

Data Safe Haven User Guide

Apache Guacamole

The Alan Turing Institute

Aug 30, 2022

Table of Contents

1	💡 Introduction	5
1.1	Definitions	5
2	🔑 Set up your account	7
2.1	👉 Prerequisites	7
2.2	🔒 Set a password	8
2.3	👤 Set up multi-factor authentication	13
3	💻 Access the Secure Research Environment	21
3.1	👉 Prerequisites.	21
3.2	👤 Log into the research environment	21
3.3	👤 Log into the Secure Research Desktop	24
4	💻 Analysing sensitive data	29
4.1	🎁 Pre-installed applications	29
4.2	⌨️ Keyboard mapping	30
4.3	🔓 Access applications	30
4.4	🐍 Available Python and R versions	31
4.5	📦 Install R and python packages	32
5	🔗 Share files with collaborators	35
5.1	📁 Shared directories within the SRE	35
5.2	📁 Bring in new files to the SRE	36
6	👯 Collaborate on code using CoCalc	37
6.1	💻 Access CoCalc.	37
7	🔖 Versioning code using GitLab	39
7.1	🗃️ Maintaining an archive of the project	39
7.2	👤 Access GitLab	39
7.3	🏷️ Public repositories within the SRE	40
7.4	🌐 Support for GitLab use.	40
8	📖 Collaborate on documents using CodiMD	43
8.1	👤 Access CodiMD.	43
8.2	👤 Editing other people's documents	43
8.3	⚡ Troubleshooting CodiMD.	45
9	ℹ️ Access additional SRDs	47
10	💻 Access databases	49
10.1	💻 Microsoft SQL.	49
10.2	💻 PostgreSQL	49
10.3	🌐 Connecting using Azure Data Studio	49
10.4	dbeaver Connecting using DBeaver	50
10.5	🐍 Connecting using Python	53
10.6	🐍 Connecting using R.	54
11	🐞 Report a bug	55

Table of Contents

11.1	Help us to help you.	55
12	Acknowledgments	57
13	Appendix A: Command Line Basics	59
14	Appendix B: Gitlab tutorial notes	61
14.1	Getting started with Git	61
14.2	Repositories	61
14.3	Add your Git username and set your email	61
14.4	Cloning projects	62
14.5	Branches	62
14.6	Downloading the latest changes in a project	63
14.7	Add and commit local changes	63
14.8	How to create a Merge Request	64
15	Appendix C: Troubleshooting	67
15.1	! No applications available	67
15.2	! Unexpected certificate error	67

1 Introduction

 Welcome to the Turing Data Safe Haven! 

Secure research environments (SREs) for analysis of sensitive datasets are essential to give data providers confidence that their datasets will be kept secure over the course of a project. The Data Safe Haven is a prescription for how to set up one or more SREs and give users access to them. The Data Safe Haven SRE design is aimed at allowing groups of researchers to work together on projects that involve sensitive or confidential datasets at scale. Our goal is to ensure that you are able to implement the most cutting edge data science techniques while maintaining all ethical and legal responsibilities of information governance and access.

The data you are working on will have been classified into one of five sensitivity tiers, ranging from open data at Tier 0, to highly sensitive and high risk data at Tier 4. The tiers are defined by the most sensitive data in your project, and may be increased if the combination of data is deemed to require additional levels of security. You can read more about this process in our policy paper: *Arenas et al, 2019, arXiv:1908.08737*.

The level of sensitivity of your data determines whether you have access to the internet within the SRE and whether you are allowed to copy and paste between the secure research environment and other windows on your computer. This means you may be limited in which data science tools you are allowed to install. You will find that many software packages are already available, and the administrator of the SRE will ingress - bring into the environment - as many additional resources as possible.

Important Please read this user guide carefully and remember to refer back to it when you have questions. In many cases the answer is already here, but if you think this resource could be clearer, please let us know so we can improve the documentation for future users.

1.1 Definitions

The following definitions might be useful during the rest of this guide

Secure Research Environment (SRE)

the environment that you will be using to access the sensitive data.

Data Safe Haven

the overall project that details how to create and manage one or more SREs.

Username domain

the domain (for example `projects.turingsafehaven.ac.uk`) which your user account will belong to. Multiple SREs can share the same domain for managing users in common.

SRE ID

each SRE has a unique short ID, for example `sandbox` which your System Manager will use to distinguish different SREs in the same Data Safe Haven.

SRE URL

each SRE has a unique URL (for example `sandbox.projects.turingsafehaven.ac.uk`) which is used to access the data.

2 **Set up your account**

This section of the user guide will help you set up your new account on the SRE you'll be using.

2.1 **Prerequisites**

Make sure you have all of the following in front of you when connecting to the SRE.

-  The email from your System Manager with your account details.
-  Your **username**, given in an email from your System Manager.
-  The **domain name and URL** for the SRE, given in an email from your System Manager.
-  Your computer.
-  Access to the specific wired or wireless network detailed in the email from your System Manager.
-  **Data security training** for those working on health datasets.
-  Your **phone**, with good signal connectivity.

You should also know who the **designated contact** for your SRE is. This might be an administrator or one of the people working on the project with you. They will be your primary point of contact if you have any issues in connecting to or using the SRE.

Note For example, during the Turing Data Study Groups, the **facilitator** of each SRE is the designated contact

2.1.1 Username

Your username will usually be in the format `firstname.lastname`. In some places, you will need to enter it in the form `username@<username domain>`

Tip You can find your username in the email you received from your System Manager.

Caution! If you have a hyphenated last name, or multiple surnames, or a long family name, your assigned username may not follow the same pattern of `firstname.lastname`. Please check with the designated contact for your SRE if you are unsure about your username.

Note In this document we will use **Ada Lovelace** as our example user. Her username is:

- short-form: `ada.lovelace`
- long-form: `ada.lovelace@projects.turingsafehaven.ac.uk`

2.1.2 Network access

The SRE that you're using may be configured to allow access only from a specific set of IP addresses. This may involve being connected to a specific wired or wireless network or using a VPN. You also may be required to connect from a specific, secure location. You will be told what

Set up your account

these requirements are for your particular environment.

Tip Make sure you know the networks from which you must connect to your SRE. This information will be available in the email you received with your connection information.

2.1.3 Data security training requirements

Depending on your project, you may be required to undertake data security awareness training.

Tip Check with your designated contact to see whether this is the case for you.

2.1.4 Your phone for multi-factor authentication

Multi-factor authentication (MFA) is one of the most powerful ways of verifying user identity online. We therefore use MFA to protect the project data - specifically, we will use your phone number.

Important Make sure to have your phone with you and that you have good signal connectivity when you are connecting to the SRE.

Caution! You may encounter some connectivity challenges if your phone network has poor connectivity. The SRE is not set up to allow you to authenticate through other methods.

2.1.5 Domain names

You should be given the username domain in the initial email from your System Manager. You might receive the SRE URL at this time, or you might be assigned to a particular SRE at a later point.

Note In this document Ada Lovelace - our example user - will be participating in the sandbox project at a Turing Data Study Group.

- Her **username domain** is `projects.turingsafehaven.ac.uk`.
 - Her **SRE URL** is `https://sandbox.projects.turingsafehaven.ac.uk`
-

2.2 Set a password

For security reasons, you must reset your password before you log in for the first time. Please follow these steps carefully.

- Open a private/incognito browser session on your computer.

Tip One of the most common problems that users have in connecting to the SRE is automatic completion of usernames and passwords from other accounts on their computer. This can be quite confusing, particularly for anyone who logs into Microsoft services for work or personal use.

Caution! Look out for usernames or passwords that are automatically completed, and

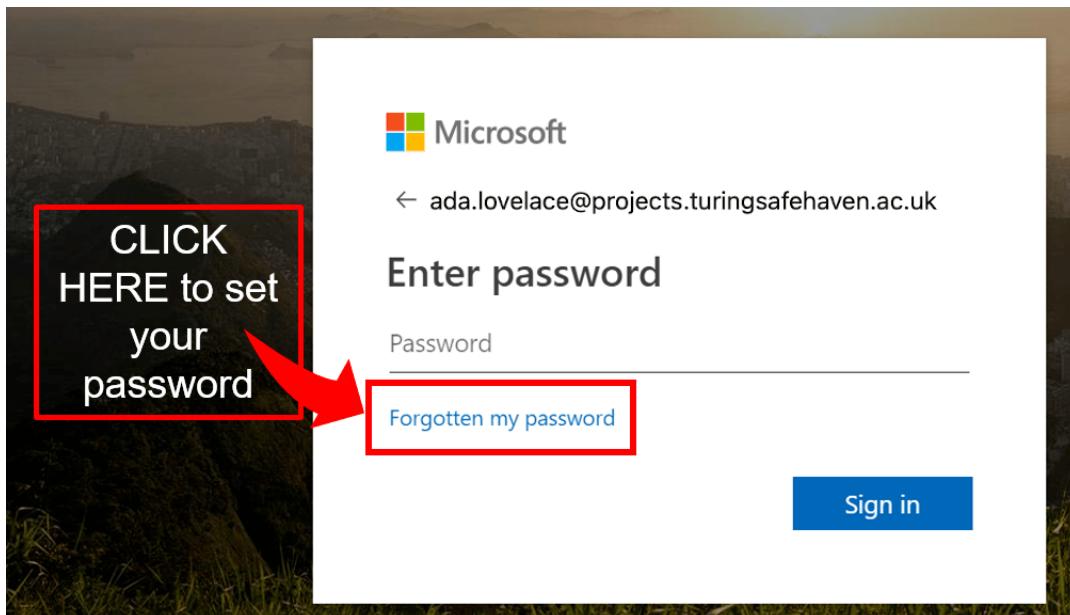
2 Set up your account

make sure that you're using the correct details needed to access the SRE.

- Navigate to the following URL in your browser: <https://aka.ms/mfasetup>. This short link starts the process of logging into your account.
- At the login prompt enter `username@<username domain>` and confirm/proceed. Remember that your username will probably be in the format `firstname.lastname`.

Note Our example user, Ada Lovelace, participating in the `sandbox` project at a Turing Data Study Group, would enter `ada.lovelace@projects.turingsafehaven.ac.uk`

- There will then be a password prompt. The first time you log in you need to click “**Forgotten my password**”.



Caution! If you reset your password, you will need to wait 5-10 mins before logging in again, to allow the user management system to sync up with the new password.

- Fill out the requested CAPTCHA (your username should be pre-filled).

Get back into your account

Who are you?

To recover your account, begin by entering your email address or username

Email or Username: *

ada.lovelace@projects.turingsafehaven.ac 

Example: user@contoso.onmicrosoft.com or user@contoso.com



Enter the characters in the picture or the words in the audio. *

Next

Cancel

- Confirm your phone number, which you provided to the System Manager when you registered for access to the environment.

Get back into your account

verification step 1 > choose a new password

Please choose the contact method we should use for verification:

- Send a text to my mobile phone number
 Call my mobile phone number

In order to protect your account, we need you to enter your complete mobile phone number (*****51) below. You will then receive a call. Please answer it to continue.

Enter your phone number

Call

- Select a password.
Your password must comply with the following requirements:

Important

- alphanumeric
- minimum 12 characters
- at least one each of:
 - **AB** uppercase character
 - **aB** lowercase character
 - **12** number
- you should choose a unique password for the SRE to ensure it is secure

Caution! Do not use special characters or symbols in your password! The virtual keyboard inside the SRE may not be the same as your physical keyboard and this can make it difficult to type some symbols.

Note that this will also ensure that it passes the Microsoft Azure AD password requirements.

Tip We recommend using a password generator like [this one](#) to create a password that meets these requirements. This will ensure that the password is different from any others that you might use and that it is unlikely to be on any list of commonly used passwords.

If your password is too difficult to memorise, we recommend using a password manager, for example [BitWarden](#) or [LastPass](#), to store it securely.

- Enter your password into the Enter new password and Confirm new password fields.

Get back into your account

verification step 1 ✓ > **choose a new password**

* Enter new password:

* Confirm new password:

Finish

[Cancel](#)

- Then continue to the next step

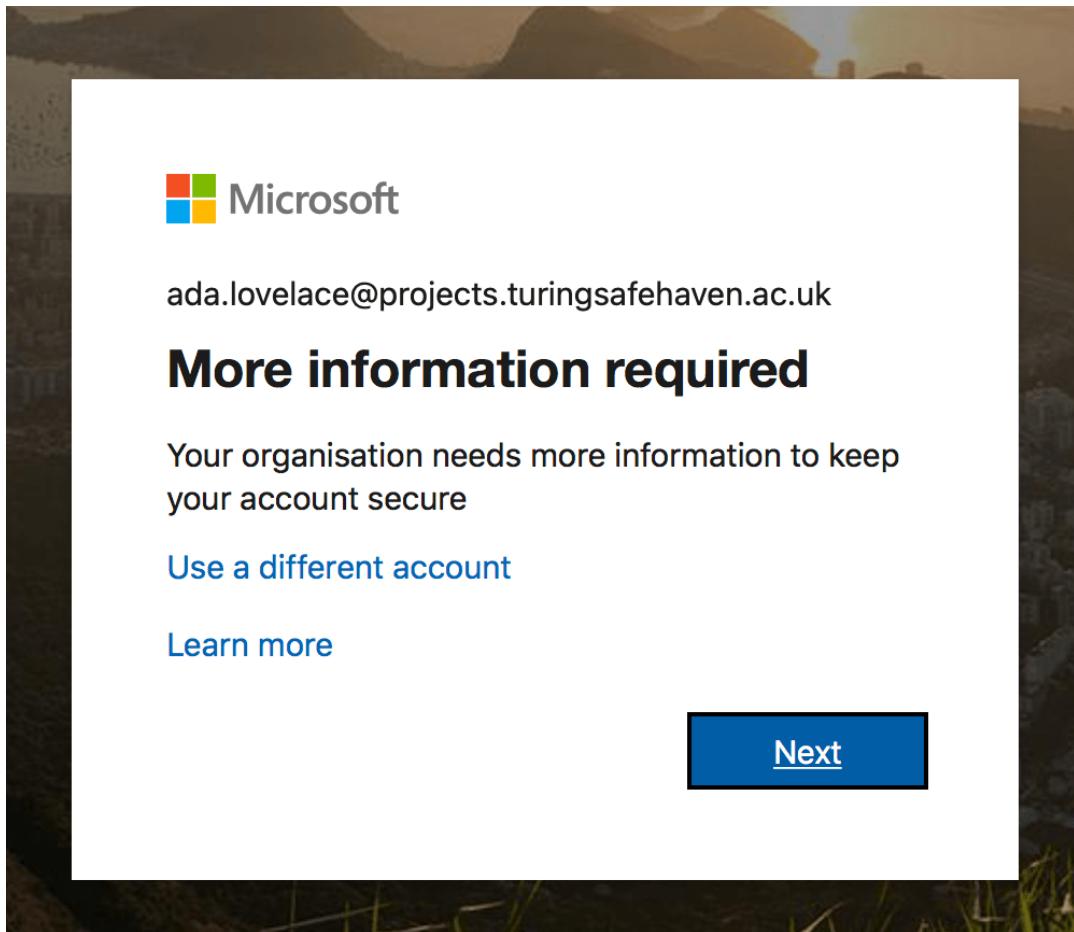
Get back into your account

✓ Your password has been reset

To sign in with your new password, [click here.](#)

Click here to
go on to the
next step

- Log into your account when prompted and at this point you will be asked for additional security verification.



