# Guide: Remote Server Access, MLC, Jupyter Notebook, File Transfer, and Basic Commands

**1. Accessing the Remote Server**

* **Connect to HS-Student WLANs**
* **SSH Access**: Use Secure Shell (SSH) to access the remote server from your local machine's terminal (use cmd).

ssh [214288@students.hertie-school.org@ds01](mailto:214288@students.hertie-school.org@ds01)

ssh 214288@students.hertie-school.org@192.168.50.152

* Enter your password
* **Exit remote server:** exit

**2. Creating a Machine Learning Container (MLC)**

* **Create MLC**: Use the **mlc-create** command to create a Machine Learning Container.

mlc-create kai-ufo-predict Tensorflow 2.11.0 [-w=/home/214288@hertie-school.lan@ds01](mailto:-w=/home/214288@hertie-school.lan@ds01)

(Note: specify a recent Tensorflow version, as they will define which python is installed. New versions of Tensorflow will come with new python versions, e.g. python 3.10)

* **Access MLC:**

mlc-open kai-ufo-predict

* **Check python version:**

python –version

* **How to close mlc:**

mlc-stop kai-ufo-predict -Y

* **List all available mlcs:** mlc-list
* **Remove mlc:** mlc-remove kai-ufo-predict (Note first you need to close the mlc)
* **Exit mlc:** exit (Note: this does not close the mlc, you just get back to home directory on the remote server)

**3. Configuring jupyter kernel**

* **Add kernel to jupyter:**

python -m ipykernel install --user --name=kai-ufo-predict --display-name="ufo-predict (Tensorflow)"

(Note: you will get a warning that it has been stored on the root directory. This is no problem, it is not the root directory of the server, but just the mlc, i.e. the docker file)

* **Check kernel specification**

Jupyter kernelspec list

**4. Set up SSH remote server access in VS Code**

* **Install the “Remote Development” Extension pack:** Go to the extension tab, type in “remote development” and install the option
* **Set up remote on VS code:** On the bottom left of the IDE, click on "Open remote window" icon. Select “Add SSH Host”. Type in “ssh [214288@students.hertie-school.org@ds01](mailto:214288@students.hertie-school.org@ds01)” and confirm password. When prompted what system to use, choose “Linux” or “Ubuntu”, not “Windows” or “Mac”
* **Install Extensions remotetly:** You will need to reinstall extensions such as Git, Python in the remote.
* **Clone your repo:** Use the inbuilt git extension to clone your repo and save it in “workspace/”
* **Open your repo on the remote:** You can open folders on the remote machine once you have installed “Remote-SSH” and “Remote-Containers” extension. Now click on an .ipynb file you want to run and it should open.

**5. Starting Jupyter Notebook in MLC and Using It in Visual Studio Code**

* **Start Jupyter Notebook**: Note within VS Code, but outside it (i.e. use cmd), open cmd and type:

jupyter notebook –no-browser –port=9090

(Note port 9090 works good, but if occupied, try another one like 9095)

* **Open a new cmd and type:** type command below and enter password

Ssh -f [214288@students.hertie-school.org@ds01](mailto:214288@students.hertie-school.org@ds01) -L 9090:localhost:9090 -N

(Note: remember to use the same port as you specified above)

* **In a browser, enter the URL provided in the first CMD you opened and return enter.** Even if the jupyter notebook does not appear in your browser, now you are ready to use it in VS Code. Trust me.
* **Select kernel in VS code**: On the top right of VS Code, click on “select kernel” or the kernel that is currently specified. Click “select another kernel” -> “Existing jupyter server” -> “Enter the URL of the running jupyter server” (this should fill out automatically) -> confirm the IP address that pops up and you are DONE!
* **Close jupyter session:** Ctrl+Fn+C on the cmd where the urls are posted and it will prompt you whether you want to close it. Type y and you are out.

**6. Installing additional packages into your mlc environment**

* **List installed packages**

Pip list

* **Install desired packages**

Pip install desired packages

* **Restart kernel (not always necessary)**

**4. Uploading Data Files Using SCP**

* **Upload Data to Remote Server**: To upload data from your local machine (Windows) to the remote server using SCP, use the following command. Replace **username**, **remote-server**, and **remote-path** as needed:

scp C:/Users/kaius/NLD.pkl [214288@hertie-school.lan@ds01](mailto:214288@hertie-school.lan@ds01):

(Note: this will upload the file to the your home directory)

* **Downloading a file:** DON’T connect to the server. Then type

scp [214288@hertie-school.lan@ds01:workspace/ufo-prediction/demo/rca-ufo-merge.csv](mailto:214288@hertie-school.lan@ds01:workspace/ufo-prediction/demo/rca-ufo-merge.csv) .

scp [214288@hertie-school.lan@ds01:workspace/ufo-prediction/demo/rca-ufo-merge.csv](mailto:214288@hertie-school.lan@ds01:workspace/ufo-prediction/demo/rca-ufo-merge.csv) C:/Users/kaius

**5. Moving Files on the Remote Server Using MV**

* **Move Files on Remote Server**: Use the **mv** command to move files within the remote server. Replace **source** and **destination** with file paths:

mv NLD.pkl workspace/ufo-prediction/demo

**Additional Commands:**

* **Change Working Directory (cd)**:
  + To navigate to a different directory, use **cd** followed by the directory path:

cd /path/to/directory

* + To go up one directory level (to the parent directory), use:

cd ..

* **List Files in a Directory (ls)**:
  + To list the files and directories in the current directory, use the **ls** command:

ls

This guide covers the basics of accessing a remote server, creating an MLC, using Jupyter Notebook, transferring files, and performing basic file and directory operations. Make sure to replace placeholders with your specific information and paths.