

SPACe Documentation

None

Table of Contents

1. Home	3
1.1 Welcome to the SPACe Documentation	3
1.2 Contact information	4
1.3 Accessibility statement	5
1.4 Download this documentation	7
2. About our services	8
2.1 Workbench Overview	8
2.2 Getting onto the Workbench	9
2.3 data.gov.scot	13
3. Analytical Workbench user guide	15
3.1 Welcome to the user guide for the SPACe Analytical Workbench	15
3.2 Getting started	16
3.3 Security	39
3.4 Using your Linux VM	43
3.5 Using the software on your VM	49
4. Shiny server user guide	64
4.1 The R Shiny Info	64
4.2 Overview	65
4.3 Application Development	66
4.4 Test-VM	72
4.5 The prod-VM	73
4.6 Deployment process for the production service	77
4.7 Information for the super user	79
4.8 Trouble Shooting	83
4.9 Docker	86
4.10 Security	88

1. Home

1.1 Welcome to the SPACe Documentation

This website hosts the user guides for the SPACe Analytical Workbench (AWB) and Shiny app hosting services.

 **Note**

This is a living document. Things may change as the service evolves, and some parts of the documentation may be incomplete.

1.1.1 About our services

For more information on the services we offer, see the "[Analytical Workbench overview](#)."

To get started with the Analytical Workbench, see the "[Analytical Workbench quick start](#)."

1.1.2 Analytical Workbench user guide

For more information, see the "[Analytical Workbench user guide](#)."

1.1.3 Shiny server user guide

For more information, see the "[Shiny server user guide](#)."

1.2 Contact information

For all enquiries about the SPACe service, please contact us by email.

1.2.1 Email

analytical.workbench@gov.scot

1.2.2 Address

Scottish Government
St. Andrew's House
Regent Road
Edinburgh
EH1 3DG

1.3 Accessibility statement

This accessibility statement applies to the SPACe user documentation. It does not apply to any of the SPACe services themselves, or any other content found on any SPACe service.

This website is run by the Scottish Government and the Scottish Public Sector Analytical Collaborative (SPACe). We want it to be accessible and usable for as many people as possible.

We try to make our site more accessible by using:

- Easier-to-read colours.
- Larger fonts.
- Images with captions and alternative text.
- Plain English, where possible.

You should be able to:

- Change the zoom level, window size, and window orientation.
- Navigate the website using a keyboard.
- Navigate the website using speech recognition software.
- Listen to all content on the website using a screen reader.

1.3.1 Feedback and contact information

Please contact us if you:

- Cannot access any part of this site.
- Want to report an accessibility problem.
- Need information on this website in a different format. Such as: accessible PDF, large print, easy read, audio recording, or braille. We will consider your request and get back to you as soon as we can.

Reporting accessibility problems with this website

We are always looking to improve the accessibility of this website.

If you find any problems not listed on this page, or think we are not meeting accessibility requirements, please contact us.

Contact information

Please contact the [SPACe helpdesk](#).

Enforcement procedure

The Equality and Human Rights Commission enforces the [accessibility regulations](#) (the Public Sector Bodies (Websites and Mobile Applications) (No. 2) Accessibility Regulations 2018).

If you're not happy with how we respond to your feedback, contact the [Equality Advisory and Support Service](#). They are an independent advice service. They will advise you on what to do next.

1.3.2 Technical information about this website's accessibility

We commit to making our websites accessible, in accordance with the [accessibility regulations](#) (the Public Sector Bodies (Websites and Mobile Applications) (No. 2) Accessibility Regulations 2018).

Compliance status

This website is partially compliant with the [Web Content Accessibility Guidelines version 2.1 A and AA success criteria](#).

Non-accessible content

The content listed below is non-accessible for the following reasons.

NON-COMPLIANCE WITH THE ACCESSIBILITY REGULATIONS

- Top level navigation cannot be navigated using a keyboard if the top level navigation bar has been scrolled off-screen.
- Some linked document formats such as PDF and Microsoft Word documents may not be fully accessible. Where possible we make these documents accessible and provide a suitable HTML alternative.

NOT WITHIN THE SCOPE OF THE REGULATIONS

- Some pages are not written in plain English. This is under review.

1.3.3 Preparation of this accessibility statement

This statement was prepared on 16 November 2022, and last reviewed on 16 November 2022.

This website was last tested on 16 November 2022. This was an internal, non-expert assessment.

1.4 Download this documentation

This PDF file contains every page on this website in a single downloadable file.

If you need the information on this website to be provided in a different format, please contact the [SPACe helpdesk](#).

 **Note**

This PDF file is a work-in-progress. Some elements may not be rendered correctly, or may not match the accessibility criteria defined in our [accessibility statement](#).

1.4.1 Download link

 [Download the SPACe documentation as a PDF file](#)

2. About our services

2.1 Workbench Overview

The Analytical Workbench platform allows access to a broad range of tools and software, tailored to the needs of public sector analysts, with a focus on open-source solutions, collaboration, and security assurance. It provides access to a world-class data centre, high performance compute and the ability to collaborate across organisational boundaries; providing the ability to share resources, code, and ideas.

The Workbench is provided by SPACe, the Scottish Public Sector Analytical Collaborative, with the following strategic objective:

To collaborate wherever possible to help people better produce, find and use data and statistics about Scotland, to make our services more efficient, and increase our impact. Our approach to this work will be open, flexible, and forward thinking, acknowledging that compromise may sometimes be necessary on all sides. Wherever possible, we will seek to build shared infrastructure to support all producers of Official Statistics in Scotland.

2.1.1 Virtual Machines

The Workbench provides access to high-performance Linux and Windows virtual machines (VMs) via your web browser, to enable data analysis across the public sector. Within each VM you get access to a range of open-source software with compute power sufficient for your analytical needs.

2.1.2 Security Zones

When you setup a project it will sit in one of three security zones: Standard Zone, Protected Zone or Private Zone, depending on your information governance needs:

- Standard zone is open to the internet and all project users can transfer data in or out of projects in this zone
- Protected zone can only connect to a limited number of websites, mainly to allow software package downloads. Data transfer out of projects in this zone is restricted.
- Private zone is disconnected from the internet and data can only be transferred in or out of the Workbench via a controlled pathway.

All zones are hosted in the Edinburgh International Data Facility, data is safe and secure with access password controlled and restricted to allow-listed organisational VPNs.

2.1.3 Contact Details

If the Workbench is something you or your organisation could use, please contact the [SPACe helpdesk](#) and we will be able to provide you with a demonstration and discuss how you could use it. If you want to start using the Workbench we can work with you to onboard your organisation and start any projects that you want to run on the Workbench.

2.2 Getting onto the Workbench

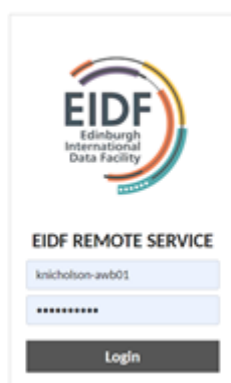
Now that your project has been created on the Workbench and you have been sent your usernames and passwords you can use the Workbench.

2.2.1 First

Login to your organisational VPN. The Workbench cannot be accessed except through your organisation's VPN.

2.2.2 Second

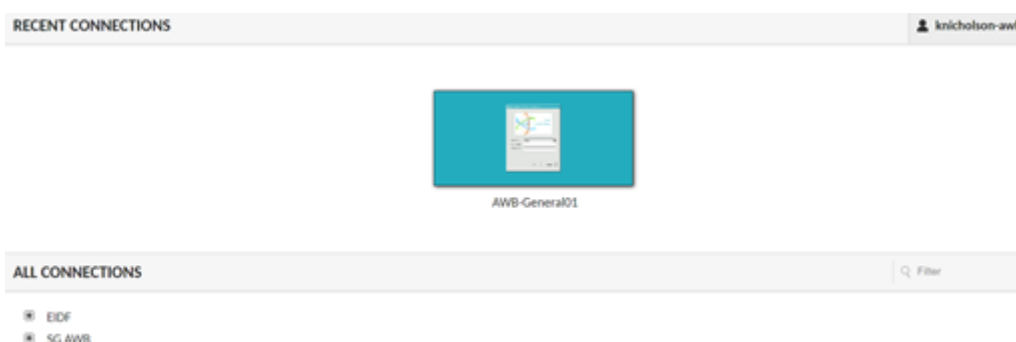
Go to the following website <https://eidf.epcc.ed.ac.uk/eidf01/#/>. This is the main login for the Workbench. If you cannot access this website even through your VPN please email [SPACe helpdesk](#) as your organisation's VPN might not have been added to the system yet. You should see the following image when you login.



EIDF Login Page

2.2.3 Third

Enter your first level (main Workbench) username and password. If this is the first time logging in or if the password has expired, you will be asked to change it. Otherwise you will be able to see all the VM you can access. You should see the following image when you login.



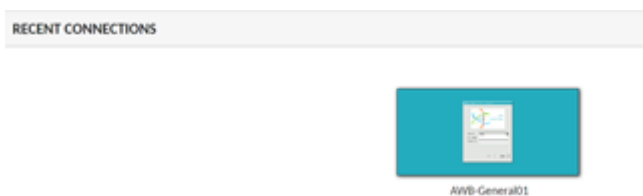
VM Connection Page

2.2.4 Fourth

Find the name of the project VM you will be using from the list of VMs you have access to. It will appear in one of the selections in the All Connections section of the page, or if you have recently used the project VM it will appear in the Recent Connections section. You should see the following image when you login.



List of VMs

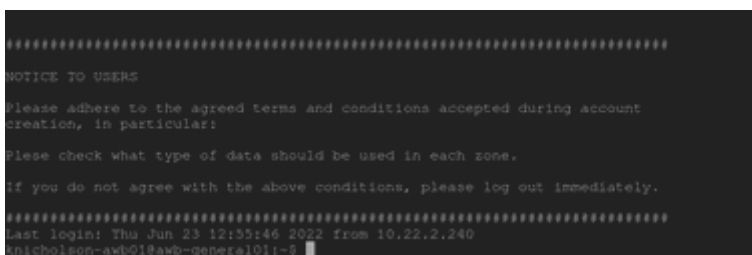


Recent Connections

2.2.5 Fifth

If this is the first time you are connecting to a Windows VM project you will be asked to change your password when you login. However, if you are connecting to a Linux VM project for the first time you will need to change your password before logging in.

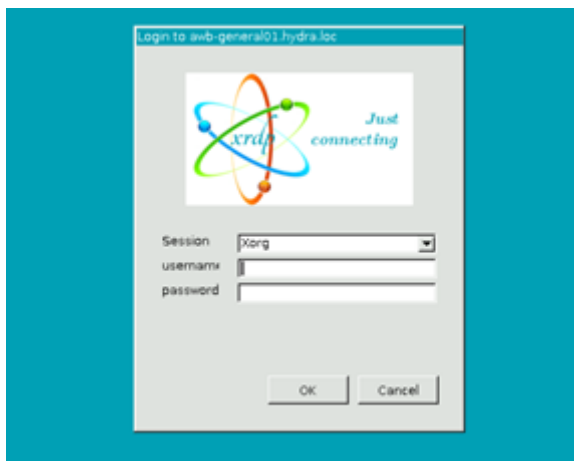
To change your Linux VM password click on the SSH connection of the Linux project VM. To login use your second layer username and password. Once in you will see a command line please type in 'passwd' and press return this will prompt you to change your password. Once you have changed your password you can exit the SSH by typing 'exit' and pressing return and selecting Home on the pop-up box this takes you back to the Connections Page.



Command Line access

2.2.6 Sixth

Select your project VM and enter your second layer (project VM) username and password. You will then be in the project VM, use the normal windows and Linux menus to find R Studio or Python and run your project.



Linux VM login

2.2.7 Seventh

To add data to your project VM it will depend on the zone of your project VM is in:

- For Standard Zone projects you can use a Serv-U account or add it through the internet access the zone has.
- For Protected Zone projects the person on the project with access to Serv-U will need to add data to the zone.
- For Private Zone projects please contact [SPACe helpdesk](#) for help to add data.

Please note that the Scottish Government has another way of adding data to the Workbench please see those guidelines for how to use that method. [Add link](#)

2.2.8 Eighth

For projects that use Serv-U to add data go to the following address <https://data-transfer.epcc.ed.ac.uk/> on the computer where your data sits. Use your Serv-U username and password to login, updating passwords where needed. Or work with the [SPACe helpdesk](#) to add data to private zone projects.



Serv-U login

2.2.9 Ninth

In Serv-U add all the files from your local computer that you want adding to your project. Then when you are in the Standard or Protected Zones of the Workbench you can access Serv-U to download your data to that project space. This process can be reverse to move files from the project space to your local computer.

2.2.10 Tenth

Run your project and when finished remove any files you want and contact the [SPACe helpdesk](#) to close the project.

2.2.11 Checkbox overview of process

- Receive login details from the Workbench helpdesk
- Login to your VPN
- Go to <https://eidf.epcc.ed.ac.uk/eidf01/#/>
- Login to the first layer of the Workbench
- Find the link to your project VM
- If first login to a Linux SPACe select SSH connection and change password
- Select your project VM and login, changing Windows password if need be
- Identify method for adding data based on project zone
- Use Serv-U to add data to Standard or Protected or contact the [SPACe helpdesk](#) about the Private zone
- Run your project in the Workbench, when you want to close project remove any files

2.2.12 Contact Details

For any other support look through the longer AWB guide or contact [SPACe helpdesk](#).

2.3 data.gov.scot

2.3.1 Introduction

Domain created in April 2020 during pandemic, as a trusted source for Scottish Government data:

- hosts interactive content to supplement static publications
- hosted by the University of Edinburgh Information Services and managed by EPCC as part of analytical workbench

2.3.2 What is hosted?

- data.gov.scot/child-poverty-system-map
- data.gov.scot/coronavirus-covid-19
- data.gov.scot/covid-19-infection-survey
- data.gov.scot/environment
- data.gov.scot/genderindex
- data.gov.scot/labour-productivity-2022
- data.gov.scot/primary-income-account
- data.gov.scot/poverty
- data.gov.scot/time-use-scotland
- data.gov.scot/wealth
- data.gov.scot/workforce-diversity-2021
- data.gov.scot/workforce-diversity-2022

2.3.3 Who does what?

- [SPACe helpdesk](#) provide [SG templates](#) and manage the service
- Analysts develop, test, build and upload content
- EPCC arrange release of content for testing and live

2.3.4 Workflow steps

#	Task	Owner	Notes
1	Discuss and record requirements in content log	Analyst / SPACe helpdesk	
2	Review styling, presentation and accessibility guidance	Analyst	See Resources
3	Develop, test and build content	Analyst	
4	Set up Objective Connect folder; invite analysts and EPCC	SPACe helpdesk	
5	Email analytical.workbench@gov.scot to request website	Analyst	At least 10 working days before website is required
6	Upload zipped content to ObjectiveConnect folder no later than 2 working days before release	Analyst	Name the .zip so it unzips as requested website name
7	Release unzipped content to make website	EPCC	

2.3.5 Resources

- [Publishing Official Statistics on data.gov.scot](#) (Yammer)
- [Digital Accessibility and public sector websites - Digital](#) (blogs.gov.scot)
- [Digital Scotland Design System](#)
- [Data visualisation: charts guidance | Digital Scotland Design System](#)
- [DataScienceScotland/sgtemplates: R markdown templates for data analysts working for the Scottish Government.](#) (github.com)
- [Interactive charts, animations and dashboards - Style.ONS](#)

3. Analytical Workbench user guide

3.1 Welcome to the user guide for the SPACe Analytical Workbench

Note

This is a living document. Things may change as the service evolves, and some parts of the documentation may be incomplete.

This document is a guide to accessing and utilising the SPACe Analytical Workbench (AWB) service.

The Analytical Workbench contains virtual machines (VMs) hosted within the Edinburgh International Data Facility (EIDF). Each virtual machine is designed to enable flexible and secure public sector data analysis. Currently the service is only available to public sector employees.

VMs are computers that you access through a web browser. They act like physical computers but use virtual devices to provide the same functionality as physical hardware. VMs get CPU and memory, video cards, access to storage, and network connectivity from the hosts they run on.

3.1.1 Operating Systems

VMs have guest operating systems to run software. The AWB allows users to choose between two types of operating system for their projects to be run on:

- Ubuntu 20 Linux VM with desktop interface and SSH command line connections
- Microsoft Windows Server 2019

3.1.2 Security

The AWB has a range of security processes to protect the projects using it. These processes include project segregation, account controls, and internet firewall allow lists.

The AWB can only be accessed through multi-factor authentication and by computers connecting through approved institutional networks (organisation VPNs).

Projects can be set up in one of three security zones, depending on the information governance required by the project. For more information, see "[Security zones](#)."

It is also the responsibility of each user to ensure that security of their project is maintained. We strongly discourage account sharing and emphasize the following:

- **DO NOT** use another person's account, nor allow other people to use your accounts, except by agreement of the [SPACe helpdesk](#).
- **DO** keep your passwords confidential, and inform us if someone else learns any of them.
- **DO** contact the [SPACe helpdesk](#) if you become aware that the security of the AWB has been compromised in any way.

3.1.3 Feedback

The Workbench is considered to be a stable production ready service. If you identify any problems or issues, please feed this back to us so we can improve the service for others, by contacting the [SPACe helpdesk](#).

3.2 Getting started

3.2.1 Requesting access

Request a new project

The project owner must complete a Project Request Form and send it to the [SPACe helpdesk](#). Each project must have a project owner.

[You can download the Project Request Form here.](#)

The Project Request Form is a Microsoft Word document. Please [contact us](#) if you need the form to be provided in a different format.

Change membership of an existing project

The project owner must update their Project Request Form and send it to the [SPACe helpdesk](#).

What are the restrictions in each zone?

For a full description of the restrictions in each zone, see [Zones](#).

PUBLIC ZONE

The public zone has full internet access. Users can download software using any of our supported methods.

PROTECTED ZONE

The protected zone has restricted internet access. This zone has an allow list of web urls that can be accessed. For more information, see [the Protected Zone allow list](#).

Data can be uploaded or downloaded to a VM in the protected zone using any of our supported methods. For more information, see ["Uploading data"](#).

PRIVATE ZONE

The private zone has no internet access.

Data can only be uploaded or downloaded to a VM in the private zone using our supported methods. For more information, see ["Uploading data"](#).

How do I determine what zone I should request?

This decision is based on how sensitive the data you will be analysing is. For a full description of data security types and their associated zone requirements, see [Zones](#).

What software comes pre-installed on the VM?

A list of the software (including the current versions) that comes pre-installed on a VM is given in the [list of available software](#).

What if I need to install additional software?

If your project is to live in the public zone, you will be able to download any software that you need at any stage during your project.

If your project is to live in the protected zone, you will be able to download any software that you need at any stage during your project from the allowed list of URLs. See [the Protected Zone allow list](#).

If your project is to live in the private zone, then this additional software must be installed **at the start of the project**. Only the project owner will be able to download this additional software.

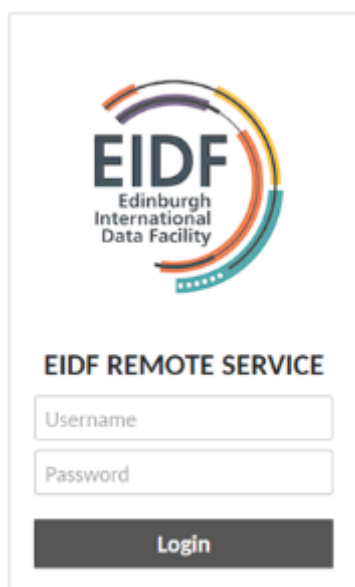
3.2.2 Log in

Accessing the Workbench

Access to any AWB requires a user to submit an access request to the [SPACe helpdesk](#), who will discuss your account and project requirements.

The Analytical Workbench Service can **only** be accessed via an allow-listed organisational VPN. Attempts to access the EIDF and AWB services from outside of an allow-listed network will be blocked, for security purposes. Please contact the [SPACe helpdesk](#) to discuss how to do this if you are unsure.

Once connected to your organisation's VPN, you will be able to navigate to the EIDF Remote Service login page at the following address: <https://eidf.epcc.ed.ac.uk/eidf01>.



EIDF Login Page

AWB Gateway Connections

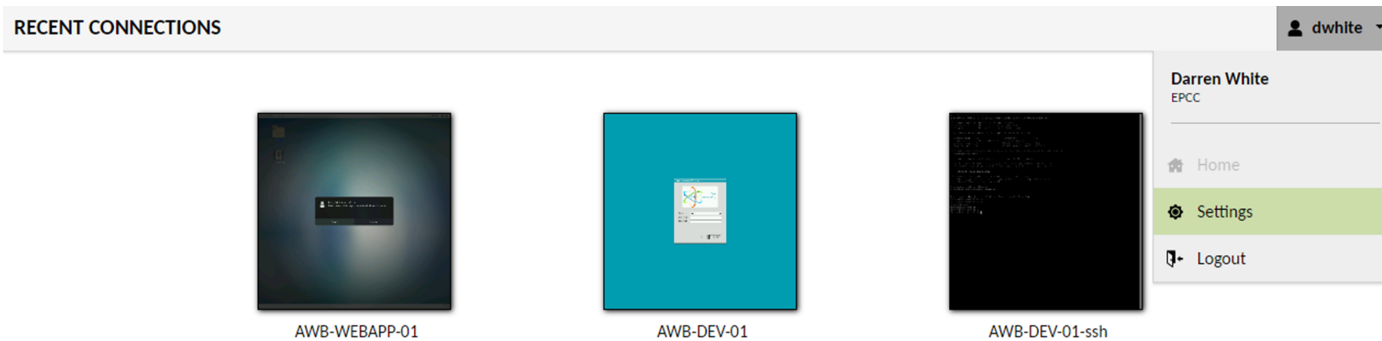
After logging in to the EIDF Remote Service you will be presented with what we will from here refer to as the AWB Gateway¹. From the gateway you can launch connections to individual VMs, enabling either a command line (`AWB-PROJECT-ssh`) or desktop (`AWB-PROJECT` or `AWB-PROJECT-RDP`) interface.