2.3 Set up multi-factor authentication

The next step in setting up your account is to authenticate your account from your phone. This additional security verification is to make it harder for people to impersonate you and connect to the environment without permission. This is known as multi-factor authentication (MFA).

2.3.1 Phone number registration

- In order to set up MFA you will need to enter your phone number

2 Set up your account

Keep your account secure

Your organization requires you to set up the following methods of proving who you are.

Phone

You can prove who you are by answering a call on your phone.

What phone number would you like to use?

Call me

Message and data rates may apply. Choosing Next means that you agree to the [Terms of service](#) and [Privacy and cookies statement](#).

[Next](#)

- Once you click next you will receive a phone call straight away.

Phone

We're calling +44 [REDACTED] now.

[Back](#)

Tip The call might say *press the pound key* or *press the hash key* both mean hit the # button.

- After following the instructions you will see the following screen

Keep your account secure

Your organization requires you to set up the following methods of proving who you are.

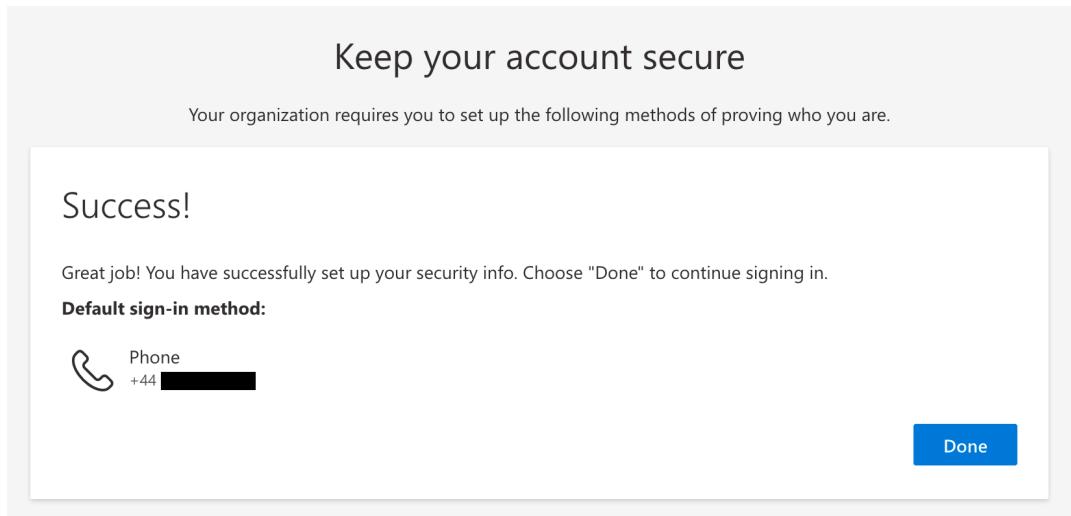
Phone

Call answered. Your phone was registered successfully.

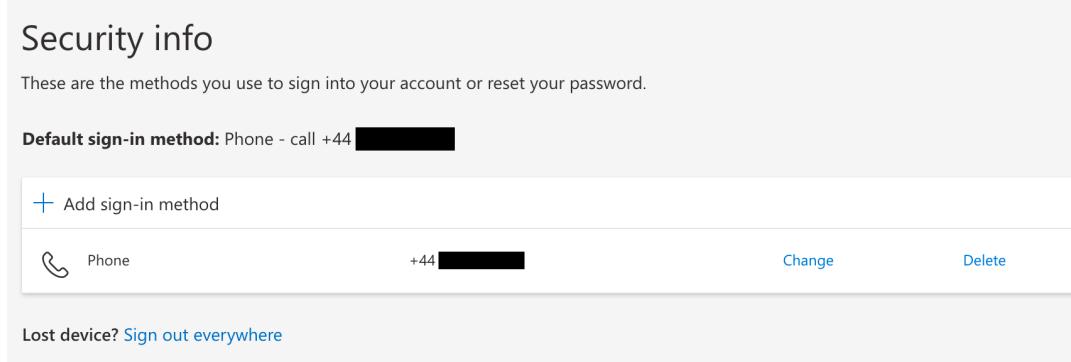
[Next](#)

Set up your account

- Click Next to register this phone number for MFA



- You should now see the Security Information dashboard that lists all your verified MFA methods

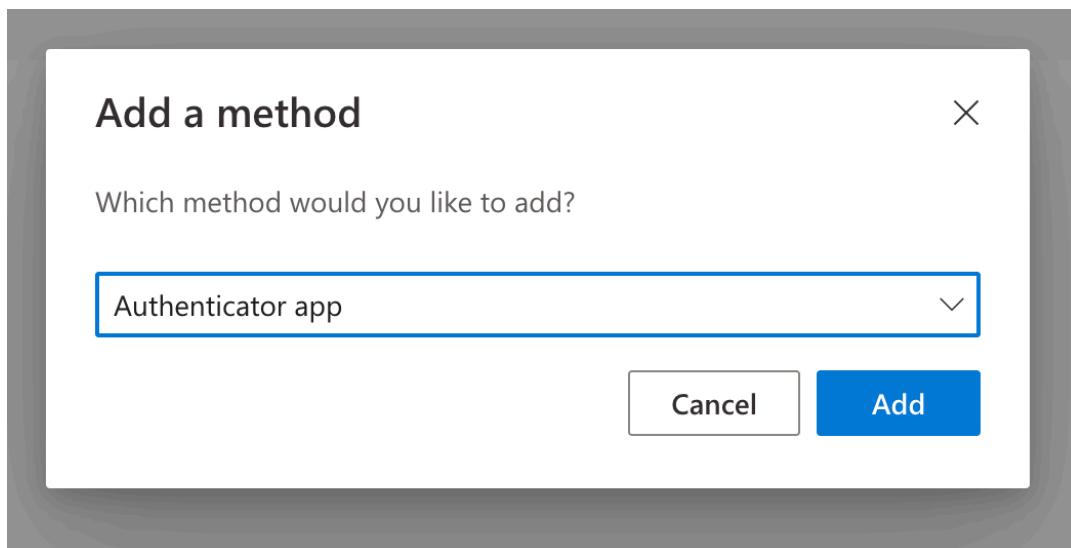


A screenshot of the 'Security info' dashboard. The title 'Security info' is at the top. Below it, a note says 'These are the methods you use to sign into your account or reset your password.' Under 'Default sign-in method:' is a row with a phone icon, the word 'Phone - call', and the number '+44 [REDACTED]'. Below this is a row with a plus sign icon, the text 'Add sign-in method', and icons for 'Change' and 'Delete'. At the bottom left is a note 'Lost device? [Sign out everywhere](#)'.

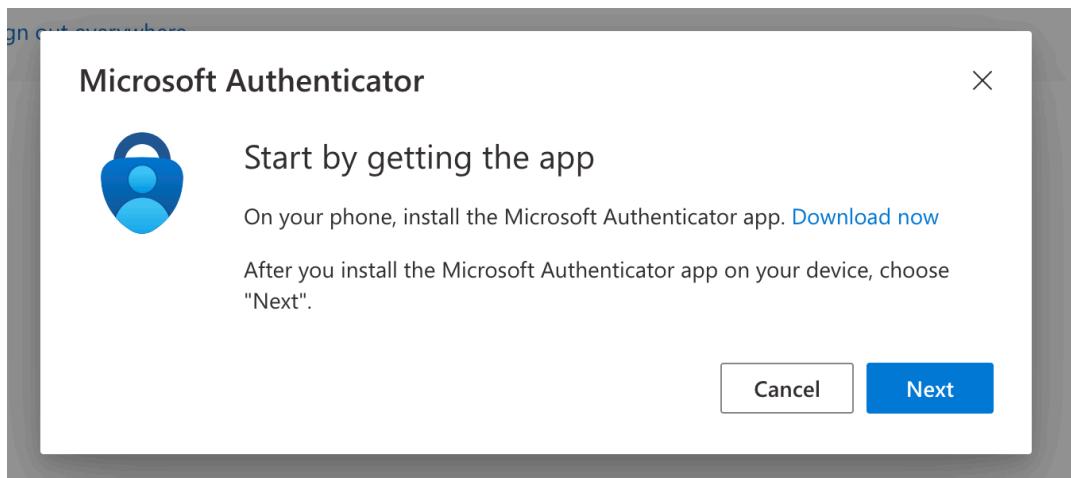
2.3.2 Authenticator app registration

- If you want to use the Microsoft Authenticator app for MFA (which will work if you have wifi but no phone signal) then click on + Add sign-in method and select Authenticator app

Set up your account

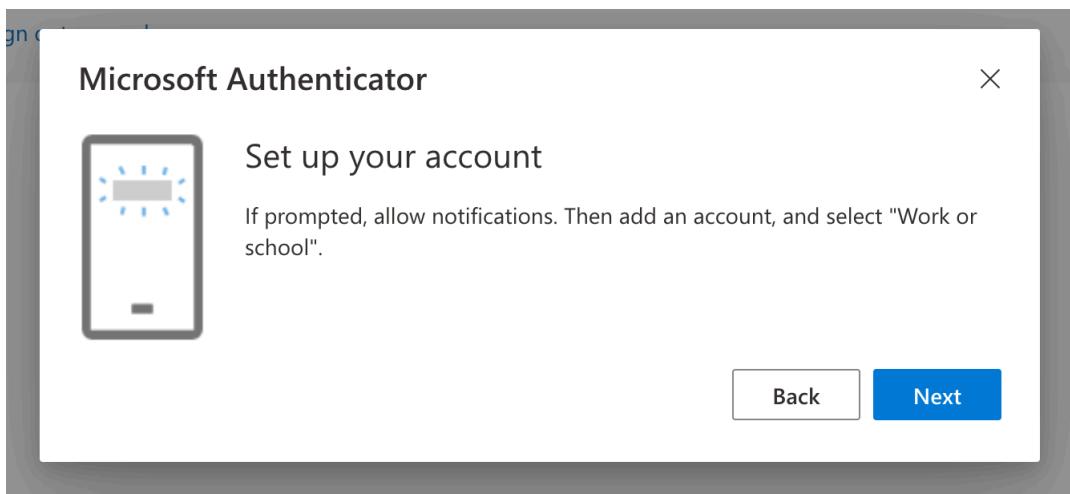


- This will prompt you to download the Microsoft Authenticator phone app.



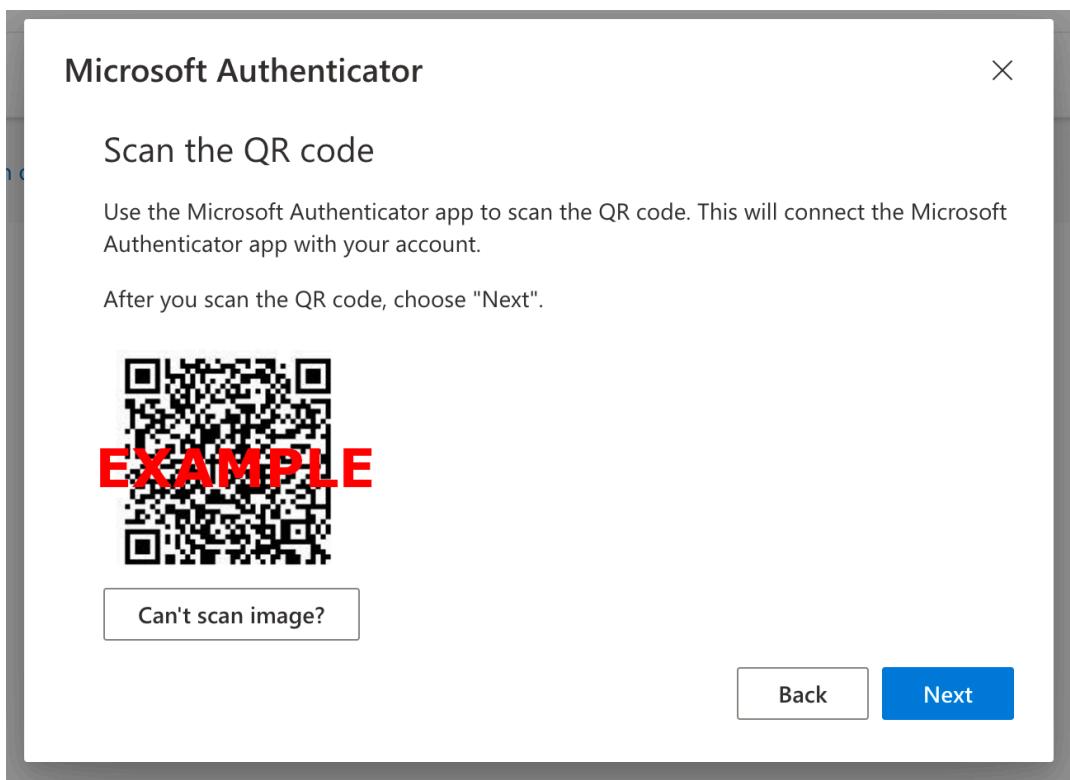
- You can click on the link in the prompt or follow the appropriate link for your phone here:
 -  iOS: <https://bit.ly/iosauthenticator>
 -  Android: <https://bit.ly/androidauthenticator>
 -  Windows mobile: <https://bit.ly/windowsauthenticator>
- You will now be prompted to open the app and:
 - To allow notifications
 - Select Add an account
 - Select Work or School account

2 Set up your account



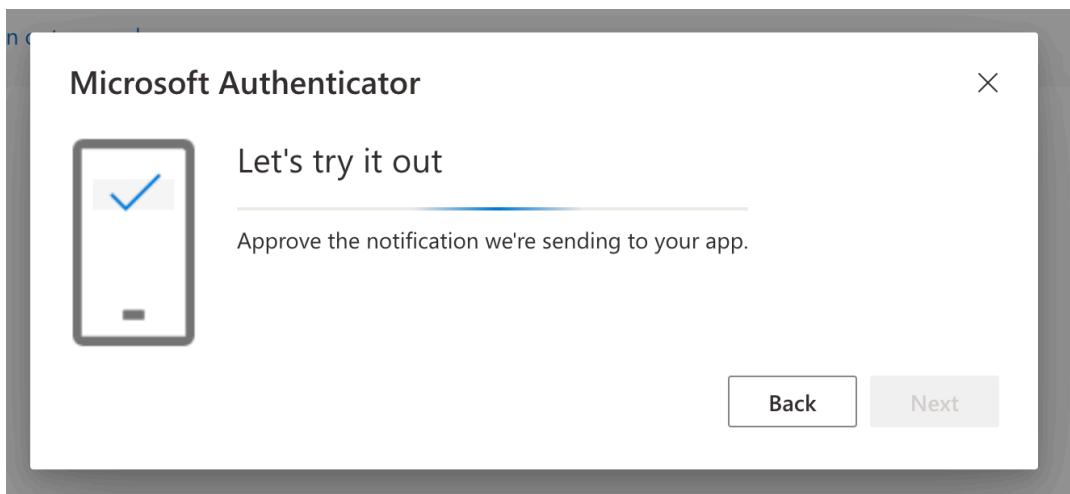
Important You must give permission for the authenticator app to send you notifications for the app to work as an MFA method.

