To login to the server and go to the desired directory:

Go to terminal of ubuntu and type
ssh e27890@wight.seg.rmit.edu.au
Terminal will ask for password. It is
*ens$026#* Now it's mea#620$

You are now inside a non-GPU instance. Now jump to a GPU server. Type
ssh segsresap07
Terminal will again ask for password. Same password now.
ens$026#

Now type
cd /research/remote/petabyte/users/eunus

Here create a directory for your thesis, so that your work doesn't get mixed up with others' files.

** The complete directory path will look like this.
e27890@wight.seg.rmit.edu.au/research/remote/petabyte/users/eunus

Most probably the hard drive is shared between GPU instances and non-GPU instances, so the path will be the same for both the GPU and non-GPU instance.

** Anaconda (now default python 3.5), CUDA, cudnn, tensorflow-gpu and keras have already been installed in the server. If you need to work with python 2, create a new python environment inside Anaconda and install everything inside it.

** You can use sftp to get a GUI mode of files and directories. There you can open files, see them, and edit them, create new files and folders- just like you do in your pc.
Click "Connect to server" in Ubuntu and type
sftp://e27890@wight.seg.rmit.edu.au/research/remote/petabyte/users/eunus
and click "Connect".
It will ask for the password.

You are now in a non-GPU instance. You may stay here and edit files, or you may go to a GPU instance. To go to a GPU instance, right click and select "Open in remote terminal". Now jump to a GPU server in the way mentioned in the 2nd paragraph
(Jumping to a GPU server will take to back to the Home directory in the terminal, so you need to use
cd /research/remote/petabyte/users/eunus
to go to the desired directory)
EDITING FILES CAN BE DONE EITHER FROM GPU INSTANCE OR NON-GPU INSTANCE. Both have the same effect. But their configurations (CPU, RAM etc) are NOT the same.

##

Copy files to and from server:

Copying a directory from your pc to the server (and vice versa) takes a lot of time if it has a lot of small files, because files are copied one by one. If the files are of kB range size, then copying speed will be in kB/s which is very slow. In that case, create a zip of that folder, and copy that zipped file to the server. The server will consider the zipped file as a single file, so it will try to upload (or send) it quickly.

To copy a file from your pc to the server, use scp command.
scp file_path_in_your_pc file_path_in_the_server

file_path_in_the_server will look like this:
e27890@wight.seg.rmit.edu.au:/research/remote/petabyte/users/eunus/your_directory_in_the_server/Your_folders_and_files

To copy a file from the server to your pc:
scp file_path_in_the_server file_path_in_your_pc

** For the destination, only the parent folder path should be given in the command.

To unzip a zipped file, type:
unzip your_zipped_file_name

To zip a folder:
zip -r your_zipped_file_name the_folder_you_want_to_zip

your_zipped_file_name should contain .zip at the end.

For extracting a tar.gz archive,
tar -xvzf your_zipped_file_name

Here*, your_zipped_file_name* should contain .tar at the end.

You can do all these simply in GUI mode like you do in your local computer using sftp, but it will probably take much more time.

** **For windows users:**
Install PUTTY from here- https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

Select one of the MSI (Windows Installer) from "Package files" according to your pc configuration.
Install PUTTY in a suitable directory according to your choice.
Go to the directory where you installed Putty and open putty.exe (Application).
Click "Session" and type
e27890@wight.seg.rmit.edu.au
in the "Host Name (or IP address)" textfield. Select the radio button for SSH. For safety, I would prefer selecting "Always" for "Close window on exit". Putty doesn't save this configuration for later use, however. You have to do this every time you start Putty.
Click "Open". If any window pops up, click "Yes". Now it will ask for password.

(Copying text from your pc to the putty terminal is somewhat strange. Copy from your pc using ctrl+c, then just press the right mouse button in the putty terminal to paste it there)

Now you can jump to a GPU server, as mentioned previously.

To copy files from your windows PC to the server, simply open a command prompt (not putty) and use "pscp" instead of "scp".

sftp is possible in Putty using psftp, but you can't get any GUI mode most probably, so you have to do all your work in terminal.

Training a model without your presence:

Training a model may take a lot of time, and you may not want to keep your PC awake all the time. Use "screen" to let the training occur at your absence.

Before starting the training, create a screen.

screen -S any_name

YES, IT'S CAPITAL S. Now start training and detach your PC from the screen (wait, I ain't talking about any physical connection detaching !!). Press 3 buttons as once: CTRL+A+D . You have now detached your PC from the screen. Server is doing its work. Note down the PID. Now you can safely close the terminal (ubuntu or putty terminal) and even turn off your PC.

To retrieve the screen you created, type

screen -r PID_you_noted.the_name_you_gave

after you log in to the server.

Once you have logged in to the server, you can get the list of active screens by typing

screen -ls

Here you will find both the PID and the name of the screen.

To kill a screen, type

screen -X -S PID_of_the_screen.name_you_gave quit

(source: https://stackoverflow.com/questions/1509677/kill-detached-screen-session)

Some peculiar problems:

- If you stop training a model midway, the GPU memory doesn't get freed. In that case, you have to free the memory manually.

  Type

  ```
  nvidia-smi -q
  ```

  in the terminal. It will give you a list of all the processes (and their PIDs) occupying the GPU memory (go to the memory section of GPU). Now kill all these processes.

  kill -9 PID_of_the_process_you_want_to_kill

  If you don't do this, you may eventually run out of memory and problem like this can occur- https://stackoverflow.com/questions/48514608/tensorflow-failed-to-create-session-in-server/48515395

  If you find errors like ResourceExhaustedError or see that tensorflow is not able to create any session at all, kill the processes occupying the GPU.

- Occasionally you may not be able to login to the server. Normally it's a temporary problem.

Installing:

Installing Anaconda:

Download the installer in your pc and use scp to copy it to the server. Then follow the instructions of the official site.

** Let Anaconda change the path variable.

Installing tensorflow-gpu and keras:

Tensorflow works best with Python 3.5 . But currently, the default Anaconda comes with Python 3.6.3.

So install Python 3.5 after installing Anaconda by typing this.
conda install python=3.5

Now install cuda and cudnn by typing this.
conda install cudnn

Now install tensorflow-gpu.
conda install tensorflow-gpu

Install keras now.
conda install keras

You may need pillow and h5py packages.
pip install pillow
conda install h5py

For image augmentation, you may need the Augmentor package.
pip install Augmentor

Augmentor documentation:
https://augmentor.readthedocs.io/en/master/userguide/mainfeatures.html (Don't rely on it completely. See the comments and documentation in the source code)
http://augmentor.readthedocs.io/en/master/_modules/Augmentor/Pipeline.html (source code)

(Always try to use conda install. Some packages, like pillow, Augmentor etc cannot be installed using conda. In those cases, use 'pip' instead of 'conda')