LHTSERVER-2 MANUAL

Server Information
1. Basic Usage
    1.1 The ABCs of SSH
    1.2 Usage of Some Installed Tools
    1.3 Location of Some Installed Packages and Softwares
        Anaconda (Python)
        CUDA
        PyTorch
        TensorFlow
        Matlab

Server Information

- **IP address:** `202.120.38.232`
- **System:** Ubuntu Server 16.04.5 LTS
- **Kernel version:** 4.4.0-77-generic

1. Basic Usage

1.1 The ABCs of SSH

The port of *ssh* connection to the server is `2022`.

- **Linux** or **Mac**

    1. In **Linux** or **Mac**, you can log in to the server by *ssh* in terminal, like

        ```
        ssh -p 2022 $USERNAME@202.120.38.232
        ```

        If you want to play a video, view images or launch MATLAB with GUI interface, use `-X` for X server forwarding:

        ```
        ssh -p 2022 $USERNAME@202.120.38.232
        ```

        It is recommended to replace `-X` with `-Y4C` to speed up SSH connections:

        ```
        ssh -Y4C -p 2022 $USERNAME@202.120.38.232
        ```

2. When you are using **ipython notebook** or some other **Web server**, you will notice that you can only connect to the server through port `2022` and all the other ports are closed. Now you may need the port forwarding (Tunnel) to make your server work as expected.

   > Local Port Forwarding : connections from the SSH client are forwarded via the SSH server, then to a destination server
   >
   > ```
   > ssh -L 8999:127.0.0.1:8999 $USERNAME@202.120.38.232 -p 2022
   > ```
   >
   > With this command, you can visit `http://localhost:8999` in you browser to connect your Web server which are listening to the port `202.120.38.232:8999`. The following command may find out which ports are used by the other users
   >
   > ```
   > nmap -sT localhost
   > ```

   You can also append the following lines to file `~/.ssh/config`:

   ```
   Host LHTSERVER-1  # you can also set another name you like
     HostName 202.120.38.232
     User $USERNAME
     Port 2022
     LocalForward 8899 127.0.0.1:8899
   ```

   an run `ssh LHTSERVER-1`, it works the same as the previous commands.

3. SSH clients will automatically be disconnected from the server and prompt the below message after being idle or inactive for several minutes. Adding these lines to `ssh_config` will keep your connection alive.

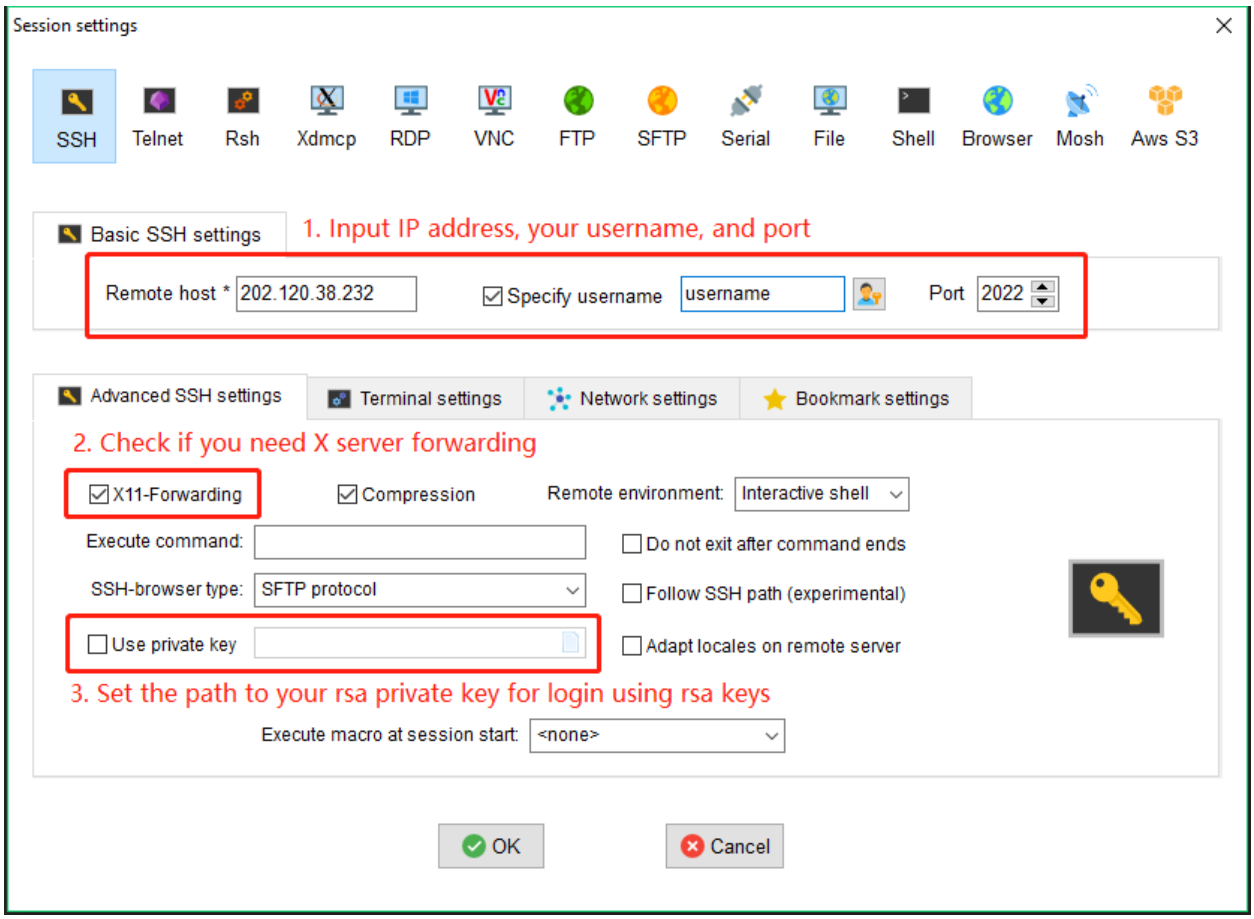
   ```
   Host *
   ServerAliveInterval 180
   ServerAliveCountMax 5
   ```