- The next prompt will give you a QR code to scan, like the one shown below
 - Scan the QR code on the screen

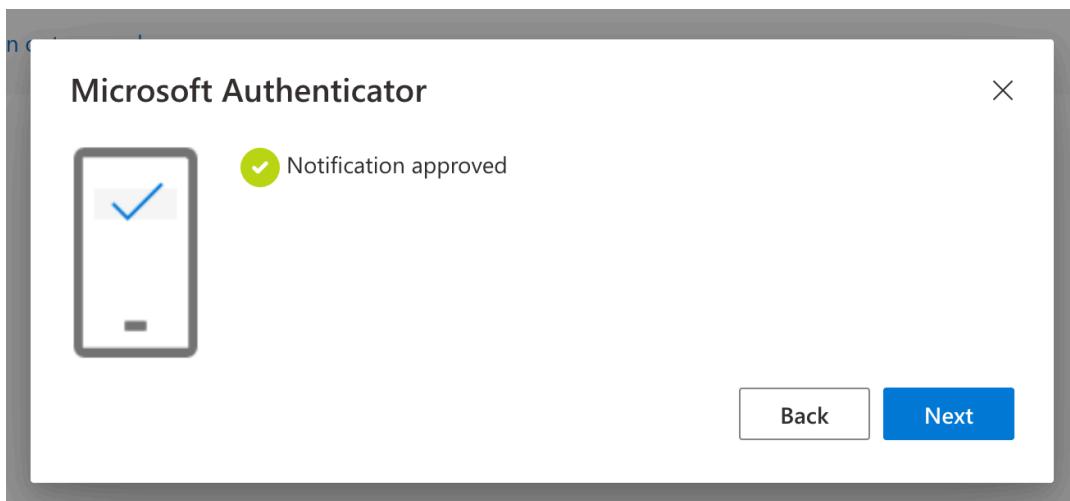


- Once this is completed, Microsoft will send you a test notification to respond to

2 Set up your account



- When you click Approve on the phone notification, you will get the following message in your browser



- You should now be returned to the Security Information dashboard that lists two verified MFA methods

Security info

These are the methods you use to sign into your account or reset your password.

Default sign-in method: Phone - call +44 [REDACTED] [Change](#)

[+ Add sign-in method](#)



Phone

+44 [REDACTED]

[Change](#)

[Delete](#)



Microsoft Authenticator

[Delete](#)

Lost device? [Sign out everywhere](#)

- Choose whichever you prefer to be your Default sign-in methods.
- You have now finished setting up MFA and you can close your browser

2.3.3 Troubleshooting MFA

Sometimes setting up MFA can be problematic. You may find the following tips helpful:

- Make sure you allow notifications on your authenticator app.
- Check you don't have *Do not Disturb* mode on.
- You have to be SUPER FAST at acknowledging the notification on your app, since the access codes update every 30 seconds.
- Sometimes just going through the steps again solves the problem

3 Access the Secure Research Environment

3.1 Prerequisites

After going through the account setup procedure, you should have access to:

- Your username
- Your password
- The SRE URL
- Multifactor authentication

Tip If you aren't sure about any of these then please return to the [Set up your account](#) section above.

3.2 Log into the research environment

- Open a **private/incognito** browser session, so that you don't pick up any existing Microsoft logins
- Go to the SRE URL given by your System Manager.

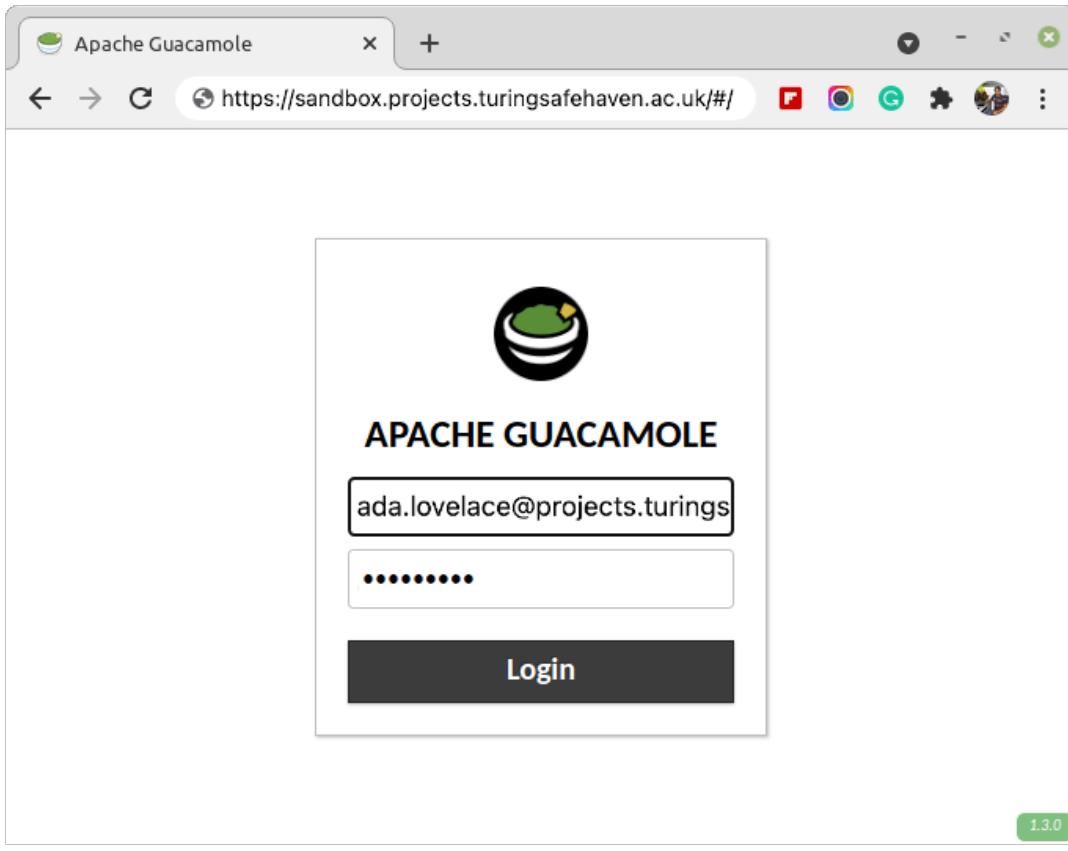
Note Our example user, Ada Lovelace, participating in the sandbox project at a Turing Data Study Group, would navigate to <https://sandbox.projects.turingsafehaven.ac.uk>.

Important Don't forget the **https://** as you will not be able to login without it!

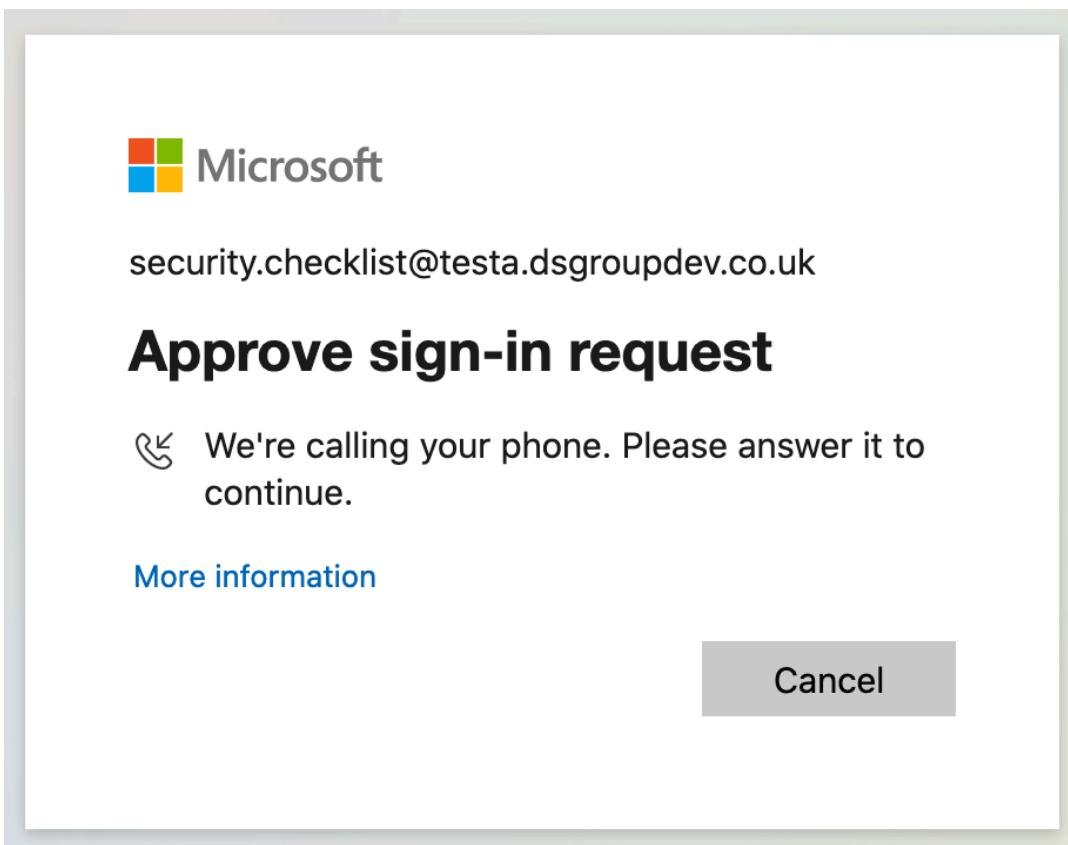
- You should arrive at a login page that needs you to enter:
 - your `username@<username domain>`
 - your passwordthen click Login.
- You should arrive at a login page that looks like the image below:

Note Our example user, Ada Lovelace, participating in the sandbox project at a Turing Data Study Group, would enter `ada.lovelace@projects.turingsafehaven.ac.uk` in the User name box, enter her password and then click Login.

3 Access the Secure Research Environment



- You will now receive a call or mobile app notification to authenticate using multifactor authentication (MFA).



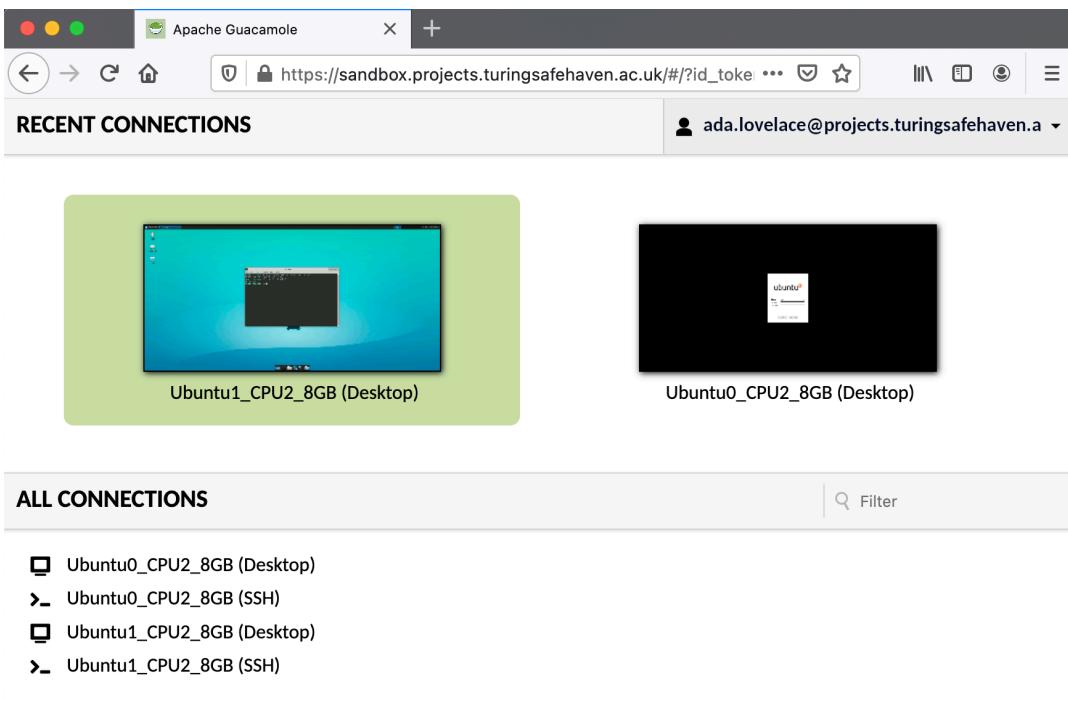
📞 For the call, you may have to move to an area with good reception and/or press the hash (#) key multiple times in-call.

📱 For the app you will see a notification saying "*You have received a sign in verification request*". Go to the app to approve the request.

Caution! If you don't respond to the MFA request quickly enough, or if it fails, you may get an error. If this happens, please retry

- If you are successful, you'll see the a screen with icons for the available apps.

3 Access the Secure Research Environment



Welcome to the Data Safe Haven! 

