Mastering Remote Management with Linux Servers

Presented by the IEEE overlords:
James Oswald et Mahnoor Amir
Written by The Eboard at 3am last night:
James, Noor, Robin, Tony, Carolyn

Table of Contents

What is SSH Interfacing?

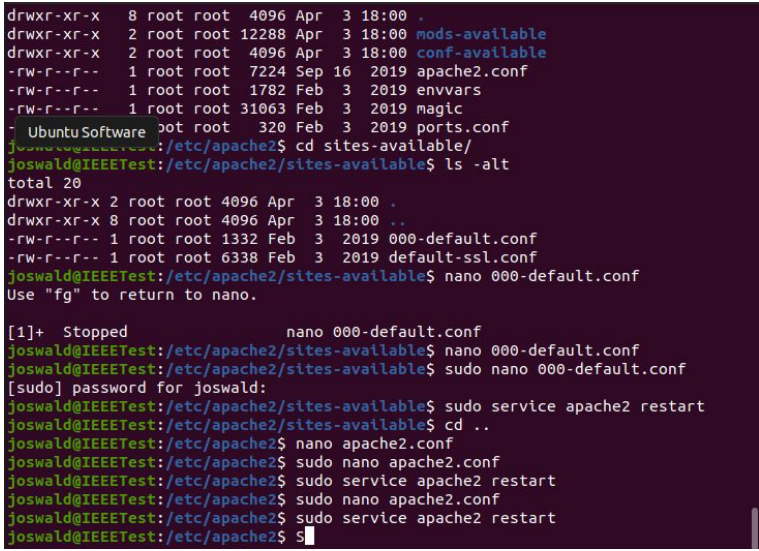
What is a Shell?

- A **Shell** is an interface for users to access programs and features provided by the computer's operating system. The shell itself is often a program provided by the OS.
- It is called a shell because it is the interface layer that sits between the OS and its users and their programs.
- **Command Line Interface (CLI) Shells** now also known as **Terminals** are the most common shells used by servers.
- Early OS's were characterized by a lack of GUI, relying solely on the shell for all interfacing with programs and features.

Examples Of Shells

- Each operating system offers its own unique shell, but the general concept of a CLI Shell / Terminal is the same across all OSs.
- Some Common OSs and their Shells:

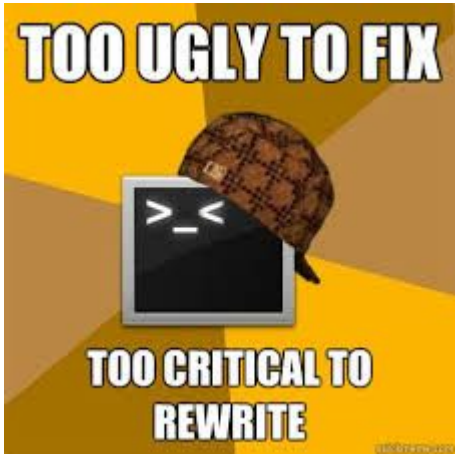
Operating System	Shell Name
Windows	cmd.exe
Linux	bash
Mac	Z shell



Picture of a Bash shell in action

Linux Server Shells

- Linux servers use a common shell known as **Bash.**
- Bash is a versatile shell based off of Unix more on how to use Bash later.
- Almost all servers lack a GUI and interface with users exclusively through a Shell, There are a number of reasons for this.



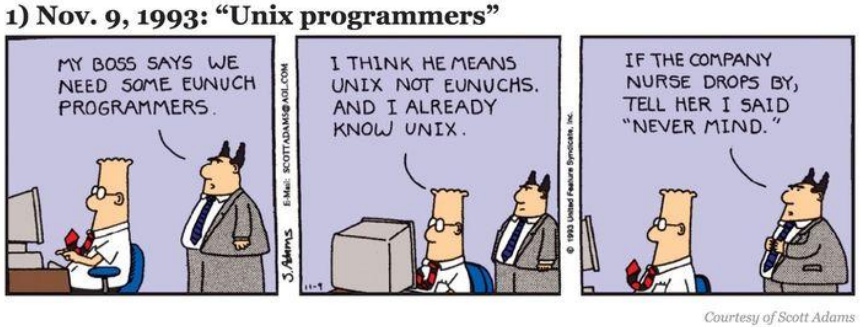
Meme provided by Sleepy Eboard

Five core reasons for foregoing a GUI:

- Reduces the amount of security vulnerabilities (reduced attack surface)
- Increases performance (less to hog CPU, Memory and Disk IOPS
- Reduces the chances of being affected by a software bug
- Reduces the amount of patching needed
- Reduces the disk space required

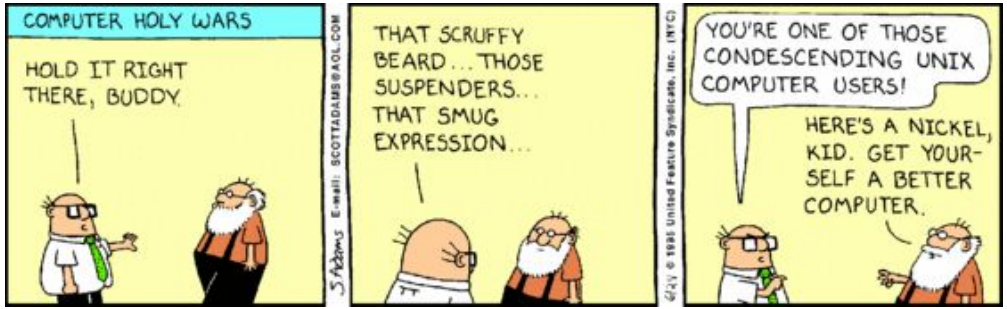
SSH (Secure Shell)

- SSH is a means of interfacing with a server’s shell from any internet connected device.
- SSH is a networking protocol, a set of standard rules that allow for communication and data sharing between two or more devices.
- SSH uses public-key cryptography to allow secure transfer of shell commands and files over unsecured networks.



SSH Clients

- A SSH client is a program that you run on your machine that provides you with remote shell access to a server using the SSH protocol.
- Both Windows and Mac come with ssh clients built in accessible from their shells
- Much like logging into a computer, Connecting with SSH requires you to select an account to log in with on the remote server, once you select an account you will be prompted to enter a password, if the account has one.



Connecting to A Linux Server

Connecting With A SSH Client

Windows

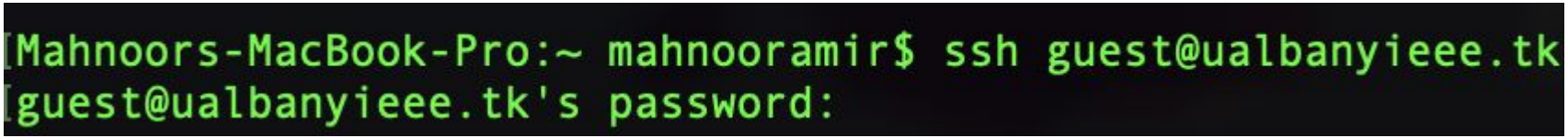
- Open command prompt
  - Type “cmd.exe” into the windows search bar and click it to run it.
  - OR windows+r, type “cmd” and run

