

Guide to Jupyter Notebooks on Viper



For students of the
Centre of Excellence for Data Science,
Artificial Intelligence and Modelling

Guide to Jupyter Notebooks on Viper

Step by step overview

This guide covers an approach designed to simplify the creation of a Jupyter enabled Python environment and the steps needed to launch Jupyter Notebook sessions on Viper.

The following are the basic steps required to run Jupyter Notebooks on Viper. Some are required to be carried out before connecting to Viper, some you would normally expect to carry out just once, and others you step through every time you want to run a Jupyter Notebook on Viper.

These steps are expanded on in more detail in the rest of this document and are included here as a simple overview and reference once the rest of the guide has been reviewed.

Before connecting to Viper

1. (if required) Apply for VPN access using the form on the support portal at [Request access to the GlobalProtect Client VPN Service](#) remembering to reference the use of Viper in the justification field.
2. Apply for a Viper account at <https://hpc.wordpress.hull.ac.uk/apply-for-an-account/>

First time (run once)

3. Connect to Viper through SSH either using [MobaXterm \(Windows\)](#) or the terminal (macOS or Linux)
4. In the Linux terminal on the login node type **ViperSetupJupyter** and press return, which will create a Conda Python Virtual Environment tagged with the current date (i.e. 2022-07-22). See [Creating a Jupyter enabled Conda Virtual Environment \(step 4\)](#)

All other times (assuming connected to Viper through SSH)

5. Start a Jupyter Notebook by typing your equivalent of ONE of the following at the Viper login node terminal, depending on the resource you need:

./StartJupyter-cpu-2022-07-22

./StartJupyter-gpu-2022-07-22

./StartJupyter-highmem-2022-07-22

(note the date at the end will be different) and pressing return. This tries to start a job on Viper to run your Jupyter Notebook session which will run for 24 hours. See [Starting a Jupyter Notebook \(step 5\)](#)

6. Assuming there is resource available, after a minute or two you should receive an email which will give details about how to set up your SSH tunnel. This SSH tunnel allows you to connect the web browser on your local computer with the Jupyter Notebook running on Viper. This is done either using MobaXterm tunnelling tool (Windows) or via the terminal (macOS and Linux). See [Setting up a tunnel \(step 6\)](#)
7. With the tunnel set up, click on the URL in the email which should open up the Jupyter Notebook session
8. Refer to the [Using Jupyter \(step 8\)](#) section for selecting the appropriate Python Kernel and information about migrating notebooks from other platforms
9. When finished working, cancel your Jupyter session by running the **scancel** command with the appropriate job ID as provided by the connection email

Contents

Step by step overview.....	2
Before connecting to Viper.....	2
First time (run once).....	2
All other times (assuming connected to Viper through SSH).....	2
Document Guide.....	3
Viper overview.....	4
Important notes about using Viper.....	4
VPN.....	4
Data.....	4
Shared resource.....	4
Creating a Jupyter enabled Conda Virtual Environment.....	5
Instructions.....	7
Starting a Jupyter Notebook.....	8
Setting up a tunnel.....	10
Connecting to Jupyter.....	11
Using Jupyter.....	12
Migrating from other platforms.....	12
Python Kernel.....	14
Installing Packages.....	14
Stopping a Jupyter Notebook.....	14
More Information.....	15
Getting help.....	15

Document Guide

This document includes screenshots as well as selections of text from various sources, which appear as follows:

```
[username@login01 ~]$ Text in this format shows what you can expect to see at the Linux terminal, i.e. Viper
[username@login01 ~]$ While text in red indicates what you should type
```

```
In []: This is text that should be typed into Jupyter cells
```

Commands to be run and shown as **CommandName** and should be typed into the Linux terminal followed by the return key

(CC 2022-09-20 v1.2)

Viper overview

Viper, the University of Hull's High Performance Computing facility, is used by research staff and students in many disciplines across the University and is a potentially significant tool for those with any sort of computational element to their research.

Viper is a 'cluster' of approximately 200 computers built into a cluster, and features more than 6000 compute cores, high memory systems and GPU technology together with dedicated high performance storage and fast interconnect between systems to meet the needs of the most computationally demanding tasks.

Some notes about

- Viper runs Linux, though to make use of Viper you do not need to be a Linux expert – being familiar with just a few commands is enough to get started.
- Viper runs a scheduler, a piece of software which manages access to the computers across the cluster, monitoring what is running and how resources such as memory, cpu processors and GPU cards are being used.
- Users run 'jobs' some of which are interactive (like Jupyter Notebook sessions) but many of which are submitted to run automatically without user interaction. Jobs usually have a batch script, a text file containing recipe for the job, for example what resource is needed and what the job should do.

At the end of this guide are links to [More Information](#) and also [Getting help](#)

Important notes about using Viper

VPN

To apply for an account or to connect to Viper you either need to be on the University network or connected via VPN. There is a form on the University support portal to request access to the University VPN service. Fill out the form at [Request access to the GlobalProtect Client VPN Service](#) and include reference to the use of Viper in the justification field.

Data

Storage on Viper is separate from your normal University storage. It is high performance storage which is available on all compute nodes across the Viper infrastructure. The storage is not suitable for storing sensitive data (e.g. personal identifiable data) and shouldn't be used as such. It should also be treated as scratch storage, that is it is not regularly backed up and you are responsible for making your own data backups

Shared resource

Viper is a shared resource, used by many research staff and students from across the University. There may be times when all resource is in use in which case you will be entered into a queue to wait until resource becomes available. We encourage Viper users to consider others when making use of Viper and only using resource as appropriate.