3.3 Log into the Secure Research Desktop

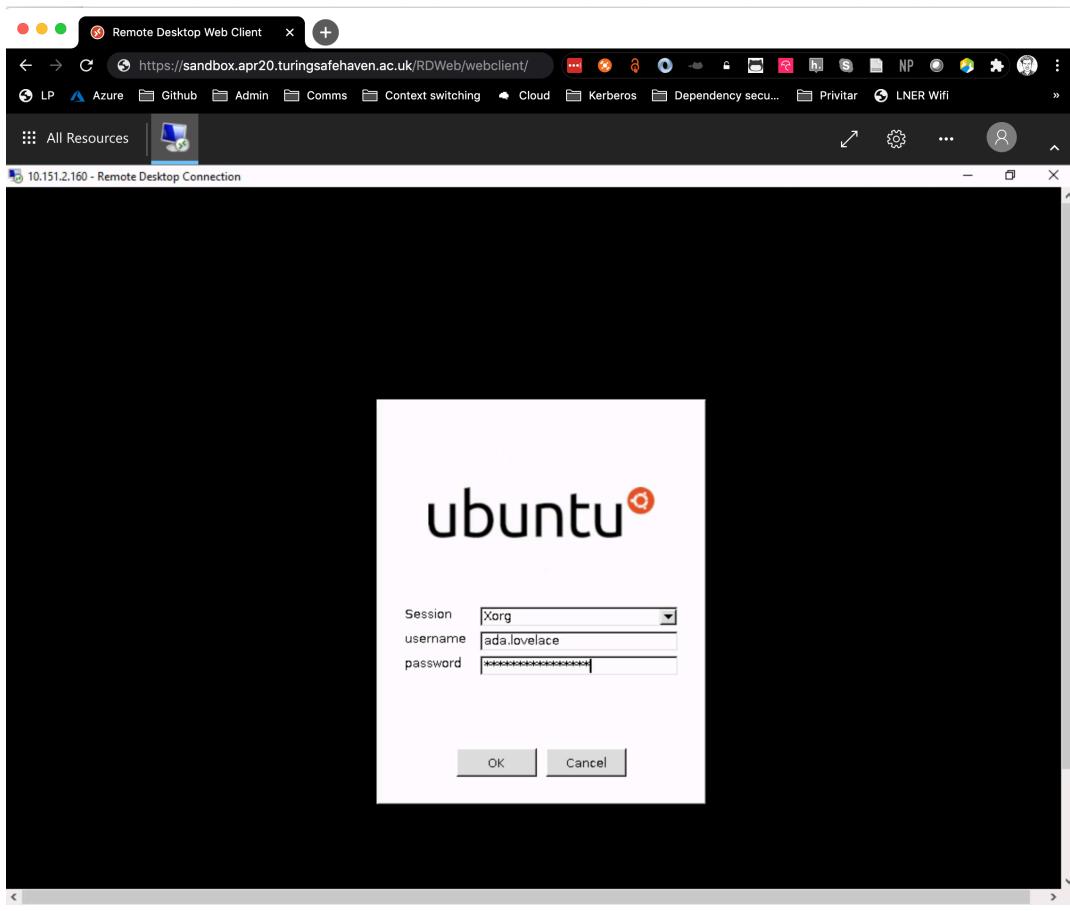
The primary method of performing data analysis in the SRE is using the Secure Research Desktop (SRD).

This is a virtual machine (VM) with many different applications and programming languages pre-installed on it. Once connected to it, you can analyse the sensitive data belonging to your project while remaining completely isolated from the internet.

- Click on one of the Desktop connections (for example Ubuntu0_CPU2_8GB (Desktop) to connect to the desktop.
- Insert your username and password.

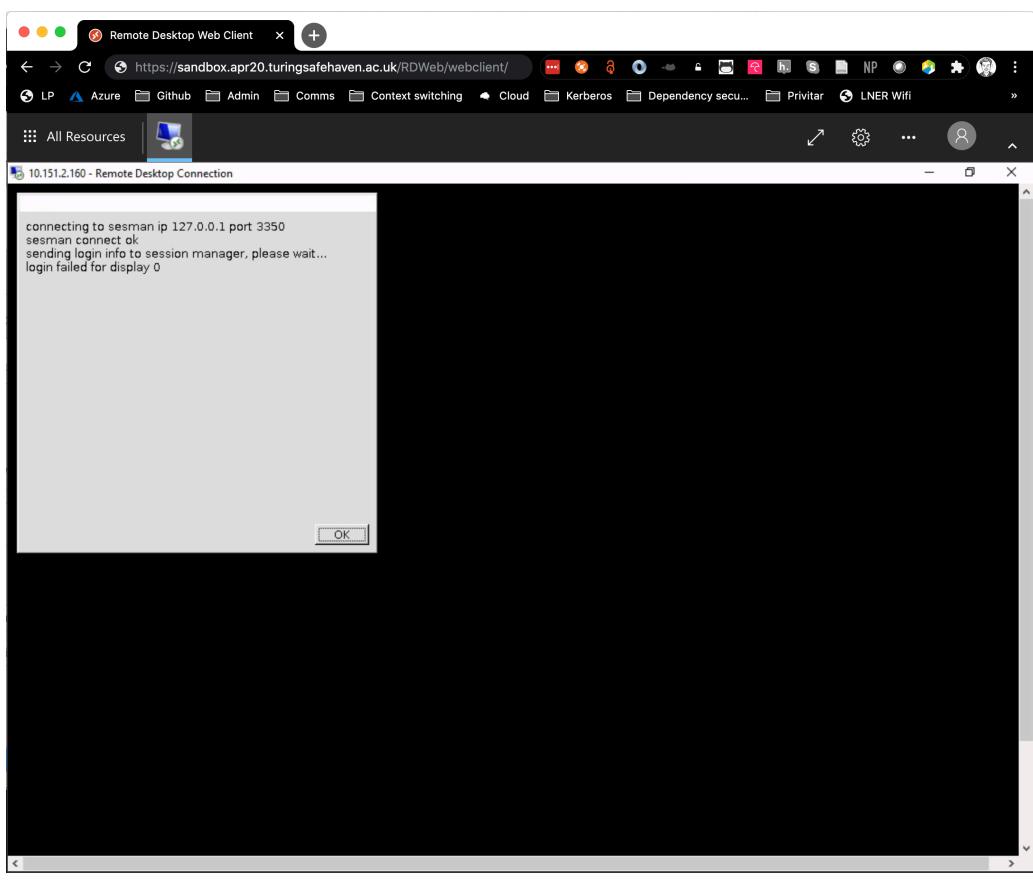
Note Our example user, Ada Lovelace, would enter `ada.lovelace` and her password.

3 Access the Secure Research Environment



Error If you enter your username and/or password incorrectly you will see a warning like the one below. If this happens, please try again, entering your username and password carefully.

3 Access the Secure Research Environment

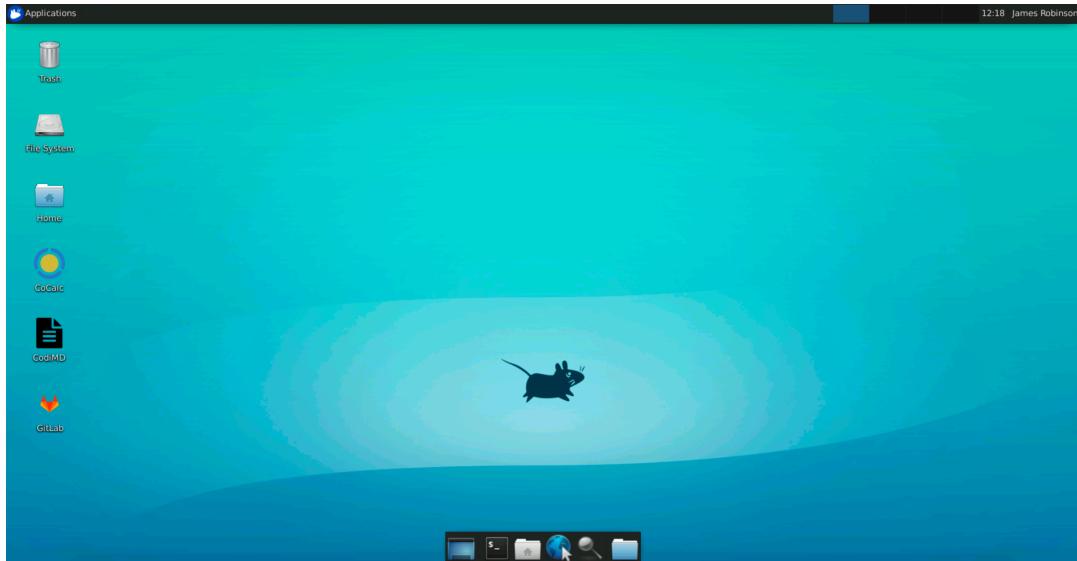


Caution! We recommend *not* including special characters in your password as the keyboard layout expected by the login screen may be different from the one you're using.

- if you want to reset your password, follow the steps defined in the [Set your password](#) section above.
- if you want to continue with special characters in your password, please test that they are being entered correctly by typing them in the username field.

- You should now be greeted by a Linux desktop.

3 Access the Secure Research Environment



You are now logged into the Data Safe Haven SRE! Welcome 

4 Analysing sensitive data

The SRD has several pre-installed applications and programming languages to help with your data analysis.

4.1 Pre-installed applications

4.1.1 Programming languages / compilers

- C
- C++
- CMake compiler
- Fortran
- gcc compilers
- Java
- Julia (plus common data science libraries)
- Microsoft .NET framework
- Python [three most recent versions] (plus common data science libraries)
- R (plus common data science libraries)
- Rust compiler with cargo package manager
- scala
- spark-shell

4.1.2 Editors / IDEs

- emacs
- nano
- PyCharm
- RStudio
- Spyder
- vim
- Visual Studio Code

4.1.3 Writing / presentation tools

- TeXStudio for technical writing
- LaTeX / pdflatex / xelatex for producing different document outputs
- LibreOffice for creating presentations

4.1.4 Database access tools

- Azure Data Studio
- DBeaver
- psql
- sqlcmd

4.1.5 Other useful software

- Firefox
- git
- weka

If you need anything that is not already installed, please discuss this with the designated contact for your SRE.

Attention! This secure research desktop SRD is your interface to a single computer running in the cloud. You may have access to additional SRDs so be careful to check which machine you are working in as files and installed packages may not be the same across the machines.

4.2 Keyboard mapping

When you access the SRD you are actually connecting through the cloud to another computer - via a few intermediate computers/servers that monitor and maintain the security of the SRE.

Caution! You may find that the keyboard mapping on your computer is not the same as the one set for the SRD.

Click on Desktop and Applications > Settings > Keyboard to change the layout.

Tip We recommend opening a text editor (such as Atom , see [Access applications](#) below) to check what keys the remote desktop thinks you're typing – especially if you need to use special characters.

4.3 Access applications

You can access applications from the desktop in two ways: the terminal or via a drop down menu.

Applications can be accessed from the dropdown menu. For example:

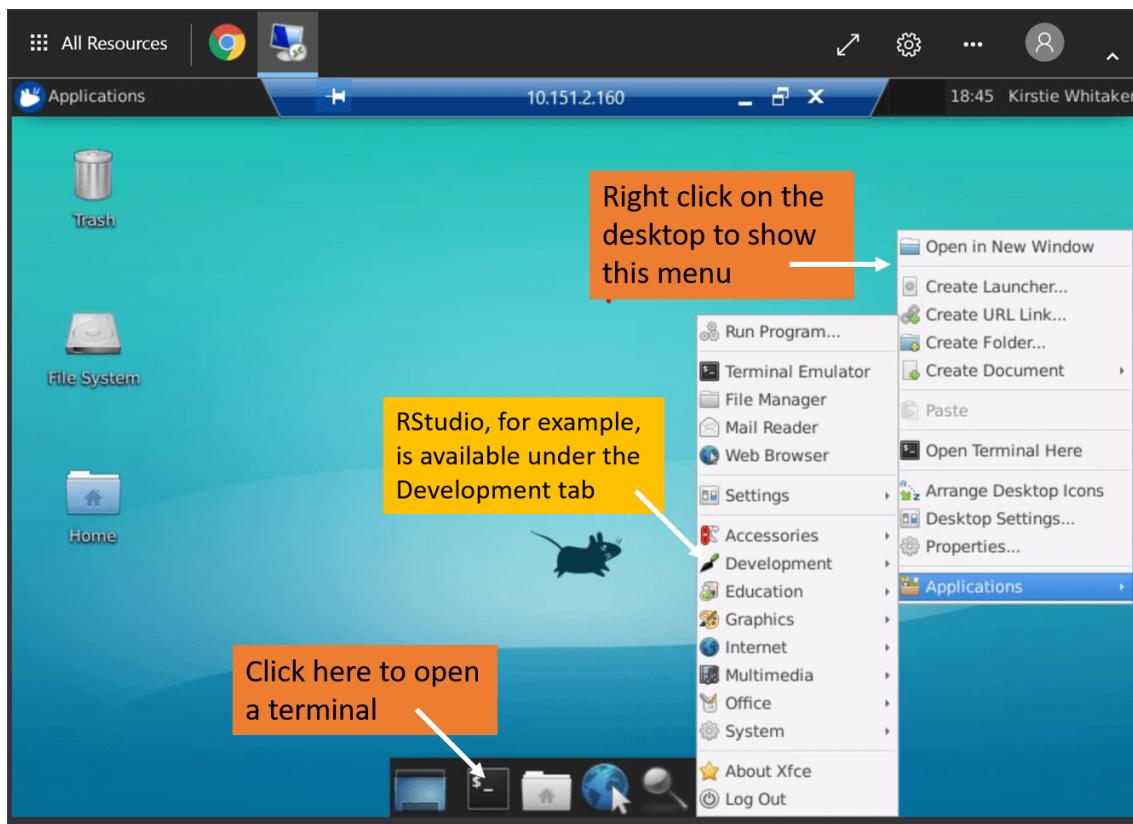
- Applications > Development > Atom
- Applications > Development > Jupyter Notebook
- Applications > Development > PyCharm
- Applications > Development > RStudio

4 🗃 Analysing sensitive data

- Applications > Education > QGIS Desktop

Applications can be accessed from a terminal. For example:

- Open Terminal and run jupyter notebook & if you want to use Python within a jupyter notebook.



4.4 💡 Available Python and R versions

Typing R at the command line will give you the system version of R with many custom packages pre-installed.

There are several versions of Python installed, which are managed through pyenv. You can see the default version (indicated by a '*') and all other installed versions using the following command:

```
> pyenv versions
```

This will give output like:

```
system
3.8.12
* 3.9.10 (set by /home/ada.lovelace/.pyenv_version)
3.10.2
```

You can change your preferred Python version globally or on a folder-by-folder basis using

- pyenv global <version number> (to change the version globally)

- `pyenv local <version number>` (to change the version for the folder you are currently in)

4.4.1 Creating virtual environments

We recommend that you use a dedicated [virtual environment](#) for developing your code in Python. You can easily create a new virtual environment based on any of the available Python versions

```
> pyenv virtualenv 3.8.12 myvirtualenv
```

You can then activate it with:

```
> pyenv shell myvirtualenv
```

or if you want to automatically switch to it whenever you are in the current directory

```
> pyenv local myvirtualenv
```

4.5 📦 Install R and python packages

There are local copies of the PyPI and CRAN package repositories available within the SRE. You can install packages you need from these copies in the usual way, for example `pip install` and `packages.install` for Python and R respectively.