Mac

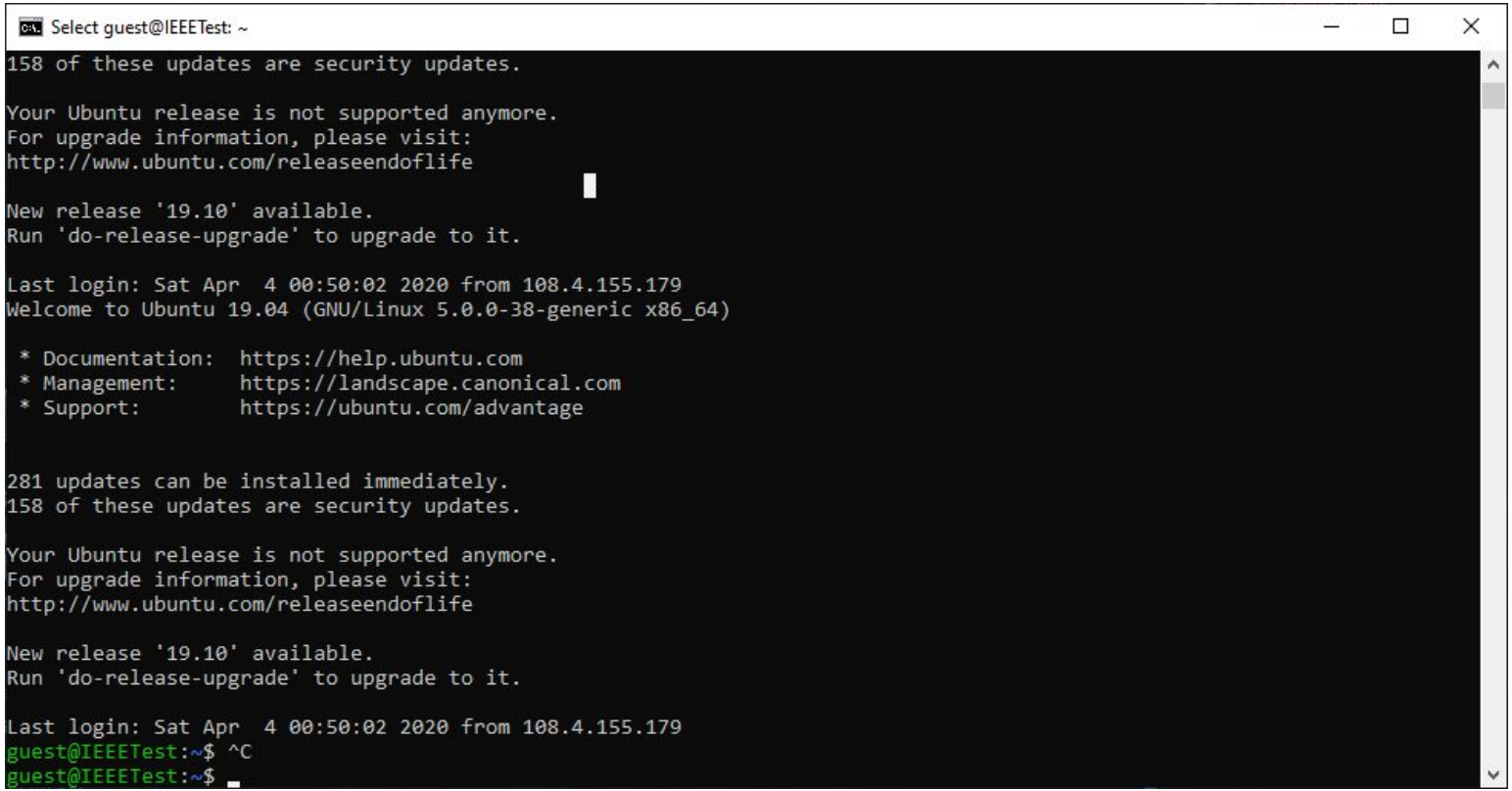
- Open terminal:
  - Open launchpad or cmd+space
  - Search terminal
  - Open terminal

For both operating systems the syntax to run the built in SSH client is: `ssh <username>@<domain>`
You will then be prompted to enter a password for the account.

To connect to the test server for this workshop: `ssh guest@ualbanyieee.tk`
You will be prompted for a password, the password is: `ieee`



Welcome To the Test Server!



File System Navigation and Manipulation

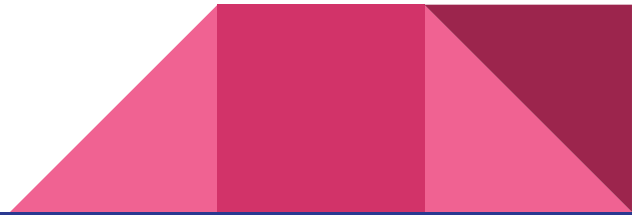
Starting off in a Bash Shell

Welcome to a Bash terminal!
The first thing you'll notice is the, `guest@IEEETest:~$`, This Is called **The Prompt.**
The prompt consists of 3 parts, the user, the **current working directory (cwd)**, and the command input

```
<username>@<servername>:<cwd>$ <command input>
```

Unlike modern OSs that use GUIs, in a CLI shell like bash, you will always be within the file system, and always inside of a directory. Navigating the filesystem and changing your current working directory is essential to managing a server.

Using Bash we run **Commands**. Commands are programs provided by the OS that let us interface with OS features, such as the file system.