Warning

The first time you login to the gateway you will need to change your password.

CHANGING THE AWB GATEWAY PASSWORD

The first thing you will be prompted to do when you first log in to the gateway is change your password. If you ever want to do this manually, select the Settings option from the menu in the top right hand corner of the main login page.



AWB Gateway Settings Page

On the settings page you should navigate to the Preferences Tab and fill out the Change Password section before selecting the Update password button.

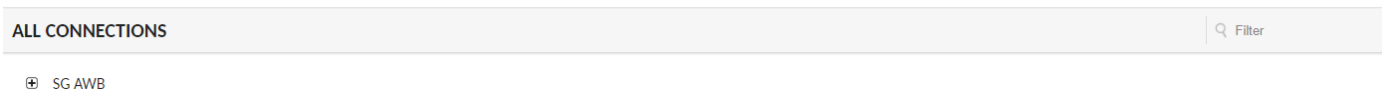
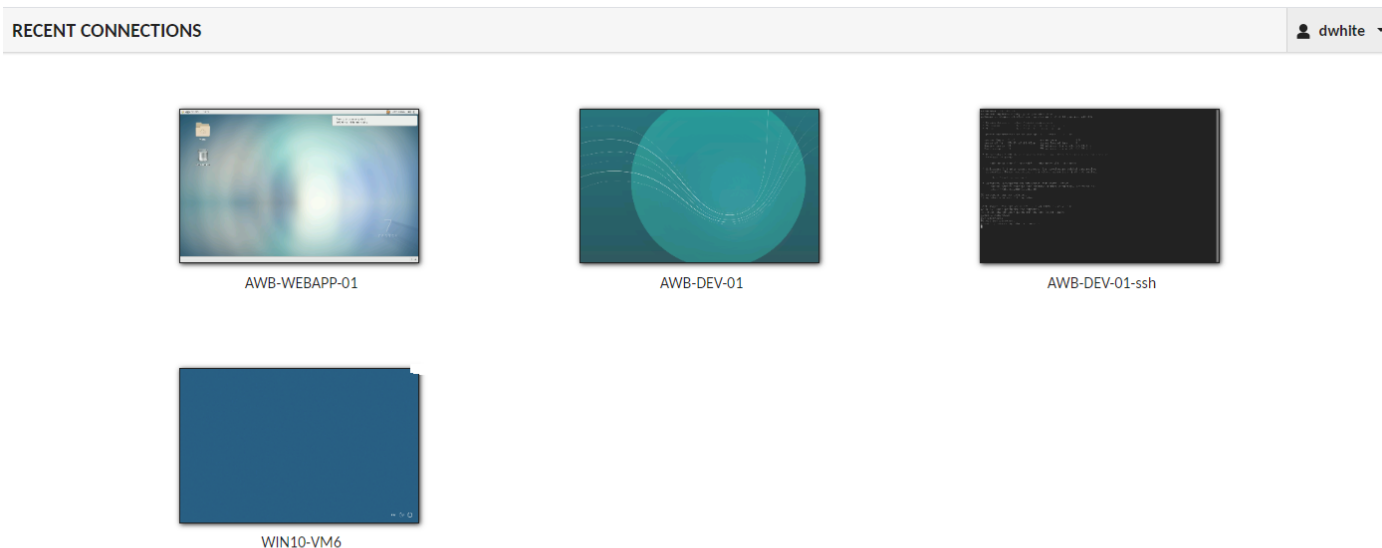
Note

The system will prompt users to update their password every 90 days. Remember that changing this password does **not** change your password on the individual VMs, which have their own security layer. To change your password for VMs see later sections of the guide.

Selecting a VM to use

One of the major advantages of the Analytical Workbench is that you can run more than one VM at the same time.

On the main gateway page, which you can select at any time by choosing the Home button when you see it, you will see all the VM connections you can have.



Gateway Landing Page

The connections are, as an example:

- AWB-PROJECT - this is for the standard Desktop view of the Linux VM
- AWB-PROJECT-RDP - this is also for the standard Desktop view of the Linux VM
- AWB-PROJECT-SSH - this is for a Command Line Interface view for the Linux VM
- WAWB - this is for the Microsoft Windows service

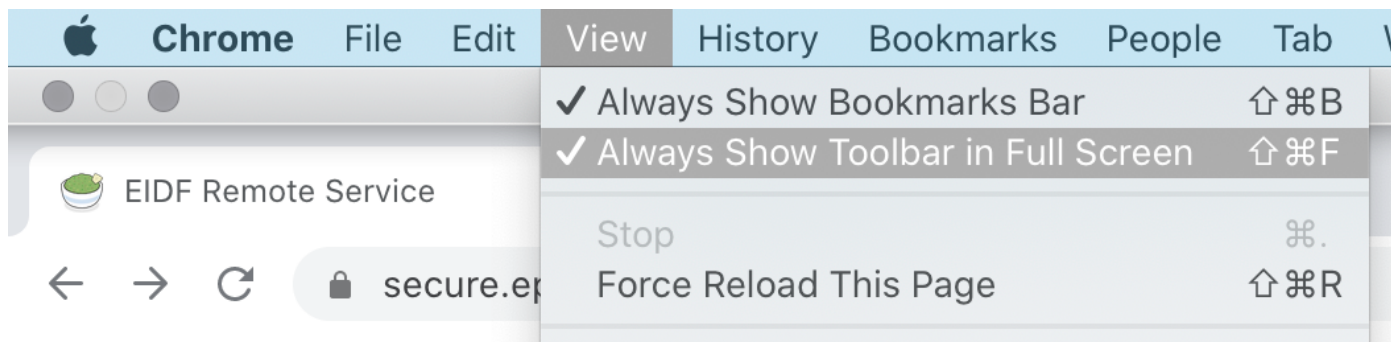
Selecting the VM you want to use will open up a login screen for your preferred connection. At present there are project-specific login credentials for each user. The username and password are the same for the Linux Desktop RDP and SSH connections, as they connect to the same underlying VM. For most users there will typically only be one VM type visible, as we are limiting projects to one VM unless there are unique use cases to require access to other VMs as the same user withing a project.

You can also select any VM that you currently have previously used from this page in the recent connection pane.

Maximising VM workspace

When starting a Workbench VM, make sure that your browser windows are at a suitable size (preferably maximised) **before** selecting the VM connection. To obtain the most screen real-estate for the VM, we recommend setting your browsers to true fullscreen modes, particularly on small/low resolution screens. On windows, this is done by simply pressing the F11 key. On Macs:

1. Maximise the browser to it's own desktop space as normal
2. On Chrome or Safari: Select View > Uncheck "Always Show Toolbar in Full Screen"



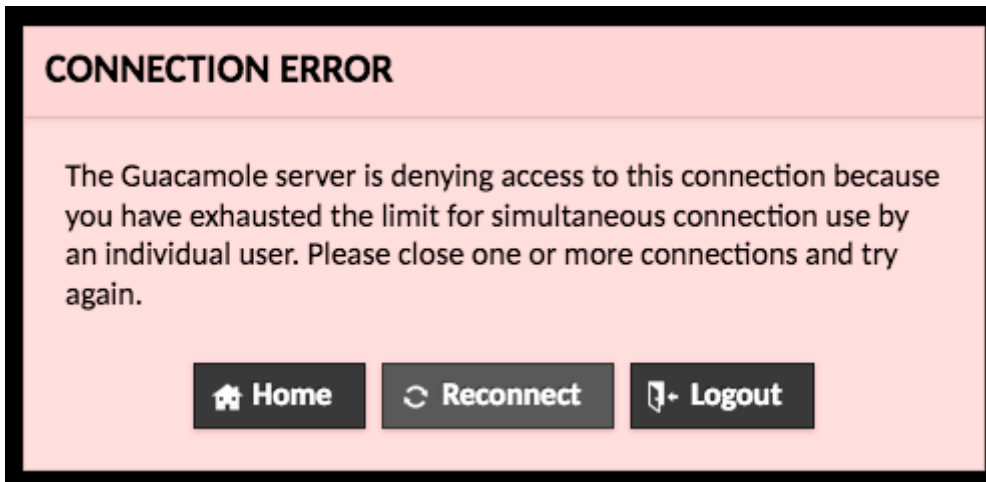
VM Maximise Display for Macs

Accessibility Options

After connecting to a VM from the gateway it is possible to access a hidden side pane for settings and accessibility options by pressing `⌘ Ctrl` + `⇧ Shift` + `⌘ Alt`.

Too many connection to the Workbench

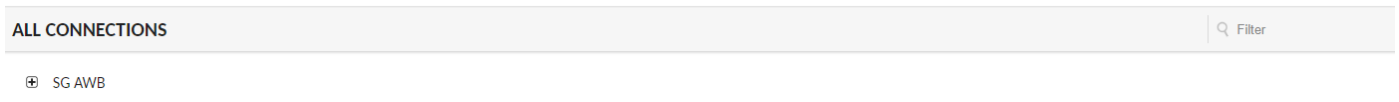
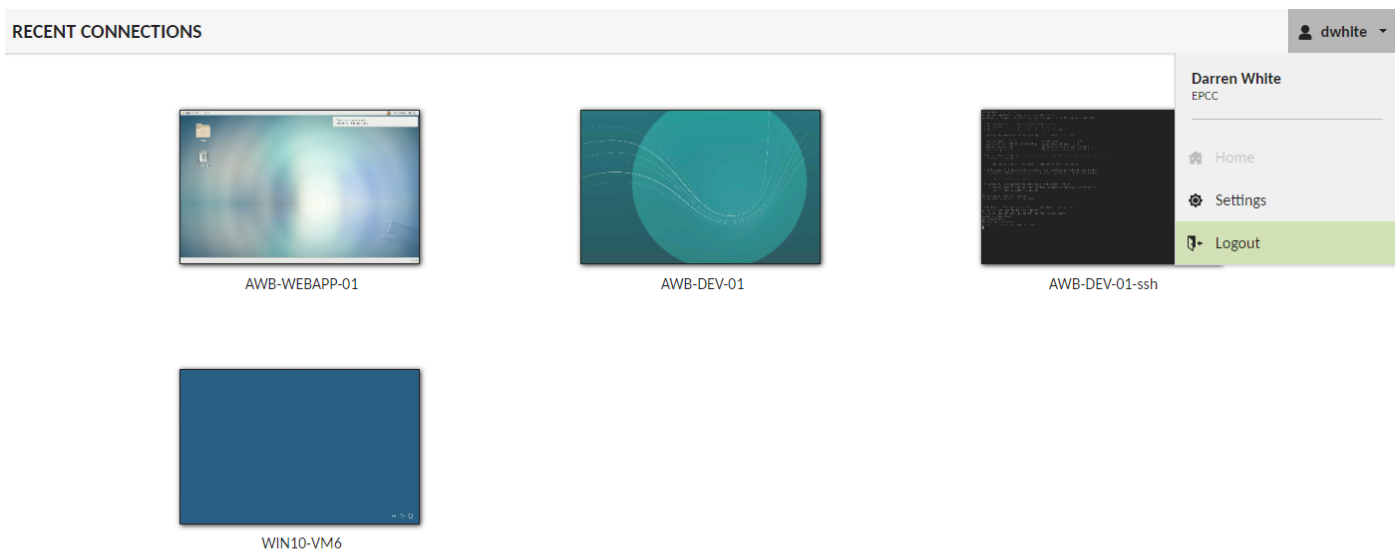
If you try to access the Analytical Workbench in too many browser windows you will see the following warning as you can only have so many open at one time.



AWB too many connections warning.

Logging out from the Workbench

To log out from the Analytical Workbench click your username at the top right and choose 'logout':



Logging out from the AWB

1. This uses the Apave Guacamole software, a remote desktop gateway, and was previously referred to as such in earlier versions of the guide. ←

3.2.3 Transferring Data

After determining whether your AWB project is in the appropriate security zone data can be transferred to each AWB project. For more information on security zones, see "[Security zones](#)."

Data can be transferred in several ways depending upon the Security Zone of your project.

	Standard	Protected	Private
Secure File Transfer Protocol (SFTP)	✓	✓	✓
Clipboard	✓	✗	✗
Objective Connect	✓	✗	✗
Serv-U	✓	✗	✗
Secure File Copy (scp)	✓	✓	✗
Open Data (via Web API)[1]	✓	✓	✗
Internet	✓	✗	✗
Github/Gitlab (git)	✓	✗	✗

[1] For more information, see "[Allow list for virtual machines in the protected zone](#)."

Using Secure File Transfer Protocol (SFTP)

ADDING DATA TO YOUR PROJECT

- You must zip the file which you wish to add to your project and name it according to the following format: zone-project-filename. For example, if you wish to add "my_file.zip" to the project "prj01" in the standard zone, you must rename it "standard-prj01-my_file.zip".

Note

Your zip file name must not contain any spaces.

- Move your file to the appropriate landing zone given to you by your [SPACe helpdesk](#) (contact your [SPACe helpdesk](#) if you don't know where this is). Scottish Government staff can access the landing zone by visiting \\sdx\AWB-Users from File Explorer.

Linux

- Your file will be moved to the `/import/<PROJECT_NAME>/` directory on your VM. Anyone who is part of your project will be able to view your file!
- Log in to your VM and copy your file from `/import/<PROJECT_NAME>/` to your home directory.

Windows

- Your file will be moved to the `S:\Import` directory on your VM. Anyone who is part of your project will be able to view your file!
- Log in to your VM and copy your file from `S:\Import` to your home directory.

MOVING DATA FROM YOUR PROJECT

- You must zip your file and name it with your organisation username.

Linux

- Move your file to the `/export/<PROJECT_NAME>/` directory. Anyone who is part of your project will be able to view your file until it is moved by the file transfer process.

Windows

- Move your file to the `S:\Export` directory. Anyone who is part of your project be able to view your file until it is moved by the file transfer process!
- Your file will be moved to the landing directory given to you by your helpdesk. During this moving process, your file will be renamed according to the following format: `inbound-zone-project-username`. For example, "username.zip" from the project "prj02" in the protected zone, will be renamed "inbound-protected-prj02-username.zip" during this process.

 **Note**

Consider password protecting your zip file to restrict access.

Clipboard

Copy and paste data to and from the VM using the clipboard.

Using Objective Connect

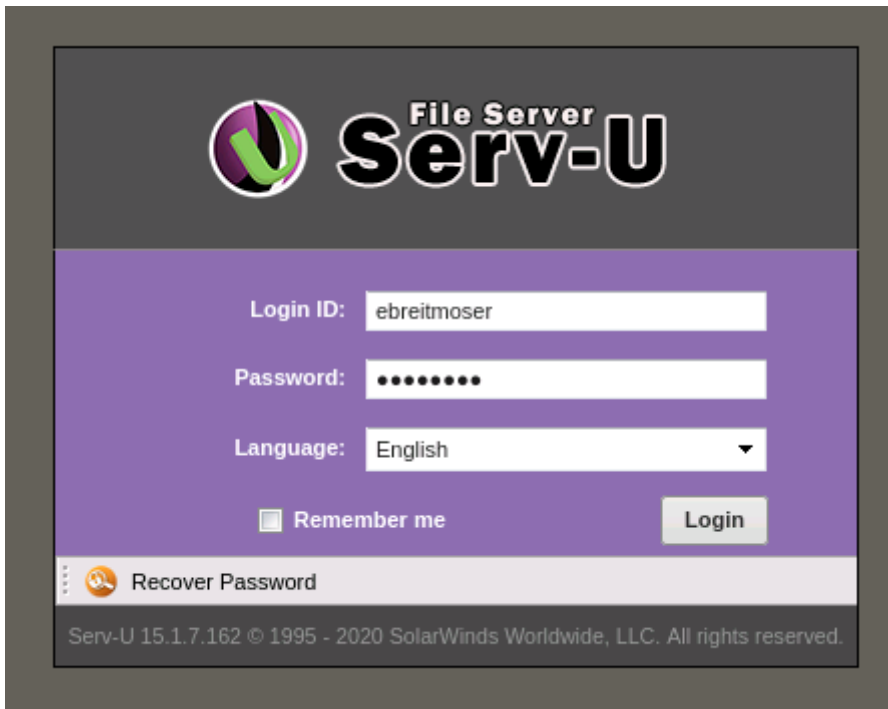
Objective Connect is a secure file sharing application. Please [contact us](#) if you would like a private workspace for sharing files.

Using Serv-U

With Serv-U you can share files or compressed (zipped) folders. You can either add files to the Desktop-VM or to the Windows-VM and you can take them off it, you use the same method for both. You will need a username and password for Serv-U, which is now created automatically for new users.

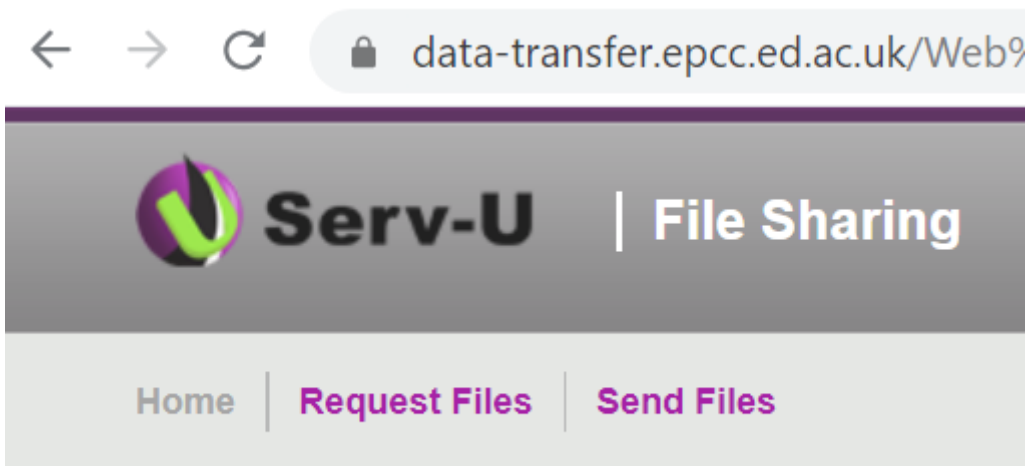
You can access Serv-U at <https://data-transfer.epcc.ed.ac.uk/>.

If you do not currently have a Serv-U account and would like one, please contact the [SPACe helpdesk](#).



Serv-U Login Page

To upload a file to Serv-U select the 'Send Files' button in the top left hand corner once you are logged in.



Access the Send Files tab.

On the Send Files page you will need to add a guest email at the bottom left (the email can be your own) and a guest password on the right in the middle. Then select the 'Next' button.

Sending a file with Serv-U

On the next screen add all the files you want to share by using the Browse button to find them on your local computer:

Add files to Serv-U

Click Upload when you are finished, then Done on the next page. This will send an email to the guest user email informing them about the availability of the files for them on Serv-U.

To download the file to the VM, open a web browser in the VM and then put in the web link from the email you received from Serv-U. You will be asked for the unique password that you set up as a guest password that you can choose.

Then choose the file(s) you want to download to the VM

Select your file(s) to download

The file(s) were uploaded on 10/03/2020 10:08:31

Select All

1  [Screen Shot 2019-09-06 at 15.41.16.png](#)

 Download

Select files for download

You will then find these files in the Download folder in your user directory on the VM.

Using *secure file copy* (scp)

Secure File Copy is a tool in Linux that allows you to copy a file between computers. If your host machine allows it, you can use `scp` to copy data from it to the Desktop-VM or vice versa. Or you can use `pscp` to copy data from it to the Windows-VM or vice versa. This is something that is being discussed with IT teams to see how this would work with organisational connections such as those at the Scottish Government.

Using Open Data

The Desktop view is connected to the internet and though downloading from email accounts and other services has been disabled you will be able to connect to open data platforms. So if the data you are looking for is on those platforms then you can use your normal code to connect to the data and download it.

For example, you can get data from the Scottish Open Data platform at statistics.gov.scot. To do this you need the endpoint of the data you want. The following example shows the API (Application Program Interface) endpoint for the SIMD open data.



Scottish Government
Riaghaltas na h-Alba
gov.scot

STATISTICS.GOV.SCOT

ATLAS

DATA

SEARCH

Explore ▾

Tools ▾

 A data cube dataset in folders : [Scottish Government](#), [Scottish Index of Multiple Deprivation](#)

Scottish Index of Multiple Deprivation - Geographic Access to Services

Mean travel time (in minutes) to key services, by car or public transport. This dataset contains indicators used in SIMD 2016 and SIMD 2020.