Caution! You **will not** have access to install packages system-wide and will therefore need to install packages in a user directory.

- For CRAN you will be prompted to make a user package directory when you install your [first package](#).
- For PyPI you will need to [install](#) using the `--user` argument to `pip`.

4.5.1 R packages

You can install R packages from inside R (or RStudio):

```
> install.packages(<package-name>)
```

You will see something like the following:

```
Installing package into '/usr/local/lib/R/site-library'  
(as 'lib' is unspecified)  
Warning in install.packages("cluster") :  
  'lib = "/usr/local/lib/R/site-library"' is not writable  
Would you like to use a personal library instead? (yes/No/cancel)
```

Enter yes, which prompts you to confirm the name of the library:

```
Would you like to create a personal library  
'~/R/x86_64-pc-linux-gnu-library/3.5'  
to install packages into? (yes/No/cancel)
```

Enter yes, to install the packages.

4.5.2 Python packages

You can install python packages from a terminal.

```
pip install --user <package-name>
```

Tip If you are using a virtual environment as recommended above, you will not need the --user flag.

4.5.3 Package availability

Depending on the type of data you are accessing, different R and python packages will be available to you (in addition to the ones that are pre-installed):

- Tier 2 (medium security) environments have full mirrors of PyPI and CRAN available.
- Tier 3 (high security) environments only have pre-authorised packages available.

If you need to use a package that is not on the allowlist see the section on how to [bring software or data into the environment](#) below.

5 🔑 Share files with collaborators

5.1 📂 Shared directories within the SRE

There are several shared areas on the SRD that all collaborators within a research project team can see and access:

- input data: /data/
- shared space: /shared/
- scratch space: /scratch/
- output resources: /output/

5.1.1 Input data: /data/

Data that has been “ingressed” - approved and brought into the secure research environment - can be found in the /data/ folder.

Everyone in your group will be able to access it, but it is **read-only**.

Important You will not be able to change any of the files in /data/. If you want to make derived datasets, for example cleaned and reformatted data, please add those to the /shared/ or /output/ directories.

The contents of /data/ will be **identical** on all SRDs in your SRE. For example, if your group requests a GPU-enabled machine, this will contain an identical /data/ folder.

Tip If you are participating in a Turing Data Study Group you will find example slides and document templates in the /data/ drive.

5.1.2 Shared space: /shared/

The /shared/ folder should be used for any work that you want to share with your group. Everyone in your group will be able to access it, and will have **read-and-write access**.

The contents of /shared/ will be **identical** on all SRDs in your SRE. For example, if your group requests a GPU-enabled machine, this will contain an identical /shared/ folder.

5.1.3 Scratch space: /scratch/

The /scratch/ folder should be used for any work-in-progress that isn’t ready to share yet. Although everyone in your group will have **read-and-write access**, you can create your own folders inside /scratch and choose your own permissions for them.

The contents of /scratch/ will be **different** on different VMs in your SRE. For example, if your group requests a GPU-enabled machine, this will have its own /scratch/ folder.

5.1.4 Output resources: /output/

Any outputs that you want to extract from the secure environment should be placed in the /output/ folder on the SRD. Everyone in your group will be able to access it, and will have **read-and-write access**. Anything placed in here will be considered for data egress - removal

5 Share files with collaborators

from the secure research environment - by the project's principal investigator together with the data provider.

Tip You may want to consider having subfolders of `/output/` to make the review of this directory easier.

Hint For the Turing Data Study Groups, we recommend the following categories:

- Presentation
 - Transformed data/derived data
 - Report
 - Code
 - Images
-

5.2 Bring in new files to the SRE

Bringing software into a secure research environment may constitute a security risk. Bringing new data into the SRE may mean that the environment needs to be updated to a more secure tier.

The review of the "ingress" of new code or data will be coordinated by the designated contact for your SRE. They will have to discuss whether this is an acceptable risk to the data security with the project's principle investigator and data provider and the decision might be "no".

Hint You can make the process as easy as possible by providing as much information as possible about the code or data you'd like to bring into the environment and about how it is to be used.

6 ⚡ Collaborate on code using CoCalc

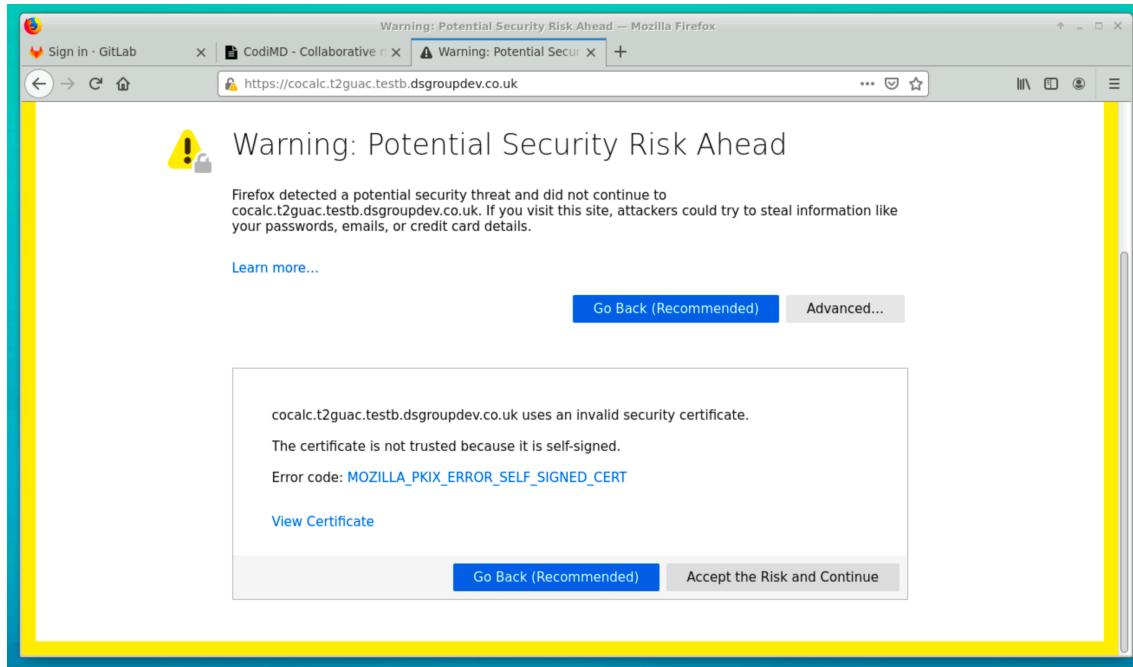
CoCalc is a collaborative calculation and data science environment. It lets you work with others on projects, using Jupyter, LaTeX, Octave, Python or R in collaborative notebooks.

The CoCalc instance within the SRE is the easiest way to work directly with others in your team (for example pair-programming) who might not be physically near you. You do not need to worry about the security of the information you upload there as it is fully contained within the SRE and there is no access to the internet and / or external servers.

Important The CoCalc instance within the SRE is entirely separate from the <https://cocalc.com> service.

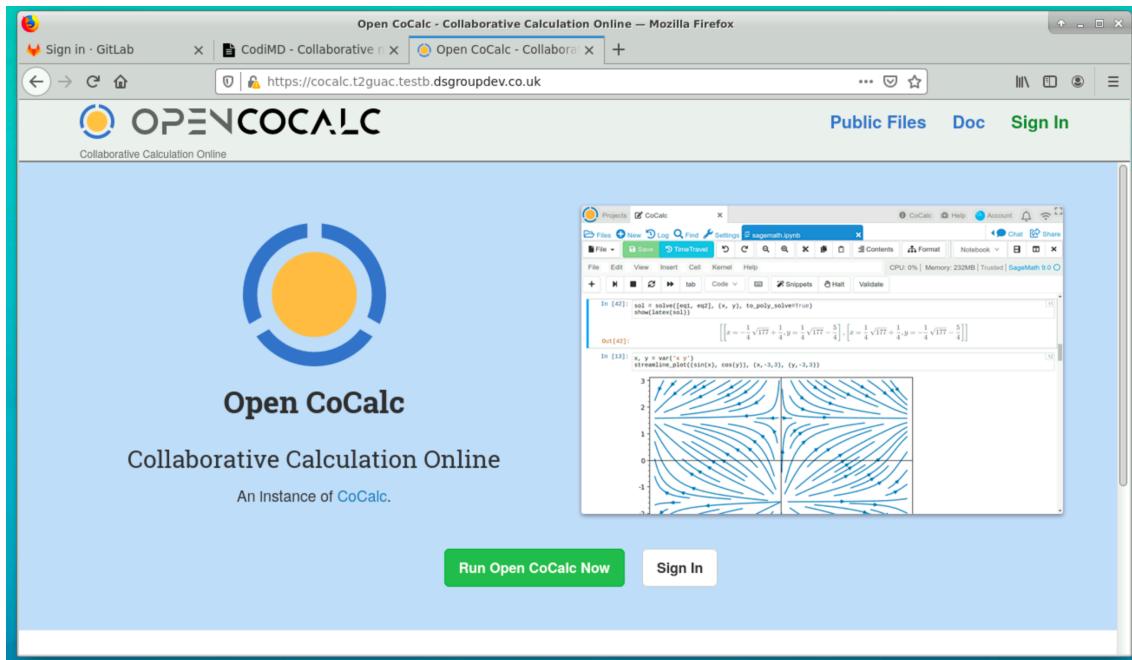
6.1 🔒 Access CoCalc

You can access CoCalc from an internet browser in the SRD using the desktop shortcut. The first time that you login, you will see a security warning. This is expected, please click on Advanced and then Accept the Risk and Continue.



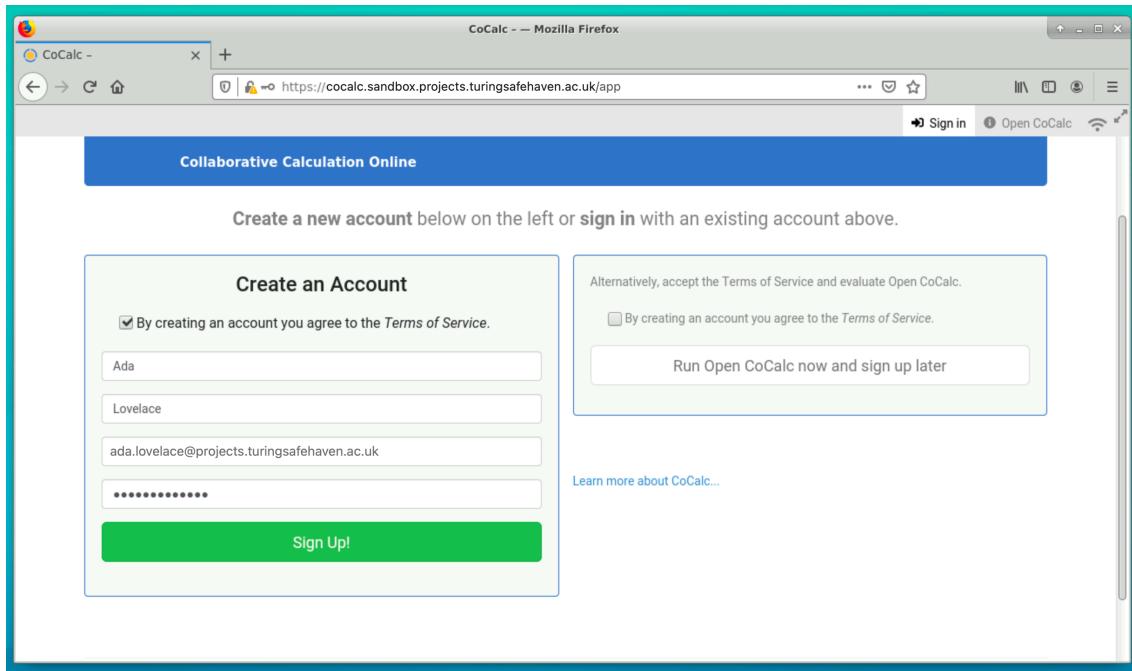
You will then get to the CoCalc homepage where you should click on Sign In

6 Collaborate on code using CoCalc



You will need to create a new account. You can use any username/password here - it is not connected to your main Safe Haven account.

Note Our example user, Ada Lovelace has used ada.lovelace@projects.turingsafehaven.ac.uk as her username and set her own password



7 Versioning code using GitLab

GitLab is a code hosting platform for version control and collaboration - similar to GitHub. It allows you to use `git` to **version control** your work, coordinate tasks using GitLab **issues** and review work using GitLab **merge requests**.

Note GitLab is a fully open source project. This information doesn't matter at all for how you use GitLab within the SRE, but we do want to thank the community for maintaining free and open source software for us to use and reuse. You can read more about GitLab at [their code repository](#).

The GitLab instance within the SRE can contain code, documentation and results from your team's analyses. You do not need to worry about the security of the information you upload there as it is fully contained within the SRE and there is no access to the internet and/or external servers.

Important The GitLab instance within the SRE is entirely separate from the <https://gitlab.com> service.

7.1 Maintaining an archive of the project

The Data Safe Haven SRE is hosted on the Microsoft Azure cloud platform. One of the benefits of having cloud based infrastructure is that it can be deleted forever when the project is over. Deleting the infrastructure ensures that neither sensitive data nor insights derived from the data or modelling techniques persist.