Back to Basics

What you need to know to navigate and manipulate files systems:

1. Make a new directory/folder: `mkdir <foldername>`
2. Make a new file: `touch <filename and extension>`
3. Delete a file: `rm <filename>`
4. Delete a directory/folder: `rmdir <foldername>` or `rm - r <foldername>` (which deletes a folder and all its contents!)
5. Move a file: `mv source destination`
    a. `mv Downloads/example.txt Documents/example.txt`
    b. You can rename a file as you move it or you can keep it in place:
    c. `mv Downloads/example.txt Documents/love_unix.txt`

More linux/unix commands

1. Print working directory: `pwd`
2. Change directory to my folder: `cd myfolder` or `cd myfolder/next/next`
3. Go up to the parent folder: `cd ..`
4. `ls` : list files
5. `ls -lh` : list files with human-readable data sizes
6. `grep <pattern> <file>`
   a. E.g.: `grep Holden Catcher_in_the_Rye.txt`
   b. Tip: you can use regex for the pattern!
7. `cat filename` : prints an entire file

Try It Out!

Create a directory for yourself in ~:

```
guest@IEEETest:~$ mkdir myDirectory
```

Look at all of the crap in ~:

```
guest@IEEETest:~$ ls -alt
```

Go into the directory you just created:

```
guest@IEEETest:~$ cd myDirectory
guest@IEEETest:~/myDirectory$
```

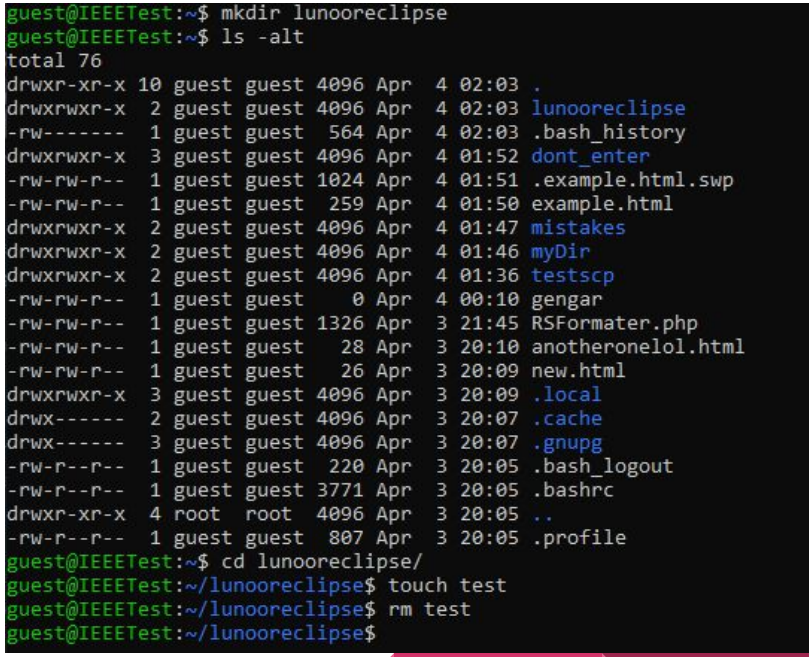
Create a useless file with nothing in it:

```
guest@IEEETest:~/myDirectory$ touch test
```

Delete it cause it’s useless:

```
guest@IEEETest:~/myDirectory$ rm test
```

Next up we’ll create a useful file



Creating and Editing Files, Text Editors Vim and Nano

What is Nano?

- Nano is a simple, easy-to-use command-line text editor.
- It’s without the learning curve of more advanced text editors, giving us access to an interface to concisely:
  - Open and create files
  - Copying and pasting
  - Searching for text
  - Save and exit

Using Nano (Create a file)

- Create and open a file
  - nano filename
- Open file path
  - nano /path/to/filename
- Save file
  - ctrl-o
- Exit file
  - Ctrl-x

Try It Out!

Lets create a small HTML File on the server.

Use nano to create a new file:

```
guest@IEEETest:~/myDirectory$ nano test.html
```

Write up a short html file,

```
<!DOCTYPE html>
<html>
<body>
<p> Hello Server! </p>
</body>
</html>
```