DATA

ABOUT

API

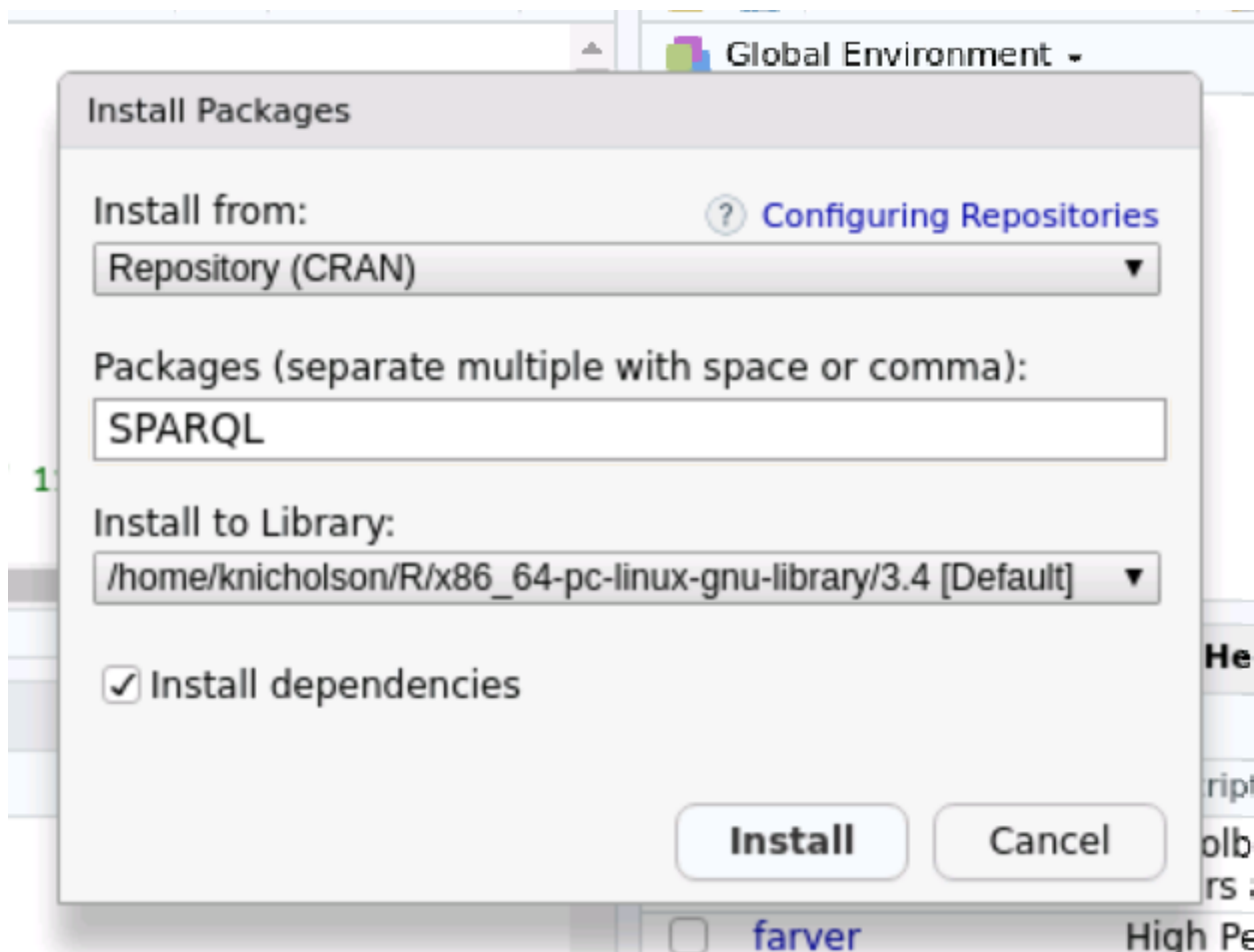
URI

This is a linked data resource: it has a permanent unique [uri](#) at which both humans and machines can find it on the our [SPARQL endpoint](#).

<http://statistics.gov.scot/data/scottish-index-of-multiple-deprivation---geographic-access-to-services-indicators>

statistics.gov.scot Open Data

The endpoint for SIMD is [here](#). The end point is a web address that you can dial out to using a programming language such as SQL, or a JSON interpreter. The main interface for SIMD is a graph database and you will need to write a SPARQL query in RStudio to connect to it.



Installing sparql for R-Studio

Using the internet

Using git repositories

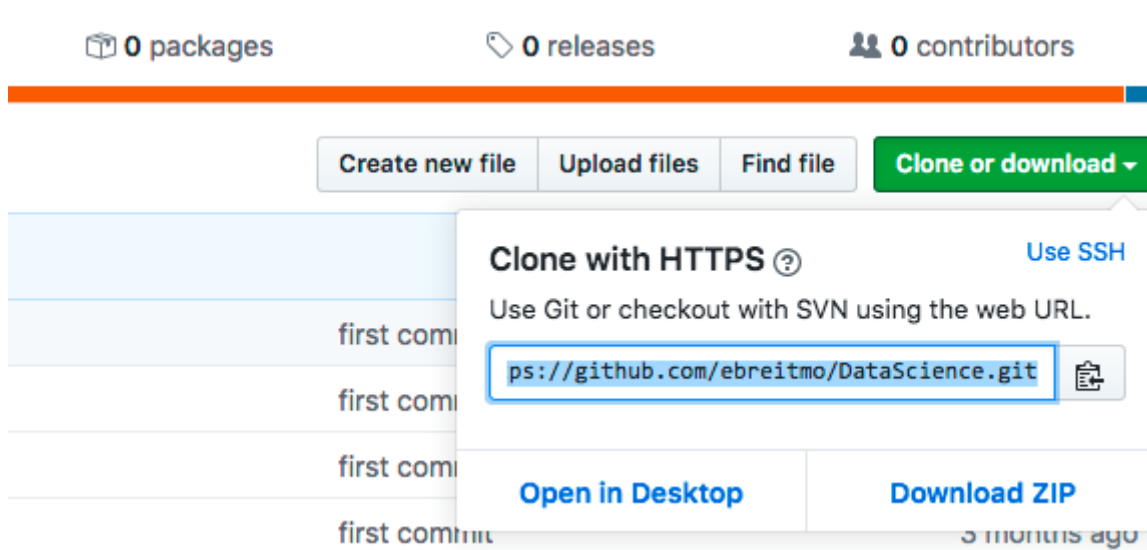
If you need to access a remote git repository you will have to interact with it via a Terminal command line. In the future that should be made easier by providing a 'git pull'-button.

To download data from an external git repository from a terminal window use the following command:

Bash

```
git clone https://github.com/ebreitmo/DataScience.git MyNewDirectory
```

This will retrieve everything from <https://github.com/ebreitmo/DataScience.git> and put it into the new directory MyNewDirectory on the VM. Omitting the new directory name will create a directory 'DataScience' as the repository's name above indicates and put the repo's content there.



Finding the Github repository address to clone.

Other Methods

Data upload methods will develop over time. This includes the longer term plan of using the data lake methodology for adding data that can be shared within the Analytical Workbench by all its users. There will be a number of information governance and data sharing challenges that will need to be dealt with but the end product should be a much easier way of accessing data.

3.2.4 Collaboration

One of the main advantages of the Analytical Workbench is the ability to collaborate with other people. Over time we want to support collaboration in many ways, but at the moment there are three key ways you can collaborate, with more detailed examples appearing below as more services become available.

Data

Over time there will be a central pool of data that people will be able to access, depending on their access rights. This central pool will accelerate collaboration across organisations. At present though the data collaboration comes from having shared project spaces where users can add their own data. For more information, see "[Uploading data.](#)"

Code

One of the major ways we can improve collaboration and improve standards and sharing across the public sector is by sharing code. The Workbench will aim to have an internal code repository server for sharing code during a project and will enable the sharing of code to an external public repository. Not only is this an essential part of the Digital First policy, but it is also the best way of collaborating with our colleagues. By being open and sharing our work we can communicate more effectively and work more as teams and less as silos. See section 3.4 below for examples on how to do this.

Co-working

As teams and projects using the Workbench develop we will want to develop better tools to support them. Technology is a way of accelerating our existing processes. We will therefore need to first develop how we work better together. This will involve re-thinking how we work in teams across our organisations and between organisations.

Once we have a better understanding of how we work we will be able to support this with better means of communication and co-working. But you will still be able to co-work in the Analytical Workbench.

You will have a shared area in the Linux Workbench, with more details found in section 4.5 of this guide. You will be able to share documents, files and data as defined above. Of course you will also be able to use the Workbench at the same time as using a web based video conferencing tool on your normal computer.

Collaborative Git Workflows

An in-depth tutorial on GitHub workflows can be found [here](#).

A simplified set of instructions can be found below for two use cases. If you wish to share your code outside of the Workbench, we recommend using an external repository hosting service. For code that will only live within the Workbench, it is also possible to set up a local repository that multiple users can work from, as detailed below in section 3.4.2.

CREATING AN EXTERNAL REPOSITORY

To create a repository you wish to share externally, you will need to have access to a Git repository hosting service, such as GitHub or GitLab. If you don't already have an account on a service, we recommend GitHub for public repositories. Go to <https://www.github.com> and follow the instructions to create an account.

Create external repository hosted on GitHub

The quickest way to get started is to create a new repository on GitHub. Head to your GitHub home screen and select New from the sidebar.

Repositories



 [dubiteatro/QuasarBNN](#)

Create a new Github repository

On the next screen, name your repository and select the options you require. It is good practice to create a `README` file, and also using a `.gitignore` file for the language your code is in. This stops Git from picking up files that are temporary and not best handled by version tracking software (e.g. for Python this includes `.pyc` files, or `.RHistory/` `.RData` session-specific files in R).

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Repository template

Start your repository with a template repository's contents.

No template ▾

Owner *



djwhiteastro ▾





Repository name *

git-test



Great repository names are short and memorable. Need inspiration? How about [probable-guacamole?](#)

Description (optional)

-  **Public**
Anyone on the internet can see this repository. You choose who can commit.
-  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:


Skip this step if you're importing an existing repository.

- Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

- Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: R ▾

- Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

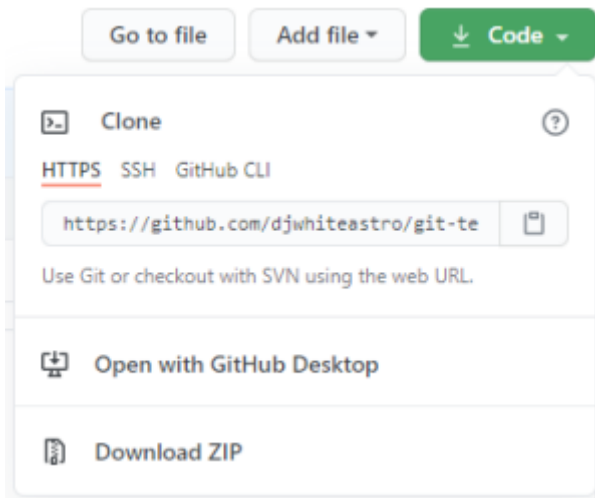
This will set  `main` as the default branch. Change the default name in your [settings](#).

Github new repository setup page

Click "Create Repository" at the bottom of the screen, and you should be taken to your fresh, new repo.

Cloning remote GitHub repository

Once your repository is created, we can view the remote URLs of our repository with the dropdown menu under the Code button:



Cloning your new repository

Depending on which method you choose to access from the Workbench, you need to set up the relevant access permissions on GitHub. Recent changes now require a Personal Access Token (PAT) for https access, and a password-protected SSH key for SSH access. The GitHub documentation you should follow is [here](#).

Once done, using a command line window on the Linux VM, move to a folder that you wish to keep your code in. In this example we create a user directory in the active users home directory called `code` and navigate into it before checking it's location with `pwd`.

Bash

```
mkdir ~/code
cd ~/code
pwd
```

From here, the newly created repository can be cloned to the local directory using the `git clone` command and the clone URL specific to your repository.

Bash

```
git clone https://github.com/user/new_repository
```

Check the contents of your cloned repository with `ls`.

Bash

```
ls new_respository/.
```

Adding and committing files

Using either the command line or the file viewer GUI interface, add/move/copy the files you wish to share to the repo folder. You then need to add these files to your local git version control in two stages. The first command `git add` specifies which files to add to the staging area. The staging area acts as a pre-upload point that is not pushed back to Github yet. To push all the staged changed to Github the `git commit` command is used. Commits are made with a message that will be stored alongside it to help you differentiate between commits when you are reverting back to earlier code. The message is entered via an editor in the command line of by using `-m` option with `git commit`.

For example, lets create a new file called `test.R` and commit it to our repository.

Enter the repository folder and create an empty file.

Bash

```
cd new_repository
touch test.R
```

Then add the file with `git add` and `commit` it to Github with the message "Test file added.".

Bash

```
git add test.R
git commit -m "Test file added."
```

Pushing changes to the remote repository

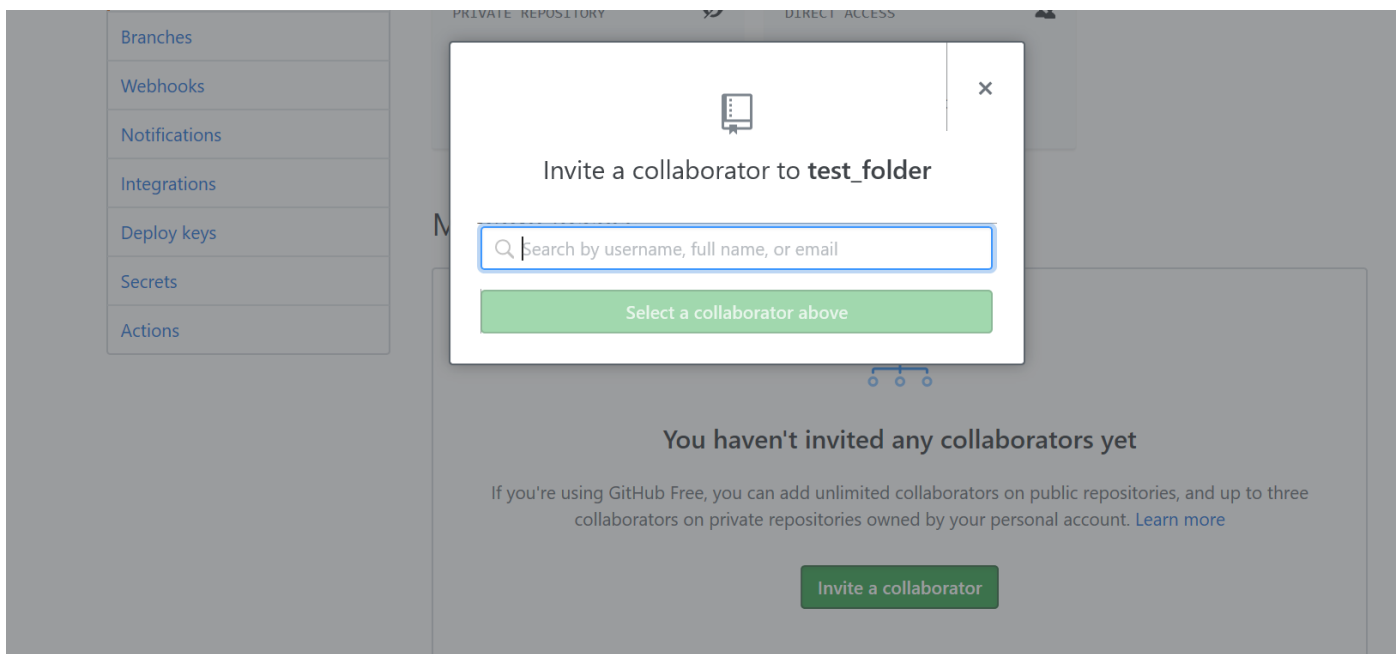
The final stage is to `push` your code to the remote repository hosted on GitHub, which will allow others to view/edit the code too.

Bash

```
git push
```

Adding Contributors

If you have created a private repository, you must specifically give others access rights. Head to the repository main menu on Github and select Settings. On the left tab, select Manage access. Then select Invite a collaborator and find your colleague by their **GitHub username**. This will give them access rights to your repository. They may connect through the same steps as above.



Adding new contributors

CREATING A LOCAL GIT REPOSITORY

It may also be useful (or necessary if external services cannot be used) to create a repository on the AWB itself that multiple project users can access.

Create a bare repository

The first step is to create a new, bare repository to act as a remote for each user's individual clones of the repo. Determine the location and name of repository (either a new or pre-existing folder) and initialise it.

Bash

```
git init --bare /shared_storage/a_local_repository/a_local_repository.git
# do something about permissions?
```

The methods in the previous section will now work by referencing the `.git` file. Cloning for example

Bash

```
git clone /shared_storage/a_local_repository/a_local_repository.git
```

Once the users clone of the repository has been initialised `git` commands should work as normal.

From here, any other user that can see the project folder should be able to use the repository as a "remote", allowing users to collaborate in the same way as with an external repository, such as creating branches to develop new features.

CREATING A NEW BRANCH

The strength of Github is that it allows you to create new branches that let you experiment with new code and commit or discard any changes you make. The main branch is known as `main` and should not be modified until your changes are stable.

If you wish to create a new branch, we suggest naming it after the individual writing it and the section of code that will be added, such as `steven_login_functions`. Creating a new branch uses the `checkout` command with the `-b` option.

Bash

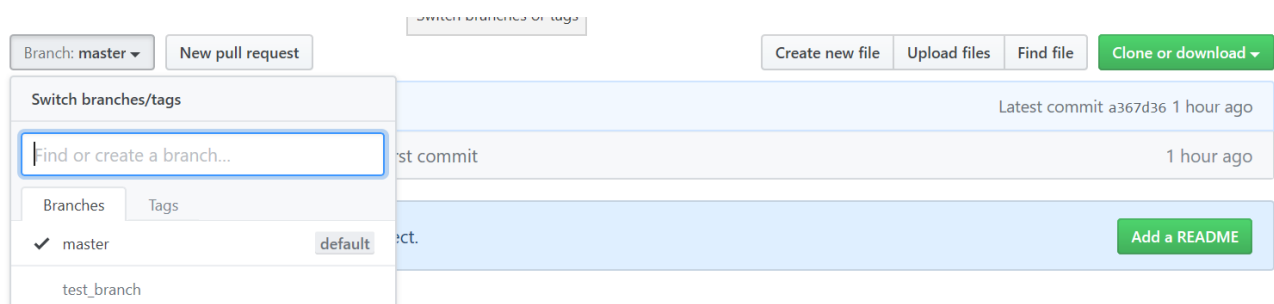
```
git checkout -b steven_login_functions
```

You are now on the new branch, and this branch currently is identical to the main branch we came from. When you make your changes to this branch using `add`, `commit` and `push` as before these changes will now only be made to this new branch. Though do make sure you tell it to push only to the new branch and not the master.

Bash

```
git push origin steven_login_functions
```

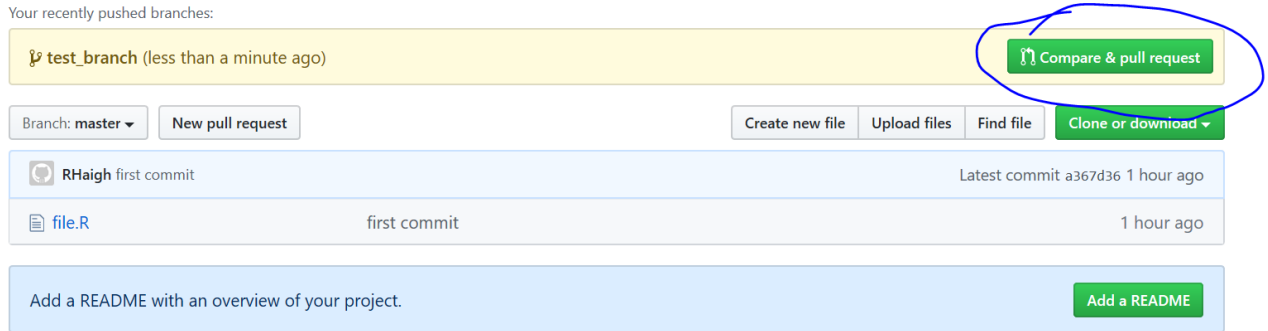
If you are also using GitHub, you can also examine these changes using the repository interface.



Exploring repo branches on Github

Merge Branch with Main

When you are happy with, and tested the changes you have made to the branch you will want to update the Master. To `merge` your changes into your main branch you first create a `pull request`. This is a request for permission to add your changes to the main branch. You can create this request by clicking on `Compare and pull request` button.

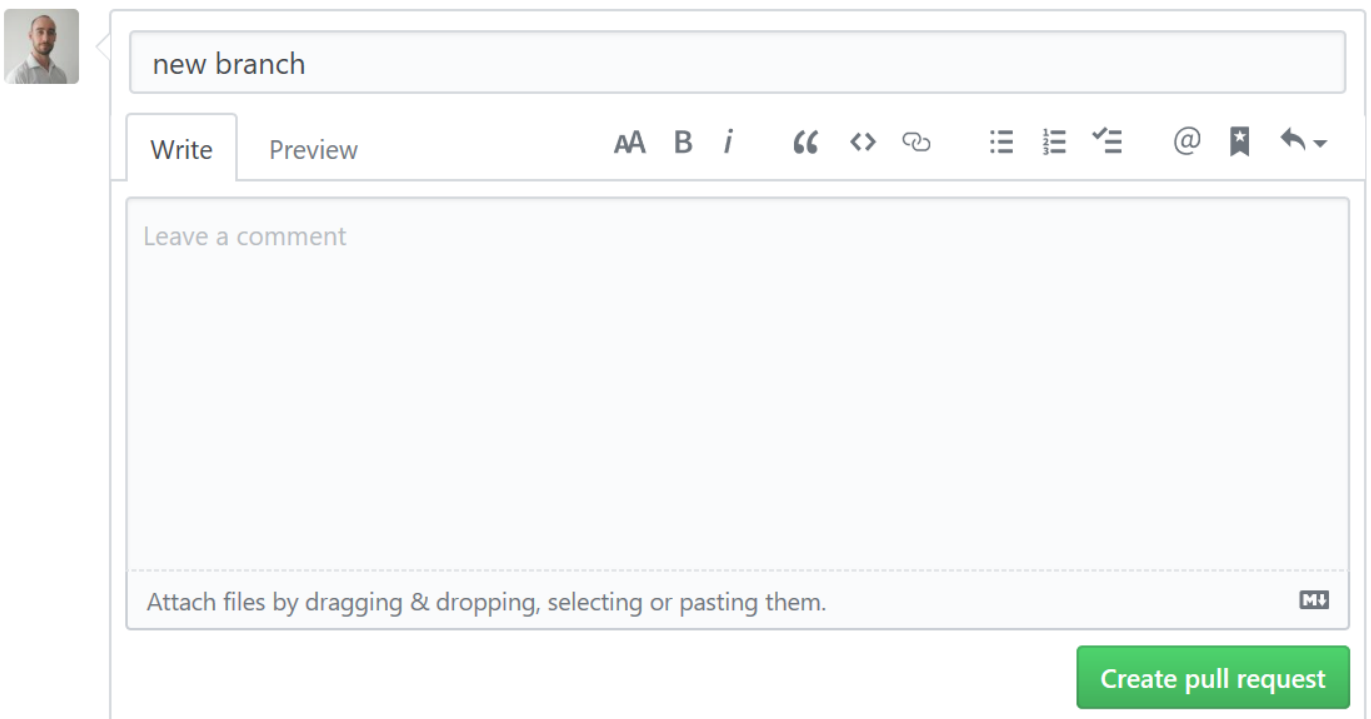


Creating a new PR

This will create a new dropdown where you can give details of the changes you have made.