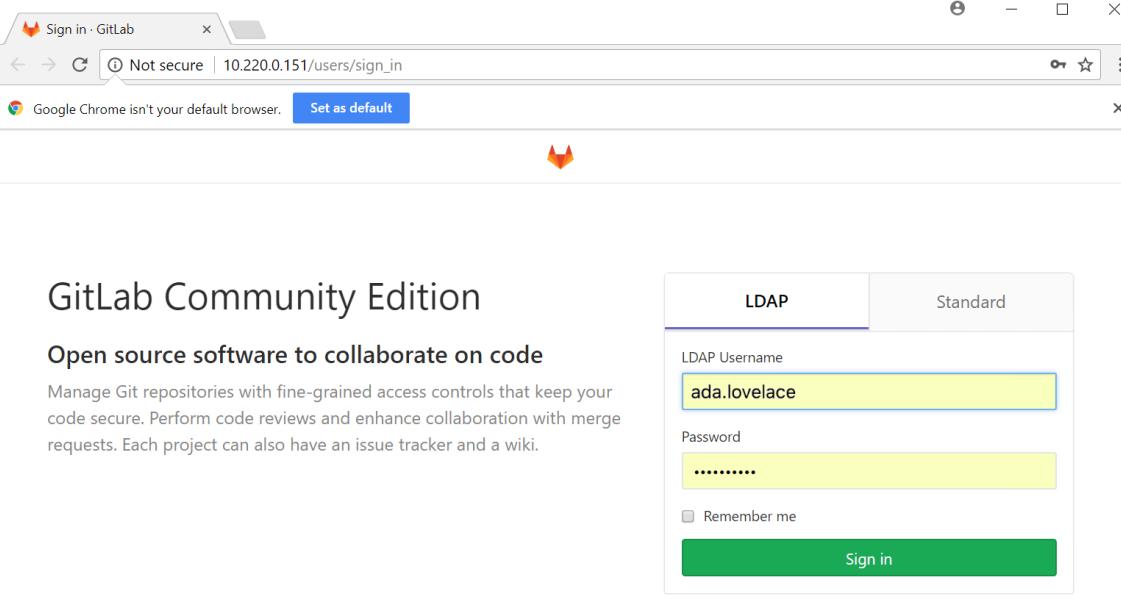
Make sure that every piece of code you think might be useful is stored in a GitLab repository within the secure environment. Any other work should be transferred to the shared /shared/ drive. Anything that you think should be considered for **egress** from the environment (eg. images or processed datasets) should be transferred to the shared /output/ drive.

Caution! If you are participating in a Turing Data Study Group, everything that is not stored in a GitLab repository or on the shared /shared/ or /output/ drives by Friday lunchtime will be **DESTROYED FOR EVER**.

7.2 Access GitLab

You can access GitLab from an internet browser in the SRD using the desktop shortcut. Login with username `firstname.lastname` (the domain is not needed) and password .

Note Our example user, Ada Lovelace would enter `ada.lovelace` in the LDAP Username box, enter her password and then click Sign in .



GitLab Community Edition

Open source software to collaborate on code

Manage Git repositories with fine-grained access controls that keep your code secure. Perform code reviews and enhance collaboration with merge requests. Each project can also have an issue tracker and a wiki.

Accessing GitLab from the browser on the SRD is an easy way to switch between analysis work and documenting the process or results.

Warning Do not use your username and password from a pre-existing GitLab account! The GitLab instance within the SRE is entirely separate from the <https://gitlab.com> service and is expecting the same username and password that you used to log into the SRE.

7.3 🌐 Public repositories within the SRE

The GitLab instance inside the secure research environment is entirely contained *inside* the SRE.

When you make a repository inside the SRE “public” it is visible to your collaborators who also have access to the SRE. A “public” repository within the SRE is only visible to others with the same data access approval, it is not open to the general public via the internet.

Tip We recommend that you make your repositories public to facilitate collaboration within the secure research environment.

7.4 🏗️ Support for GitLab use

If you have not used GitLab before:

- There is a small tutorial available as an [Appendix](#) to this user guide.
- You can find the official documentation on the [GitLab website](#).
- Ask your team mates for help.
- Ask the designated contact for your SRE.
- There may be a dedicated discussion channel, for example during Turing Data Study

Groups you can ask in the Slack channel.

8 Collaborate on documents using CodiMD

CodiMD is a locally installed tool that allows you to collaboratively write reports. It uses Markdown which is a simple way to format your text so that it renders nicely in full HTML.

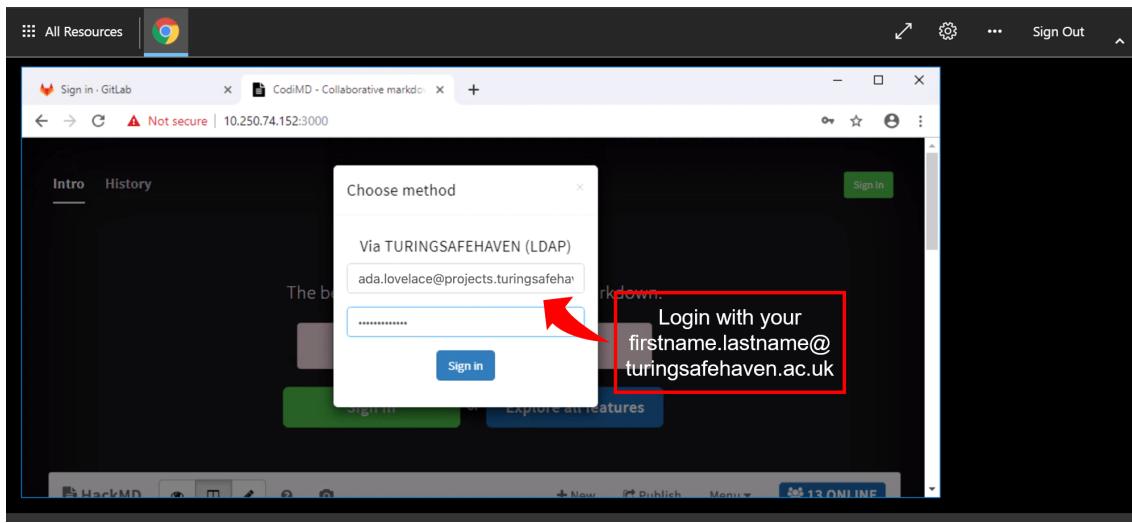
Note CodiMD is a fully open source version of the HackMD software. This information doesn't matter at all for how you use CodiMD within the SRE, but we do want to thank the community for maintaining free and open source software for us to use and reuse. You can read more about CodiMD at their GitHub repository.

We recommend [this Markdown cheat sheet](#).

8.1 Access CodiMD

You can access CodiMD from an internet browser from the SRD using the desktop shortcut. Login with your long-form username `firstname.lastname@<username domain>` and password.

Note Our example user, Ada Lovelace would enter `ada.lovelace@projects.turingsafehaven.ac.uk` in the Username box, enter her password and then click Sign in .



Accessing CodiMD from the browser on the SRD is an easy way to switch between analysis work and documenting the process or results.

8.2 Editing other people's documents

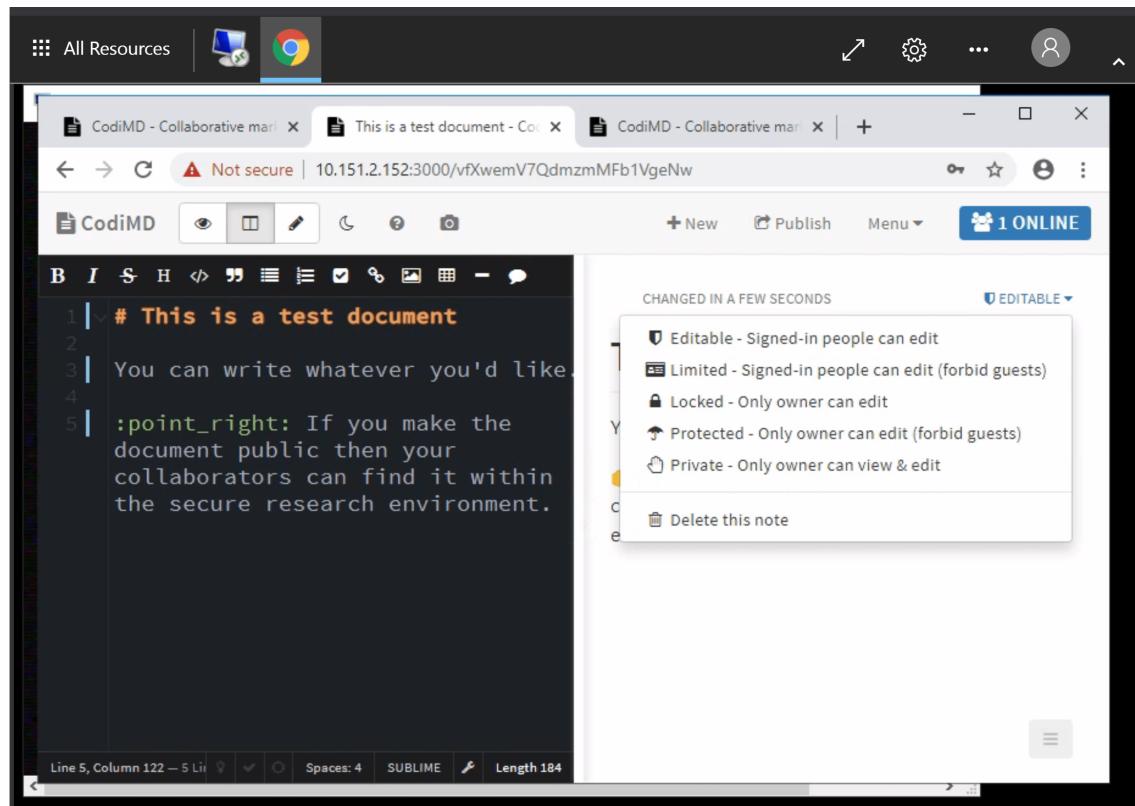
The CodiMD instance inside the secure research environment is entirely contained *inside* the SRE.

When you make a markdown document inside the SRE “editable” your collaborators who also have access to the SRE can access it via the URL at the top of the page. They will have the right

8 Collaborate on documents using CodiMD

to change the file if they are signed into the CodiMD instance.

The link will only work for people who have the same data access approval, it is not open to the general public via the internet.

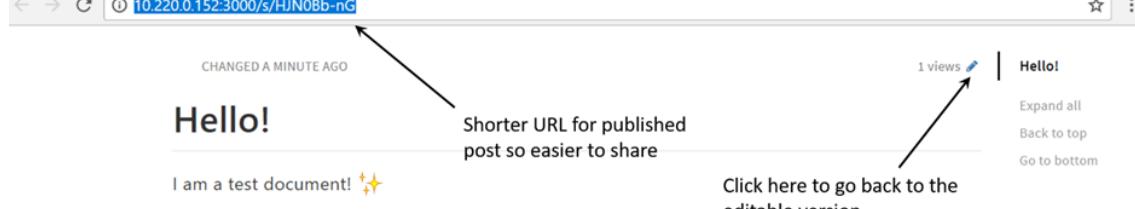


Tip We recommend that you make your documents **editable** to facilitate collaboration within the secure research environment. Alternatively, the **locked** option allows others to read but not edit the document.

The default URL is quite long and difficult to share with your collaborators. We recommend **publishing** the document to get a much shorter URL which is easier to share with others.

Click the Publish button to publish the document and generate the short URL which is easier to share with others.

Click the Publish button to publish the document and generate the short URL which is easier to share with others.



Important Remember that the document is not published to the internet, it is only available to others within the SRE.

Tip If you are attending a Turing Data Study Group you will be asked to write a report describing the work your team undertook over the five days of the projects. Store a copy of the CodiMD URL in a text file in the outputs folder. You will find some example report templates that outline the recommended structure. We recommend writing the report in CodiMD - rather than GitLab - so that everyone can edit and contribute quickly.

8.3 ↴ Troubleshooting CodiMD

We have noticed that a lower case L and an upper case I look very similar and often trip up users in the SRE.

Tip Double check the characters in the URL, and if there are ambiguous ones try the one you haven't tried yet!

Rather than proliferate lots of documents, we recommend that one person is tasked with creating the file and sharing the URL with other team members.

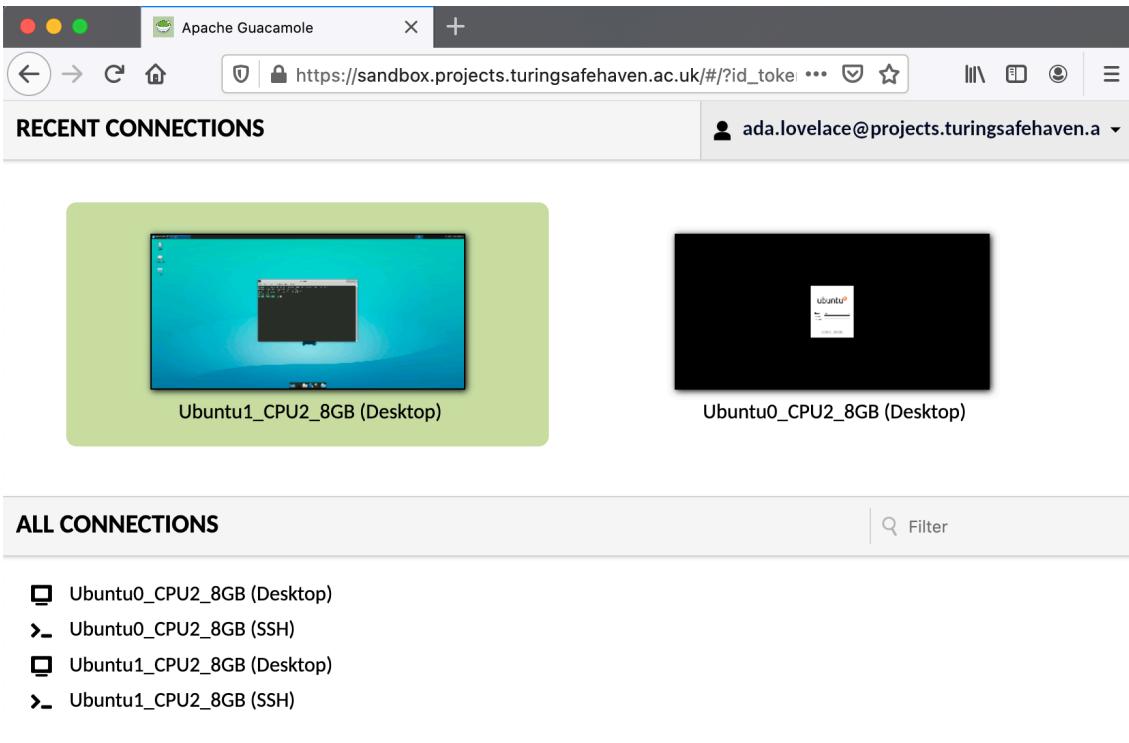
Tip You could use the GitLab wiki or README file to share links to collaboratively written documents.

9 Access additional SRDs

Your project might make use of further SRDs in addition to the main shared desktop. Usually this is because of a requirement for a different type of computing resource, such as access to one or more GPUs (graphics processing units).

You will access this machine in a similar way to the main shared desktop, by selecting a different Desktop connection.

