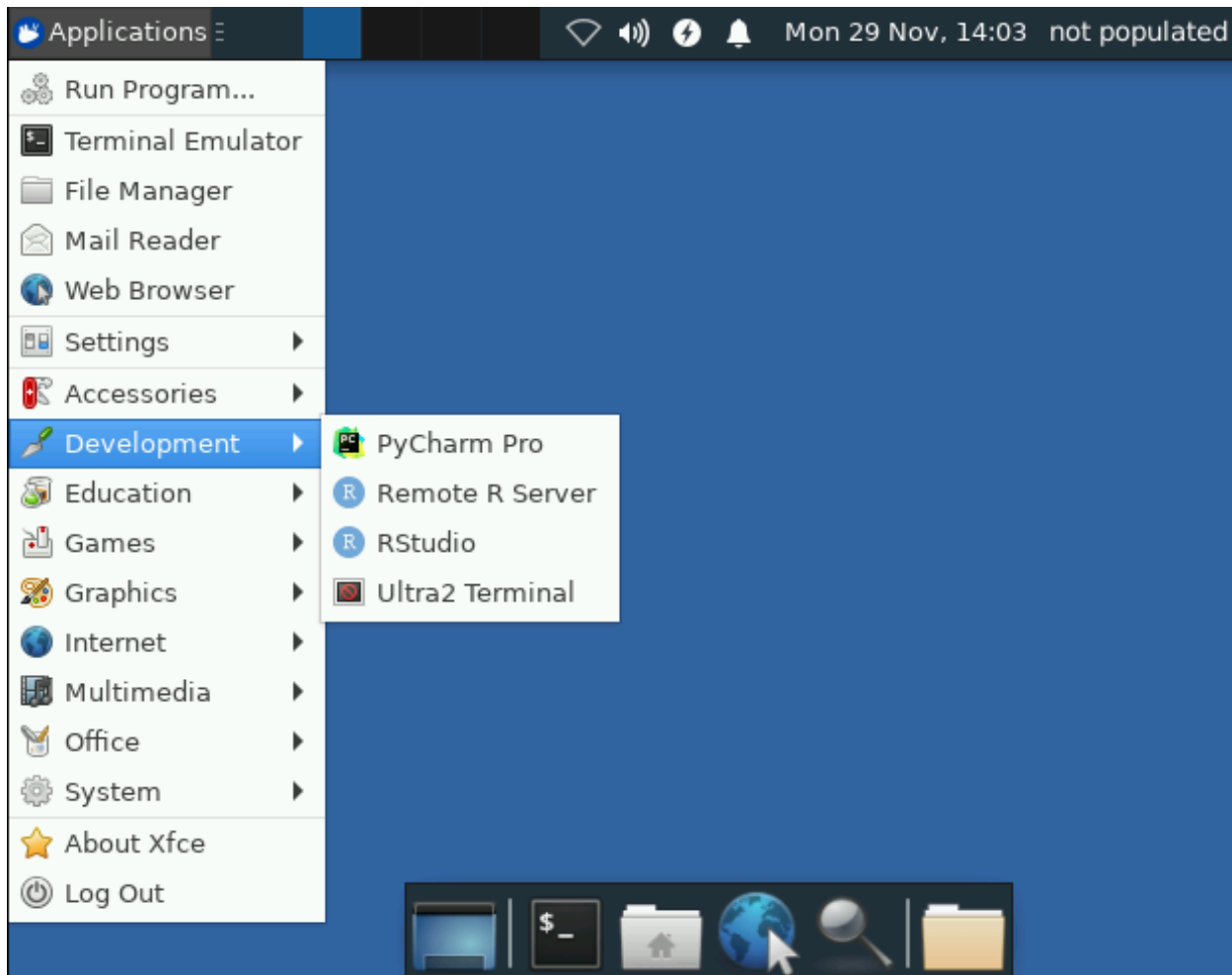
The Development menu has options for:

- PyCharm Pro – a Python development environment. You need to bring your own license for this but it is free for academics.
- RStudio – an R-language development environment.
- Remote R Server – this runs R on ultra2 which you can access using the RStudio environment, giving you the ability to run compute-intensive jobs on ultra with the flexibility of a GUI on the desktop
- Ultra2 Terminal – this opens a command-line terminal window for using the ultra2 computer. It simply runs "ssh ultra2" in a window.

