

The Knowledge

Geographic Data Science Lab

2020-05-27

Contents

1	The Knowledge	5
2	Remote Working	7
3	VPN	9
3.1	<i>What is a VPN?</i>	9
3.2	<i>How can I setup the VPN for Liverpool?</i>	9
3.3	Set up a VPN connection	9
3.4	How to use the GlobalProtect VPN Client to access the University network	10
4	SSH Unix Commands	13
5	Files	15
5.1	Backup/sync	15
5.2	Encryption	15
5.3	File Transfer Protocol (FTP)	16
6	Data Science Stack	17
7	Docker	19
7.1	Installation	19
7.2	Useful Docker Commands	20
8	JupyterLab	21
8.1	Local install	21
8.2	Remote install	22
8.3	Useful Python Docker Images	23
9	RStudio server	25
9.1	Local install	25
9.2	Remote install	26
9.3	Useful R Docker Images	26

Chapter 1

The Knowledge

In preparation to be able to stay productive while having to work fully remote, this document presents a few things jotted down in one place to help with setups, etc.

The first step is a checklist everyone should go through:

1. Do you have a computer to work on at home?
2. Are *all* your relevant files accessible from home?
3. Do you have a webcam, mic and speakers/headphones?
4. Are you able to connect to computers on campus? This involves being setup with the University VPN?

These are the basic elements you will require, so if the answer to any of the above is no, please speak with your PhD supervisor.

Chapter 2

Remote Working

Chapter 3

VPN

3.1 *What is a VPN?*

A VPN (virtual private network) connects a machine that lies outside of the university (ie. outside the firewall) to the internal network. When the VPN is running, your network traffic (e.g. Internet) is routed through the university in the same way as if the computer was on your work desk. This enables you to:

- Access journal websites like you would inside the university
- Access network drives (e.g. M Drive etc) - but be careful when transferring big files
- Access servers (e.g. over the terminal / command line / ftp)

3.2 *How can I setup the VPN for Liverpool?*

You can find a more comprehensive guide to setup your VPN on the `setup_vpn.md` document of this repository.

3.3 Set up a VPN connection

This document describes how to set up a VPN. The text below has been copied from CSD website.

To access the VPN service:

1. Register

Submit a request to register for the VPN service via CSD. You will need to explain why you require VPN access and what you intend to do with it. It may be that other services offer the solution you require instead of the VPN.

2. Download the VPN Client

Once your registration is confirmed you will need to download and install the GlobalProtect VPN Client if you are using Windows 10.

You will need admin rights in order to download the GlobalProtect VPN Client. This is how to access admin rights on your MWS PC. Please access admin rights before attempting to download the VPN Client.

The GlobalProtect VPN Client is also compatible with MacOS. Mobile and tablet devices cannot connect using this client. If you are using a Linux distribution, such as Ubuntu, you may be able to connect to the University network using VPNc.

3. Open the client and connect

Click here for instructions on how to use the GlobalProtect VPN Client to access the University network from your PC or Mac.

3.4 How to use the GlobalProtect VPN Client to access the University network

You must make sure you are registered to access the VPN service first. Once you are registered you can download and install the GlobalProtect VPN Client to connect to the University network.

GlobalProtect is compatible with Windows 10 and Macs. It is not possible to use GlobalProtect to connect to the University network on a mobile or tablet device.

To install the Client

Access Admin Rights on you PC.

Visit <https://vpn.liv.ac.uk> Enter your University username and password to login to the VPN portal.

Click the appropriate link to download the required version of the VPN client - Windows 32 bit, Windows 64 bit, or Mac OS. (To check which version you require, see your system properties on your device)

Once the file has downloaded, double-click to run the installation. Follow the steps through the installation wizard, accepting the default options.

Once installed you will see the GlobalProtect “globe” icon appear in the system tray (bottom right, near the clock). It is a globe and it will have a red x on it, showing that it is not currently connected.

Double-click on the GlobalProtectglobe icon in the system tray. In the window that opens, enter the following:

3.4. HOW TO USE THE GLOBALPROTECT VPN CLIENT TO ACCESS THE UNIVERSITY NETWORK11

Username: enter your University username Password: enter your University password Portal: vpn.liv.ac.uk

Click Apply.

The GlobalProtect VPN client will then automatically connect to the University network - the red cross should disappear from the icon in the system tray.

You can close the window: the client will stay connected. To connect and disconnect the client

Once the GlobalConnect VPN client has been installed, the icon will remain in your system tray

To connect the VPN right click the GlobalProtect icon in the system tray and choose Connect. When you have finished and want to disconnect the VPN, right click on the icon and choose Disconnect.

How to allow third party applications - like Global Protect - to install on a Mac

1. Open System Preferences and click Security & Privacy
2. Select the General tab
3. Click the lock in the lower left-hand corner
4. Enter your computer username and password, then select Unlock

Chapter 4

SSH Unix Commands

Chapter 5

Files

You need to make sure that:

- a) You have access to all of your files
- b) Your files are backed up so your setup is not entirely reliant on a single device
- c) Each device on which your files are copied or from which they are accessed is encrypted

5.1 Backup/sync

The simplest and recommended way to do this at Liverpool is to keep all your files and data on your university account at OneDrive. This is part of the Office 365 Suite available from the university, you can find more info at:

<https://www.liverpool.ac.uk/csd/working-from-home/>

There are Windows and Mac clients that work relatively well (equivalent to Dropbox client).