Note Our example user, Ada Lovelace, participating in the sandbox project at a Turing Data Study Group, might select `Ubuntu1_CPU2_8GB (Desktop)` instead of `Ubuntu0_CPU2_8GB (Desktop)`



The screenshot shows the Apache Guacamole web interface. At the top, it displays a connection to 'Ubuntu1_CPU2_8GB (Desktop)' with a green background, indicating it is currently selected. Below this, under 'RECENT CONNECTIONS', is another connection to 'Ubuntu0_CPU2_8GB (Desktop)' with a black background. At the bottom, the 'ALL CONNECTIONS' section lists four entries:

- Ubuntu0_CPU2_8GB (Desktop)
- Ubuntu0_CPU2_8GB (SSH)
- Ubuntu1_CPU2_8GB (Desktop)
- Ubuntu1_CPU2_8GB (SSH)

- This will bring you to the normal login screen, where you use the same username and password credentials as before.
- Any local files that you have created in the `/output/` folder on other VMs (e.g. analysis scripts, notes, derived data) will be automatically available in the new VM.

Tip The naming pattern of the available desktop connections lets you know their compute capabilities. For example `Ubuntu1_CPU2_8GB` has 2 CPUs and 8GB of RAM.

10 Access databases

Your project might use a database for holding the input data. You might also/instead be provided with a database for use in analysing the data. The database server will use either Microsoft SQL or PostgreSQL.

If you have access to one or more databases, you can access them using the following details, replacing <SRE ID> with the SRE ID for your project.

10.1 Microsoft SQL

- Server name: MSSQL-<SRE ID> (e.g. MSSQL-SANDBOX)
- Database name: <provided by your System Manager>
- Port: 1433

10.2 PostgreSQL

- Server name: PSTGRS-<SRE ID> (e.g. PSTGRS-SANDBOX)
- Database name: <provided by your System Manager>
- Port: 5432

Examples are given below for connecting using Azure Data Studio, DBeaver, Python and R. The instructions for using other graphical interfaces or programming languages will be similar.

10.3 Connecting using Azure Data Studio

Azure Data Studio is currently only able to connect to Microsoft SQL databases.

Note Our example user Ada Lovelace, working in the sandbox SRE on the projects.turingsafehaven.ac.uk Safe Haven, would connect using Azure Data Studio as follows:

Connection Details

Connection type	Microsoft SQL Server
Server	MSSQL-SANDBOX.apr20.decovid.turingsafehaven.ac.uk
Authentication type	Windows Authentication
User name	[Redacted]
Password	[Redacted]
<input type="checkbox"/> Remember password	
Database	<Default>
Server group	<Default>
Name (optional)	
Advanced...	
Connect Cancel	

Important Be sure to select Windows authentication here so that your username and password will be passed through to the database.

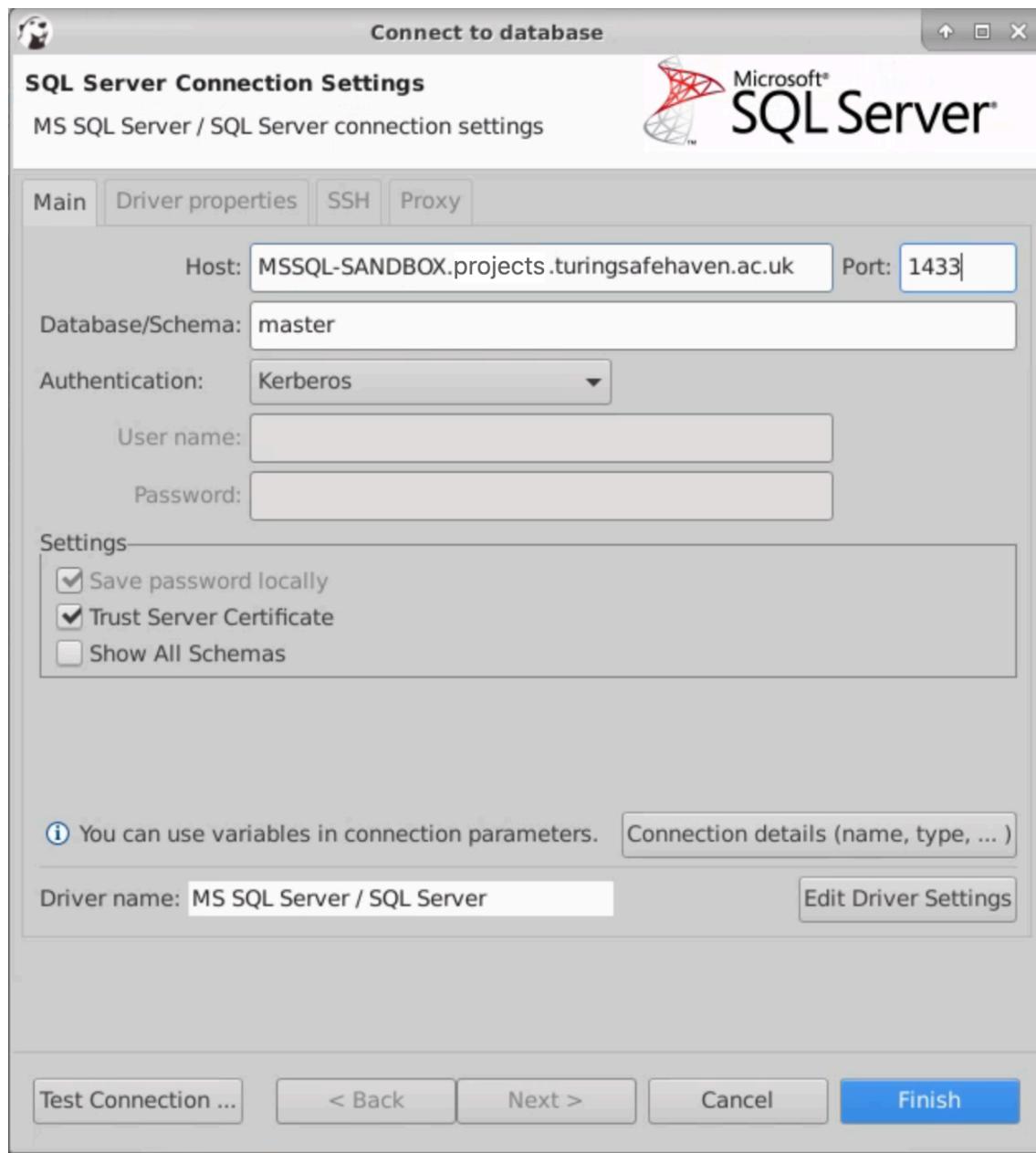
10.4 🐾 Connecting using DBeaver

Click on the New database connection button (which looks a bit like an electrical plug with a plus sign next to it)

10.4.1 Microsoft SQL

- Select SQL Server as the database type
- Enter the necessary information in the Host and Port boxes and set Authentication to Kerberos
- Tick Show All Schemas otherwise you will not be able to see the input data

Note Our example user Ada Lovelace, working in the sandbox SRE on the projects.turingsafehaven.ac.uk Safe Haven, would connect using DBeaver as follows:



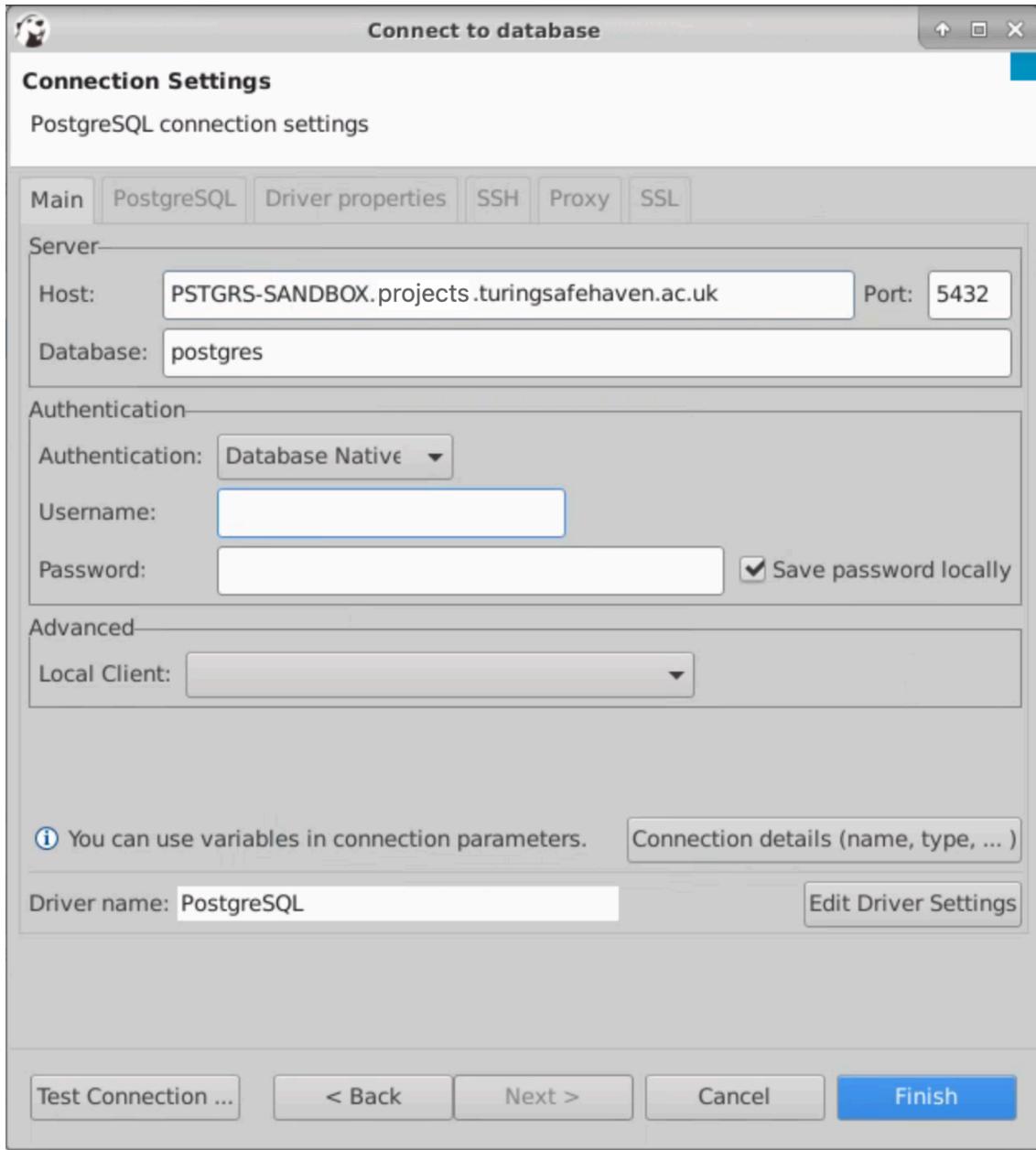
Important Be sure to select Kerberos authentication so that your username and password will be passed through to the database

10.4.2 PostgreSQL

- Select PostgreSQL as the database type
- Enter the necessary information in the Host and Port boxes and set Authentication to Database Native

Important You do not need to enter any information in the Username or Password fields

Note Our example user Ada Lovelace, working in the sandbox SRE on the projects.turingsafehaven.ac.uk Safe Haven, would connect using DBeaver as follows:



Tip If you are prompted for Username or Password when connecting, you can leave these blank and the correct username and password will be automatically passed through to the database



10.5 Connecting using Python

Database connections can be made using pyodbc or psycopg2 depending on which database flavour is being used. The data can be read into a dataframe for local analysis.

Note Our example user Ada Lovelace, working in the sandbox SRE on the projects.turingsafehaven.ac.uk Safe Haven, would connect using DBeaver as follows:

10.5.1 Microsoft SQL

```
import pyodbc
import pandas as pd

server = "MSSQL-SANDBOX.projects.turingsafehaven.ac.uk"
port = "1433"
db_name = "master"

cnxn = pyodbc.connect("DRIVER={ODBC Driver 17 for SQL Server};SERVER=" + 
server + "," + port + ";DATABASE=" + db_name + ";Trusted_Connection=yes;")

df = pd.read_sql("SELECT * FROM information_schema.tables;", cnxn)
print(df.head(3))
```

10.5.2 PostgreSQL

```
import psycopg2
import pandas as pd

server = "PSTGRS-SANDBOX.projects.turingsafehaven.ac.uk"
```

```
port = 5432
db_name = "postgres"

cnxn = psycopg2.connect(host=server, port=port, database=db_name)
df = pd.read_sql("SELECT * FROM information_schema.tables;", cnxn)
print(df.head(3))
```

10.6 Connecting using R

Database connections can be made using `odbc` or `RPostgres` depending on which database flavour is being used. The data can be read into a dataframe for local analysis.

Note Our example user Ada Lovelace, working in the sandbox SRE on the `projects.turingsafehaven.ac.uk` Safe Haven, would connect using DBeaver as follows:

10.6.1 Microsoft SQL

```
library(DBI)
library(odbc)

# Connect to the databases
cnxn <- DBI::dbConnect(
  odbc::odbc(),
  Driver = "ODBC Driver 17 for SQL Server",
  Server = "MSSQL-SANDBOX.projects.turingsafehaven.ac.uk,1433",
  Database = "master",
  Trusted_Connection = "yes"
)

# Run a query and save the output into a dataframe
df <- dbGetQuery(cnxn, "SELECT * FROM information_schema.tables;")
head(df, 3)
```

10.6.2 PostgreSQL

```
library(DBI)
library(RPostgres)

# Connect to the databases
cnxn <- DBI::dbConnect(
  RPostgres::Postgres(),
  host = "PSTGRS-SANDBOX.projects.turingsafehaven.ac.uk",
  port = 5432,
  dbname = "postgres"
)

# Run a query and save the output into a dataframe
df <- dbGetQuery(cnxn, "SELECT * FROM information_schema.tables;")
head(df, 3)
```

11 Report a bug

The Data Safe Haven SRE has been developed in close collaboration with our users: you!

We try to make the user experience as smooth as possible and this document has been greatly improved by feedback from project participants and researchers going through the process for the first time. We are constantly working to improve the SRE and we really appreciate your input and support as we develop the infrastructure.

Important If you find problems with the IT infrastructure, please contact the designated contact for your SRE.

11.1 Help us to help you

To help us fix your issues please do the following:

- Make sure you have **read this document** and checked if it answers your query.
 - Please do not log an issue before you have read all of the sections in this document.
- Log out of the SRE and log back in again at least once
 - Re-attempt the process leading to the bug/error at least twice.
 - We know that “turn it off and turn it back on again” is a frustrating piece of advice to receive, but in our experience it works rather well! (Particularly when there are lots of folks trying these steps at the same time.)
 - The multi-factor authentication step in particular is known to have quite a few gremlins.
 - If you are getting frustrated, log out of everything, turn off your computer, take a 15 minute coffee break, and then start the process from the beginning.
- Write down a comprehensive summary of the issue.
 - A really good bug report makes it much easier to pin down what the problem is.