Connecting to Viper (step 3)

For Windows users, we advise downloading [MobaXterm](#) which is available free. It provides the key functionality required to run on Viper, in particular an SSH connection (to provide the command prompt), a built-in file explorer, and a mechanism for creating SSH tunnels (see step 6).

Setting up a connection

Follow the above link, or Google MobaXterm and go to the MobaXterm website. Click on the “Download now” button on the “Home Edition”, then on the following page select either Portable edition (blue link) or Installer edition (green link). The installer edition requires administrator rights, so if you are using a University PC where you don’t have admin rights, select the portable edition. Once running, there is very little difference between the two versions.

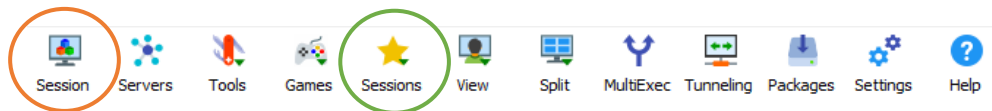


Figure 1 - MobaXterm menu icons – Session icon

To setup a connection:

1. After opening MobaXterm, click on the Session icon near the top left of the window (see figure 1).
2. In the new windows that opens, click the icon labelled SSH near the top left, then set the following:
 - a. In the remote host box enter: viper.hull.ac.uk
 - b. Check the box next to Specify Username
 - c. Replace <default> in the username box with your 6 digit university user ID
3. With the above set, click OK

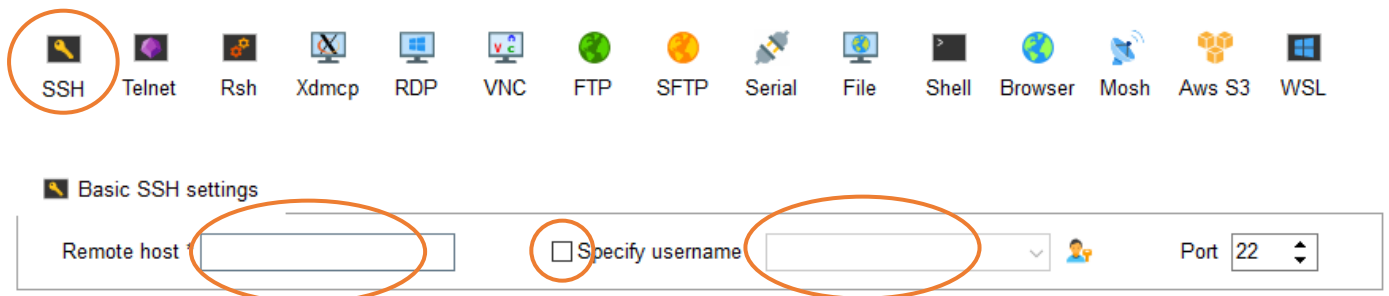


Figure 2 - MobaXterm menu icons – Session Setup

Notes on connecting:

- You need to connect with your unique 6 digit university user ID, e.g. 465449
- MobaXterm will prompt if you want to save your password – it is suggested you click **no**
- The first time you connect you may see a window stating the server’s host key is unknown and asking if you trust the key. In this instance it is ok to click **yes**
- When you type in your password, nothing is shown on the screen. This is a security feature and is by design. Just type your password and press return
- You will probably see an error message about a file .Xauthority not existing, this is fine and can be ignored. This file is created after first login and the error message won’t be shown again.

- After the first time you set up the session as above, the next time you need start MobaXterm you can find the session saved by clicking on the sessions icon (yellow star highlighted in green in figure 1) and then clicking on the session labelled viper.hull.ac.uk

The MobaXterm Window

Once connected to Viper the MobaXterm window will become two panels as shown in figure 3. On the right-hand side is the black panel of the main console or terminal, showing the Linux command prompt where you type the commands to run. On the left-hand side is a file explorer panel, providing a simple tool to allow you to transfer files to and from Viper.

