How To Connect With Putty Via SSH

Setting up a connection within PuTTY

> *Download PuTTY from, http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html*
> *Select putty.exe from the list.*
> *Once you have it downloaded open it up.*
> *The information you need to enter to connect to your account is the following*
> *Hostname: yourdomain.com or yourIPAddress*
> *Port: 22*

Common Commands

In the below examples commands will be surrounded by quotes, please do not utilize the quotes when entering the commands.

Auto Complete Feature
The auto complete feature allows you to utilize tab when running commands. I will refer to this feature in below commands and when it is very useful.

pwd
This command is utilizing by simply typing pwd. It gives you the current working directory or the directory you are currently in. When you first SSH to a server you are placed in your home directory which is located at /home/username/ There is an alias to get you to your home directory which is the “~” character. This command can be utilized from anywhere and will always give you your full working path.

passwd
This command is used to change your password. When you run the command “passwd” and it will prompt you for your current password and then a new password.\

Simple Help With Commands

Keep in mind when utilizing any Linux command you may want further information on the command itself. There are multiple ways to get this information and I recommend the first method. The first method is simply run the command with the –help switch. For instance “passwd –help” will give you a lot of the common switches and uses for the command incase you forget how it is used. The second method is to look at the man page. You can get into a man page for a given command with the following “man passwd”. This will open an editor that allows you to scroll through the man or manual page that has a very indepth and detailed description of the commands. Usually it gives too much information and is unhelpful. Another resource is google, however it will generally just return a man page.

cd
The next command that will be of great use is the “cd” command which stands for change directory. This command will allow you to move around in the server and get to the desired location. Its use is as follows “cd /path/to/change” where the /path/to/change/ is actually the place you are trying to get to on the server. For example if you wanted to get to the /home/username directory you’d use the command “cd /home/username” and it will then change the directory to that current directory. A quick pwd will show the results.

By using the command “cd /home/use” and pressing tab it should autocomplete to /home/username automatically. If it does not, that means there are probably more than 1 match for /home/use and if you press tab again it will list all the folders in home that match username such as user username used or any other foldernames that begin with use. This can be very handy when trying to get to a primary directory.

ls
This command is the list command. This is one of the commands you will utilize most frequently. It does just as it sounds and lists the directories contents. When I run the ls command I always run “ls –alh”. The switch “a” stands for all which will list all files including those with a leading . meaning they are hidden files. The “l” switch lists the files one per line rather than a bunch horizontally. The “h” switch stands for human readable which will lists the file size in kilobytes and megabytes instead of bytes. For example instead of 4096 it will show 4.0K. The ls –alh command shows a lot of information. We will cover the permissions section in more depth below. The syntax of this command is “ls –alh” or “ls –alh /path/on/server” where you give it a folder/path you’d like listed out. I personally like to be in the folder and run the ls command.

stat
The stat command can be used to gain more detail about a file. Generally there will be much more detail than you will ever need. The handy use for this is if you are unsure what permissions a file has set simply run the command on a file. Its syntax is as follows, “stat filename”. It will provide you with a lot of information about the file that may or may not be helpful.

Permissions

Permissions are a broad topic, this is not comprehensive.

Dissecting the output of ‘ls –l’

Lots of data is presented with ‘ls –l’ – we’ll focus strictly on the owner/group and the impact that setting has on the permissions of the file. One of the drawbacks to using ls –l to determine the owner/group of a file is that it will only display the first 8 characters of the username and group name. If the names are longer than 8 characters, you must add –n to the options of ls (ls –ln) to see the userid and groupid in place of the username and group name.

UNIX File Permissions
Unix files have 9 “slots” to determine the permissions applied to that file, plus one extra on the left that tells you what type of file you’re looking at. Here’s the breakdown:

> *r = read*
> *w = write*
> *x = execute*

The file-type bit will usually be either a “d” or a “-“ (there are exceptions which are beyond the scope of this document).

If the object is a directory, the following considerations should be remembered:

> *Execute permission is required on directories. (Think of “execute” on a directory as the ability to use it).*
> *Removing the ‘read’ bit on a directory will prevent the associated entity from viewing the contents of a directory.*
> *The ‘write’ bit on a directory signifies the ability to add files to a directory, and delete files from a directory. If a person has ‘write’ permission, they can delete any file in that directory, regardless of whether or not they own it, or have write permission to that specific file.*

If the object is a file, the following considerations should be remembered:

> *Execute permission is required if it is a program.*
> *If the file is a script read permission is also required (unlike a binary executable). Remember, scripts are read in, then compiled, then run, so it’s critical that the user have permission to execute AND read the script.*
> *The write bit gives a user the ability to delete a file, rename a file, and modify a file.*