Please include:

- Your computer’s operating system and operating system version.
 - Precise condition under which the error occurs.
 - What steps would someone need to take to get the exact same error?
 - A precise description of the problem.
 - What happens? What would you expect to happen if there were no error?
 - Any workarounds/fixes you have found.
- Send the bug report to the designated contact for your SRE.

Hint We very strongly recommend “rubber ducking” this process before you talk to the designated contact for your SRE. Either talk through to your imaginary rubber duck, or find a team member to describe the error to, as you write down the steps you have taken. It is amazing how

11 Report a bug

often working through your problem out loud helps you realise what the answer might be.

12 **Acknowledgments**

This user guide is based on an initial document written in March/April 2018 by Kirstie Whitaker.

Updates:

- December 2018 by Catherine Lawrence, Franz Király, Martin O'Reilly, and Sebastian Vollmer.
- March/April 2019 by Miguel Morin, Catherine Lawrence, Alvaro Cabrejas Egea, Kirstie Whitaker, James Robinson and Martin O'Reilly.
- November 2019 by Ben Walden, James Robinson and Daisy Parry.
- April 2020 by Jules Manser, James Robinson and Kirstie Whitaker.
- November 2021 by James Robinson

13 ↴ Appendix A: Command Line Basics

If you have never used a Linux desktop before, you might find some of the following commands useful if you are using a terminal.

Go into a project directory to work in it

```
cd NAME-OF-PROJECT
```

Go back one directory

```
cd ..
```

List what's in the current directory

```
ls
```

Create a new directory

```
mkdir NAME-OF-YOUR-DIRECTORY
```

Remove a file

```
rm NAME-OF-FILE
```

Remove a directory and all of its contents

```
rm -r NAME-OF-DIRECTORY
```

View command history

```
history
```

Show which directory I am in

```
pwd
```

Clear the shell window

```
clear
```


14 Appendix B: Gitlab tutorial notes

GitLab can be thought of as a local version of GitHub - that is a git server along with useful features such as:

- **Project wiki** - exactly what it says
- **Project pastebin** - share bits of code
- **Project issue tracker** - track things TODO and bugs
- **Pull requests** - Way to keep track of changes individuals have made to be included in master

Some teams design their entire workflows around these things. A comparison in terms of features can be found [here](#).

14.1 Getting started with Git

If you have never used `git` before, you might want to take a look at an introductory guide. There are multiple `git` cheat sheets such as [this one from the JIRA authors](#) and [this interactive one](#) and .

14.2 Repositories

A repository is usually used to organize a single project. Repositories can contain folders and files, images, videos, spreadsheets, and data sets – anything your project needs. We recommend including a `README`, or a file with information about your project. Over the course of the work that you do in your SRE, you will often be accessing and adding files to the same project repository.

14.3 Add your Git username and set your email

It is important to configure your `git` username and email address, since every `git` commit will use this information to identify you as the author. On your shell, type the following command to add your username:

```
git config --global user.name "YOUR_USERNAME"
```

Then verify that you have the correct username:

```
git config --global user.name
```

To set your email address, type the following command:

```
git config --global user.email "your_email_address@example.com"
```

To verify that you entered your email correctly, type:

```
git config --global user.email
```

You'll need to do this only once, since you are using the `--global` option. It tells Git to always

use this information for anything you do on that system. If you want to override this with a different username or email address for specific projects, you can run the command without the `--global` option when you're in that project.

14.4 Cloning projects

In git, when you copy a project you say you “clone” it. To work on a git project in the SRD, you will need to clone it. To do this, sign in to GitLab.

When you are on your Dashboard, click on the project that you'd like to clone. To work in the project, you can copy a link to the git repository through a SSH or a HTTPS protocol. SSH is easier to use after it's been set up, [you can find the details here](#). While you are at the Project tab, select HTTPS or SSH from the dropdown menu and copy the link using the Copy URL to clipboard button (you'll have to paste it on your shell in the next step>).



Go to your computer's shell and type the following command with your SSH or HTTPS URL:

```
git clone <PASTE HTTPS OR SSH HERE>
```

14.5 Branches

Branching is the way to work on different versions of a repository at one time. By default your repository has one branch usually named `master` or `main` which is considered to be the definitive branch. We use branches to experiment and make edits before committing them to `main`.

When you create a branch off the `main` branch, you're making a copy, or snapshot, of `main` as it was at that point in time. If someone else made changes to the `main` branch while you were working on your branch, you could pull in those updates.

To create a branch:

```
git checkout -b NAME-OF-BRANCH
```

Work on an existing branch:

```
git checkout NAME-OF-BRANCH
```

To merge the `main` branch into a created branch you need to be on the created branch.

```
git checkout NAME-OF-BRANCH  
git merge main
```

To merge a created branch into the `main` branch you need to be on the created branch.

```
git checkout main  
git merge NAME-OF-BRANCH
```

14.6 Downloading the latest changes in a project

This is for you to work on an up-to-date copy (it is important to do this every time you start working on a project), while you set up tracking branches. You pull from remote repositories to get all the changes made by users since the last time you cloned or pulled the project. Later, you can push your local commits to the remote repositories.

```
git pull REMOTE NAME-OF-BRANCH
```

When you first clone a repository, `REMOTE` is typically `origin`. This is where the repository came from, and it indicates the SSH or HTTPS URL of the repository on the remote server. `NAME-OF-BRANCH` is usually `main`, but it may be any existing branch.

14.7 Add and commit local changes

You'll see your local changes in red when you type `git status`. These changes may be new, modified, or deleted files/folders. Use `git add` to stage a local file/folder for committing. Then use `git commit` to commit the staged files:

```
git add FILE OR FOLDER  
git commit -m "COMMENT TO DESCRIBE THE INTENTION OF THE COMMIT"
```

To add and commit all local changes in one command:

```
git add .  
git commit -m "COMMENT TO DESCRIBE THE INTENTION OF THE COMMIT"
```

To push all local commits to the remote repository:

```
git push REMOTE NAME-OF-BRANCH
```

For example, to push your local commits to the `main` branch of the `origin` remote:

```
git push origin main
```

To delete all local changes in the repository that have not been added to the staging area, and leave unstaged files/folders, type:

```
git checkout .
```

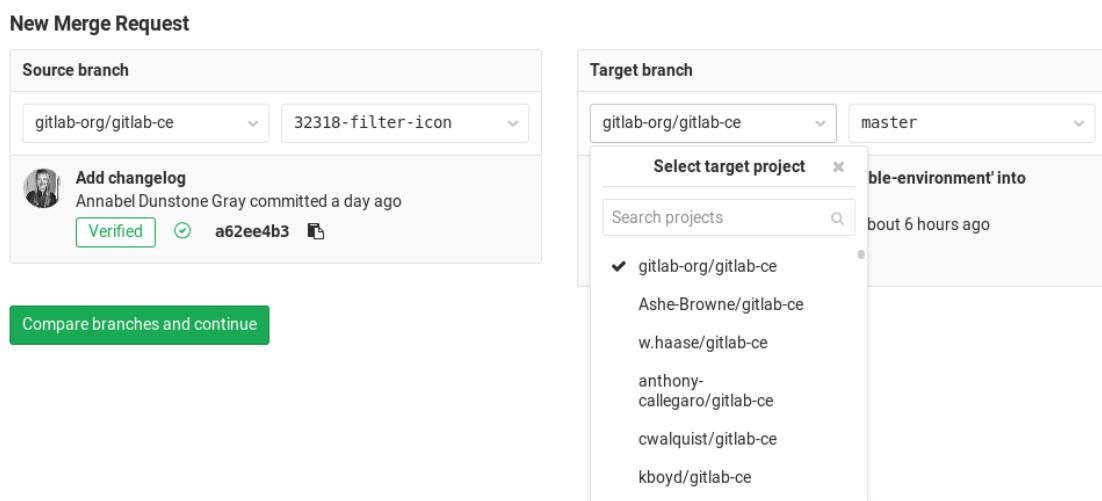
Note: The `.` character typically means all in Git.

14.8 How to create a Merge Request

Merge requests are useful to integrate separate changes that you've made to a project, on different branches. This is a brief guide on how to create a merge request. For more information, check the [merge requests documentation](#).

- Before you start, you should have already created a branch and pushed your changes to GitLab.
- Go to the project where you'd like to merge your changes and click on the Merge requests tab.
- Click on New merge request on the right side of the screen.
- From there on, you have the option to select the source branch and the target branch you'd like to compare to.

The default target project is the upstream repository, but you can choose to compare across any of its forks.



- When ready, click on the Compare branches and continue button.
- At a minimum, add a title and a description to your merge request.
- Optionally, select a user to review your merge request and to accept or close it. You may also select a milestone and labels.

14 Appendix B: Gitlab tutorial notes

New Merge Request

From **feature/issues-board** into **master**

Change branches

Title Feature/issues board

Start the title with **WIP:** to prevent a **Work In Progress** merge request from being merged before it's ready.

Description

Write **Preview**

Write a comment or drag your files here...

B I " <> []

Styling with **Markdown** and **slash commands** are supported

Attach a file

Assignee Assign to me

Milestone Create new milestone

Labels Create new label

Source branch

Target branch Change branches

Remove source branch when merge request is accepted.

Submit merge request Please review the [contribution guidelines](#) for this project. **Cancel**

Commits 29 Builds 43 Changes 46

Most recent commits displayed first

11 Oct, 2016 29 commits

 Updated to pass the board ID with the boards root to save conflicts with EE
Phil Hughes authored about an hour ago



- When ready, click on the Submit merge request button.

Your merge request will be ready to be approved and merged.

15 ↴ Appendix C: Troubleshooting

15.1 ! No applications available

15.1.1 Symptom

- You can successfully log into the remote desktop web interface

Note For our example user, Ada Lovelace, this would be <https://sandbox.projects.turingsafehaven.ac.uk>.

- You do not see any available connections



15.1.2 Cause

You have not been added to the correct SRE security group.

15.1.3 Solution

Ask your System Manager to add you to the appropriate SRE security group.

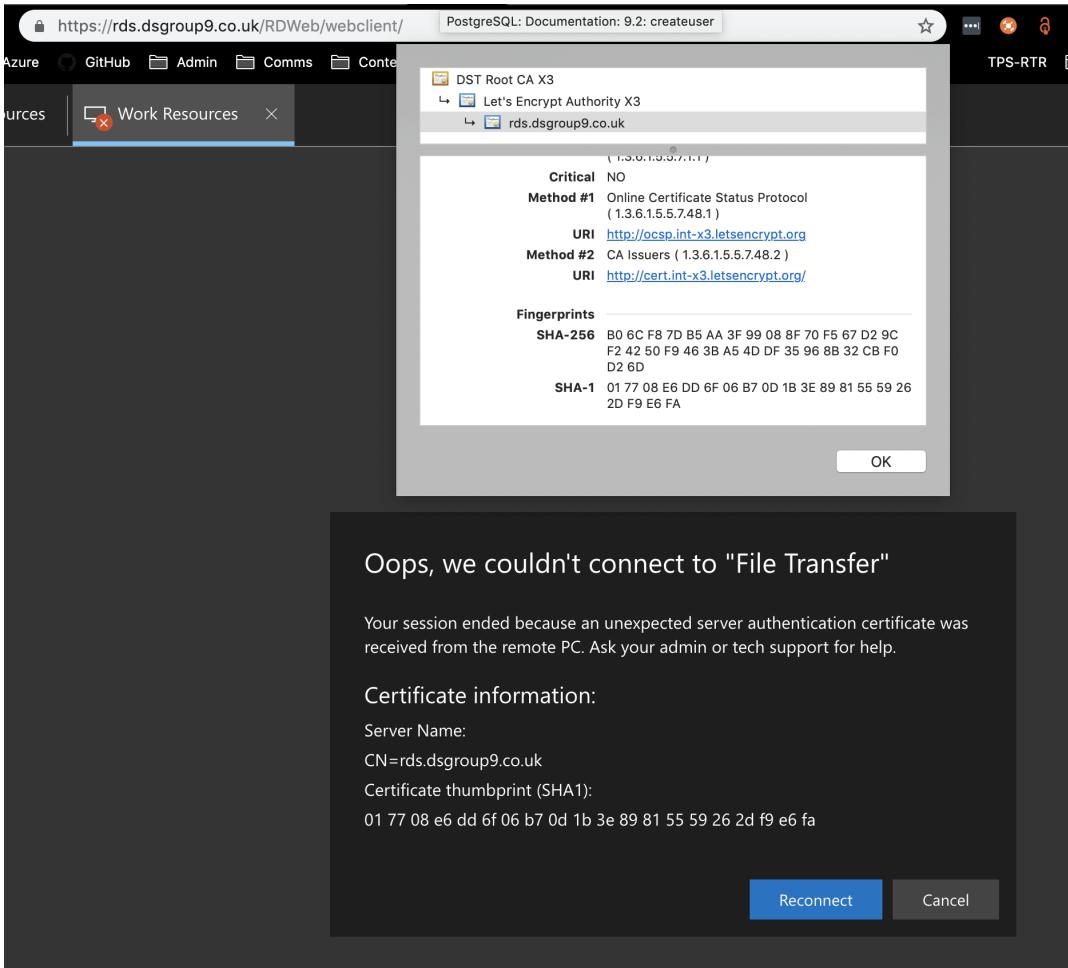
15.2 ! Unexpected certificate error

15.2.1 Symptom

- You can successfully log into the remote desktop web interface

Note For our example user, Ada Lovelace, this would be <https://sandbox.projects.turingsafehaven.ac.uk>.

- You can see several apps, but when you try to launch one of them, you receive an error saying “Your session ended because an unexpected server authentication certificate was received from the remote PC.”
- When you click on the padlock icon in the address bar and view the SSL certificate, the “SHA-1 Fingerprint” in the certificate matches the “SHA-1 Thumbprint” in the error message.



15.2.2 Cause

The SSL certificate protecting your connection to the RDS webclient expires every three months and is renewed every two months. The new SSL certificate is seamlessly picked up by your browser when connecting to the web page. However, the webclient downloads a separate copy of the certificate for its own use to validate connections to the apps it serves. This downloaded certificate is cached by your browser, which means that the old certificate will continue to be used by the web app when the browser is allowed to load things from its cache.

15.2.3 Solution

Get your browser to do a **hard reload** of the page (instructions vary by browser and operating system). You may also need to **clear your cache** for this site. In either case, removing locally cached data should mean that you retrieve a copy of the new certificate.