4. To transfer data between the remote server and your local computer, you can use `scp` or `sftp` in the terminal. For example, Copy a file from your computer to the server:

   ```
   scp /path_to_file $USERNAME@202.120.38.232:/home/$USERNAME/somefolder/ -P 2022
   ```

   Get more information about the usage from **Usage of scp** and **Usage of sftp**.

- **Windows** In **Windows**, you can use some GUI SSH terminals with X server (like MobaXterm) or without X server (like PuTTY), which may take some extra time to learn about. You might also want to try Cmder or Cygwin, both of which can connect the server by ssh in command line in the same way as **Linux** or **Mac** (no X server). Here is an example for how to set up MobaXterm



1.2 Usage of Some Installed Tools

- Aria2 [Aria2] is a lightweight, multi-connection download utility. You can also download a file by `wget`. For example, downloading a zip file by using 2 connections per host:

```
aria2c -x2 http://foo.org/bar.zip
```

- feh `feh` is an image viewer for console users (X11 forwarding required). You can view an image in the terminal by

```
feh /path_to_image/image.jpg
```

- htop You can dynamically monitor the CPU and Memory in `htop` or `top`. **When you are running some programs and have no idea how much resource it will require, run `htop` at first.** This is important to avoid a server crash.
- mplayer `mplayer` is a video player for console users (X11 forwarding required). For example,

```
mplayer /path_to_video/video.mp4
```

1.3 Location of Some Installed Packages and Softwares

Anaconda (Python)

- DIR: `/usr/local/anaconda3`
- Version: **Pyhon 3.6.5** It is recommended to set the *Anaconda* as your default python interpreter since many third-party packages (like numpy, sklearn, PyTorch, TensorFlow, etc) are installed in it. Please add the following line to your bash file (`~/.bashrc`) before you use,

  ```
  export PATH="/usr/local/anaconda3/bin:$PATH"
  ```

  Then, execute the following command to activate your configuration,

  ```
  yourusername@lhtserver-2: source ~/.bashrc # activate your change made in ~/.bashrc
  ```

  Now you have finished the configuration, and you can verify your python environment by executing the following command and see the results like below:

  ```
  yourusername@lhtserver-3: which python
  /usr/local/anaconda3/bin/python # print the full path of your python3 interpreter
  ```

  We recommend you to use **Python 3** instead of Python 2 for better official support in the future. However, we also installed **Python 2** as a virtual environment named **py2** of *Anaconda3* so you can switch your python environment between *Python 2* and *Python 3* conveniently.

  ```
  yourusername@lhtserver-3: source activate py2 # switch to python2 from python3
  (py2) yourusername@lhtserver-3: which python
  /usr/local/anaconda3/envs/py2/bin/python # print the full path of your python2 interpreter
  (py2) yourusername@lhtserver-3: source deactivate # switch to python3 from python2
  ```

CUDA

So far, we have installed 3 different versions of CUDA for different deep learning platforms:

- **CUDA 8.0**: `/usr/local/cuda-8.0` , with **cuDNN 7.0.4**
- **CUDA 9.0**: `/usr/local/cuda-9.0` , with **cuDNN 7.1.4**, in support of **TensorFlow 1.9.0**
- **CUDA 9.2**: `/usr/local/cuda-9.2` , with **cuDNN 7.1.4**, in support of **PyTorch 0.4.1**
- **GPU Driver**: 396.37

Before running your deep learning program, you need to config your CUDA environment correctly. For example, if you want to use **PyTorch**, you should make sure that your **$PATH** and **$LD_LIBRARY_PATH** variables are configured correctly, ```bash export PATH="/usr/local/cuda-9.2/bin:$PATH" export LD_LIBRARY_PATH="/usr/local/cuda-9.2/lib64:$LD_LIBRARY_PATH" ``` You can also add the above lines to your `~/.bashrc` file to make it effective for all terminals

PyTorch

- **Version**: **0.4.1**
- **CUDA**: **CUDA 9.2**
  PyTorch is installed for both *Python 3* and *Python 2*. Import by `import torch` . For more help, See Docs and Forum.

TensorFlow

- **Version**: **1.9.0**
- **CUDA**: **CUDA 9.0**
  TensorFlow is installed for both *Python 3* and *Python 2*. Import by `import tensorflow`

By default, **TensorFlow will take up all available GPU resources** which may bring some trouble to other users. It is recommanded to add the following codes in your program to limit the GPU resources:

```
import tensorflow as tf
import os
os.environ['CUDA_VISIBLE_DEVICES'] = '0' # only use one GPU: gpu_id = 0
config = tf.ConfigProto()
# allocate GPU memories exactly according to your program
config.gpu_options.allow_growth = True
# set the percentage of maximum gpu memories used for running this program
config.gpu_options.per_process_gpu_memory_fraction = 0.5 # 50% of total gpu memories
sess = tf.Session(config=config)
```

Matlab

DIR: `/usr/local/MATLAB/R2016b` MATLAB is already added to the `$PATH`.