SSH and Stuffs

Introduction

SSH into a network

```
ssh user@remote_address
```

Copy file to ssh connection

```
scp file_name user@remote_address:~/copy_directory
```

View image over ssh

Requirement

- Check the server's **sshd_config** *(/etc/ssh/sshd_config)* make sure `X11Forwarding yes` is not commented out.
- Run ssh with extra parameter -Y `ssh -Y user@remote_address`
- open image using **eog** `eog image.jpg`

Running Jupyter notebook on remote ssh and using on local machine

- start jupyter notebook on the remote host using --no-browser parameter

```
remote_user@remote_host$ jupyter notebook --no-browser --port=8889
```

- make sure to copy the token from the terminal.

```
[C 15:21:43.664 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///home/masum/.local/share/jupyter/runtime/nbserver-6188-open.html
    Or copy and paste one of these URLs:
        http://localhost:8889/?
token=0b3fbf10217d5c78b94529ca2ef07623ea28f8911fee9a30
     or http://127.0.0.1:8889/?
token=0b3fbf10217d5c78b94529ca2ef07623ea28f8911fee9a30
```

- Run shell tunnel to forward port

```
local_user@local_host$ ssh -N -f -
L localhost:8888:localhost:8889 remote_user@remote_host
```

> Explanation: The first option -N tells SSH that no remote commands will be executed, and is useful for port forwarding. The second option -f has the effect that SSH will go to background, so the local tunnel-enabling terminal remains usable. The last option -L lists the port forwarding configuration (remote port 8889 to local port 8888).

- Now open `localhost:8888` on your local machine.

> Note: Remember its **8888** on your machine that is forwarded to **8889** of the remote server.

Screen

While using remote connection to run a continuous session i.e. running a development server that needs to keep running even if the ssh session is not active or disconnects. Go see Screen Notes

Authorize local machine to remote server

- Generate rsa key if not done already. Check by using

```
cat ~/.ssh/id_rsa.pub
```

- If the file doesn't exist

```
ssh-keygen -t rsa -b 4096 -C <email>
```

> follow => link

- Add key to remote machine as authorized client

```
ssh-copy-id name@hostname
```

Develop remotely using VSCODE

Microsoft-notes

Install the extension REMOTE-SSH from marketplace. Configure the SSH configuration file for local user.

> VSCODE version need in both machine needs to match.

Add fingerprint to remote server as trusted user

Use ssh tunnels and stuffs without prompt to enter password everytime.

Pre-requisite

- SSH Key generattion: its a fine article. In short to create a `rsa` ssh you need to do the following

```
ssh-keygen -t rsa -b 4096 -C <email@host.com>
```

Remember email is not necessarily a verified one. But its better to use the one you will be using for push and other operation.

- Checking for existing key Another fantastic article. In short check if there is anything in `~/.ssh` directory -

```
ls -al ~/.ssh
```

The thing we need is the *id_rsa.pub* (In case of RSA key).

Now add the key as trusted device

The public key will be added to trusted device fingerprint list on the remote host.

```
ssh-copy-id -i ~/.ssh/id_rsa.pub user@host
```

SSH using multiple public-private key pairs

We can access different machine using different public-private key pair just so they stay more safe. i.e. Accessing a running EC2 instance requires to use the private key provided by AWS console.

```
ssh -i <key_file_path> user@host
```

-i indicates the input-file as key parameter. SSH protocol demands that the permission of a key file is set 400.