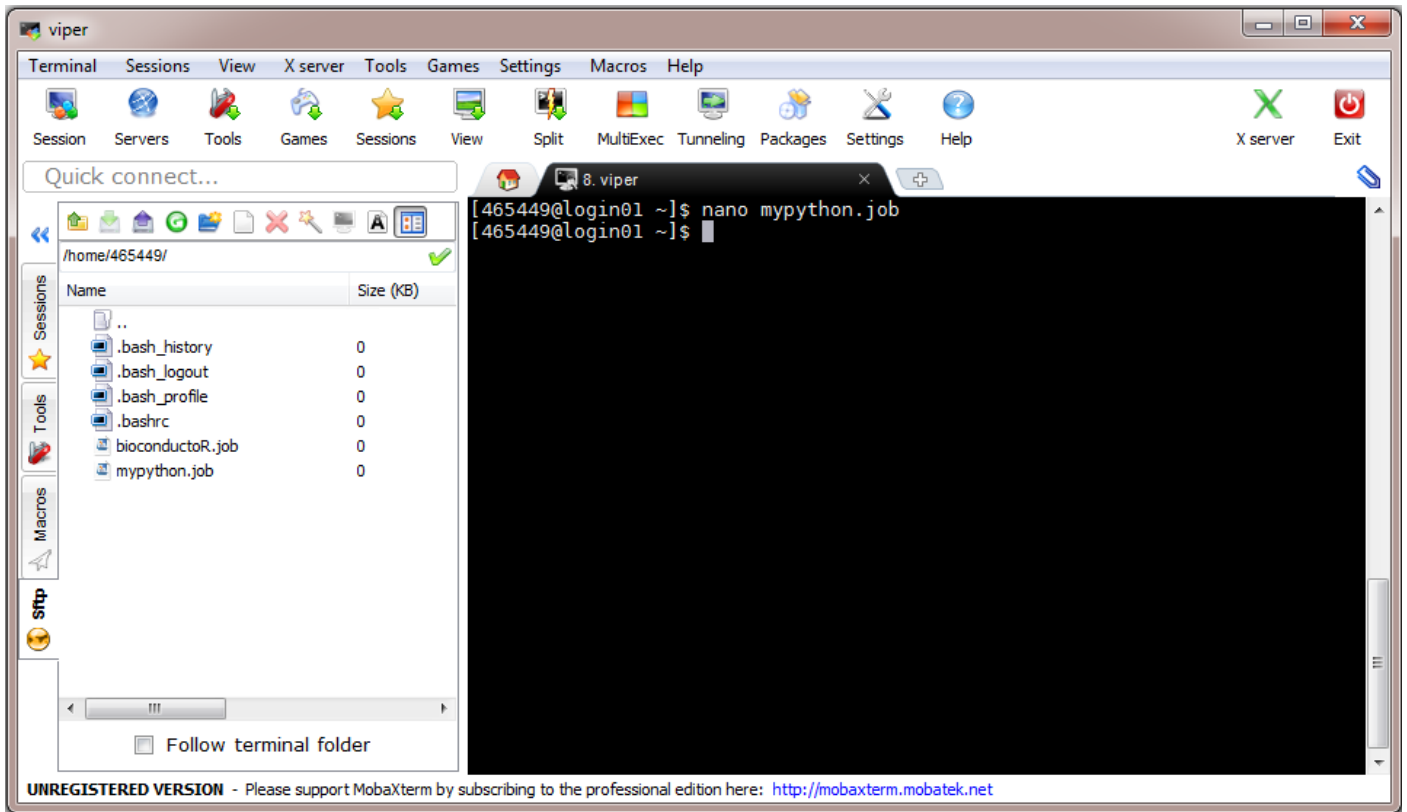


Figure 3 - MobaXterm Window

To transfer large amounts of data (either file count or file size) it is recommended to use a tool such as FileZilla, another tool that can be downloaded for free. There is more information on FileZilla available on the Viper wiki.

Creating a Jupyter enabled Conda Virtual Environment (step 4)

A Python virtual environment allows you to create a custom environment in which they can have the packages, and versions of those packages that are required. Virtual environments make it easy to cleanly separate different projects and avoid problems with different dependencies and version requirements across components.

It is possible to run Jupyter notebooks on Viper, however there are a number of steps involved in doing so. A simplified way of creating a Jupyter enabled virtual environment is available using the command **ViperSetupJupyter** on the Viper login node.

The virtual environment which will be based on Python 3.10 and which has Jupyter enabled also includes the following commonly used packages (you can install additional packages can be installed):

- matplotlib
- pandas
- numpy
- scipy
- tensorflow

As well as creating a Conda virtual environment, the **ViperSetupJupyter** command also sets up the ability to start Jupyter Notebooks. This includes commands to submit jobs that start a 24-hour Jupyter Notebook running on either a standard compute node, high memory node or GPU node.

Instructions

From the Viper login node command prompt type **ViperSetupJupyter** and press **return**. You will then be asked to confirm by pressing **y** after which a new Conda environment will be created with the suffix matching the date the command was run.

```
[username@login01 ~]$ ViperSetupJupyter
This script will create a Jupyter enabled Python Virtual Environment and supporting files
The Jupyter environment and scripts will all be tagged 2022-07-22
Continue [Y or y] or Quit [any other key] y
Continuing
Preparing to generate Jupyter notebook - this may take around 15 minutes
Collecting package metadata (current_repodata.json): done
```

...

```
Generated Conda environment

Using Jupyter Virtual Environment jupyter-2022-07-22

To start a Jupyter notebook that uses just CPU, type (followed by enter/return):
./StartJupyter-cpu-2022-07-22

To start a Jupyter notebook that requires GPU, type (followed by enter/return):
./StartJupyter-gpu-2022-07-22

To start a Jupyter notebook that requires large amounts of memory (over 128GB up to 1TB), type (followed by enter/return):
./StartJupyter-highmem-2022-07-22

If resource is available, you should receive an email after a few minutes with connection details

Now you have run this command once and everything is setup, in future you only need to run the ./StartJupyter-cpu-2022-07-22 type commands to start a Jupyter notebook, you do not need to re-run ViperSetupJupyter again unless you need to restart from scratch

More information can can be found in README-Jupyter-2022-07-22.txt

If you have issues, please raise a ticket on the support portal at https://support.hull.ac.uk and search for Viper
```

Figure 4 - Creating your Jupyter enabled virtual environment

Starting a Jupyter Notebook (step 5)

As well as creating the Conda environment the ViperSetupJupyter command also creates a number of additional files, including specific commands to start Jupyter sessions. These are:

To start a session on standard compute nodes (for jobs that require up to 28 cores and 128GB of RAM) type (followed by return)

`./StartJupyter-cpu-YYYY-MM-DD`

To start a session on a high memory compute node (up to 1TB of RAM) type (followed by return)

`./StartJupyter-highmem-YYYY-MM-DD`

To start a session on a GPU node and make use of an Nvidia K40M GPU card, type (followed by return)

`./StartJupyter-gpu-YYYY-MM-DD`

Notes:

- The appropriate YYYY-MM-DD suffix depends on the date the ViperSetupJupyter command was run, NOT the current date.
- The capitalisation of the commands is important, capital S on Start and J on Jupyter.
- When typed into the Viper login node Linux terminal, the command needs to have `/` (dot forward slash) at the beginning.