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across for](#)







Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).




Writing a PR note

Then is will add the new code to the master branch once approved by the moderator of the repository (the creator). If you are the moderator, you will have the option to merge this pull request in the conversations tab:

new branch #1


 Open RHaigh wants to merge 1 commit into `master` from `test_branch` 


 Conversation **0**  Commits **1**  Checks **0**  Files changed **1**


 RHaigh commented now  
No description provided.

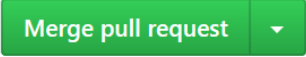

  new branch 54d1164

Add more commits by pushing to the `test_branch` branch on `RHaigh/test_folder`.



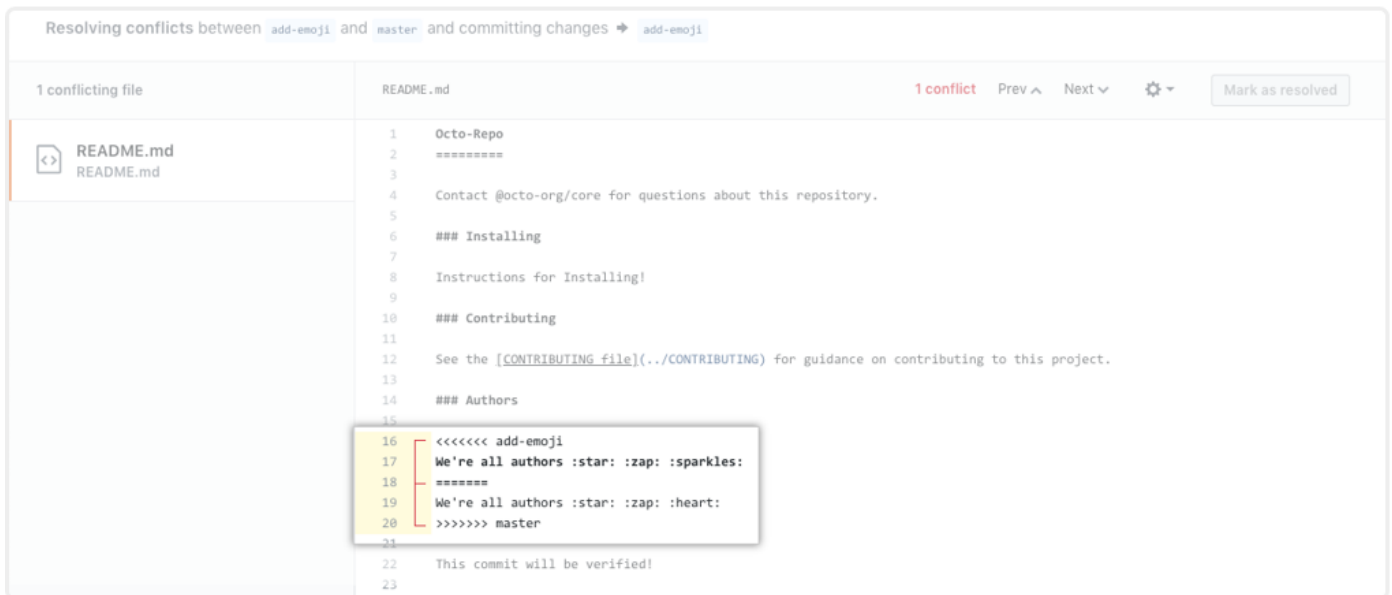
 **Continuous integration has not been set up**
GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.

 **This branch has no conflicts with the base branch**
Merging can be performed automatically.

 Merge pull request  You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

PR merges OK

If you have made changes to the existing code that was already in the master branch then git will not know which to keep. This is called a merge conflict. If you are the moderator of the repo, you need to instruct git which of these changes to keep and which to discard.



Resolving conflicts between `add-emoji` and `master` and committing changes → `add-emoji`

1 conflicting file

README.md 1 conflict Prev ^ Next v ⚙️ Mark as resolved

```
1 Octo-Repo
2 =====
3
4 Contact @octo-org/core for questions about this repository.
5
6 ### Installing
7
8 Instructions for Installing!
9
10 ### Contributing
11
12 See the [CONTRIBUTING file](../CONTRIBUTING) for guidance on contributing to this project.
13
14 ### Authors
15
16 <<<<<< add-emoji
17 We're all authors :star: :zap: :sparkles:
18 =====
19 We're all authors :star: :zap: :heart:
20 >>>>>> master
21
22 This commit will be verified!
23
```

Looking at code differences from a PR

Once these have been resolved, you may merge and create a new merged master branch. This provides a strong versioning methodology for code which can be trusted and shared.

3.3 Security

3.3.1 Security zones

Depending upon your project and the data being accessed, SPACe can offer projects with different levels of security. For projects which exist in the standard and protected zones, you will most likely work within a shared VM. These VMs have fixed system-wide software, but users can install their own packages and tools within their user space. For more information, see:

- ["Installing new software with Anaconda."](#)
- [RStudio](#)
- [Other software](#)

For the Linux-based private zone, each project will exist on it's own, unique VM. This allows project owners the option to request the ability to install software on the VM system-wide. The VM will be placed in the standard zone whilst the project owner installs the required software before the VM is moved to the private zone.



Warning

The level of security becomes fixed after any data has been moved on the VM.

	Standard	Protected	Private
Access to the internet	✓	Limited by Allowlist ¹	✗
Install packages	✓	Limited by Allowlist ¹	✗
Install system software	✗	✗	✓

¹ For more information, see ["Allow list for virtual machines in the protected zone."](#)

Which Zone to Use

The Workbench sits in the Edinburgh International Data Facility (EIDF). This meets a range of security requirements, these requirements can be provided on request. The Workbench is accessed through organisational VPNs and multi-factor authorisation, with each project having unique usernames and passwords.

All projects are isolated from each other, and all data added to a project is encrypted in transit. And when projects have been completed data sanitisation and disposal processes are used to remove them from the Workbench ensuring that the data has been deleted.

Each project can sit in one of three zones: Standard, Protected, and Private. The following provides a description of each zone. It is up to each project to determine which zone to use for their project based on the information governance of the projects lead organisation. The following provides information to help with this decision.

STANDARD ZONE

The standard zone is open to the internet and all project users can transfer data in or out of projects in this zone. It therefore provides open access to use open data or similar data and it provides the ability to trial using the Workbench, or use it for training purposes.

PROTECTED ZONE

The protected zone can only connect to a limited number of websites mainly to allow software package downloads. You can restrict who can or cannot add data to this zone. It should allow a wide range of data to be used though do check with the lead organisations data protection team to confirm. It is expected that this zone will be the zone most used by projects.

PRIVATE ZONE

The private zone is disconnected from the internet and data can only be transferred in or out of the Workbench via a controlled pathway. This zone is therefore similar to the National Safe Haven and theoretically any kind of data could be used in this zone. But do confirm this with the lead organisations data protection team. This zone enables projects to be sure their data is safe.

**Note**

For more information on data upload procedures, see "[Uploading data](#)".

3.3.2 Allow List for Virtual Machines in the Protected Zone

The allow list is configured to allow access to the following services from each virtual machine (VM) in the Protected Zone.

- Pip: The Python package manager and repository.
- Conda: The Anaconda Python package manager and repository.
- CRAN: The R package repository.

3.3.3 Adding Software to your Private Zone VM (Project Owners Only)

Private zone VMs come pre-installed with certain software. For more information, see "[Available software](#)."

If you are a project owner and you would like additional software on your Private Zone VM, the process to do so is described below.

Process for customising a Private Zone VM

1. Your VM will initially be set up in the **standard** zone. Do NOT put any sensitive data on the VM in this zone.
2. The project owner will be given a **temporary** user account with a limited set of privileged security rights and commands. Using these commands, you can configure your project VM in any way you like: you can remove existing software, add additional software, set up custom environments etc. These changes can be applied across the whole VM for **all** users. If you need help configuring your VM, please contact the [SPACe helpdesk](#).
3. Once you have configured your VM, contact the [SPACe helpdesk](#) and ask that your VM be moved from the Standard Zone to the Private Zone.
4. The AWB Helpdesk will then contact EPCC and a ticket will be raised to:
5. Remove your temporary user account.
6. Perform a security scan of your newly configured VM.
7. Move the configured VM into the private zone.
8. Issue you with a new local user account.
9. You will then login to the Private zone via <https://shs.epcc.ed.ac.uk/eidf01-private/> and your VM will be configured according to your changes in the standard zone.

3.4 Using your Linux VM

3.4.1 Linux desktop (RDP)

Desktop view

The desktop view is the VM connection to use for graphical interface software such as R Studio or Orfeo toolbox. This VM is based on the Linux operating system, and has a similar underlying feel to Apple software (which is also based on a Linux precursor called Unix). It has also been styled to look similar to Windows after a little while you can get used to how it works.

New users

For new users, you must first connect to a Linux VM via the Command Line/SSH connection option, where you will be prompted to update your password. For more information, see "[Command line \(SSH\) Interface](#)."

Once completed, you can reconnect to a desktop/RDP view and log in with your new password.

Access to the Linux Desktop view

To connect to the Desktop view select your preferred VM RDP connection (such as AWB-General01) from the gateway, and enter your username and password when prompted.

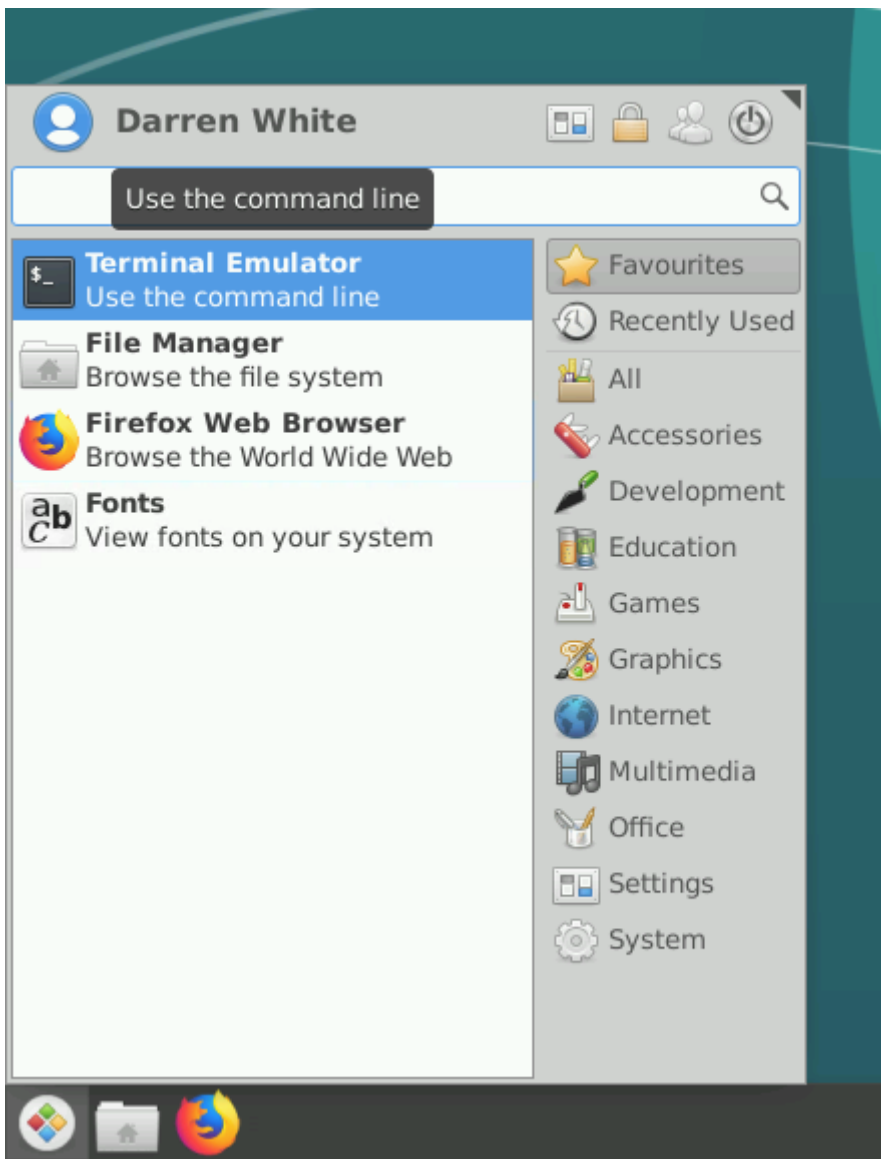
If you see the following pop-up screen, then you have entered an incorrect username/password combination. Remember, changing your Guacamole password does not change your VM password.



Connection failure due to incorrect credentials

Changing your Linux password via the Desktop view

If you wish to change your password, this can be done by opening a command line view using the program Terminal. This can be selected from the Applications menu in the top left hand corner and click on Terminal Emulator.



Open a new shell from the start menu

With a terminal open, use the `passwd` command to obtain a prompt to change the user password.

```
Bash
```

```
passwd
```

Warning

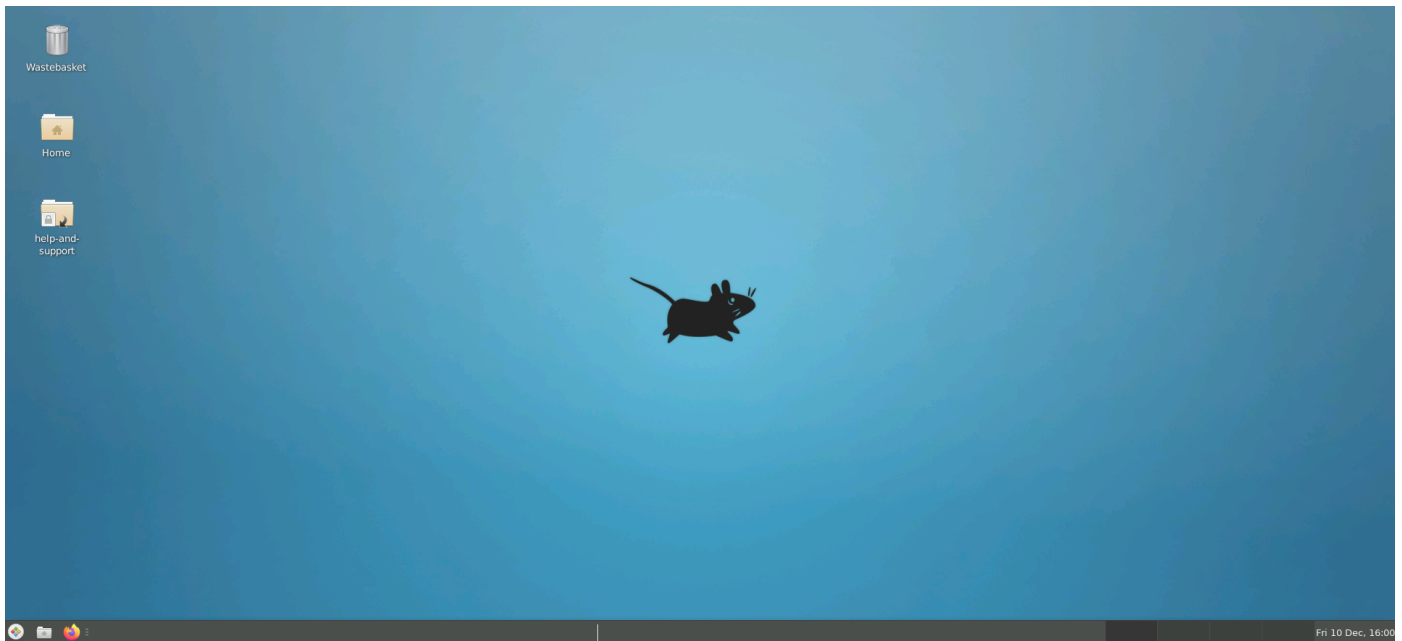
This new password cannot be based on a word from a dictionary. This will change the password for all VMs you access using the same username (but not the EIDF login).

Note

Passwords are set to expire every 90 days. If you are certain that you are using the correct VM password but cannot log in via the Desktop (RDP) view, please try accessing the SSH Command Line view via the project Manager, as shown in section 6. If your password has expired, this is where you will be able to update it.

Using the Linux Desktop

After logging on you will see the something very similar to the following screen.



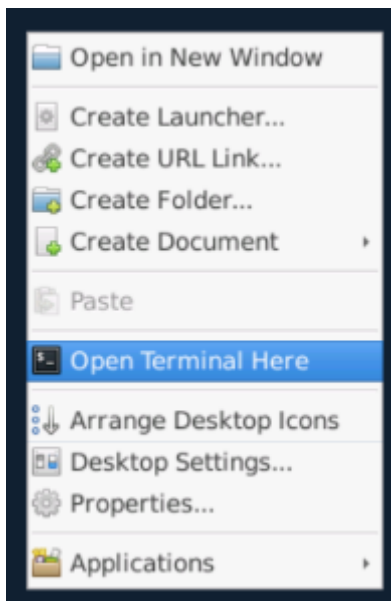
Linux VM Desktop

The look and feel is similar to a Windows desktop, with applications accessible from a start menu in the bottom left. Unfortunately, it won't be possible to extend your desktop across two monitors. But there are workspaces which allow users to have multiple virtual desktops. Users can switch between these using buttons in the bottom right hand corner.



Desktop workspaces

You can also open a Unix command-line terminal, right click with your mouse on the desktop background or in any folder location and choose Open Terminal Here. This can be used to support any work you are doing.



Right-click and open shell

Home space

For the shared VMs in the Standard and Protected Zones, in order to provide users with enough home space, user directories are mounted on a Network File System (NFS). These directories are found under the path `/md3_01/<project>/home/<user>`. For example, within this guide, we assume people belong to the default `awb01` project, so the paths for someone with the username "APerson-awb01" would be `/md3_01/awb01/home/aperson-awb01`, either on the command line or navigated to via the file browser: File System > md3_01 > awb01 > home > aperson-awb01.

For most scenarios this should be dealt with automatically, but if you are using code/scripts/etc with hardcoded paths, care needs to be taken, as this is different from the standard `/home/<user>` location for most Linux operating systems.

Shared drive space

To enable collaborative working, in the Standard and Protected Zones, the Linux Desktop connection has a large capacity, high performance shared disk drive attached, found under the path `/beegfs_10/<project>/data` via command line, or using the file browser under File System > beegfs_10 > > data.

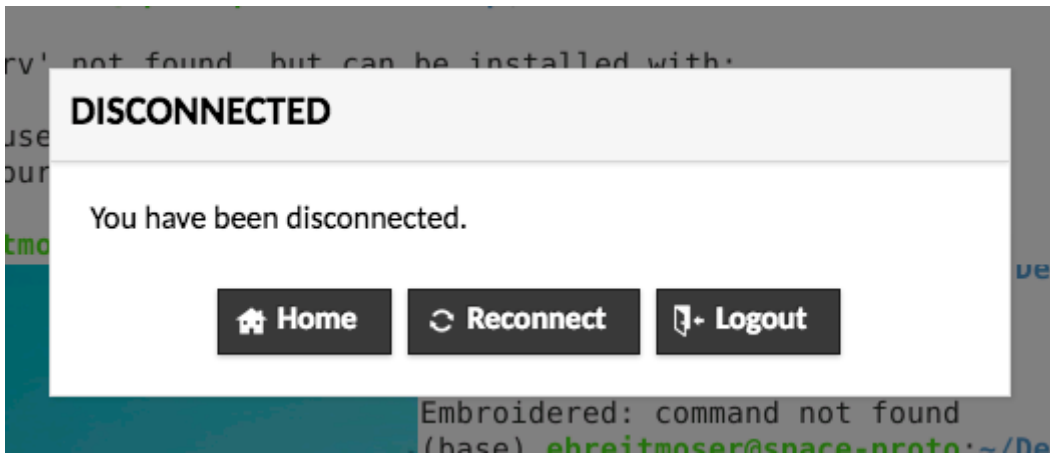
In this space, large files can be shared among users. If you are part of a project that has requested shared directories, there should be a folder that you can access and share files, which are controlled by group-level permissions. If you think you should have access to certain project folders, please contact either the leader of the project in the first instance, or the [SPACe helpdesk](#) if you do not know who the relevant person is, to organise access.

You can also create your own folder, and control permissions on access, from personal access only to access based on groups you are a member of, or even completely open to other users within the Workbench. This can be done by right-clicking on the folder, selecting the permissions tab, and making your selections there. Advanced users can also use the command line tool `chmod` if you are familiar with it.

Logging out from the Desktop

To log out from your Desktop connection click the start menu icon in the bottom left corner and click the Power icon and confirm when asked.

You can then return to the AWB Gateway by choosing Home or select Reconnect if you want the session to start again, or logout if you want to completely close the session.