Changing Permissions:

chmod
chmod is the command used to modify permissions on a file or directory. The syntax is: ‘chmod xxx object’ (where xxx is the new permissions and object represents a file or directory). UNIX permissions are bit-masked to achieve the desired level of security. (Adding together the appropriate values achieves the final result). Here are the values used in UNIX permissions:

> *x (execute) = 1*
> *w (write) = 2*
> *r (read) = 4*

When you assign a file a specific permission setting, you give it three numbers (four is possible, that again, that is beyond the scope of this document). The first number represents the bitmasked permissions for the owner, the second represents the group, and the third represents the permissions for everyone else. Total permissions are achieved by adding the values above together (a value of 5 would be read+execute, a value of 7 would be read+write+execute).

```
[westhost]% chmod 644 filename.txt

[westhost]% ls –l filename.txt

-rw-r–r– 1 root root 0 Oct 11 16:06 filename.txt

[westhost]% chmod 771 filename.txt

[westhost]% ls –l filename.txt

-rwxrwx–x 1 root root 0 Oct 11 16:06 filename.txt
```

chown
The “chown” command is very similar to the chmod command. I want to note to take extra precaution before utilizing this command. Ownership is very important in Linux and if you mistakenly change the owner of a file you can cause it to not be accessible by anyone else. You utilize the chown command as follows, “chown username:groupname /home/USERDIRECTORY” where the username is what user you want to assign and groupname is the group you want to assign. There are certain directories such as .htpasswd and the public_html that have special ownership requirements and the group portion of it needs to be set to “nobody”. Please keep in mind anytime you are using the chown command always copy/paste the file names or utilize autocomplete to avoid any typos that may have server wide effects.

mv
mv is the Move command. This will allow you to move files from one location to another. It’s use is as follows, “mv file /path/to/new/home/”. The mv

command is also utilized to rename files. You rename a file by the following “mv filename newfilename” this will successfully rename the file “filename” to “newfilename”. When moving files be cautions of the files you are moving and where you are moving them to.

When moving files please always remember to have the trailing / if you are moving it into a directory so the system knows you are moving the file into that directory and not renaming it to that foldername.

cp
cp is the Copy command. This will allow you to copy a file from one location to another while leaving the file in its original location. Copy is a command that is meant to be utilized on a single file. When I utilize copy I almost always utilize it with the –a switch. The syntax is as follows, “cp –a filename /path/to/copy /”. The “a” switch is short for utilizing the “dpR” which will recurse and copy all files in a directory and the directory itself which is the “R” part of the switch, the “dp” is used to preserve attributes, timestaps, and symbolic links. This will leave the timestamps and ownership the way they are before the copy. With the “a” switch you can successfully copy files and directories with no problems.

find
Find is a very powerful command that can be utilized to do what its name suggests, find. The basic syntax of this command is “find /where/to/look -switch filename”. This is the find command in its very basic form. There are many different switches to this command, I will only discuss three of them in this training. The first of which is iname and is used like so, “find . -iname example.php”. This command says to look from my current location which is denoted by the . the switch is iname which indicates I am searching for the name of the file case insensitive and the file name I am looking for is example.php. Utilizing the iname switch it will find any files named example.php, Example.php, EXAmple.php and so forth.

The next switch is simply the name switch and it does the same thing as iname except it is case sensitive. The last switch that may be very helpful is the mtime switch. This is the last modified time. It is used like so, “find . -mtime +3” which will look in the current directory denoted by the . and search for all files and folders that were modified more than 3 days ago. The last thing you can do to add to the find command is utilize a switch named exec. This executes a system command on the information returned from find. It is used like this, “find . -iname test.php -exec rm -rf {} \;” which will find all files named test.php and the execute the remove function and delete the files.

rm
This is the remove command. It is another one of those commands that you need to be VERY careful when utilizing. You could very easily delete your entire account with a misplaced forward /. The remove command is utilized as follows, “rm filename”. This will not work on a directory, in order to remove a directory you need to utilize a different switch. “rm -rf directory” will remove the directory. You can also use the -rf switch for removing a simple file name. The “r” portion of the switch recurses into the directory and deletes all files and folders in that directory including the directory itself. The “f” switch stands for force which will remove all files ignoring any errors or warnings. Please be very very cautious once again when using this command.

touch
The touch command will allow you to create an empty file with the name provided. This is very handy when needing to create php.ini or .htaccess files. The command is used like this, “touch filename” and then if you perform an ls in that directory you will see the newly created empty file. Please note if you create this file as root it will be owned by root and you will need to utilize a chown command in order to change the ownership accordingly.

cat
The cat command will take a files contents and spit them out to the screen. The commands syntax is as follows, “cat filename”. This will display the entire file to the console window so be prepared to be overloaded with data if it is a large file. If you do not need to see the entire file a better command would be either head or tail which I will describe next.