These commands act as launchers that run job scripts (located in your home directory in a folder called to jupyter-scripts-YYYY-MM-DD) which in turn submit jobs to Viper requesting it start a Jupyter Notebook session.

Typical output from one of these commands is shown below.

```
[username@login01 ~]$ ./StartJupyter-gpu-2022-07-22
Submitted Jupyter notebook task 3590571, waiting for job state information

Job started, you should receive an email with connection details shortly
You can cancel this job by running: scancel 3590571
```

Figure 5 - Starting a Viper Jupyter Notebook

If there isn't spare resource available, for example if all GPU nodes are busy, you will receive a message stating this along with an estimation of when your task will start.

In order to access notebook, a tunnel must be created that allows your local web browser to connect to the compute node running the notebook. Shortly after the job has started, you should receive an email which contains the bespoke connections details needed to connect to your running notebook. An example of the email you can expect to receive is shown below:

Fri 22/07/2022 22:25

N noreply-hpc@hull.ac.uk
Jupyter Connection Details Job ID 3590571

To Chris Collins

The following information will enable you to connect to your Jupyter Notebook by from job ID 3590571

The notebook will be available for 24 hours. If you are finished with your notebook before 24 hours, please cancel the allocation by running `scancel 3590571`

To connect:

For Windows, using MobaXterm click on the Tunneling icon at the top, go to New SSH Tunnel and set as follows:

Forwarded port: 61127

SSH server: viper.hull.ac.uk
SSH login: 524070
SSH port: leave blank (default 22)

Remote server: gpu02
Remote port: 61127

Click Save and then start the tunnel with the green arrow, entering your password when prompted.

For Mac or Linux, please create a tunnel using the command:

```
ssh 524070@viper.hull.ac.uk -L 61127:gpu02:61127
```

Use URL
<http://127.0.0.1:61127/?token=a648b5de65ccdaa2192b94955eb546c6edfe5a155e13436a>

Check you are using the required ipython kernel (shown top right, or select via Kernel > Change Kernel > select:

Python [conda env:.conda-jupyter-2022-07-22]

If you have issues, please raise a ticket via the support portal at <https://support.hull.ac.uk/tas/public/ssp/content/detail/service?unid=04cebf66128e4c8c8df023229c23b64a>

Figure 6 - Viper Jupyter Notebook connection email

- The Information in the orange box explains how to create an SSH tunnel using MobaXterm on Windows.
- The information in the green box is the command to run on the macOS/Linux terminal to create an SSH tunnel
- The URL in the blue box links to your Jupyter notebook and includes a secure token

Setting up a tunnel (step 6)

Windows with MobaXterm

An SSH tunnel can be created between your PC and a compute node using MobaXterm. On the menu bar at the top of the window, click on the Tunneling icon (see figure below):

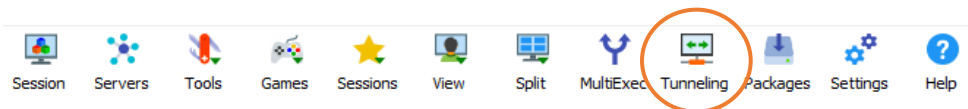


Figure 7 - MobaXterm menu icons – Tunneling icon

In the next window go to New SSH Tunnel (highlighted in green):

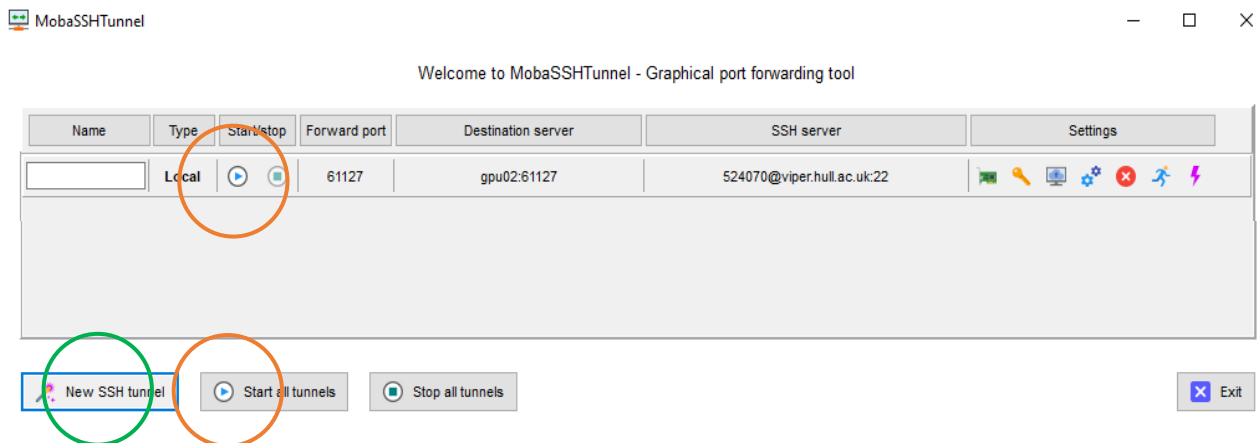


Figure 8 - MobaSSHTunnel tool

In the new MobaSSHTunnel tool configuration window that opens, you need to use the information contained in the email (figure 6) to configure the tunnel, as shown below.