Save the File

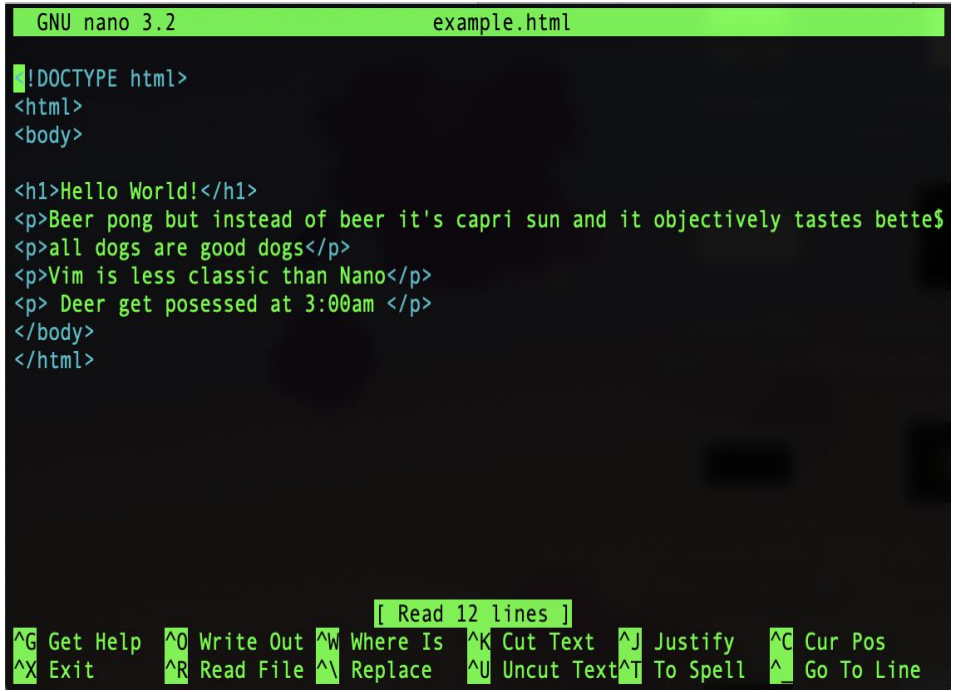
```
Ctrl-o
```

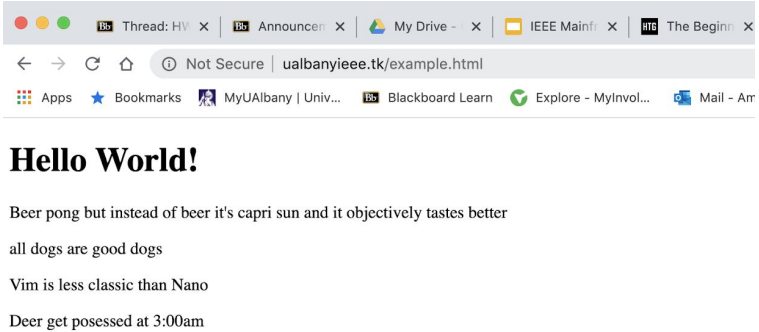
Exit nano

```
Ctrl-x
```

Take a look at your HTML file in a web browser, at http://UAlbanyIEEE.tk/, Navigate to your HTML file.

Hello Server!





What is Vim?

- Vim, like nano is text editor to be used in the command-line
- It can be used as a standalone application in a graphical user interface
- Vim is considered to be one of the best text editors of all time, being both extremely powerful and notoriously hard to use at the same time.

Creating a file with Vim

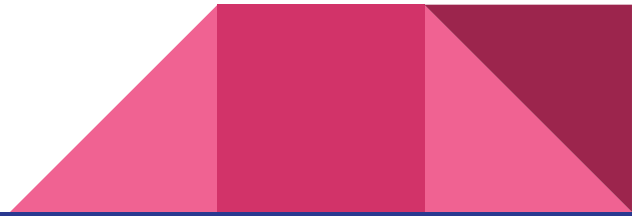
Command to create or open vim:

- vim filename.extension or vi filename.extension
- vim /path/to/filename or vi /path/to/filename

Vim has two modes: command mode and edit mode. Hit “a or i” to insert text. Hit “Esc” to to get back into command mode.

Basic vim command - :**q** quit vim :**w** save file : **wq** save file and quit

:**q!** quit vim without save



Additional Vim Commands

dd	Cut line	yy	yank(copy)	x	cut one letter
w	Jump to start of word	G	Go to end of file	gg	Go to start of file
p	paste	s	Delete one letter and change to insert mode	$	Go to end of line

Try it out!

Let's create and run a python program in vim!