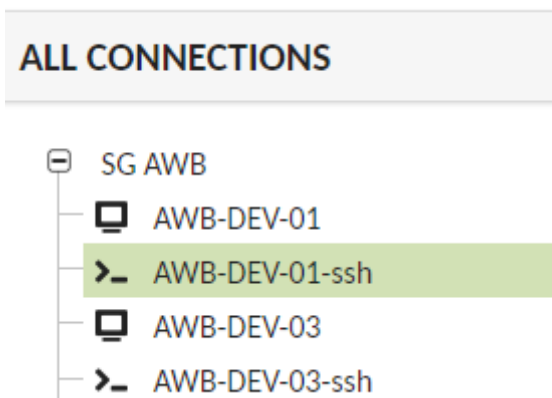
Disconnect from session

3.4.2 Command Line (SSH) Interface

The command line view can be useful for low-bandwidth options, and is also required for logging in if passwords need changing (e.g. for first logins or after your password has expired).

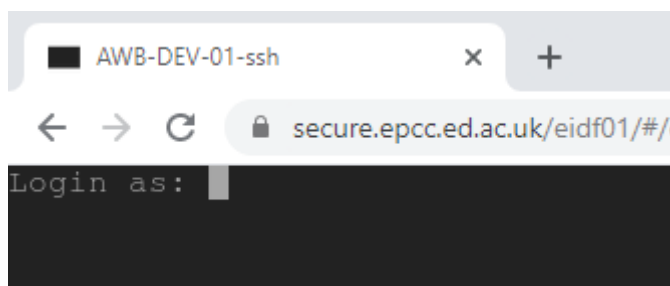
Access to the Linux Command Line view

To log into the command line view select the SSH connection to our Linux VM from the main Analytical Workbench page.



Starting into a VM Shell session

You will be prompted for your Linux VM username and password (this is the same as your password for the Desktop VM).



Logging into a VM Shell session

This will give you a command line view, which you can use like any other command line.

For new users or those with expired passwords, the command line will ask you for your current password, and request that you create a new password. This will apply to all VMs that you log in with the same username.

Changing your VM-password

To change your VM password type in the command `passwd` and then enter your existing password, your new password and confirmation of your new password.

Logging out from Command Line

To logout type in the command `exit` into the shell and hit

You will then see a prompt menu. As with the Linux Desktop VM, select Home if you want to return to the project Manager, select Reconnect if you want the session to start again, or logout to exit the session.

3.5 Using the software on your VM

3.5.1 Available Software

Each Vm you access has a default set of software installed. These are kept up to date where possible through regular security patching of the VMs during maintenance windows. An overview of the types of software available is below.

Linux

PROGRAMMING

- Anaconda Python (Python, conda)
- Jupyter Notebooks/Lab
- R and R Studio
- Visual Studio Code

DATABASE

- PostGIS
- PostgreSQL

OTHER

- Libre Office (Word, Excel, Powerpoint equivalents)
- QGIS
- Web Browser

Windows

PROGRAMMING

- Anaconda Python (Python, Conda)
- Jupyter Notebooks/Lab
- R and R Studio
- Visual Studio Code

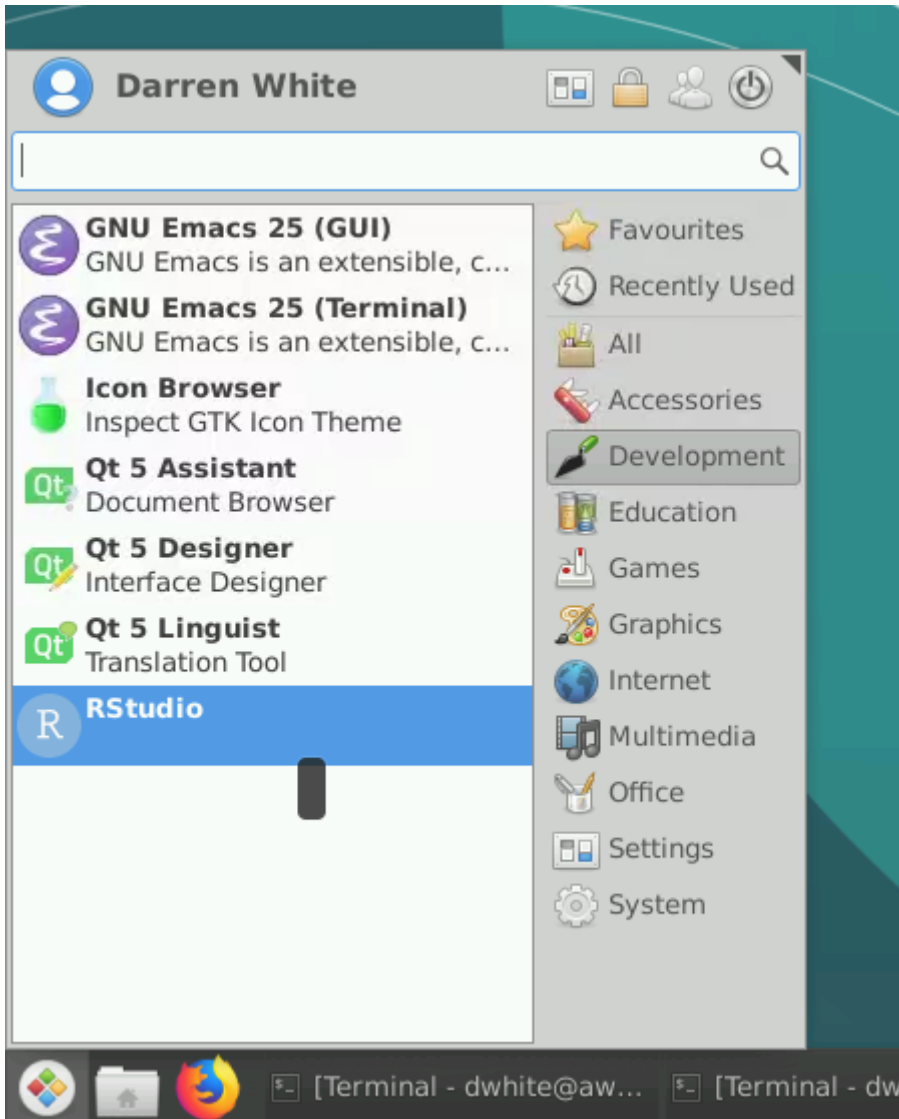
OTHER

- Microsoft Office
- NVivo
- QGIS
- Stata
- Web Browser

3.5.2 RStudio

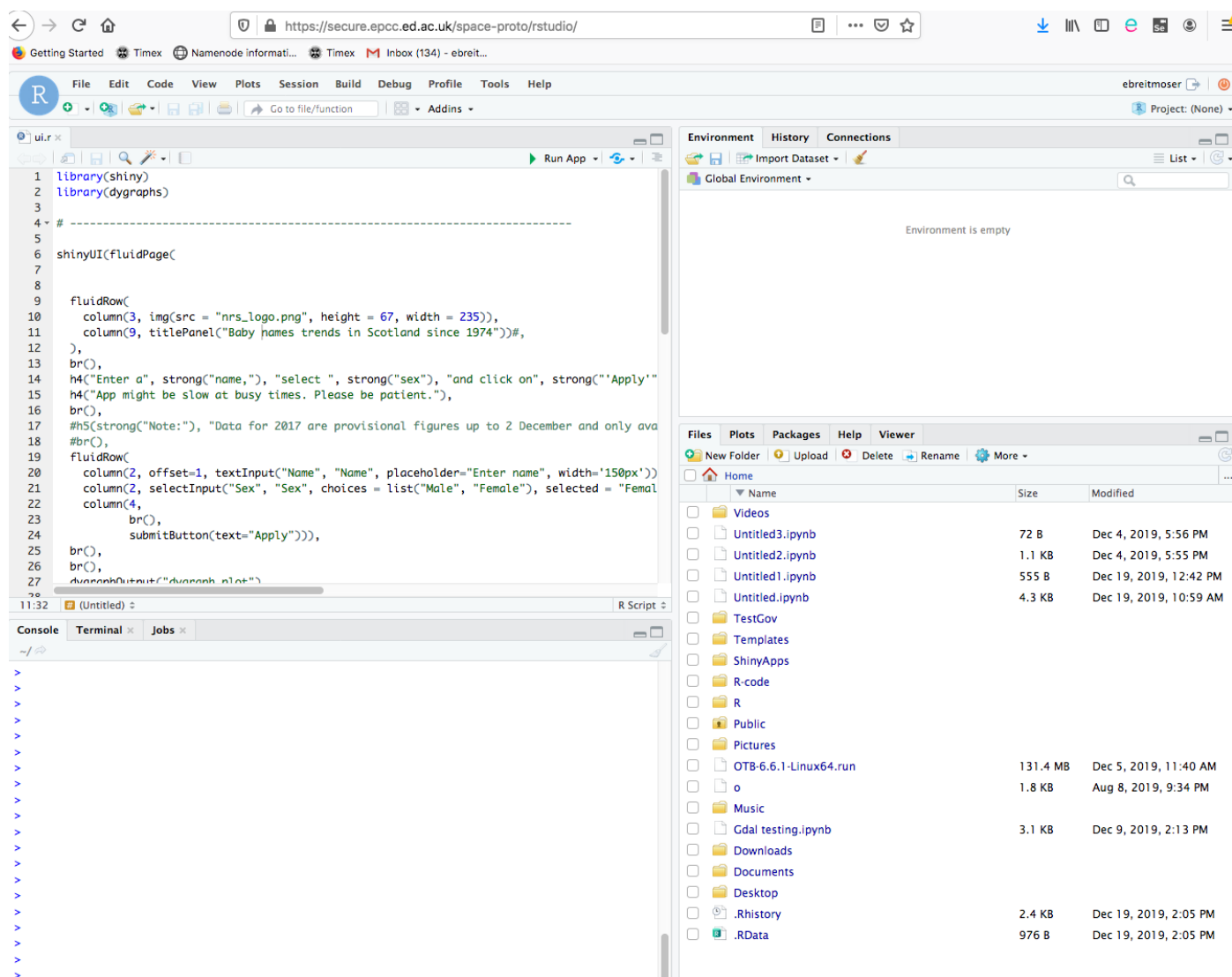
R with RStudio is one of the key tools in the Desktop view. You can access it using the Applications menu in the bottom left hand corner, R is under Graphics and RStudio is under Development.

The current installed version of R is 4.0.5. For more information on installing and accessing other versions of RStudio, see "[Other software](#)."



Start R-Studio

You will now have an RStudio session open and this should work as you would expect.



R-Studio open session

To add data to your RStudio session you can get data in the usual way from the local areas on the VM. This data you will have added to the VM through the methods described in "[Uploading data.](#)"

There are a number of Shortcuts that can be used with RStudio. These shortcuts involve using the **Shift** button please avoid the **Shift** being the first key pressed. For example by using this order

Ctrl + **Shift** + **R**

You will open a new terminal (the order listed in the shortcuts help page is **Shift** + **Ctrl** + **R**). The following shortcut keys for RStudio seem to work fine, whereas some others don't. This is under investigation and is related to shortcuts in the browser window. The following keys seem to work, using Firefox 73.0:

- **Alt** + **Shift** + **G**
- **Alt** + **Shift** + **K**
- **Alt** + **Shift** + **T**
- **Ctrl** + **Shift** + **1**

You can also use R directly this will give you the normal command line access to R.

Adding Packages to R and RStudio

One of the major benefits of the Analytical Workbench is the freedom to add whatever R packages you need to use when you want to use them. Packages should mainly be installed from CRAN aka Comprehensive R Archive Network.

← → ↻ cran.r-project.org/web/packages/available_packages_by_name.html

Available CRAN Packages By Name

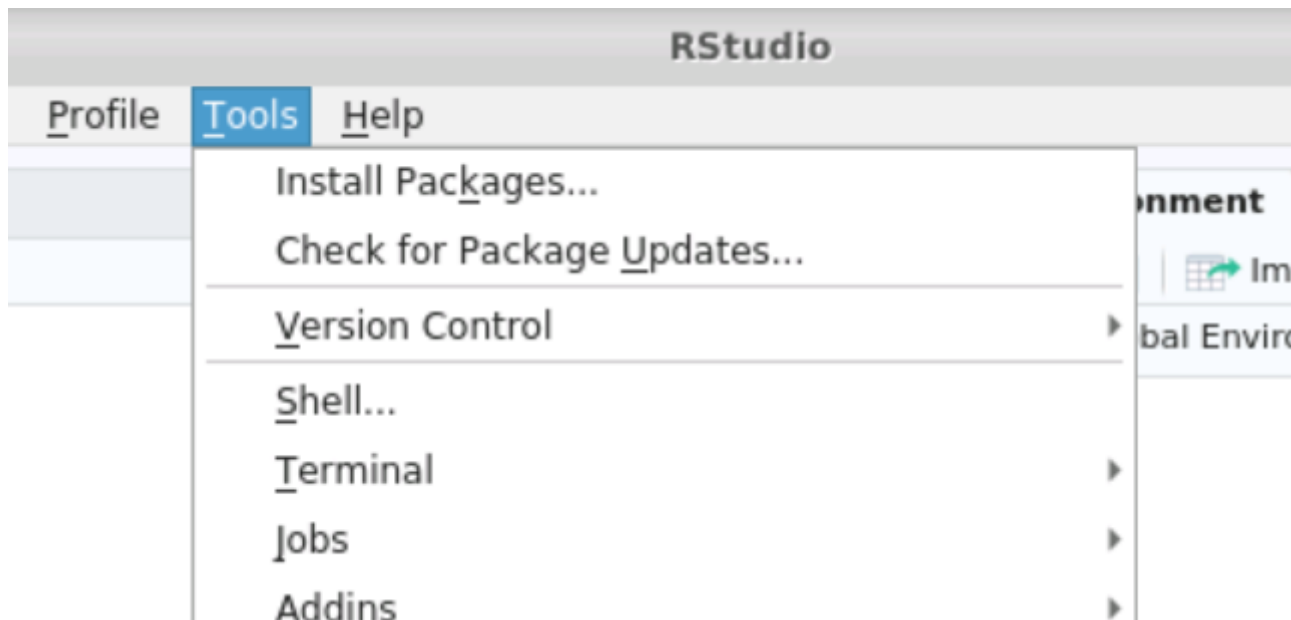
[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [Z](#)

A3	Accurate, Adaptable, and Accessible Error Metrics for Predictive Models
aaSEA	Amino Acid Substitution Effect Analyser
ABACUS	Apps Based Activities for Communicating and Understanding Statistics
abbyyR	Access to Abbyy Optical Character Recognition (OCR) API
abc	Tools for Approximate Bayesian Computation (ABC)
abc.data	Data Only: Tools for Approximate Bayesian Computation (ABC)
ABC.RAP	Array Based CpG Region Analysis Pipeline
abcADM	Fit Accumulated Damage Models and Estimate Reliability using ABC
ABCanalysis	Computed ABC Analysis

CRAN Package Repository for R

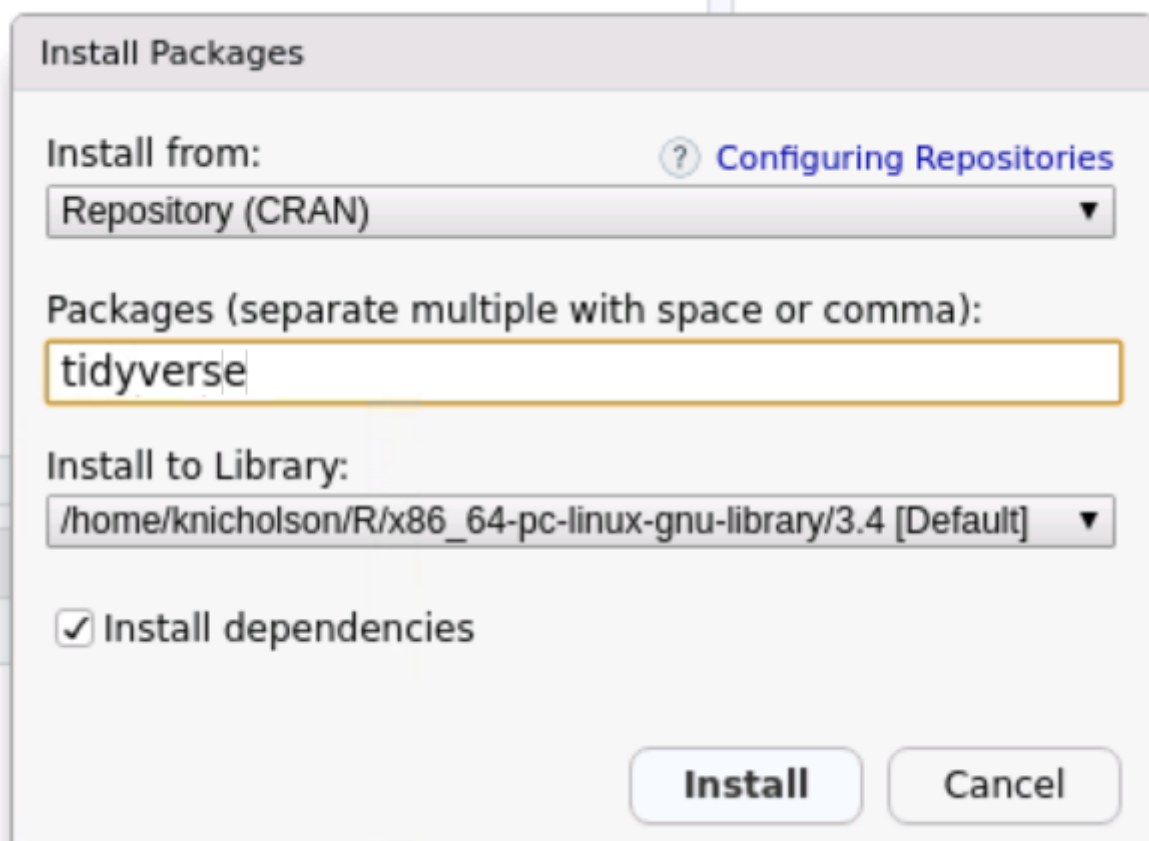
There are many packages available on CRAN that provide a wide range of functionality to R. There are two ways you can install these packages to R.

The first way is to use the dropdown menu to install a package from CRAN. From the Tools menu select Install Packages...



R-Studio open session

This will give you a dialogue box into which you can type the package you want. RStudio will help you to find the package as you type. Please note it might take some time to install the package depending on how large it is. You will see text in the console showing the package being installed. When it says that 'The downloaded source packages are in' you know it has finished.



Selecting a package to install

```
Console Terminal x Jobs x
~/ ↵
*** copying figures
** building package indices
** installing vignettes
** testing if installed package can be loaded
* DONE (tidyverse)

The downloaded source packages are in
  '/tmp/RtmpsBQ792/downloaded_packages'
> |
```

Console output after installing a package

Another to install packages is directly in the console within RStudio. Type `install.packages("NameOfPackage")` to install the package from CRAN. It is also possible to install R packages using the command line version of R, and also within your own anaconda environments, if using [Conda](#).

3.5.3 Python

A full version of Anaconda is installed for all users by default, which comes with Python 3.7.3 along with a variety of default packages. If users only require basic python, this should be sufficient to use. However, if users wish to install and use specific python packages beyond this (or even different versions of python/R/etc, [see R section](#)), we highly recommend doing so within a self-contained `conda` virtual environment.

Anaconda includes the tool `conda`, which we can make use of to create our own environments. An environment isolates software from other dependencies installed on the system and allows multiple versions of software to be installed and run side-by-side. This is useful if you want to test new versions of software, have older scripts that rely on specific versions or if you are doing development work.

Check out the [conda workflow](#).

3.5.4 Conda

Conda is initialised in your command line by running.

```
Bash  
conda init bash
```

This adds some startup code to your terminal so that `conda` will be able to work. You should not need to run this command again.

Default environment directory

By default, `conda` will try to create environments where the default or `base` environment exists. Regular users do not have permissions for this directory so we must direct our environments to a location in each users home directory. This is done by adding a few lines to the `.condarc` file which is where the user configuration for `conda` is stored.

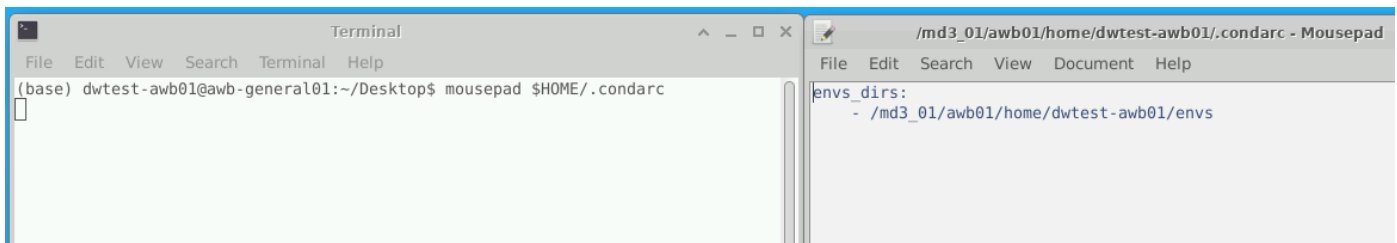
Edit the file `.condarc` file using `mousepad`

```
Bash  
mousepad ~/md3_01/<project>/home/<user>/.condarc`
```

If the file is empty, add the following lines, replacing the username as appropriate. If it is not empty, add the line under the `envs_dirs` heading, putting it at the top of the list:

```
Bash  
envs_dirs:  
- ~/md3_01/<project>/home/<user>/envs
```

Save the file to disk using `File > Save`.



Save the modified `.condarc` file.

Once this has been done, any `conda` environments you create will be installed into this location. `conda` environments can then be created as normal and they will be installed to the location specified in the `.condarc` file.

Create a new conda environment

Conda has many subcommands that allow for the creating, management and removal of environments. The full `conda` manual is [here](#).

For example, to create a new environment

```
Bash  
conda create -n MyEnv python=3.7
```

which creates a new environment with the name `MyEnv`, and installs a fresh copy of your selected Python version with its dependencies (in our case version 3.7).