How to use Ultra

What you need to know

- There are two computer systems you will use. The "ultra2" computer is a HPC system (High Performance Computing) with a vast amount of memory and processing power. The virtual desktop "c19-desktop" is much more limited and shared with other users, but it does have RStudio and PyCharmPro. Please try to do your analysis on ultra2 if you can, **not** on the desktop, out of consideration for other users. See below for details.
- Your account will be a member of a sub-group of u036, either u036-isaric, u036-phosp or u036-collab. By default the files in one sub-group *cannot* be read by members of a different sub-group.
- Your home directory should not be used for storing project files, please use one of the shared directories.
- Project files are visible to everyone else in the project but to nobody else.
- No personally identifiable data may be stored on the system. Whilst it is a secure environment, it is also shared and it is explicitly not a *safe haven* so is not authorized to hold unconsented PII.

Directories

Home directories and project files for the u036 (c19-isaric) project live under /home/u036. The same paths and files are available from both ultra2 and the c19-desktop so you can share files between the two systems.

There are two sub-projects, "isaric" and "phosp", and there is an additional sub-project called "collab" which is for external collaborators.

Your project files will be in /home/u036/u036-*subgroup*/shared/… These files are *only* accessible to members of your sub-group (isaric/phosp/collaborator).

To share files across the whole project, i.e. members of u036-isaric and u036-phosp, you can use /home/u036/shared.

Summary:

/home/u036

/home/u036/shared – files accessible to members of every sub-group

/home/u036/*username* – your personal files

/home/u036/u036/shared – files accessible to members of every sub-group

/home/u036/u036-isaric/shared – files accessible to members of ISARIC only

/home/u036/u036-phosp/shared – files accessible to members of PHOSP only

/home/u036/u036-collab/shared – files accessible to members of external collaborators only

How to import and export data

The environment is deliberately restricted to prevent the extraction of data. This is for security reasons and also to prevent publication of data which is not yet approved for publication. The restriction on extraction also implies that data cannot be imported, and thus there is no internet access. However data managers do have permission to import and export data on your behalf.

To import data please contact your data manager.

To export data please contact your data manager.

Using Anaconda for R and Python

A shared copy of anaconda3 has been installed and can be used by issuing the command:

```
source /home/u036/u036/shared/anaconda3/bin/activate
```

That will activate the base conda environment giving you access to additional environments. Your command prompt will now show (base) to indicate this.

Then you can activate a specific environment to get additional software, for example to use R you can issue the command:

```
conda activate Rv4
```

You will see your prompt change from (base) to (…Rv4).

Use conda deactivate when finished with that environment (or simply logout).

Using R Studio

RStudio can be started from the Applications | Development menu. It is using the Anaconda version of R; see above.

You can install additional R packages into your personal directory using the normal command but it may be necessary to specify the package source like this:

```
install.packages('DOSE', repos='https://stats.bris.ac.uk/R/')
```

As mentioned above, the resource constraints on the desktop mean that data-intensive and CPU-intensive work must be performed on the ultra2 computer. This can be done using RStudio as the GUI and connecting to an R Server running on ultra.

The first step is to ask EPCC's HPC Systems Team for a port number to be allocated to you (it will be something like 60123). (When logging the query ask them to check with abrooks).