Create and open a new file with vim:

```
guest@IEEETest:~/testscp$ vim helllo.py
```

Enter ~Insert~ mode by pressing "`i` or `a`"

```
robin = 5
tony = 2

if tony + robin == 7:
    print("Welcome to IEEE")
~
~
~
~
~
~
~
~
~
-- INSERT --
```

Write a Small Python program in Vim

←

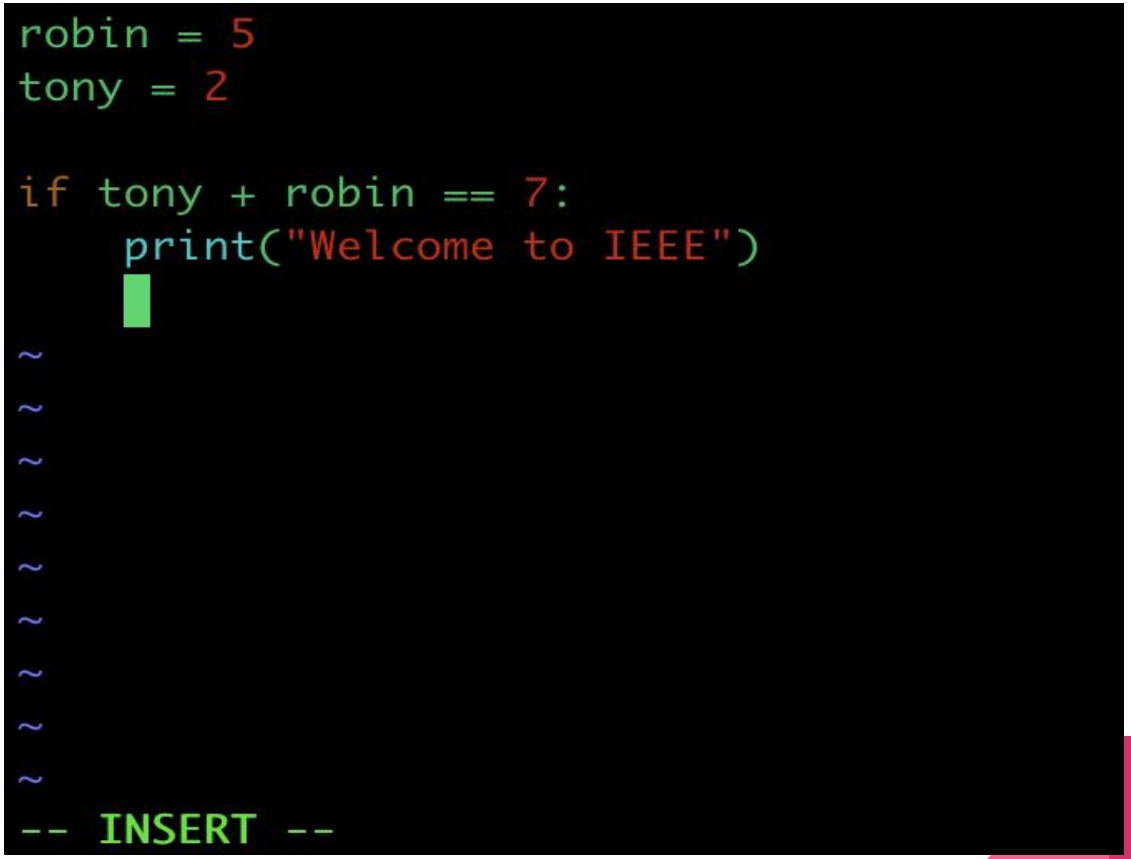
Enter command mode by pressing "esc"

Save the file with `:w`

Quit vim using `:q`

Test your program out my executing it from the terminal

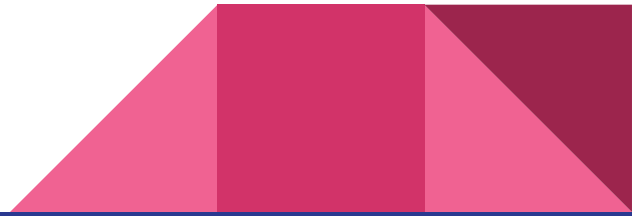
```
guest@IEEETest:~/testscp$ python3 helllo.py
Welcome to IEEE
```


robin = 5
tony = 2
if tony + robin == 7:
print("Welcome to IEEE")
-- INSERT --

Uploading Files with SCP

What is SCP

- SCP (Secure Copy Protocol) is a protocol based is SSH protocol to transfer files between machines.
- Like SSH, as a protocol SCP is a set of rules defining the data transfer format and security.
- SCP allows you to move any file from your host machine to a server
- SCP is analogous to more modern technologies like FTP and SFTP