head
The head command allows you to view the first x number of lines in a file. The default on the WestHost servers are 10 lines so if you run the command “head filename” you will get the first 10 lines of that file displayed in the terminal window. You can give it a switch with a number such as “-20” and it will show that many files, in this case 20.

tail
This command works exactly like head above. One very important feature of this command is the “-f” switch you can utilize. The “f” switch refers to follow and will allow you to follow all input into that file. This is extremely useful when you want to watch a mail log or an access log to see what happens when you send an e-mail or visit someones website. To get out of the follow you simply use the “ctrl + c” which will terminate the follow.

vi/pico
vi is an extremely powerful text editor provided in most linux installations. To use vi you simply type “vi filename” and it will then open the filename provided. If the file does not exist it will create it and open the file. vi is not completely straight forward and does not work how you would expect. With vi there are 2 different modes, command mode which is the mode you start in when first entering a file, and input mode which allows you to edit the file. To switch from command mode to insert/input mode you simply press the “i” key and then you will be allowed to edit the file. To get back to command mode simply press the escape key.

A few helpful commands to utilize within vi when in command mode are,

> *“:set nu” This command will allow you to show the line numbers in a file*
> *"dd" This command will delete the current line*
> *“5dd” This will delete the current line and the next 4 lines (a total of 5)*
> *“yy” This will copy a line*
> *“pp” This will paste the line.*
> *“0 Shift+G” this will take you to the end of the file*
> *“53 Shift+G” this will take you to the 53 line.*
> *“/searchterm” this will search for the specified searchterm. So if you did /hello it would find the first instance of the word hello. If you press “n” after searching it will go to the next instance of the word you are searching for.*
> *"Shift + ZZ" This will save and exit vi*
> *“:q!” This command will quit vi without saving and the ! is to tell the system you really want to exit without saving.*
> *“:wq” This command will write and quit, same as Shift + ZZ*

There are numerous other commands but these out to be enough for now. Keep in mind if you are using vi the most frustrating thing is getting used to command mode and insert mode. Esc takes you from insert back to command and “i” will take you from command to insert.

Pico is another editor which is much easier to use and has most of the same functionality. It however does not even get another word about it in this training besides how to use it, “pico filename”.

php
A few commands to help with php questions. You can utilize “php -v” to display information about the php version currently running on the server. “php -m” is another handy dandy command that allows you to view all the installed php modules on the server.

zip
To zip files into a .zip file you do the following. “zip nameofzipfile.zip folder” nameofzipfile.zip can be anything you want to name the zip file, however it does need to end in .zip. The folder can be a name of a file or folder. Using the zip command will leave the directory in place and make the .zip file. To unzip use the syntax “unzip zipfile.zip”. If the .zip file contains files that already exist it will prompt you if you want to replace the file, all files, none, or rename them.

gzip
gzip is similar to zip however it only works on single files. It does not gzip folders. To use gzip simply run the command like “gzip filename”. Using the gzip command will gzip the file and not leave the original file. You will then see a file named filename.gz which indicates it is gzipped. To unzip a gzipped file run the command “gunzip filename.gz”. When you gunzip if the file already exists it prompts you if you would like to overwrite the file. Also when you gunzip the .gz file will disappear and you’ll just be left with the file.

tar
This is the compression command you will be utilizing most of the time as it is the norm in Linux. The command is very simply when you break it down and look at it but it may appear confusing at first. To create a tar file run the command “tar -czvf zipfile.tar.gz foldername” and this will zip up the foldername into a zip file named zipfile.tar.gz. The tar command keeps all of the files on the server and also places a copy of them in the tar.gz file. You’ll notice the .gz extension on the end as the file is actually gzipped.

The switches associated with the tar command are as follows, the “c” tells the system to create a tar file, the “z” tells it to also gzip the .tar file when it is finished, the “v” tells it to do the zip verbose and show you what it is zipping (very hand for a lot of files), and the “f” is once again force. So after the command runs if you left of the “z” switch you’d have a .tar file but as the “z” switch is there it moves the .tar into a .gz file. To unzip a tar.gz file you use the same command as zipping it except an “x” for extract instead of the “c”. So it would look as follows, “tar -xvzf filename.tar.gz”. When you run this command it will output what it is extracting as the “v” switch is present.

****Keep in mind** with this command it WILL NOT prompt you to overwrite files it will simply do it. So PLEASE verify none of the files exist where you are extracting it so you do not overwrite important files. Also after extraction the .tar.gz file will still exist it does not magically disappear into the night.

whereis
This command can be utilized to find where certain applications are installed. It is used as follows, “whereis perl” and it will provide you a path to perl. This does not work for every installed application but it does work on most.