Once you are set up, copy all your files onto your OneDrive account, which will create a copy of them in Microsoft's secure cloud. The exception is where you have data that has requirements to be managed in particular ways - e.g. only from a single machine etc; not in the cloud.

Please, be sure to speak with your PhD supervisor if you access data that may pose some challenges when moving from local machines or within the university network (remember OneDrive is in the Cloud, not the university servers!).

5.2 Encryption

[Add guide to encryption: what it is and how to set up on Win-

dows/macOS/Linux/iOS/Android]

5.3 File Transfer Protocol (FTP)

If you need to move large and/or many files from a local machine to a remote server (e.g. from your laptop to a Linux machine at the lab), you will probably want to use something like FTP.

[Add FTP guide here]

Chapter 6

Data Science Stack

Once you have access from home to all your files and (remote) university computers, next step is easily being able to bootstrap a full data science stack that allows you to carry out scientific work. There are several ways of achieving this, but our preferred strategy is to rely on container technology, in particular on Docker. This will allow you to rapidly install the platform and set of libraries you are familiar with in a way that can be easily reproduced and redeployed (e.g. on a remote computer on campus).

Here are a series of pages that will help you get a stack ready to go:

- `setup_docker.md`: instructions to install and get Docker up and running on different platforms
- `setup_jupyterlab.md`: instructions to install and run a JupyterLab server both on local (e.g. laptop) and remote (e.g. server) machines
- `setup_rstudio.md`: instructions to install and run a JupyterLab server both on local (e.g. laptop) and remote (e.g. server) machines

Chapter 7

Docker

This document describes how to install and use Docker on different platforms.

7.1 Installation

If you are on Mac, Linux or Windows 10 Pro/Student editions, installing Docker is relatively straightforward:

- Mac
- Linux official instructions
- Windows 10 Pro/Student

It is important to note that, on Mac and Windows, Docker runs under a virtual machine so it will not use up all of the resources of your machine (conversely, it'll equate to be working on a more limited machine). This can be changed. But if you need more firepower, the idea is that you develop on your laptop and scale to a server (e.g. running out of the lab).

The steps to install Docker include:

- Obtain a copy of Docker and install it:
 - **Windows10 Pro/Enterprise:** Install Docker Desktop for Windows
 - **macOS:** Get started with Docker Desktop for Mac
- Once Docker is successfully installed, make sure to enable access to your main drive (e.g. C:\):\ul>- **Windows10 Pro/Enterprise:** Open the preferences for Docker and click the “Shared Drives” tab; click on the drive you want to add and then “Apply”
- **macOS:** this feature is automatically enabled

7.2 Useful Docker Commands

See what containers are running (this also shows you the ID - this is useful to know then R-studio crashes...)

```
docker ps
```

Stop and remove a particular container - replace ID; with the specific ID from the above (listed under CONTAINER ID) - if everything has crashed

```
docker stop ID  
docker rm -f ID
```

Chapter 8

JupyterLab

This document shows how to install and run a JupyterLab server locally and remotely.

8.1 Local install

This guide assumes you meet the following requirements in your personal machine (eg. laptop):

1. You have admin rights over your machine
2. You are running either Windows 10 Pro, macOS, or Linux

Assuming Docker is up and running (check `setup_docker.md` for that), you can install an “image”, which is the install that will let you run containers, by typing on a command line (`Terminal.app` or `PowerShell` are both good options):

```
docker pull darribas/gds:4.0
```

This will take a while to download but, once finished, you will be ready to go.

Once the command above has finished installing your GDS stack, you are ready to go! To get a Jupyter session started, you can follow these steps:

1. Run on the same terminal as above the following command:

```
docker run --rm -ti -p 8888:8888 -v ${PWD}:/home/jovyan/work darribas/gds:4.0
```

The command above spins up a container of the `gds` image, version 4.0 and ensures it is connected through two main bridges:

- Mapping your laptop’s file system from where you have launched the command (`${PWD}`) to a folder called `work` on the home directory of the container. When you login to Jupyter (see below), you will see a `work` folder

and, if you click into it, you should see the content of your laptops folder in there.

- Mapping port 8888 from the container to your laptop, so you can connect to it through a browser.

It is important to know this command starts a Jupyter server on your machine and keeps it running, so please do not quit the window until you are done using Jupyter, otherwise it will crash.