Start the R Server using the Applications | Development | Remote R Server menu. This will prompt you for your personal port number. If you don't have one, please ask the helpdesk. Do not use somebody else's number!
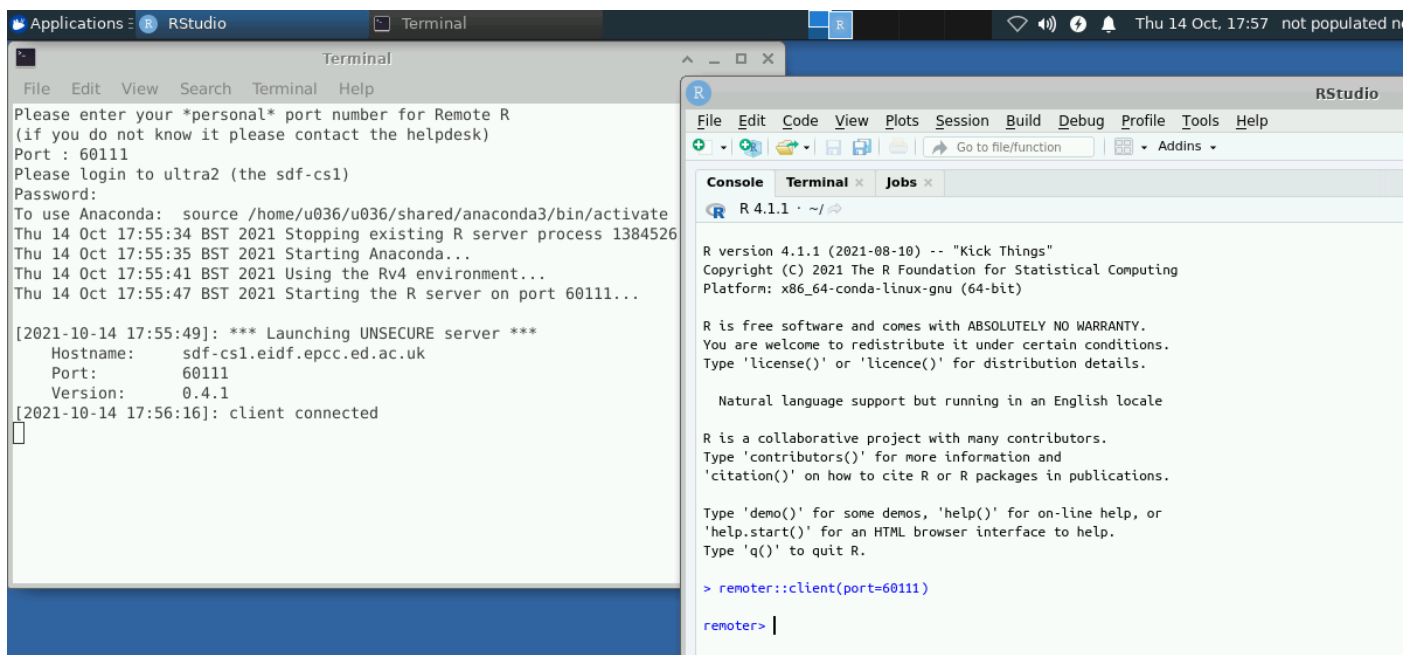
When the server is running you can start RStudio and type:

```
remoter::client(port=N)   # where N is your personal port number as above
```

Now all your variables are stored on ultra and all your R code will execute on ultra.

You can transfer a variable from the remote to the local using: `s2c(varname)` on the server.

You can see plots by using the `rpng()` command first, making the `plot()`, then retrieving it with `rpng.off()`. See the manual for more options.

When you have finished you can leave the remoter environment by typing:

exit()

and then close the Server window.

NOTE: If you have some analysis which will take a long time to run then please use the job scheduler; see the Slurm section below.

Troubleshooting
Documentation:

https://cran.r-project.org/web/packages/remoter/vignettes/remoter.pdf
https://cran.r-project.org/web/packages/remoter/vignettes/remote_machines.pdf
https://cran.r-project.org/web/packages/remoter/remoter.pdf

Bind failed: address already in use – this means that the R server is already running, please check you are using the correct port number, and if so then you don't need to start a new server. To see if the server is already running on ultra use this command and see if the output includes the command you used to start it: `pgrep -au$(id -u)`

channel 3: open failed: connect failed: Connection refused – this might mean that a client process is still running, i.e. inside your RStudio. If restarting RStudio does not help then the simplest solution is to reboot your computer.

Connection refused – this means that the R server is not running. If you previously started it then it may have crashed (this can happen due to uncaught R errors or if it would require interaction, such as trying to install a package without using the repos parameter). Try starting the server again, or exiting your RStudio.

Incompatible package versions – this happens when the versions of 'remoter' and 'pbdZMQ' on your RStudio do not match the versions on Ultra. In fact if your RStudio has newer versions than Ultra you will not see this message (however, see below). These packages are already installed so please contact the helpdesk.

Argument is of length zero (get.status("method_plot_rpng") == "rasterImage") when plotting using rpng.off() – this happens when your RStudio version of 'remoter' is newer than the one on Ultra, typically if ultra is 0.4.0 and RStudio is 0.4.1. The solution, for now, is to downgrade your 'remoter' package in RStudio using the instructions above.

The R server keeps crashing – this happens when you try to execute an unknown command, particularly if a package has not been installed or loaded yet. In particular getting the parameters wrong for ggplot() will cause it to crash. This is a known bug, see https://github.com/RBigData/remoter/issues/50 and a fix has been applied 2021-02-22.

If the server has crashed then you can restart it; there is no need to logout or login again.

If you wish to see error messages as they occur you can use the manual method for starting the R server as given above: login to ultra with ssh, source conda, activate Rv4, use the Rscript command to start the server. After a crash simply run the Rscript command again.