To activate the new env

Bash

```
conda activate MyEnv
```

You can then install packages using `conda install` or `pip install`.

Conda Channels

In order to install new software and packages, conda makes use of package "channels". These channels can be broad collections of more recent software (such as conda-forge) that hasn't made into the main conda repository, or be domain specific (e.g. bioconda or astroconda). New channels can typically be added with

Bash

```
conda config --add channels <channel-name>
```

Not all packages will be in conda, but conda also contains many non-python tools and packages that users can install (e.g. `pip`).

3.5.5 Anaconda

Installing new software with Anaconda

Due to the shared nature of the existing Analytical Workbench, installing new software system-wide is not possible for users. However, it is possible to make use of virtual environments to enable users to install local versions of software that they can then use. As an example, we are going to use the `conda` environment and package management system to install a new version of R.

Please follow the [previous instructions](#) to make sure you have initialised your `conda` package manager correctly.

First create a new environment

Bash

```
conda create -n R3.6.2 python=3.7
```

this creates a new environment with the name `R3.6.2` (or whichever name you choose), and installs a fresh copy of your selected Python version (in our case version 3.7).

In order to install new software, `conda` makes use of package "channels". These channels can be broad collections of more recent software (such as `conda-forge`) that hasn't made into the main `conda` repository, or be domain specific (e.g. `bioconda` or `astroconda`). In this example we will make use of the `conda-forge` channel and add it to our configuration file.

Bash

```
conda config --add channels conda-forge
```

then activate the environment

Bash

```
conda activate R3.6.2
```

Then install a new base version of R

Bash

```
conda install r-base=3.6.2
```

Make sure to include version numbers when it is important to have specific versions. This command will install the package selected, as well as all prerequisite software (dependencies).

Once done, this version of R can now be used in the command line. In order to use RStudio with this version, we must tell RStudio which R installation to use. This is done by exporting an environment variable, and then running `rstudio` from the command line to launch the GUI

Text Only

```
export RSTUDIO_WHICH_R=$(which R)
rstudio
```

This is only an example on updating the base version of R. It is also possible to install a variety of R packages and tools alongside this using the `conda` package manager. A good place to start learning `conda` is the [conda docs](#), which contains tips, cheat sheets, and links to other resources.

3.5.6 Jupyter Notebook

Jupyter Notebooks are open source web applications that allow you to create and share documents that contain live code, equations, visualisations and narrative text. They are an ideal way of sharing ideas, training, and learning to code. You can run a wide range of programming languages in Jupyter but currently only Python3 and R are available on the Linux Desktop VM. You can also open files, folders and even access the terminal within the notebook.

Accessing Jupyter Notebook

The Jupyter Notebook can be accessed via the Terminal.

Bash

```
jupyter-notebook
```

This will start the notebook server and open a web browser (or new tab) containing your jupyter notebook to work in, as described in the next section.

Accessing Jupyter Lab

Jupyter Lab is the replacement packages for Notebooks. It offers many of the same features but also includes a more fully featured development environment to improve the user experience in notebook management, debugging and state exploration. It can be started from the command-line

Bash

```
jupyter lab
```

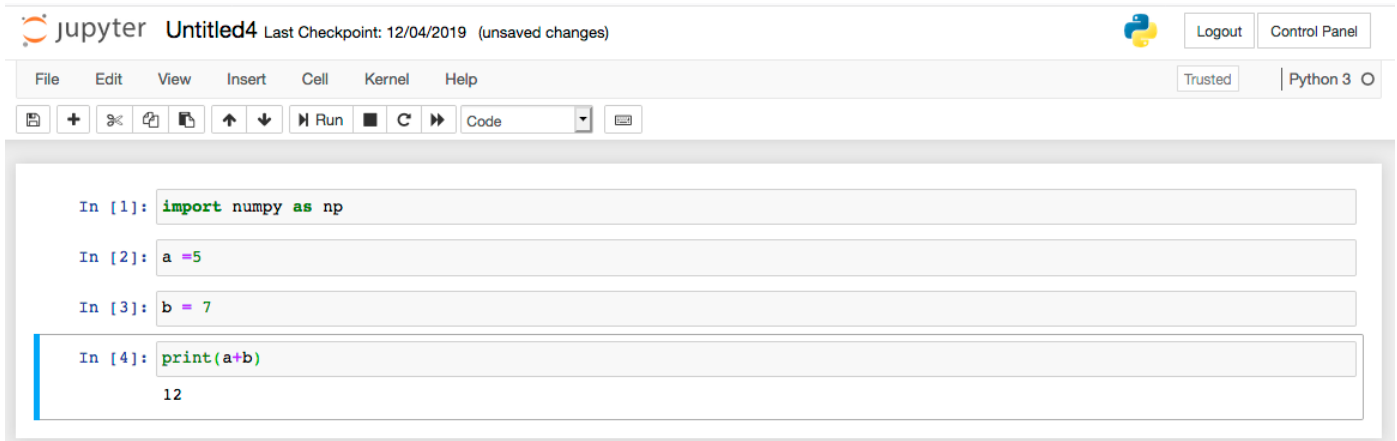
Using Notebooks

The first view you will see is your VM file structure. From here you can choose to open any existing Jupyter Notebooks or you can open a new notebook in a coding language such as Python3 or Terminal by clicking on the New button in the right top corner.



Jupyter start page

Python 3 opens a Python Notebook which you will be used to



The screenshot shows a Jupyter Notebook window titled "Untitled4" with a last checkpoint of "12/04/2019 (unsaved changes)". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Help), a toolbar with icons for file operations and execution, and a code editor. The code editor contains four input cells:

```
In [1]: import numpy as np
In [2]: a = 5
In [3]: b = 7
In [4]: print(a+b)
12
```

Example notebook

Whereas Terminal opens a Unix Command Line from where you could for example clone an external git-repository



The screenshot shows a Jupyter Notebook terminal window with the Jupyter logo and "Logout" and "Control Panel" buttons. The terminal displays the output of a git clone command:

```
(base) ebreitmoser@space-proto:~$ git clone https://github.com/ebreitmo/DataScience.git TestDir
Cloning into 'TestDir'...
remote: Enumerating objects: 17, done.
remote: Counting objects: 100% (17/17), done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 17 (delta 4), reused 17 (delta 4), pack-reused 0
Unpacking objects: 100% (17/17), done.
(base) ebreitmoser@space-proto:~$
```

Using the Jupyter command line

As with RStudio, data you upload to the VM via the data ingest methods mentioned in section 3 can be accessed within your Notebooks.

You can also install packages for use with jupyter-notebook, but we recommend doing so within a virtual environment and installing a fresh jupyter-notebook in your environment if doing so.

3.5.7 QGIS

QGIS is an open source tool for mapping and other GIS functions. There are two ways of accessing QGIS you can either use the command line to directly access QGIS by using a Terminal window and typing:

Bash

```
qgis &
```

Or you can select QGIS from the applications start menu.

The current version of QGIS is 3.22.

3.5.8 Orfeo Toolbox

Orfeo Toolbox (OTB) is an open source tool for the processing of remote sensing images. This tool can be accessed through the the Terminal and can be loaded using

Bash

```
source /opt/otb/OTB-7.1.0-Linux64/otbenv.profile
```

Orfeo tools can then be launched as normal. For example:

Bash

```
otbApplicationLauncherQt SplitImage
```

3.5.9 Other Software

LibreOffice

The full LibreOffice open source tool is available in the Desktop view, under the 'Education' section of the start menu.

LibreOffice is a tool that is similar to the Microsoft Office suite and all these tools will open any Microsoft document. It includes a range of tools that you can use including:

- LibreOffice Calc -- Spreadsheet tool compatible with `.xlsx` files.
- LibreOffice Math -- To visualize mathematical equations.
- LibreOffice Writer -- Similar to word and compatible with `.docx` files.

Web browsers

The default installed browser in the Desktop view is Firefox. Google Chrome is also installed.

Building software from source

For more technical users, it is also possible to download, build and install software into your home directories, without affecting other users, by using the '--prefix' option during the configure stage.

Installing system-wide software on project-specific VMs

Users accessing AWB VMs will be standard users with no sudo rights.

Specified sudo commands can be assigned to users if and as when required and on a single user basis, such as when creating custom, project VMs. Users can get access to specific commands but never root access to the VM itself. This will allow us to control access to data across all VMs and prevent data breaches. If you are setting up a project-specific VM and you wish to install additional software, please contact [SPACe helpdesk](#) to request privileged user rights to do so.

4. Shiny server user guide

4.1 The R Shiny Info

This page covers the internal-facing, non-public shiny server for SPACe. We distinguish between three roles:

- The data analyst: The people who are developing apps and want to have them deployed on behalf of the SG on the shiny server. All the development work will be done exclusively on the dev-VM. Obviously, they should also be able to see and interact with the app on the production/live VM at <https://eidf.epcc.ed.ac.uk/space-shiny/> as if they were a normal user.
- The user: The people accessing the apps on the production/live VM at <https://eidf.epcc.ed.ac.uk/space-shiny/> and interacting with them there.
- The sudo user: SG person who helps the data analyst in the deployment process. Only they will be able to move the app to the test and production/live VMs.

This guide covers the processes from the data analysts' perspective.

Contact the [SPACe helpdesk](#) if you have any questions.

4.2 Overview

There are three VMs in total for the shiny app development and deployment for the SPACE project:

	dev-VM	test-VM	production-VM
VM name	space-rshiny-dev	N/A	N/A
URL to view apps	https://localhost:3939 (only locally from browser in the dev-VM)	https://eidf.epcc.ed.ac.uk/space-test/login	https://eidf.epcc.ed.ac.uk/space-shiny/login
Purpose	development of SPACE apps	Establish if app is ready for deployment on production-VM. In case of failure return to dev-VM.	External users with whitelisted IP addresses can see/interact with the apps. Currently: simple password protection/login required from those users)
People with access rights	SPACE app developers/data analysts	N/A	N/A
Access	via guacamole/desktop GUI	N/A	N/A

4.3 Application Development

4.3.1 Getting a User Account

Project-based usernames are used to make sure accounts can only see folders/data in that project.

If a user is working on two projects, then they would have two separate accounts.

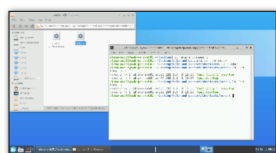
Users who require access to the shiny app development should contact the [SPACE helpdesk](#).

4.3.2 How to deploy your app on the dev-VM

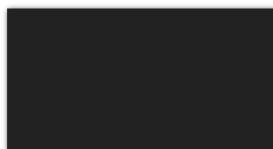
For the current shiny server we use [rshiny proxy](#) and version 2.5.0, i.e. `shinyproxy-2.5.0.jar`.

To access the VM apply for an account via the [SPACE helpdesk](#).

The VM can be accessed the same way as the other SPACE VM(s) in use for SPACE users and as described in the [Analytical Workbench user guide](#). It is visible to you after logging onto the guacamole level. Its name is `space-rshiny-dev`.



AWB-General01



WIN10-VM3



AWB-DEV-02

ALL CONNECTIONS

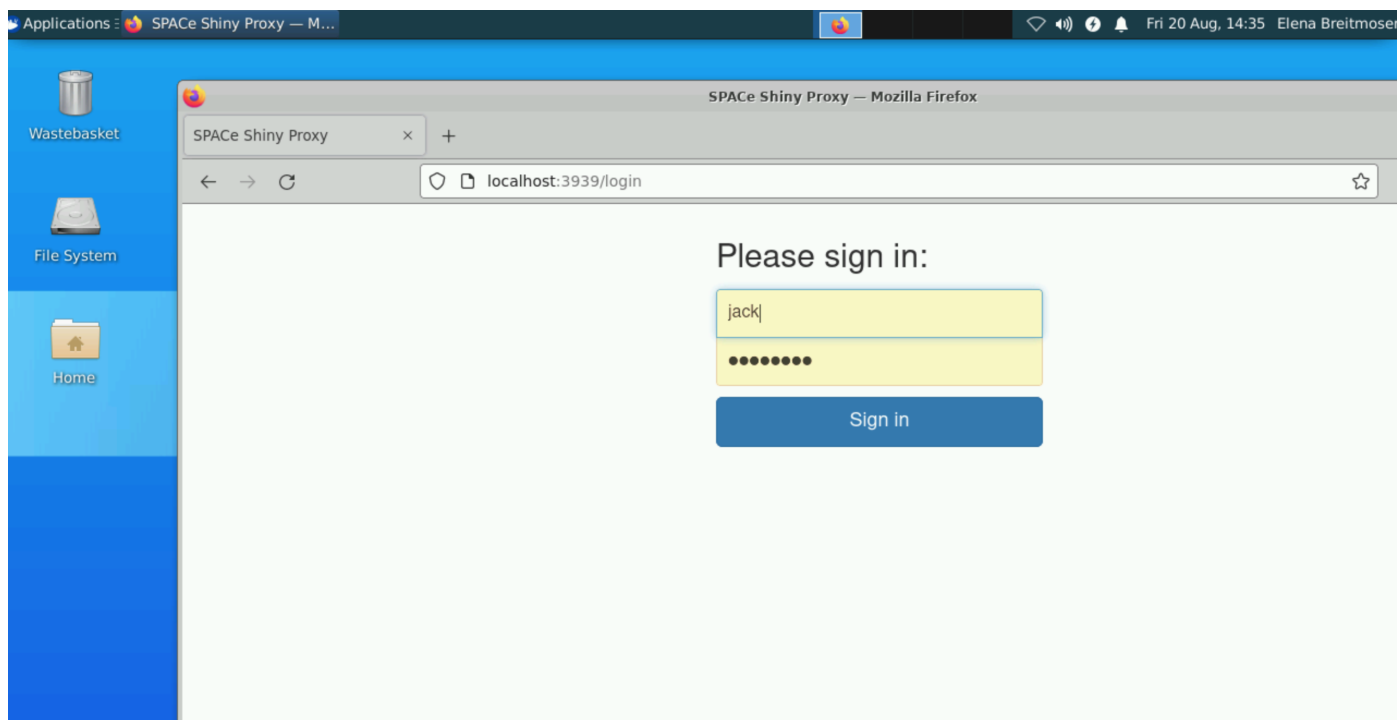
Filter

- SG AWB
 - AWB-DEV-02
 - AWB-DEV-02-ssh
 - AWB-DEV-03
 - AWB-DEV-03-ssh
 - AWB-General01
 - AWB-General01-ssh
 - AWB-SERVICES-01
 - AWB-SERVICES-01-ssh
 - AWB-WEBAPP-01
 - AWB-WEBAPP-01-ssh
 - space-rshiny-dev
 - space-rshiny01-ssh

Logging into an `rshiny-dev` session

You also need to have a username and password to be able to access the website `https://localhost:3939` to see how your app - once deployed - would look to the external user.

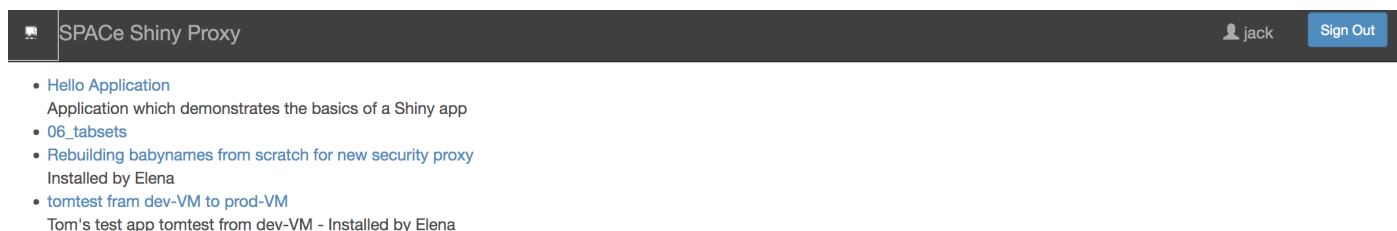
Accounts can be created by contacting the [SPACE helpdesk](#).



Logging into an `rshiny-proxy`

Once you are logged onto the server:

Clicking on the title bar 'SPACe Shiny Proxy' (in the example below), will always bring you back to the landing page, which shows you all the apps you can view.



`rshiny-proxy` landing page

Where to put your app

In the following we assume there is an app `myappname`, which is to be deployed from the dev-VM, with the following content:

Bash

```
-rwxr--r-- 1 username  username849588 Mar 20 2018 mydata.rdata
-rwxr--r-- 1 username  username 1177 Dec 5 2018 server.R
-rwxr--r-- 1 username  username 3189 Dec 5 2018 ui.r
drwxr--r-- 3 username  username 55 Dec 5 2018 www
```

It is essential that you have a `ui.R` and a `server.R` file for your app (see [deploying apps](#)). It is fine to have additional `*.R`-files, data, images, etc. as part of your app.

If your app doesn't have this file structure you will have to split your existing app into the right files. If you are using the `flexdashboard` package/`rmarkdown` please read [What's different when you use the flexdashboard package/rmarkdown](#) below.

You can transfer this app to the dev-VM and the correct directory - as outlined below - while it is compressed, e.g. by using `tar` and `zip/gzip` or by e.g. cloning an existing git repository of yours.

If you are not used to the editors available on `space-rshiny-dev`, you may prefer to add and adapt the Dockerfile (see points 4. and 6. below) before you transfer everything to `space-rshiny-dev`. Note that the directory-level structure (the Dockerfile in relation to the app code) will be important on `space-rshiny-dev` (see points 2. to 5. below) so that you can build your app successfully.

1. Open a terminal window.
2. Go to your HOME directory `/home/<username>` using the `cd` command. That command brings you to your HOME directory. Use the command `pwd` to check which directory you are currently in.
3. Create a new directory with the name of the app `myappname` using the command `mkdir myappname`. You should now have a directory `/home/<username>/myappname` (please use lowercase names for these directories, the procedure is case sensitive!).
4. Copy the [Dockerfile template](#) into the directory `/home/<username>/myappname` (The spelling and use of upper/lowercase letters is crucial, so please call it exactly 'Dockerfile'! If you open this document within a browser launched from within the EIDF Service VDI you can directly copy and paste.) So now you have the following directories and files:

Text Only

```
- /home/<username>/myappname/
- Dockerfile
- myappname/
  |_ mydata.rdata
  |_ server.R
  |_ ui.r
  |_ www/
```

Copy the app `myappname` into the directory `/home/<username>/myappname`. Use `ServU` or `git` to transfer your app to the VM.

So, assuming the app name was `shinyservertestapp` the file structure would be:

Top level app folder And the 'shinyservertestapp' directory could contain, for example:

![Image requires alt text](images/ShinyServerFileStructure2.png)

Second level app folder

Configuring Docker Edit and adapt the Dockerfile template to be updated (Use e.g. 'emacs', 'vi' or 'nano' to edit the file) in `/home/myappname/`` a) Replace all occurrences of ``myappname`` with the name of your app, which should be identical to the directory it is in (lines 32, twice in line 33, 37) b) Replace all occurrences of ``` with your username (lines 31, 32, 33, 37). c) Add all the packages your app requires to be installed, in the example below, the app ``myappname`` requires the packages 'dygraphs' and 'stringr'. (lines 25, 26) ### Running Docker

Bash

```
docker build -t myappname . |tee BuildOutput.txt
```

To build the docker image (**the first time you do this for a given app it might take a while and produce a lot of output on the command line**); so only include essential packages to the Dockerfile to save compilation time!). The second part of the command ``tee BuildOutput.txt`` will write the output to the file `BuildOutput.txt` as well; this will make it easier to spot any error messages for (you and/or) the [SPACE helpdesk](../contact.md) team in case there will be any. You have to re-run this command any time you make changes to your app (i.e. the code) and want them to be deployed. You will see from the output when re-building it that it might use the build cache. Use ``docker build --no-cache -t myappname .`` to avoid this. Run the command ``groups`` to see if ``docker`` is one of the groups you belong to

Bash

```
[ebreitmoser@space-rshiny-dev myappname]$ groups
ebreitmoser docker
```

In case you haven't been added to the docker group you will get an error message, contact the [SPACe helpdesk](../contact.md).
Check your docker image When the command completed successfully, you can check the new image `myappname` with the docker command `docker images`.

