

Transferring files from and to an UCSD's DSMLP Pod session

Written by Charles Deledalle on October 1, 2019

1 From your machine to DSMLP and vice-versa

A difficulty on DSMLP, is that all files you create inside your pod session cannot be easily accessed. In particular, you cannot easily use `scp` or `ftp` from your laptop to copy a file from your DSMLP pod session. Your only option is to transfer the file from your pod session to an accessible server.

In the following I am assuming you are using a recent Debian/Ubuntu Linux distribution, Bash and you have python and pip. I believe the procedure is very similar on Mac OS X.

1.1 Option 1 - Git

In most cases, basically to transfer code, LaTeX, and small files, the best option is to use a Git server (such as GitHub or Bitbucket) and pull/push your files to/from your pod session from/to that server. This makes even more sense when you are working on a common project with your classmates.

1.2 Option 2 - SshServer+VPN+SCP

For larger files (such as pretrained networks), you may want to avoid storing them on a Git repository. An alternative for you is then to install an SSH server on your machine (alternatively you could install an FTP server). Here are the instructions:

Installation and configuration (to do once): First of all, make sure to have a strong password on your machine (you can use the command `passwd` to change your password). Then, install an SSH server:

```
$ sudo apt install openssh-server
```

To verify that the SSH service is running

```
$ sudo systemctl status ssh
```

This should install and start an SSH server listening on port 22. If not running, enable the ssh server and start it as follows:

```
$ sudo systemctl enable ssh
$ sudo systemctl start ssh
```

You can stop the service as

```
$ sudo systemctl stop ssh
```

Once the SSH server is on, anyone on your subnetwork, knowing your IP, can access your laptop by SSH (on port 22). Of course, he or she needs to have an account on your machine and will be asked a login/password.

Transfer: By connecting your machine onto UCSD's VPN, anyone on campus can now access your machine via SSH, including yourself! After connecting to UCSD's VPN with **Cisco AnyConnect Secure Mobility Client**, look for your IP under the item **Client Address (IPv4)** (alternatively you can identify your VPN's IP via `ifconfig`). It probably looks like `137.110.XX.XX`. Note that your IP may change each time you restart the VPN. Now that you have determined your IP, connect to DSMLP and open your pod session. You can now transfer your file from DSMLP to your machine as

```
$ scp file_on_DSMLP login_on_your_machine@137.110.XX.XX:~/path_on_your_machine
```

or from your machine to DSMLP as

```
$ scp login_on_your_machine@137.110.XX.XX:~/path_on_your_machine file_on_DSMLP
```

In both cases, you execute `scp` on the pod session. Use `scp -r` to transfer directories.

1.3 Option 3 - Jupyter

If none of the above options work for you, a very convenient alternative is to use Jupyter Homepage to upload or download a particular file. You can use Jupyter like a browser based file navigator, select the file you want to download and then click on download. However, unfortunately, it only allows downloading one file at a time, so you may want to compress a set of files together using `zip` or `tar` and then download them using `jupyter`.

2 From internet to DSMLP

2.1 Case 1 - Direct download: `wget` and `curl`

Any files available with a direct URL link can be downloaded by typing on your pod:

```
$ wget http://url_of_the_file_you_want_to_download
```

or if you want to specify a destination filename

```
$ curl http://url_of_the_file_you_want_to_download -o output_filename
```

2.2 Case 2 - From GoogleDrive

In some situations you want to transfer a file from Google Drive. For large files, you may not even be able to download the file by clicking as the transfer often fails.

A first solution is to download my script (on your pod session with `wget` for instance) <https://bitbucket.org/cdeledalle/misc/src/master/sh/curldrive.sh> that runs as

```
$ bash curldrive.sh LINK FILENAME ACCESSTOKEN
```

To get the `LINK` go on Google Drive, right click on the file you want to transfer and click on `get sharable link`. An URL of the form <https://drive.google.com/open?id=XXXXXXX> will be copied to your clipboard, this is the first argument `LINK`. The second argument `FILENAME` will be the destination filename of your choice (it does not have to be the same as the source one). To get the last argument `ACCESSTOKEN` go to <https://developers.google.com/oauthplayground/>, in the right

panel choose Drive API v3 and select <https://www.googleapis.com/auth/drive>. Then click the button **Authorize API**. Choose the Google Account linked to your Google Drive, click on **Allow** and then **Exchange authorization code for tokens**. Finally copy and paste the **Access Token**.

A second solution is to look at this project <https://github.com/gdrive-org/gdrive>. I did not try it, but it is probably a good alternative to my previous solution.

2.3 Case 3 - From Kaggle

Installation and configuration (to do once): Install the official kaggle API

```
$ pip install --user kaggle
```

which will install the Kaggle command here `.local/bin/kaggle`. If not done already, add `.local/bin` in your `PATH` environment variable by adding to your `.bashrc`:

```
$ export PATH="$HOME/.local/bin:$PATH"
```

and type

```
$ source ~/.bashrc
```

Now you have to indicate Kaggle information relative to your Kaggle account and UCSD's proxy. Edit the file `.kaggle/kaggle.json`

```
{
  "username": "your_kaggle_login",
  "key": "your_kaggle_key",
  "proxy": "http://web.ucsd.edu:3128"
}
```

Make sure the file is protected

```
$ chmod 600 .kaggle/kaggle.json
```

Transfer: Once Kaggle is installed and set up, you can download kaggle datasets as

```
$ kaggle datasets download datasetkey
```

for more information refer to <https://github.com/Kaggle/kaggle-api>.