Using PyCharm on Ultra

It is not possible to use PyCharm on Ultra itself, because it is not a desktop environment, but it is possible to use PyCharm on the desktop and have it run the programs on Ultra.

The recommended way to use PyCharm on Ultra is to run it on the desktop and connect to a Python interpreter running on Ultra. This method has the benefit of a fast, responsive Python IDE running on the desktop, plus a Python interpreter running on the same machine as the data – the best of both worlds. You will need a full PyCharm license for this but it's free to students/teachers/etc. The full instructions are on the JetBrains website (links below) but the quick summary is:

File | New Project… | Na"e "*remote_ul"ra*"
File | New... | Python File | Na"e "*remote_ultra_test"py*" and add some code OR re-use existing project
File | Settings | Project: *name* | Project Interpreter
 click the cog at the end of the Project Interpreter | Add...
 In the Add Python Interpreter window choose SSH Interpreter in the left column
 Enter Host: ultra2 and Username: your existing username on ultra, click Next
 Enter your ultra Password: and tick Save Password, click Next
 Choose a Python interpreter, the default /usr/bin/python is v2.7.5 (old!),
 or choose a Python interpreter from an installed Conda environment, such as
 /home/u036/shared/conda_environments/<environment name>/bin/python, or
 /home/u036/shared/anaconda3/bin/python, which is v3.7.6
 Sync folders: click on the folder icon at the end of the Sync folders:
  click in the Remote Path entry and change it from /tmp/pycharm_project_N to
  /home/u036/<your username>/PycharmProjects/<temporary project name>, click Finish.
 (Note! Change /home/u036 to your own home directory)
 (Note! Don't use the same project name as your local copy or they will clash)

File | Settings | Appearance and Behaviour | System Settings | HTTP Proxy
  enter hostname c19-desktop-proxy and port 800

File | Settings | Build, Execution, Deployment | Deployment
 click on the Mappings tab,
 change the Deployment path: to the same path you entered in Sync folders.
 Click OK (Wait until the Network Transfer tab has finished uploading all the deployment configuration to Ultra.)

Run | Run... | select the name of the configuration to run your code directly on ultra.

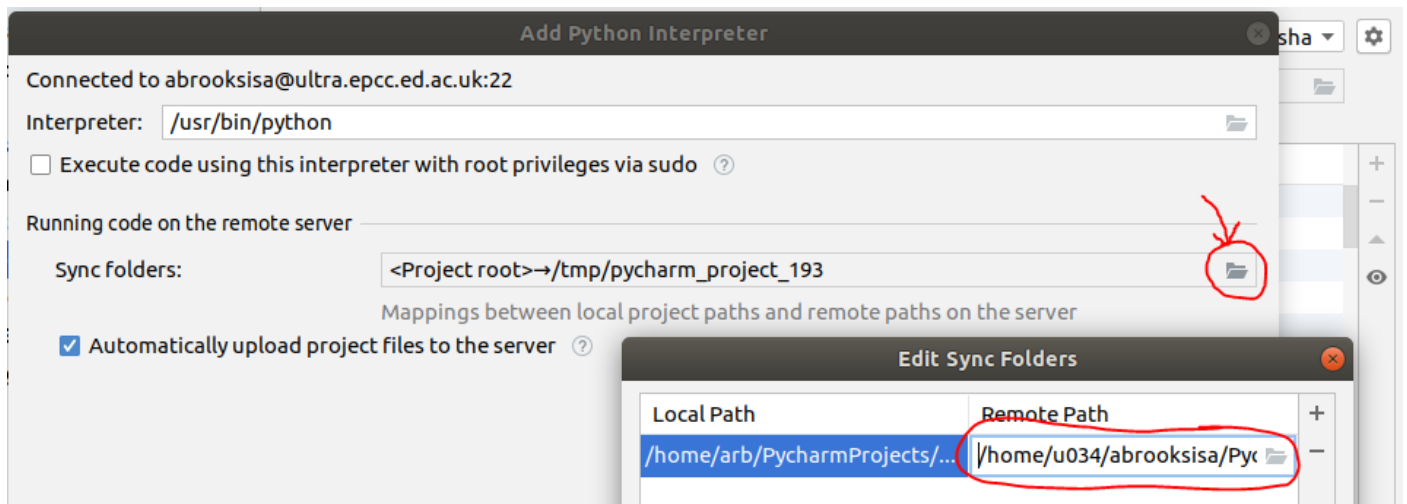Control the upload of files to ultra from the Tools | Deployment menu.