- (a) Forwarded port: <your provided port number>
- (b) SSH server: viper.hull.ac.uk
- (b) SSH login: <your username>
- (b) SSH port: leave blank (default 22)
- (c) Remote server: <the name of the computer your notebook is running on>
- (c) Remote port: <your provided port number>

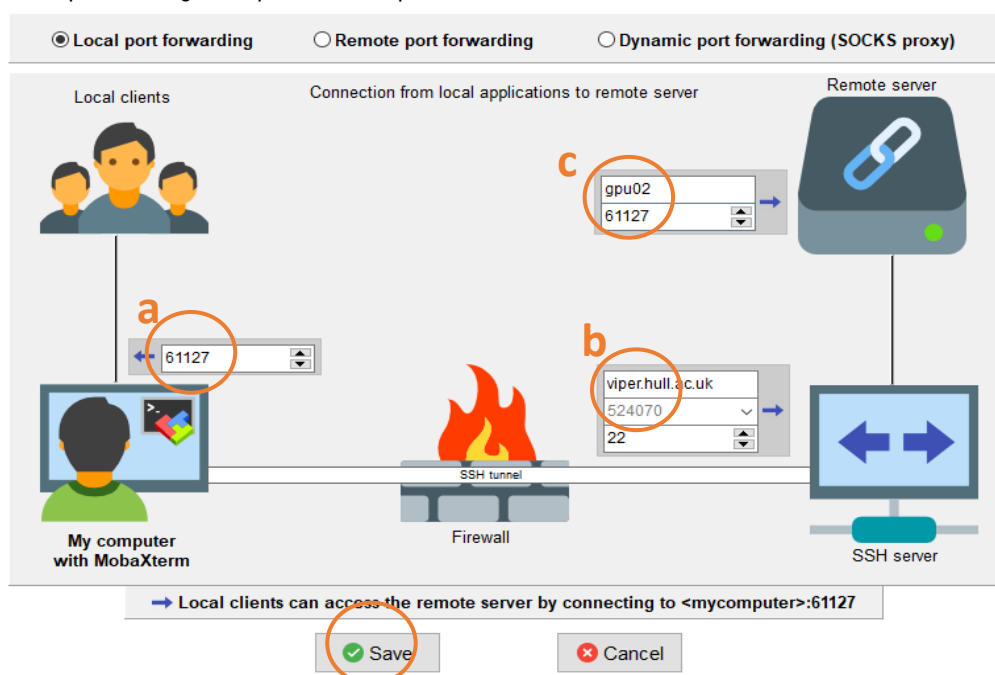


Figure 9 - MobaSSHTunnel tool configuration window

After clicking save, you will return to the MobaSSHTunnel tool where your configured tunnel will now be shown. To start your tunnel, you can either run the small arrow next to the configured tunnel, or on the “start all tunnels” button near the bottom (both options highlighted in orange in the figure MobaSSHTunnel tool). You will then be prompted to enter your password.

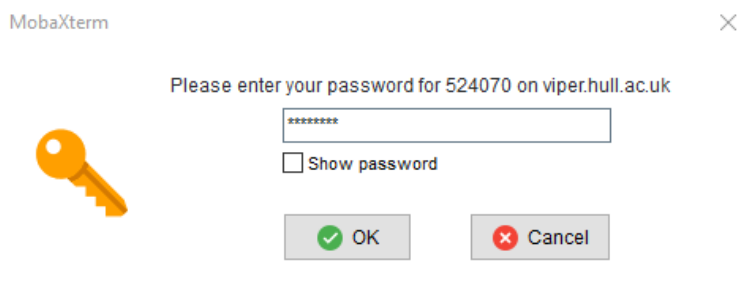


Figure 10 - MobaSSHTunnel password window

When you run subsequent Jupyter Notebook sessions, most of the information for the tunnel will be the same. In which case, clicking on the blue cogs on an existing tunnel configuration and making the minor changes may save time.

Mac or Linux

Creating a tunnel on a macOS or Linux is relatively simple. Firstly start a terminal on your local computer, and either copy from the email, or type out the SSH command, for example:

```
ssh your_UOH_username@viper.hull.ac.uk -L your_port:your_remote_host:your_port
```

Entering your username, your port and the remote host as provided in the email.

Connecting to Jupyter

Once the tunnel is active you can click on the URL in the email, and the main Jupyter page should open in your browser. The URL contains the secret token that allows only you to connect to the running Jupyter Notebook session. This secret token is different on every new Jupyter Notebook session.

At this point you can open up a notebook or create a new notebook and start work:

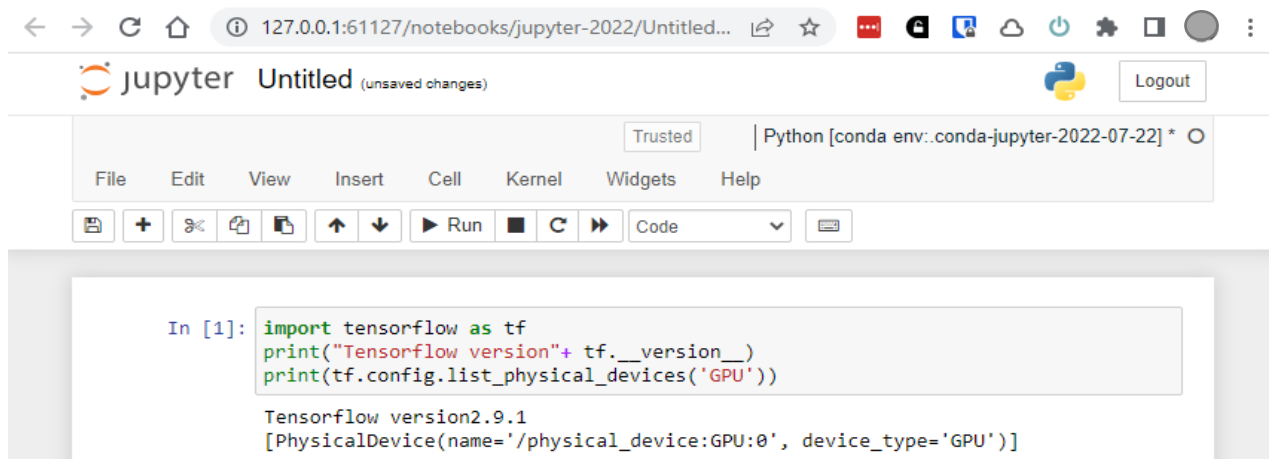


Figure 11 - Working Jupyter Notebook

Using Jupyter (step 8)

Migrating from other platforms

Common platforms used for machine learning such as Google Colab, Kaggle and other Jupyter servers allow existing code to be downloaded as notebooks, in the format of a .ipynb file. These files can then be uploaded to your Jupyter environment on Viper. Examples are shown below:

Jupyter

Open your notebook, go to File > Download As > Notebook (.ipynb)