Bash

```
$ docker images

REPOSITORY TAG IMAGE ID CREATED SIZE
myappname latest cd2b192c3cdb 2 second ago 945 MB
```

Another optional check to see if the docker image was built really successfully is to run

Bash

```
$ run -p 3838:3838 "myappname"
```

from the command line. The output would either display an error message - or if successful:

Bash

```
...
Listening on http://0.0.0.0:3838
```

(It is possible that the port 3838 is already used by another process, you would get a related error message, which would have nothing to do with the app not having built correctly, though. In this case you would have to ask the [SPACe helpdesk](../contact.md) to run the command for you instead.) Then - to see what your app would look like - open `localhost:3838` in a web browser on the dev-VM. If there are further errors they will be displayed on the command line from where you ran the docker image. Have the new app added to the configuration file 'application.yml' by contacting the [SPACe helpdesk](../contact.md). Once this has been done the app will be accessible at `https://localhost:3939` by all users listed in the configuration file and belonging to the same `group`. **### Adding the app to the rshiny-proxy configuration** For the production service the app creator has to add the following information when contacting the [SPACe helpdesk](../contact.md):

Info	Description
Id	Name of the docker image, run `docker images` to [check](#check-your-docker-image) what you as developer above
Display-name	display name of the app on the landing page, developer's choice
Description	Description which shows below the display name on the landing page, developer's choice
Container-cmd	["R", "-e", ":",run('/root//,host='0.0.0.0',port=3838)"] Technical details about the app, rmarkdown, etc., look in your Dockerfile what you used
Container-image	Name of the docker image
Access-groups	Which group(s) should be allowed to view the app and interact with it (see [here](#how-to-make-apps-visible-to-certain-users-only))

What's different when you use the 'flexdashboard' package/rmarkdown In the following we assume there is an app consisting of one file 'flexdashboard.Rmd' (you can name it anything you like plus the extension '.Rmd'), which is to be deployed from the dev-VM and later from the production/live VM. [Follow the app deployment instructions](#where-to-put-your-app), only this time the file 'flexdashboard.Rmd' will be in the *same* directory as the Dockerfile. [Configure docker](#configuring-docker), with the following differences - the name after the COPY command has the extension 'Rmd' - the syntax of the last line is slightly different

Text Only

```
CMD R -e "rmarkdown::render('flexdashboard.Rmd')"
# copy the app to the image
RUN mkdir /root/<username>
COPY flexdashboard.Rmd /root/<username>/flexdashboard.Rmd

EXPOSE 3838

CMD ["R", "-e", "rmarkdown::run('/root/<username>/flexdashboard.Rmd',shiny_args = list(port = 3838, host='0.0.0.0'))"]
```

[Run docker](#running-docker) with the modified command `docker build -t flexdashboard .`. Then [follow](#check-your-docker-image) the remaining steps and let the [SPACe helpdesk](../contact.md) know that it is a flexdashboard app. **## The Dockerfile Template**

Text Only

```
FROM openanalytics/r-base

LABEL maintainer="Name, Email address"

RUN apt-key adv --keyserver keyserver.ubuntu.com --recv-keys E298A3A825C0D65DFD57CBB651716619E084DAB9
```

```
# system libraries of general use
RUN apt-get update && apt-get install -y \
  sudo \
  pandoc \
  pandoc-citeproc \
  libcurl4-gnutls-dev \
  libcairo2-dev \
  libxt-dev \
  libssl-dev \
  libssh2-1-dev \
  libssl1.1

# system library dependency for the myappname app
RUN apt-get update && apt-get install -y \
  libmpfr-dev

# basic shiny functionality
RUN R -e "install.packages(c('shiny', 'rmarkdown'), repos='https://cloud.r-project.org/')"

# install dependencies of the myappname app
RUN R -e "install.packages('dygraphs', repos='https://cloud.r-project.org/')"
RUN R -e "install.packages('stringr', repos='https://cloud.r-project.org/')"

# copy the app to the image
RUN mkdir /root/<username>
RUN mkdir /root/<username>/myappname
COPY myappname /root/<username>/myappname

EXPOSE 3838

CMD ["R", "-e", "shiny::runApp('/root/<username>/myappname', host='0.0.0.0', port=3838)"]
```

****Notes on docker configuration commands**** - A parent image is the image that your image is based on. It refers to the contents of the FROM directive in the Dockerfile. Each subsequent declaration in the Dockerfile modifies this parent image. Most Dockerfiles start from a parent image, rather than a base image. - COPY This command copies a file from your host OS to the docker container - RUN builds your application - CMD is used to run a command as soon as a container is launched. It is different from RUN as RUN is used at the time of building an image. - EXPOSE this command is used to specify the port number upon which the container is running its process for networking with the outside world (your host). #usage: EXPOSE [port] ## Choose your version for specific R-libraries Assume you would want to use a specific version of the R-library `stringr`, version 1.3.0 in this example. The Dockerfile template file have to be changed accordingly:

Text Only

```
# Install dependencies of the myappname app
RUN R -e "install.packages('dygraphs', repos='https://cloud.r-project.org/')"
RUN R -e "install.packages('remotes', repos='https://cloud.r-project.org/');library(remotes);remotes::install_version('stringr', version = '1.3.0',
repos='https://cloud.r-project.org/')"
```

Choose your version of R For shiny-proxy, the desired version of R can be accessed from within the Dockerfile. As a prerequisite [`Rocker`](<https://github.com/rocker-org/rocker/wiki>) has to be installed from within the Dockerfile. This image allows specifying the R version in the docker tag:

Text Only

```
# Install R version 3.6
#FROM openanalytics/r-base
FROM "rocker/r-ver:3.6.3"

LABEL maintainer="Name, Email address"

#RUN apt-key adv --keyserver keyserver.ubuntu.com --recv-keys E298A3A825C0D65DFD57CBB651716619E084DAB9

# System libraries of general use
...
```

Please speak to the [SPACE helpdesk](././contact.md) for further information. R Versions and package versions TBD ## Install packages from github You can install github packages. Assume your repository is `https://github.com/` you add the following lines into Dockerfile:

Text Only

```
...
RUN R -e "install.packages('remotes', repos='https://cloud.r-project.org/')"
RUN R -e "remotes::install_github('myrepo/myfiles')"
...
```

How to work on an app as a group Data analysts who want to collaborate on a project in a shared directory. Ask/email to request - The creation of a new shared directory, providing the directory/project name `` to be used and the usernames `` of all

people to get access to this new shared directory - New users with `` to be added to an existing shared directory in `/home/shared`
- Existing users with `` to be removed from an existing shared directory in `/home/shared` ## Using Rstudio In case you want to work on your app using RStudio on the dev-VM. Note that RStudio is not needed for the deployment of the app on the shiny server.

*[AWB]: Analytical Workbench *[CLI]: Command-line interface *[CPU]: Central processing unit; used throughout this site as a catch-all term for computer processing power *[CRAN]: The Comprehensive R Archive Network *[EIDF]: Edinburgh International Data Facility *[gateway]: The initial log-in page for the Analytical Workbench services *[Gateway]: The initial log-in page for the Analytical Workbench services *[Guacamole]: The initial log-in page for the Analytical Workbench services *[HTML]: Hyper Text Markup Language *[MS]: Microsoft *[OS]: Operating system *[PDF]: Portable Document Format *[pip]: A package management and installation system for Python *[Pip]: A package management and installation system for Python *[RDP]: Remote Desktop Protocol; used throughout this site to refer to the graphical user interface for the virtual machines *[SG]: Scottish Government *[SPACE]: Scottish Public Sector Analytical Collaborative *[SSH]: Secure Shell Protocol; used throughout this site to refer to the command-line interface for the Linux virtual machines *[VM]: Virtual machine *[VMs]: Virtual machines *[VPN]: Virtual private network *[VPNs]: Virtual private networks *[W3C]: World Wide Web Consortium *[WCAG]: Web Content Accessibility Guidelines

4.4 Test-VM

4.4.1 How to access the test-VM

The test/prod machines wont have a GUI. Because they will only be accessible via the dev-VM which means it is ssh only.

1. Log on to space-rshiny-dev
2. Open a terminal window
3. From the command line `ssh space-rshiny-test` has a key set up and will log you in directly (without asking for your password)
4. Although ssh-key authentication is in place from dev to prod and test, we recommend changing the passwords the first time you log on, using 'passwd' from the command line.

4.4.2 How to transfer apps from the dev-VM to the test-VM

To transfer an app from one VM to another, it is sufficient to transfer the docker image of the app.

Check the app's docker image name on the dev-VM

Bash

```
ebreitmoser@space-rshiny-dev:~$ docker images
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE
tomtest             latest     cade5d53c27e  42 hours ago  1.05GB
babynames           latest     9cbbe0be983b  6 days ago    1.05GB
openanalytics/r-base latest     dc8921f0456d  3 months ago  588MB
hello-world         latest     d1165f221234  5 months ago  13.3kB
openanalytics/shinyproxy-demo latest     9a83d7d80ebf  2 years ago   925MB
```

Create a tarball of the docker image you want to transfer, e.g. 'babynames'.

Text Only

```
docker save babynames > babynames.tar
```

Copy the tarball to the test-VM

Text Only

```
scp -r babynames.tar ebreitmoser@space-rshiny-test:/home/ebreitmoser
```

Load the image into the local registry using command 'docker load ' and check if it is there by running 'docker images'

Text Only

```
docker load < babynames.tar

ebreitmoser@space-rshiny-test:~$ docker images
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE
babynames           latest     9cbbe0be983b  6 days ago    1.05GB
```

Then the same procedures as described under 503328396503328396 has to be followed to add the app to the configuration file application.yml on the test-VM. You will need some information from the dev-VM/the developer:

- the display name the app is supposed to get (TBC by SPACe who is responsible for deciding titles etc)
- the description
- the container-cmd

4.5 The prod-VM

4.5.1 How to access the prod-VM

The test/prod machines wont have a GUI. Because they will only be accessible via the dev-VM which means it is ssh only.

- Log on to `space-rshiny-dev`
- Open a terminal window
- From the command line `ssh space-rshiny-prod` has a key set up and will log you in directly (without asking for your password).

Although ssh-key authentication is in place from dev to prod and test, we recommend changing the passwords the first time you log on, using 'passwd' from the command line.

4.5.2 How to transfer apps from the dev-VM to the test-VM

Follow the same process as described in 503328396503328396, only anything rleated to 'test-VM' has to be replaced by 'prod-VM'.

4.5.3 Usage Statistics

Simple usage statistics

If you have admin rights (as defined in `/usr/bin/shiny/application.yml` in `admin-groups:`) go to the `/admin`, e.g. `https://eidf.epcc.ed.ac.u/space-shiny/admin`. This will show you what apps are currently active and who is accessing them. One or more groups (as defined in the authentication back-end) can have access to the administrative interface of ShinyProxy e.g. to view and manage active sessions.

In the example below user 'jack' is currently accessing the app '01_hello', and it has just started running (see Uptime).



Active Proxies

ID	User	Appname	Endpoint	Uptime
dacb2711-2159-4ae4-b7f5-5f79ce71b27e	jack	01_hello	{dacb2711-2159-4ae4-b7f5-5f79ce71b27e=http://localhost:20000}	0:00:00

Example active shiny proxies

More in-depth usage statistics on the prod-VM using Micrometer

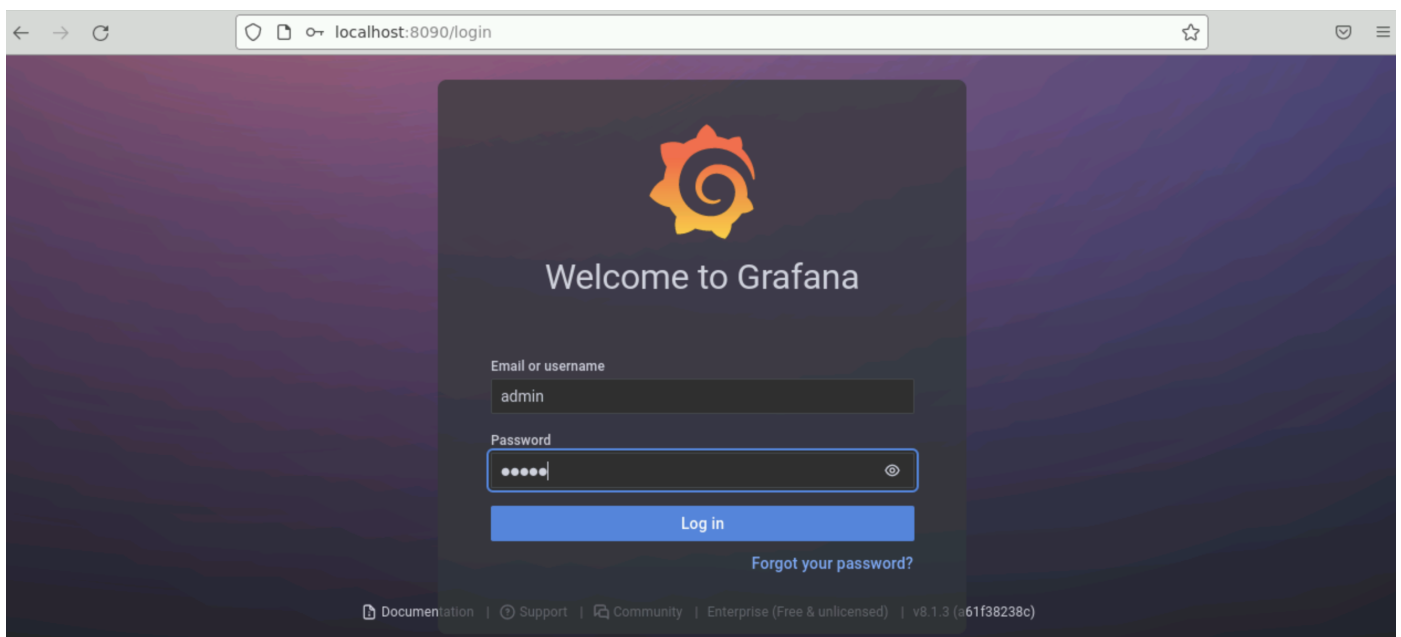
Following <https://www.shinyproxy.io/documentation/usage-statistics/>. Starting from ShinyProxy 2.5.0 an extra plug-in was added that supports for Micrometer.io. Micrometer itself is a modular library that allows to support many popular monitoring system. Our focus is to export Prometheus Metrics.

ShinyProxy can keep track of the following actions:

- User Login
- User Logout (also by expiring session)
- Authentication failure
- Application start
- Application stop
- Duration of application startup
- Duration of application usage
- Application start failures

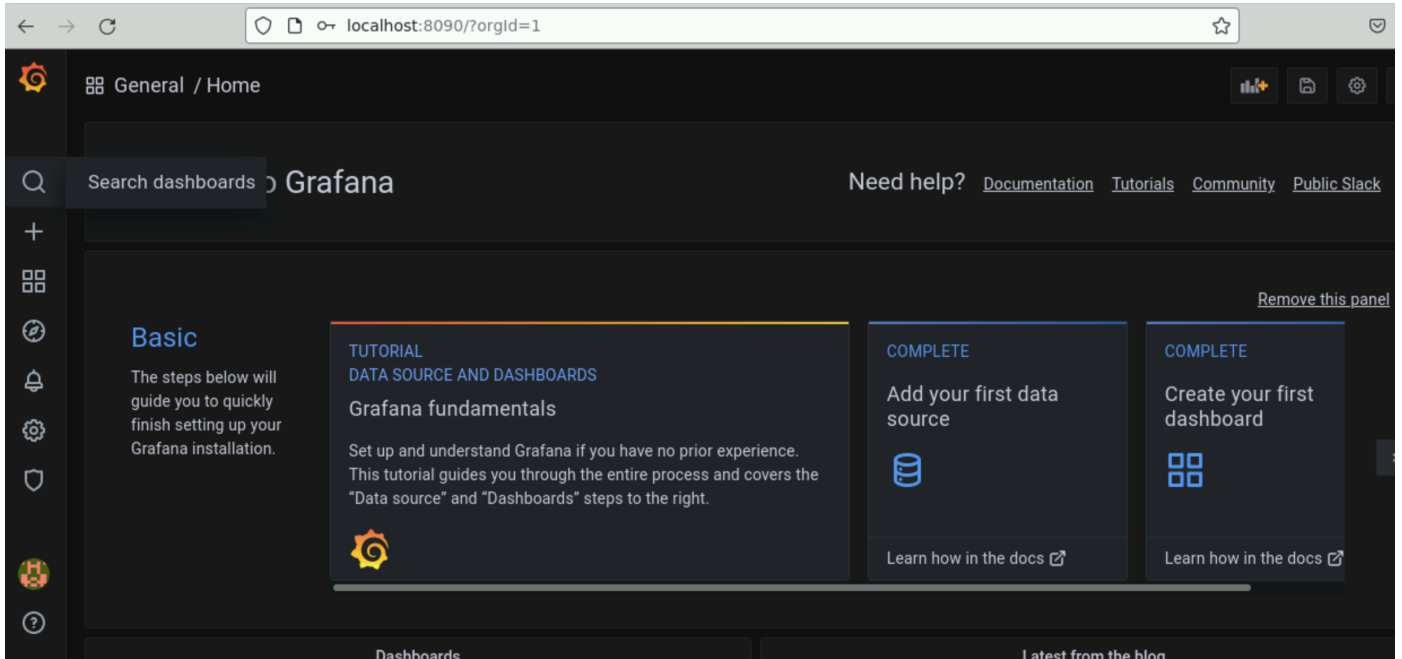
4.5.4 How to access the monitoring tools (using Prometheus and Grafana)

- Log onto the space-rshiny-dev VM desktop
- Open a terminal window
- Open a web browser
- In the terminal window type: 'ssh -L 8090:localhost:3000 space-rshiny-prod' // This logs you onto the prod-VM and port 3000 on the prod-VM can be displayed as port 8090 in the local browser on the dev-VM now
- In the web browser (your browser is on the dev-VM) enter as URL: 'http:localhost:8090'
- This should bring you to the Grafana login page on the prod-VM. Ask [SPACE helpdesk](#) for an account



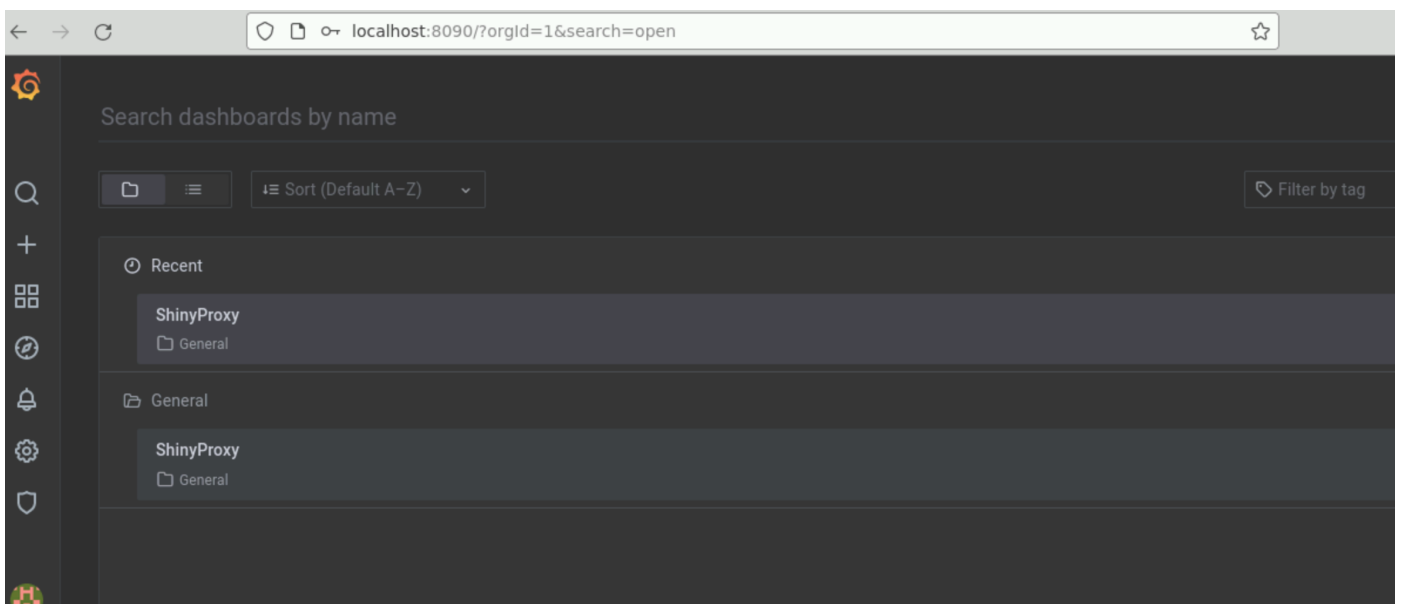
Grafana login

Choose 'Search dashboards' when hovering over the looking glass on the left hand side



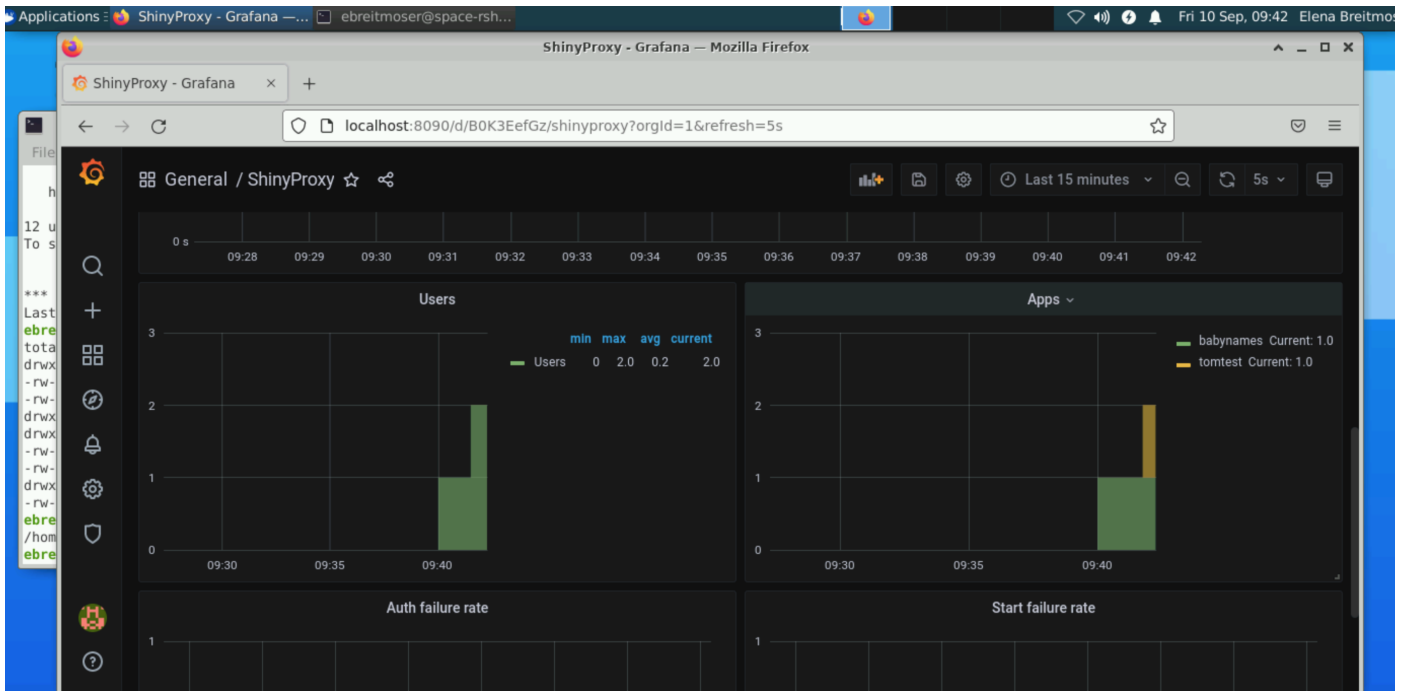
Grafana Dashboard

Choose 'ShinyProxy' (the only dashboard available)



Select shiny Proxy

Now you should see the dashboard



Monitoring Example

4.6 Deployment process for the production service

Warning

This page is a work in progress!