References:

https://www.jetbrains.com/help/pycharm/configuring-remote-interpreters-via-ssh.html

and https://www.jetbrains.com/help/pycharm/remote-debugging-with-product.html

The Sync folders dialogue box:

NOTE: If you have some analysis which will take a long time to run then please use the job scheduler; see the Slurm section below.

Running large jobs on Ultra2 with Slurm

Large, compute-intensive jobs should be run through Ultra's batch job scheduler, which on Ultra2 is Slurm. To submit a large job, e.g. an R script on a large dataset, put all the commands used for your analysis into a batch file, e.g. example_job.slurm:

```
#!/bin/bash
#SBATCH --job-name=example_job
#SBATCH --cpus-per-task=2
#SBATCH --mem=8GB
#SBATCH --time=1:00:00
#SBATCH --output=/home/u036/u036/me/job_output.log

# go to the directory the script is being submitted from
cd $SLURM_SUBMIT_DIR
source /home/u036/u036/shared/anaconda3/bin/activate
conda activate Rv4
Rscript your_analysis.R
```

Documentation on using Slurm, including features like job arrays, can be found at https://slurm.schedmd.com/documentation.html. For information on migrating PBS scripts from Ultra 1 to Slurm, see SchedMD's Rosetta Stone of Workload Managers at https://slurm.schedmd.com/rosetta.pdf.

Accessing Ultra 2 from the desktop VM

To submit Slurm scripts, it is necessary to log into the main Ultra 2 machine as opposed to the desktop VM. This can be done on the graphical desktop by opening 'Application -> Development -> Ultra2 Terminal', or on the command line by running `ssh ultra2` or `ssh sdf-cs1` (note that `ssh sdf-cs1.epcc.ed.ac.uk` will not work). Once logged into Ultra, scripts can be submitted by running, e.g, `sbatch example_job.slurm`.

You can view the status of scheduled and currently running jobs with the command `squeue`, and finished jobs can be checked with the command `sacct`, or `sacct -j <job_id>`.

Access to external databases from Ultra

A database for ISARIC has been created on a separate host, called c19-isaric01. This host is accessible from the desktop VMs and from Ultra2 using the hostname c19-database-proxy. If it does not resolve then the IP address (as seen from Ultra2) is 172.16.28.56 and from the VMs it is 192.168.134.48

To request access to the ISARIC database, raise a request with the EPCC helpdesk with your Ultra username and whether or not you need to be able to insert data. You will receive back your login credentials once you have been given the relevant level of access.

The database name is "isaric" and the schema name is "isaric". Once connected to the "isaric" database you can refer to tables as "isaric.my_table".

Access from the command line

You can connect to the database from the Ultra command line by running:

```
psql -h c19-database-proxy -U <pg_username> -d isaric
```

where pg_username is the username you were given when requesting database access. As per the [Postgres docs](#), you can also store your database credentials in the file `.pgpass` in your home folder, in the format `hostname:port:database:username:password`, and you will not need to supply your password each time. The port in this case will be the default Postgres port 5432. Postgres will only accept this file if it's accessible only by you and nobody else. To assign the right file permissions, run:

```
chmod go-rwx ~/.pgpass
```

Once in you can run SQL queries, as well as psql commands like:

- `\dt isaric.*` to list tables
- `\dv isaric.*` to list views
- `\d isaric.<table_or_view_name>` to show column information for a table or view
- `\password` to change your Postgres password

The `\d...` commands can also be appended with a '+' to view extra information like any given table/column descriptions, or the explicit SQL query that a view is made up of:

- `\dt+ isaric.*`
- `\d+ isaric.<table_or_view_name>`

Access from R

The database server is PostgreSQL, so to connect to it from R on ultra:

```
library('RPostgreSQL')

pg_con <- DBI::dbConnect(RPostgreSQL::PostgreSQL(), dbname="isaric", host="c19-database-proxy", user="myusername", password="mypassword")

library('tidyverse')

my_tbl <- tbl(pg_con, sql('select * from my_table'))
```

OR

```
my_tbl <- tbl(pg_con, 'my_table')

my_tbl %>% select(stuff) %>% filter(stuff)
```

```
dbDisconnect(pg_con)
```

to dynamically create SQL statements using R syntax.

The database name is "isaric" and the schema name is "isaric". Once connected to the "isaric" database you can refer to tables as "isaric.my_table".