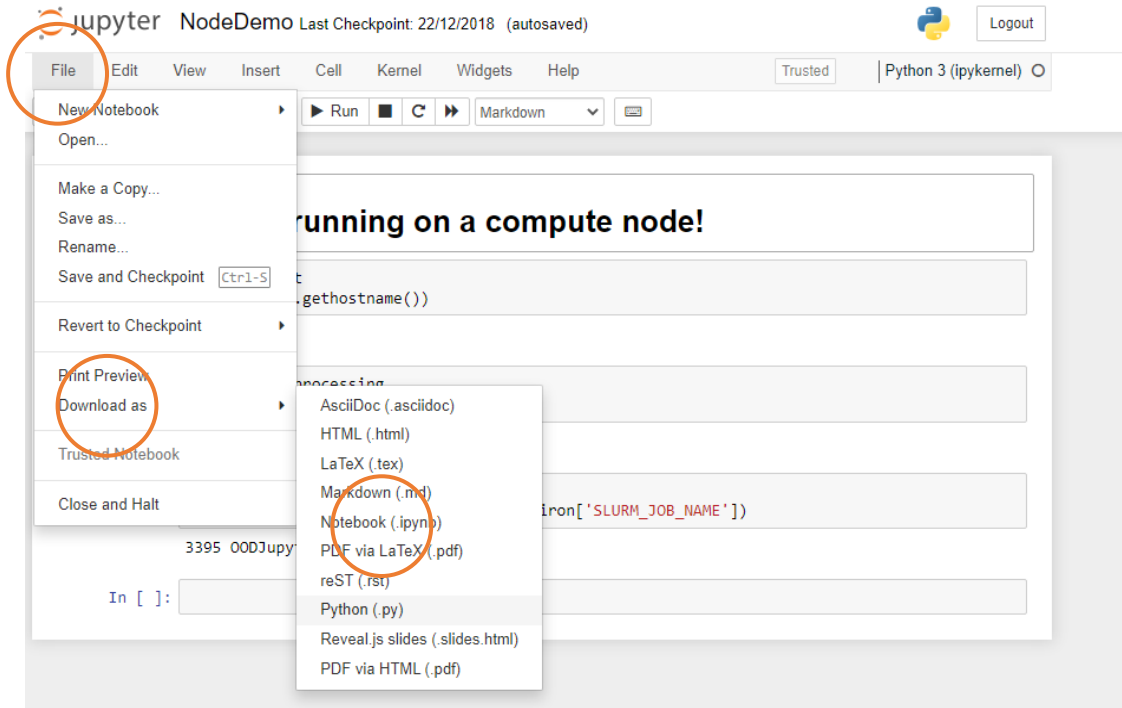


Figure 12 - Jupyter Notebook notebook download

Kaggle

Open your notebook, go to File > Download Notebook

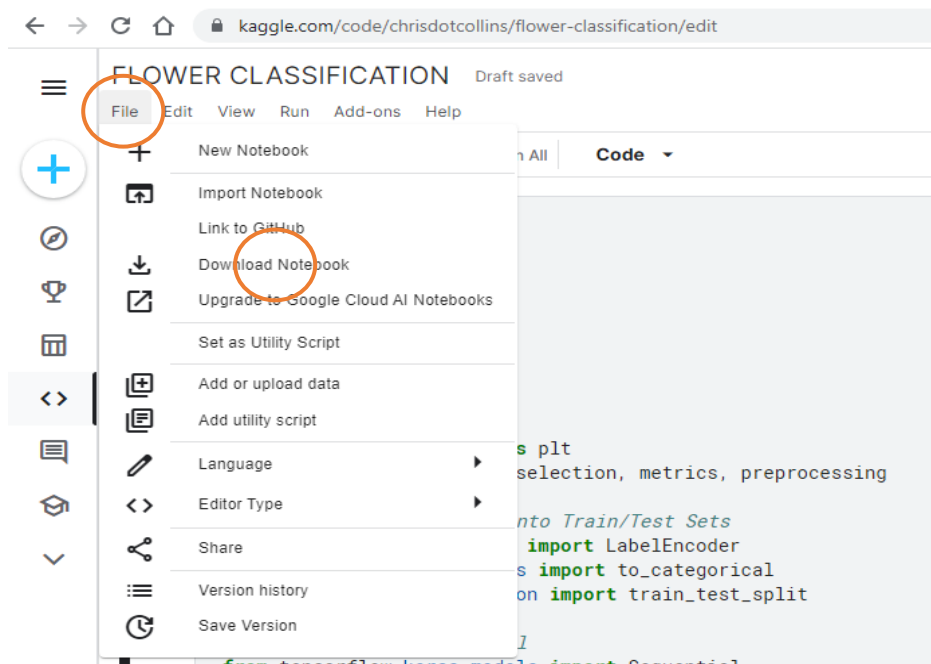


Figure 13 - Kaggle notebook download

Google Colab

Open your notebook, go to File > Download > Download .ipynb

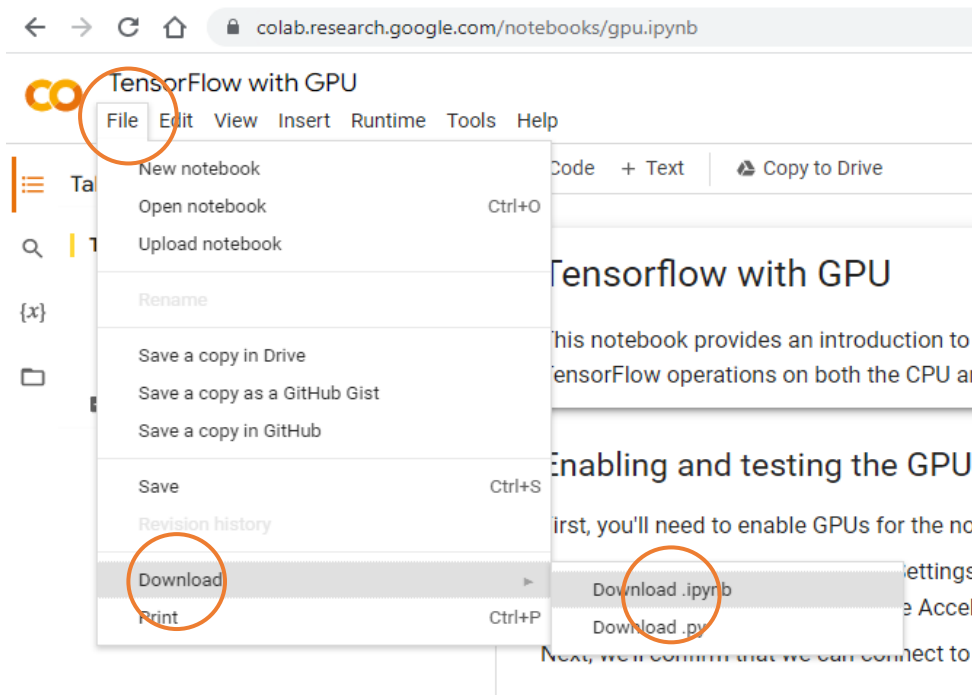


Figure 14 - Google Colab notebook download

Uploading to Jupyter on Viper

Once downloaded, you can then use the upload function on the Viper version of Jupyter to upload the notebook .ipynb file along with any supporting data files:

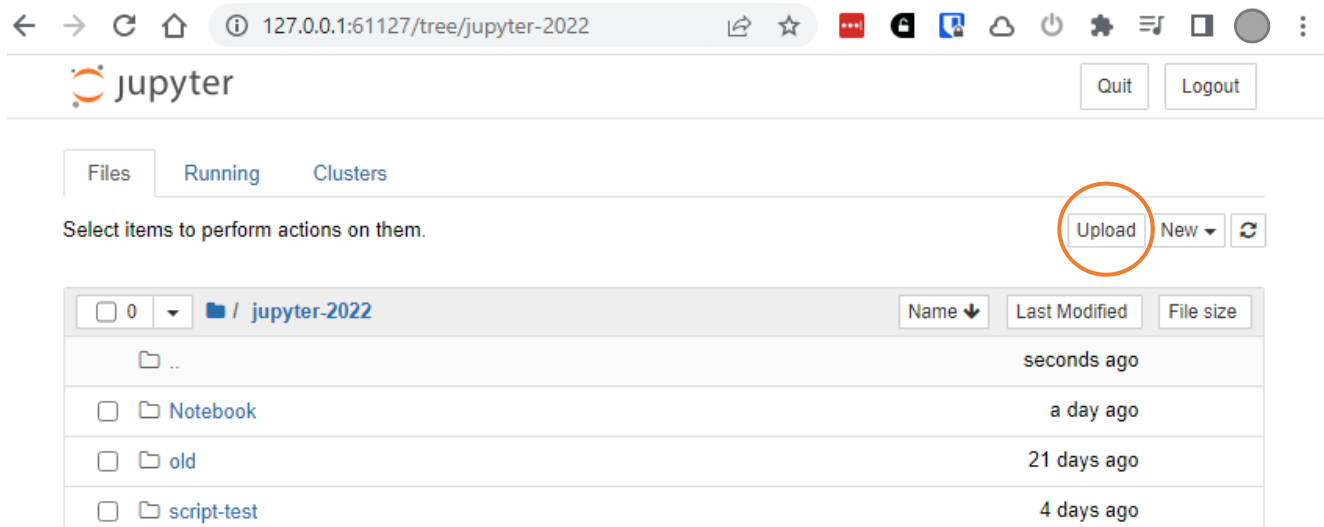


Figure 15 - Jupyter Notebook upload notebook

Python Kernel

When running your notebook, you should ensure you are using the correct Python Kernel. The current Kernel is shown in the top right of the Jupyter page (highlighted in the below screenshot). To change Kernel, go to the Kernel option, then Change Kernel and select the Kernel that matches your Jupyter Environment, this will be marked Python [conda env: conda-jupyter-YYYY-MM-DD] where YYYY-MM-DD matches when you originally ran the ViperSetupJupyter command.