2. Open your favorite browser (preferably Firefox or Chrome) and point it to `localhost:8888`
3. You will be asked for a password or a token. To find the correct one, check the terminal where you started the `docker run ...` command in 1) and look for the long token in the logs. Your prompt should look something (albeit not exactly) like this:

```
docker run --rm -ti -p 8888:8888 -v ${PWD}:/home/jovyan/work darribas/gds:4.0
Executing the command: jupyter notebook
[I 11:38:40.234 NotebookApp] Writing notebook server cookie secret to /home/jovyan
[I 11:38:41.328 NotebookApp] Loading IPython parallel extension
[I 11:38:41.612 NotebookApp] JupyterLab extension loaded from /opt/conda/lib/python3.7/site-packages/jupyterlab
[I 11:38:41.612 NotebookApp] JupyterLab application directory is /opt/conda/share/jupyter/lab
[I 11:38:43.091 NotebookApp] Serving notebooks from local directory: /home/jovyan
[I 11:38:43.091 NotebookApp] The Jupyter Notebook is running at:
[I 11:38:43.091 NotebookApp] http://ee20e7549b49:8888/?token=4dc814ee44c64383d5d32dfd439fe62bbc17d9803d9a
[I 11:38:43.091 NotebookApp] or http://127.0.0.1:8888/?token=4dc814ee44c64383d5d32dfd439fe62bbc17d9803d9a
[I 11:38:43.091 NotebookApp] Use Control-C to stop this server and shut down all
[C 11:38:43.114 NotebookApp]
```

To access the notebook, open this file in a browser:

`file:///home/jovyan/.local/share/jupyter/runtime/nbserver-6-open.html`

Or copy and paste one of these URLs:

`http://ee20e7549b49:8888/?token=4dc814ee44c64383d5d32dfd439fe62bbc17d9803d9a`

or `http://127.0.0.1:8888/?token=4dc814ee44c64383d5d32dfd439fe62bbc17d9803d9a`

The token you want to copy is the long series of letter and numbers right after `?token=`, starting by `4dc814ee`.

4. The token should let you into your Jupyter Lab session. Congratulations! You can then access the files in your computer through the `work` directory on the left-side pane.

8.2 Remote install

It is also possible to start a Jupyter server as above but, instead of run it on your local machine, it can run on a remote machine and you connect to that through your browser over the internet. The process in this context is a bit

more intricate because you need to ensure that the connection is secure, but overall it follows a similar pattern. The following steps below assume you can login to the remote server where you want to run Jupyter through `ssh` and the server already has a Docker image installed, ready to be run.

- Login to the remote machine:

```
ssh <username>@<server.ip.address>
```

1. Launch the container:

```
docker run --rm -ti -p 8888:8888 -v ${PWD}:/home/jovyan/work darribas/gds:4.0 start.sh
```

Note we are appending `start.sh` so it drops us into the command line of the container rather than launching the server directly

2. Run `jupyter notebook --generate-config`

3. Generate SSH keys with: `openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout mykey.key -out mycert.pem`

4. Generate password as in the official tutorial

5. Update `/home/jovyan/.jupyter/jupyter_notebook_config.py`

```
“python # Set options for certfile, ip, password, and toggle off # browser
auto-opening c.NotebookApp.certfile = u'/home/jovyan/mycert.pem'
c.NotebookApp.keyfile = u'/home/jovyan/mykey.key' # Set ip to ' to
bind on all interfaces (ips) for the public server c.NotebookApp.ip = "
c.NotebookApp.password = u'sha1:bcd259ccf...' c.NotebookApp.open_browser
= False

# It is a good idea to set a known, fixed port for server access
c.NotebookApp.port = 8888 “
```

6. Launch secure Lab: `jupyter lab`

7. On your own machine (laptop/tablet), log in to `<server.ip.address>:8888` with the password you have set

8.3 Useful Python Docker Images

- `gds_env`: a containerised platform for Geographid Data Science in Jupyter (Python & R)
- `jupyter-stacks`: official Jupyter stacks (the `gds_env` is based on these)

Chapter 9

RStudio server

This guide will help you set up a RStudio server running with Docker. The benefits of this approach is that it is more reliable and only involves one install, as opposed to several independent ones.

9.1 Local install

If you are an R user; this is a great image to get you up and running with the tidyverse + various geospatial packages + Rstudio server:

```
https://hub.docker.com/r/rocker/geospatial
```

To install...

```
docker pull rocker/geospatial
```

This will require a good internet connection and will take a while, but you only need to run it once.

Once ready, you can start the instance as follows (Mac example):

```
docker run -d --name rstudio -v $HOME:/home/rstudio/alex -e PASSWORD=secret -p 8787:8787 rocker/g
```

This maps your local home drive (\$HOME) to a given directory in the container (in this case - /home/rstudio/alex). Using this approach, you can store files on your laptop's drive, and access, edit or create new ones from the container (ie. using RStudio). The command above also sets up a password (**secret**) and username (**rstudio**) to use when you login to RStudio.

On Windows, this is similar (change alexa to your account name - look at your directory structure):

```
docker run -d --name rstudio -v c:\users\alexa:/home/rstudio/alex -e PASSWORD=secret -p 8787:8787
```

If this runs ok; you access Rstudio server through a browser at: <http://localhost:8787>

9.2 Remote install

[Add here specifics to make this work across the wire in a secure way]

9.3 Useful R Docker Images