4.6.1 dev-VM

All developers/data analysts only get access to this VM (and not the other two VMs) to build their apps here and create the docker images.

(MG/SPACE) to approve of new dev/analysts and request to get access for them via [SPACE helpdesk](#). Make sure to state clearly that this is for the non-public facing shiny server, `space-rshiny-dev`.

Creating a new account

Users should submit a query to [SPACE helpdesk](#).

4.6.2 To have HPC-systems security scan an app once it is ready for release

Make sure to state clearly that this is for the non-public facing shiny server, `space-rshiny-dev`.

Include the name of the docker image.

Make sure the docker image has been deployed on the dev-VM is in configuration file `application.yml` and displays correctly on localhost

(EPCC/HPC-systems) to scan both the URL and the Dev host once a new app is created. If it passes the scans as expected, EPCC do not need to be involved any further down the chain. All apps moving from the dev-VM to the test- and/or prod-VMs need to be scanned, independently of prior versions of the same app having been on the test or prod-VMs before.

4.6.3 test-VM

Only a managing group (assigned by SPACE and consisting of SPACE staff, abbreviated to 'MG' in the following text) has access.

(MG) have to approve the publication of each app from the SPACE point of view.

(MG) have to be notified by SPACE re any updates or end-of-life to existing apps and arrange for this.

(MG) port the docker images from the dev-VM to the test-VM (using the process Elena used, details to be added to documentation asap).

(MG) check everything looks and works as expected.

(MG) Add data analyst to the configuration file with credentials, so they can see their app deployed on the test-VM `https://eidf.epcc.ed.ac.uk/space-test/login`

(MG) Let data analyst know when everything is ready to be checked.

4.6.4 prod-VM

Only a managing group (assigned by SPACE, see above under 2)) has access.

After (MG) check everything on the test-VM, (MG) migrate the docker image (using the process Elena used, details to be added to documentation asap) to here.

(MG) to approve of new users to get access to the URL (that's not the VM!) for all, or a subset of all apps and to add these users to the config file for shiny ('simple authentication' what we have used so far).

(MG) Server restart required for each new added app - possibly best done outwith working hours, e.g. via cron job.

IT-whitelisting is set the same for all three VMs, but can be adjusted if needed by (EPCC/HPC-systems).

In the event of any technical problems, (MG) to contact EPCC via [SPACe helpdesk](#).

(EPCC) To sort out unexpected or new technical issues.

4.7 Information for the super user

4.7.1 Find out if you have sudo rights

To check if you have sudo user rights on `awb-webapp01` run

Bash

```
sudo -l -U <username>
```

If you don't you get 'User is not allowed to run sudo on awb-webapp01.'

If successful you get the message

Text Only

```
...
User <username> may run the following commands on awb-webapp01:
  (ALL : ALL) ALL
```

4.7.2 Creating new users locally on the VM

Since the move to the "red zone" within interim services, FreeIPA managed accounts no longer work. Therefore, we can create user accounts ourselves with

Text Only

```
sudo adduser <username>
```

and following the prompts, where the username is either (for individual accounts) or - (for accounts belonging to a project, see below for access to project directories).

When new users are needed for the shiny dev, the helpdesks only need to request accounts for the VDI and for that account to be able to connect to the space-rshiny-dev VM. The local user accounts can then be created and passed on by anyone within the support group with sudo rights as needed.

We recommend to add all new (project)users to the table at the bottom of Current User Accounts.

4.7.3 Shared project directories

If data analysts want to collaborate on a project using a shared project directory on space-rshiny-dev.

Text Only

```
sudo mkdir -p /home/shared/<projectname>           //The -p will create the directory and would ignore any errors if the directory already exists.
sudo groupadd <projectname>                        //Create the new group
sudo chgrp -R <projectname> /home/shared/<projectname> //The -R recursive for subdirectories
sudo chmod -R 2770 /home/shared/<projectname>/      //The 2 inheritance for new subfiles
sudo usermod -a -G <projectname> <username>        //The -G a list of supplementary groups which the user is also a member of. Not gid group!
```

4.7.4 Where to find the server

The server 'shinyproxy-2.5.0.jar' is in '/usr/bin/shiny/'.

4.7.5 How and when to (re-)start the server

The server needs to be restarted if any changes to the configuration file 'application.yml' are to take effect.

Check the server status: "sudo systemctl status shinyproxy"

To stop the server: "sudo systemctl stop shinyproxy"

(Re-)start the server: "sudo systemctl start shinyproxy". The server must be stopped before restarting for changes to take effect.

4.7.6 The configuration file

The configuration file 'application.yml' and the server 'shinyproxy-2.3.1.jar' must be in the same directory, and can both be found in '/usr/bin/shiny/'.

The configuration file should not be readable by 'group' or 'other'.

Text Only

```
root@space-rshiny-dev:/usr/bin/shiny# ls -lrt
total 99848
....
-rw----- 1 root      root          2312 Sep 15 14:39 application.yml
```

Otherwise this can be changed by the sudo user with the command "chmod go= application.yml".

The configuration file looks like this:

Text Only

```
proxy:
  title: SPACe Shiny Proxy
  logo-url: /home/ebreitmoser/ediunilogo.png
  landing-page: /
  bind-address:
  heartbeat-rate: 10000
  heartbeat-timeout: 60000
  port: 3939
  authentication: simple
  admin-groups: scientists
  # Example: 'simple' authentication configuration
  users:
    - name: <user1>
      password: <password1>
      groups: scientists
    - name: <user2>
      password: <password2>
      groups: maths
  # Docker configuration
  docker:
    cert-path: /home/none
    url: http://localhost:2375
    port-range-start: 20000
  specs:
    - id: 01_hello
      display-name: Hello Application
      description: Application which demonstrates the basics of a Shiny app
      container-cmd: ["R", "-e", "shinyproxy::run_01_hello()"]
      container-image: openanalytics/shinyproxy-demo
      access-groups: scientists
    - id: 06_tabsets
      container-cmd: ["R", "-e", "shinyproxy::run_06_tabsets()"]
      container-image: openanalytics/shinyproxy-demo
      access-groups: scientists
    - id: euler
      display-name: Euler Application
      description: Euler Application for Elena
      container-cmd: ["R", "-e", "shiny::runApp('/root/shinyproxy-template/euler', host='0.0.0.0', port=3838)"]
      container-image: openanalytics/shinyproxy-template
      access-groups: scientists, maths

logging:
  file:
    shinyproxy.log

server:
  servlet:
    context-path: /eidf.epcc.ed.ac.uk/space-shiny
```

Please do NOT change anything in the application file below 'server:'.

4.7.7 How to edit the configuration file

It's advisable to save a copy of the application.yml before editing, as a state to return to if breaking changes are made.

The file can be edited in the user's choice of editor, the below example is for emacs:

- Run: `sudo emacs -nw application.yml`
- Make necessary changes (paying close attention to indentation)
- To save changes: `Ctrl + x, Ctrl + s`
- To exit emacs editor: `Ctrl + x, Ctrl + c`

4.7.8 How to add new users (aka data analysts)

If you want new users to be able to see and interact with the apps on the website <https://eidf.epcc.ed.ac.uk/space-shiny/> you need to add them to the configuration file.

- Edit 'application.yml' and add the new user below 'users'. Note that indentations are important! We recommend NOT to use the same username and password as the one to access awb-webapp01!

Text Only

```
users:
- name: <userN>
  password: <passwordN>
groups: scientists
```

- For the changes to become effective, the server has to be restarted.
- You might want to add the new user to the docker group as well (see [Howtoaddanewusertothedockergroup](#) of how to do this)

4.7.9 How to add new apps to the VM

Any new app needs to be added to the configuration file 'application.yml' to become visible and useable on the website.

- Edit 'application.yml' and add the new app below 'specs'. Note that indentations are important!

Text Only

```
- id: myappname
  display-name: My new Application
  description: The new Application doing x
  container-cmd: ["R", "-e", "shiny::runApp('/root/<username>/myappname', host='0.0.0.0', port=3838)"]
  container-image: myappname
  access-groups: scientists
```

- For the changes to become effective, the server has to be restarted.

4.7.10 What's different for 'flexdashboard' package/rmarkdown

If you are to deploy an app consisting of rmarkdown and making use of the flexdashboard package.

Edit 'application.yml' and add the new app below 'specs'. Note that indentations are important! The only differences between other apps and rmarkdown ones are in the line for 'container-cmd'

Text Only

```
- id: myappname
  display-name: My new Application using rmarkdown
  description: The new Application doing x with rmarkdown
  container-cmd: ["R", "-e", "rmarkdown::run('/root/<username>/flexdashboard.Rmd', shiny_args = list(port = 3838, host='0.0.0.0'))"]
  container-image: myappname
  access-groups: scientists
```

For the changes to become effective, the server has to be restarted.

4.7.11 How to make apps visible to certain users only

The 'groups' in the configuration file 'application.yml' determines which apps a given user can see and interact with. In our case " belongs to the group 'maths', only the app 'euler' can be accessed by him:

Text Only

```
- name: <user2>
  password: <password2>
  groups: maths
[...]
- id: euler
  display-name: Euler Application
  description: Euler Application for Elena
  container-cmd: ["R", "-e", "shiny::runApp('/root/shinyproxy-template/euler', host='0.0.0.0', port=3838)"]
  container-image: openanalytics/shinyproxy-template
  access-groups: scientists, maths
```

4.7.12 How to change the title of the landing page

Currently the title of the landing page is 'SPACe Shiny Proxy'. This can be changed in the configuration file 'application.yml' by editing 'title: SPACe Shiny Proxy'.

Text Only

```
proxy:
  title: SPACe Shiny Proxy
  logo-url: /home/ebreitmoser/ediunilogo.png
```

4.7.13 How to change the logo of the landing page

Currently the logo of the landing page is set to the University of Edinburgh one. This is can be changed in the configuration file 'application.yml' by editing 'logo-url: /home/ebreitmoser/ediunilogo.png'.

Text Only

```
proxy:
  title: SPACe Shiny Proxy
  logo-url: /home/ebreitmoser/ediunilogo.png
```

4.7.14 How to add a new data analyst to the docker group

In case a new data analyst hasn't been added to the docker group automatically:

Text Only

```
sudo usermod -aG docker <username>
```

For the changes to take effect the user will have to log off and back in.

To check who is in the docker group either

Text Only

```
more /etc/group|grep -in docker
80:docker:x:998:ebreitmoser,sdt,tcrines,mbarnes2,mrake2,twilliamson2
```

or

Text Only

```
members docker
ebreitmoser sdt tcrines mbarnes2 mrake2 twilliamson2
```

4.8 Trouble Shooting

4.8.1 FAQ

When you try to access the website <https://eidf.epcc.ed.ac.uk/space-shiny/> but get: "502 Bad Gateway nginx"

This implies that the shiny server is down. Email [SPACe helpdesk](#) to check and/or restart the server

Shinyproxy server won't start up again.

If changes have been made to the configuration file 'application.yml' it is very likely that the syntax is incorrect. Check the logs 'sudo journalctl -u shinyproxy' for more detailed information.

Running a docker command and getting a permission denied error

Ask the sudo user to have you added to the docker group and log back in again for it to take effect.

Running 'docker build -t myappname .' and receiving the following error message

Text Only

```
W: GPG error: https://cloud.r-project.org/bin/linux/ubuntu bionic-cran40/ InRelease: The following signatures were invalid: EXPKEYSIG 51716619E084DAB9 Michael Rutter <marutter@gmail.com>  
E: The repository 'https://cloud.r-project.org/bin/linux/ubuntu bionic-cran40/ InRelease' is not signed.
```

`RUN apt-key adv --keyserver keyserver.ubuntu.com --recv-keys E298A3A825C0D65DFD57CBB651716619E084DAB9` Is missing in the Dockerfile

Running `docker build -t myappname .` and getting a "Error writing to output file - write (28:No space left on device)" error

Contact the sudo user describing the error message

Sudo user to check if disk space is full by running the command 'docker system df' (see Docker for more commands)

Sudo user to get in touch with EPCC helpdesk

Running `docker build -t myappname .` and getting a "E: Package 'libssl1.0.0' has no installation candidate" error (this could apply to any other package in the future, the error is only demonstrated for the case of libssl here)

The libssl package has been upgraded to libssl1.1 in Ubuntu 19.04 (and later) (see <https://askubuntu.com/questions/1173587/e-package-libssl1-0-0-has-no-installation-candidate>). Update the package in the Dockerfile accordingly.

Trying to access the app on the server after the docker image seemed to have built fine, the app was added to the configuration file 'application.yml', the server was restarted and the app title shows on the landing page, then after clicking on it this error occurs

(see also <https://www.shinyproxy.io/faq/>) or

Text Only

```
504 gateway timeout nginx
```

It is possible that the start-up time for the app needs longer than the limit set in the configuration file 'application.yml' ('heartbeat-timeout' and/or 'container-wait-time' in our example at Theconfigurationfile). These values can be increased and the server needs to be restarted to take effect. Often this is not the actual problem, though.

Probably there was an error during the build phase of the docker image, even though the docker image was created. In this case you would have to have to take a look at the probably very lengthy output from 'docker build -t myappname .' And there will be messages like:

Text Only

```
configure: error: gdal-config not found or not executable.

ERROR: configuration failed for package 'rgdal'
removing '/usr/local/lib/R/site-library/rgdal'

The downloaded source packages are in
'/tmp/RtmpDodZJN/downloaded_packages'

Warning message:
In install.packages("rgdal", repos = "https://cloud.r-project.org/")
```

In this particular example the `rgdal` package was not installed as needed.

Another and probably faster way to obtain more insight into the problem is to run the app from the command line with the following command:

```
sudo docker run -p 3838:3838 myappname R -e "shiny::runApp('/root/ebreitmoser/myappname',host='0.0.0.0',port=3838)"
```

If this produces the error message that something else is already listening at port 3838 contact the EPCC helpdesk.

Continuing with our example the output would look like so:

Text Only

```
The following object is masked from 'package:ggplot2':
last_plot

The following object is masked from 'package:stats':
filter

The following object is masked from 'package:graphics':
layout

Error in library(rgdal) : there is no package called 'rgdal'
Calls: <Anonymous> ... sourceUTF8 -> eval -> eval -> ..stacktraceon.. -> library
Execution halted
```

Which confirms the issue with the `rgdal` package. Whereas a successful launch would give (note the last line in particular:

Text Only

```
The following object is masked from 'package:sp':
show

The following object is masked from 'package:shiny':
runExample

The following objects are masked from 'package:methods':
removeClass, show

Attaching package: 'plyr'

The following objects are masked from 'package:plotly':
arrange, mutate, rename, summarise

Loading required package: XML
Loading required package: RCurl

Attaching package: 'RCurl'

The following object is masked from 'package:shinyjs':
reset

Warning in OGRSpatialRef(dsn, layer, morphFromESRI = morphFromESRI, dumpSRS = dumpSRS, :
  Discarded datum OSGB_1936 in Proj4 definition: +proj=tmerc +lat_0=49 +lon_0=-2 +k=0.9996012717 +x_0=400000 +y_0=-100000 +ellps=airy +units=m +no_defs

Listening on http://0.0.0.0:3838
```

Then you can still open `localhost:3838` in your local web browser, which should display the app if it works correctly or show new error messages on the command line.

Googling the error message shows that four more packages are required for the `rgdal` package (e.g. see <https://gist.github.com/dncgst/111b74066eaea87c92cdc5211949cd1e>). They have to be added to the Dockerfile

Text Only

```
# system libraries of general use
RUN apt-get update && apt-get install -y \
  sudo \
  pandoc \
  pandoc-citeproc \
  libcurl4-gnutls-dev \
  libcairo2-dev \
  libxt-dev \
  libssl-dev \
  libssh2-1-dev \
  libssl1.1 \
  gdal-bin \
  proj-bin \
  libgdal-dev \
  libproj-dev
```

Docker port is already allocated

This says that port `0.0.0.0:3838` is already in use. Check if the port is used by a currently running docker container:

Text Only

```
docker container ls
CONTAINER ID   IMAGE                                COMMAND                                CREATED        STATUS        PORTS
NAME
64d65e213b18  accessibility-statement-shiny-server "R -e 'markdown::ru..." 4 minutes ago  Up 4 minutes  0.0.0.0:20002->3838/tcp
hopeful_driscoll
1e0f8d4bd53f  stm_industry                        "R -e 'markdown::ru..." 5 minutes ago  Up 5 minutes  0.0.0.0:20001->3838/tcp
keen_chaplygin
d1644205de1c  stm_industry                        "R -e 'markdown::ru..." 49 minutes ago  Up 49 minutes  0.0.0.0:20000->3838/tcp
jovial_goldwasser
4f54dd33c945  stm_process                         "R -e 'markdown::ru..." 2 hours ago    Up 2 hours    0.0.0.0:3838->3838/tcp, :::3838->3838/tcp
practical_goldstine
```

To free the port:

Text Only

```
docker rm -f <container-ID> // In general
docker rm -f 4f54dd33c945 // For example above
```

4.9 Docker

Some additional docker commands, which might be useful. For more detailed information search the docker website <https://docs.docker.com/>.

Which version of docker is being used?

Bash

```
[ebreitmoser@awb-webapp01 how_toy]$ docker -v
Docker version 1.13.1, build 0be3e21/1.13.1
```

Remove all stopped containers (beware that will stop all images for all users belonging to the docker group!):

Text Only

```
docker container prune
```

See [here](#) for further information.

Otherwise beware that they can take up a lot of disk space. To check the total disk space:

Bash

```
df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            7.8G   0 7.8G   0% /dev
tmpfs           1.6G  1.9M  1.6G   1% /run
/dev/mapper/ubuntu--vg-ubuntu--lv 29G   20G  7.3G  74% /
...
```

So in this case we have a total of 29GB available of which 74% are already used.

Display information regarding the amount of disk space used by the docker daemon (see [here](#)).

Bash

```
[ebreitmoser@awb-webapp01 how_toy]$ docker system df
TYPE                TOTAL          ACTIVE          SIZE           RECLAIMABLE
Images              20             0              6.42 GB       6.42 GB (100%)
Containers          0             0              0 B           0 B
Local Volumes       0             0              0 B           0 B
```

The RECLAIMABLE amount of disk space is the total size consumed by unused images. To remove

- all stopped containers
- all volumes not used by at least one container
- all networks not used by at least one container
- all dangling images

Run (that's less than all unused images! But you do NOT want to delete all images)

Bash

```
docker system prune
```

Remove one or more docker images.

First list them all, then delete the ones you want to by using their Image ID (see <https://docs.docker.com/engine/reference/commandline/rmi/>):

Bash

```
[ebreitmoser@awb-webapp01 how_toy]$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
how_toy             latest             88aa985b958c      35 minutes ago    1.02 GB
<none>             <none>            44e1665e9b8f      13 hours ago      1.02 GB
<none>             <none>            3863d1b46f5d      14 hours ago      580 MB
```

then

Bash

```
[ebreitmoser@awb-webapp01 how_toy]$ docker image rmi -f 44e1665e9b8f 3863d1b46f5d
```

You see that the docker image typically take ~1GB of space.

Runtime metrics

See <https://docs.docker.com/config/containers/runmetrics/>. Look which containers are currently running. Then choose one and see how many %CPUs, memory etc it is currently using on the system.

Bash

```
docker container ls
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
d1644205de1c  stm_industry  "R -e 'markdown::ru..." 18 minutes ago Up 18 minutes 0.0.0.0:20000->3838/tcp  jovial_goldwasser

docker stats d1644205de1c

CONTAINER ID   NAME          CPU %     MEM USAGE / LIMIT   MEM %     NET I/O       BLOCK I/O   PIDS
d1644205de1c  jovial_goldwasser  2.86%    221.9MiB / 15.63GiB  1.39%     353kB / 986kB  0B / 1.06MB  4
```

Warning

Work in progress/needs updating

4.10 Security

For the SPACe-internal, password-protected website

4.10.1 Webpage

What users can access on the webpage is only rshiny (on port 3939). It sits behind a reverse proxy which displays only what we have allowed to be sent via this port. So nothing else is accessible. The only things that are displayed is what we allow via the apps and nothing else that sits on the VM. In other words, no shiny app user can actually access the data, which are on the VM.

In addition, the website is locked down so you require username and password to access it. Along with the page itself being locked down to specified IP addresses.

4.10.2 Shiny Proxy Server

For the shinyproxy server see <https://www.shinyproxy.io/documentation/security/> (To be checked by Elena as of 9th March 2021).

Secure the underlying infrastructure

- "never bind the docker daemon API on 0.0.0.0, only on the loopback interface (127.0.0.1)":

Created /etc/docker/daemon.json

Text Only

```
{
  "debug": true,
  "hosts": ["tcp://127.0.0.1:2375", "unix:///var/run/docker.sock"]
}
> sudo systemctl restart docker.service

> netstat -tunlp|grep -in listen
'tcp 0 0 127.0.0.1:2375 0.0.0.0:* LISTEN - '
```

Secure the communication between ShinyProxy and the LDAP directory

Secure the communication between ShinyProxy and the Docker daemon

Reverse proxy

reverse nginx proxy

Setting the bind-address in the application.yml