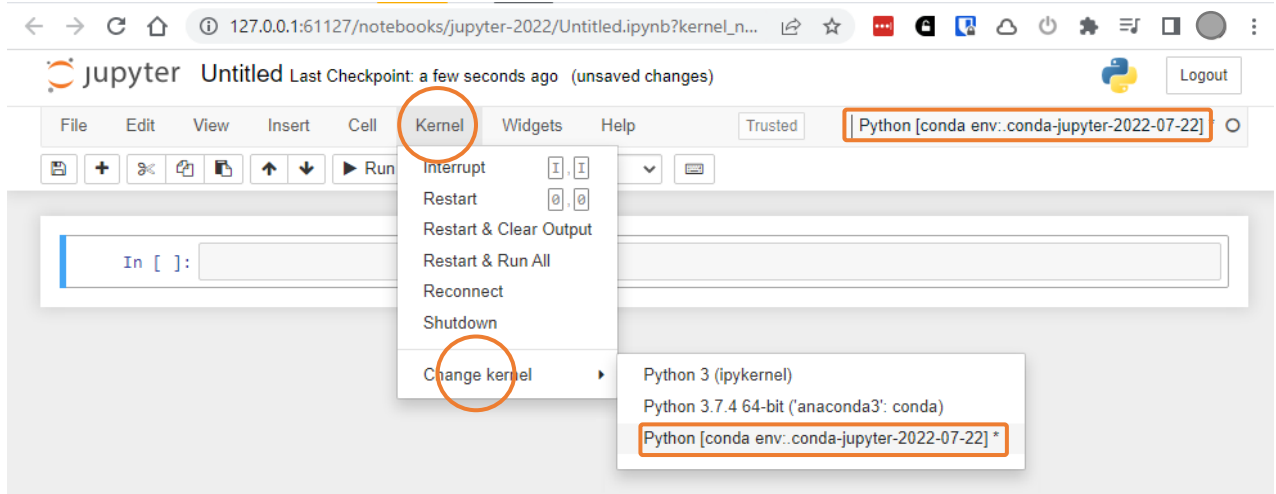


Figure 16 - Jupyter Python Kernel

Installing Packages

The Python virtual environment includes a number of standard packages such as tensorflow, numpy, scipy, matplotlib and pandas. Addition packages can be installed using the conda command, as shown below.

```
In [ ]: conda install packagename
```

Of

```
In [ ]: !conda install --yes packagename
```

Once a package is installed in the virtual environment it is available for future sessions so these commands only need to be run once per package.

If Conda install doesn't work, i.e. the package cannot be found, then pip install may work.

```
In [ ]: pip install packagename
```

Stopping a Jupyter Notebook

Viper is a shared facility and GPU resource in particular is at times in high demand. By default your Jupyter session started using the commands in this guide will run for 24 hours. If you finish working and you still have time left, we would appreciate it if you terminated your Jupyter session returning the resource for others to use. To do this, from your Viper Linux terminal, you need to use the command **scancel**.

The command to run is provided at the top of the 'Viper Jupyter Notebook connection email' (figure 6) and also at the bottom of the output when you run StartJupyter command (figure 5):

```
[524070@login01 ~]$ scancel 3590571
```

More Information

https://hpc.mediawiki.hull.ac.uk/Training/Tutorial_Videos

There are a set of introduction to Linux and introduction to HPC videos. Each are relatively short (5-10 minutes) going into detail about different topics. The first Linux video 1-Intro_to_Linux-Video_1-Fundamentals video includes information about connecting to Viper. The introduction to HPC videos will hopefully provide some useful background information about using Viper.

https://hpc.mediawiki.hull.ac.uk/Main_Page

There is a Viper Wiki which contains lots of information about using Viper.

https://hpc.mediawiki.hull.ac.uk/Training/Getting_Connected

The wiki page includes information about setting up MobaXterm on Windows as well as how to connect from macOS and Linux

Getting help

The University Support portal available at <https://support.hull.ac.uk> is the primary way of getting assistance.

For VPN access issues, please use the search bar on the support portal and search for “Working remotely” and you will then see links to [Report an Issue \(Remote Working\)](#) or [Request access to the GlobalProtect Client VPN Service](#)

For Viper issues, including if you are having problems running Jupyter Notebooks, then please use the search bar and search for “Viper HPC” and then use the [Report an Issue with a Viper HPC Job](#)

When reporting a problem with Viper or Jupyter on Viper, please provide as much information as possible. The following is particularly useful:

- The job number or ‘ID’ of the session which can be found in the subject of the email or among the output when running the StartJupyter commands and is typically a 7 digit number.
- The name (or filename) of the Jupyter notebook that you are having problems with.
- The nature of the problem, including any behaviour you are seeing and what you would expect to see.
- Details of any particular packages you are using.
- Information on whether this task has worked before, including on other platforms (Google Colab, Kaggle etc) and what if any differences there are in this code.