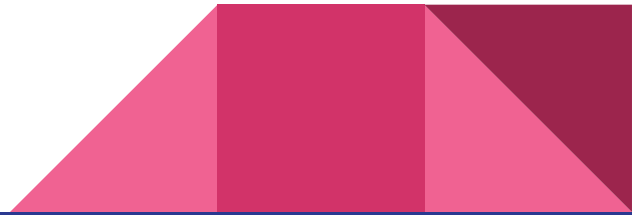
SCP Command Syntax

SCP [OPTION] [user@ source file] [user@ destination file]

Options :

-p	Preserves files modification and access times
-r	Use this option if you want to suppress the progress meter and no-error messages
-q	This option will tell scp to copy directories recursively

Example command : scp -p file.txt remote_user@ip_address:/directory/destination



Uploading Files with SCP

Create file using vim or nano

Type command:

scp /your directory/filename.extension guest@ualbanyieee.tk:/home/guest/testscp

Go to ssh client and check your file at the testscp directory

Check your file contents by opening file with vim or nano

To learn more about SCP type command “man scp”

Try It Out!

Lets add an image to our html file!

Start off by opening your terminal uploading your image to the server with scp

```
scp /path/meme.png guest@ualbanyieee.tk:/home/guest/myDir
```

Use nano or vim to add the following line into the body of your html file:

```
<img src=”meme.png”/>
```

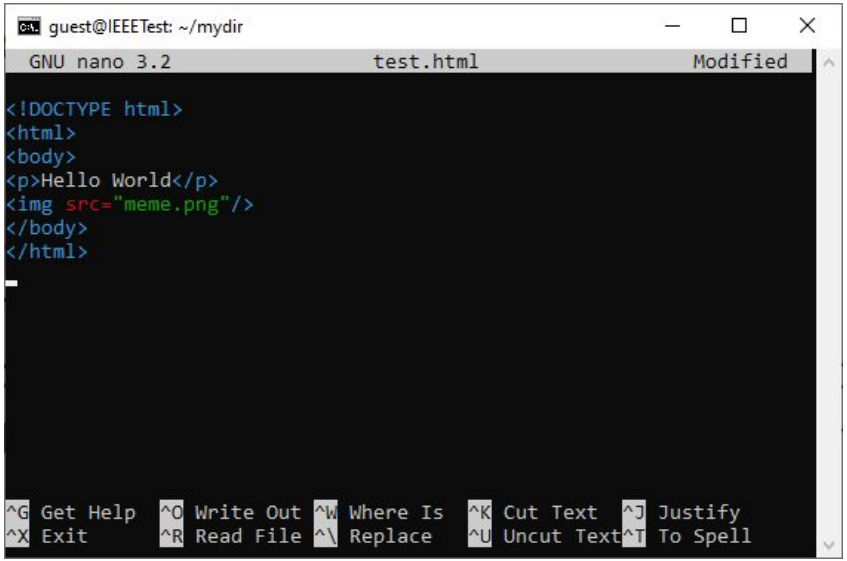
Save the html file.

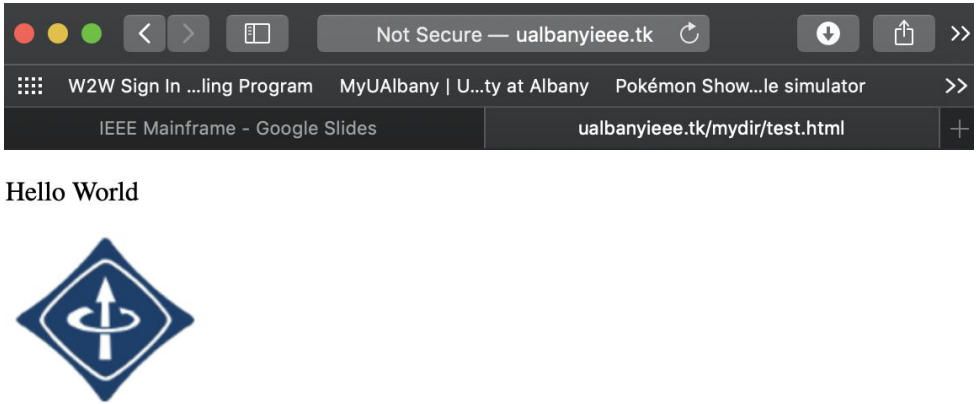
Take a look at your HTML file in a web browser, at http://UAlbanyIEEE.tk/,

Navigate to your HTML file. You should see the image you uploaded

SCP Try it out example

```
C:\Users\James>scp C:\Users\James\Pictures\meme.png guest@ualbanyieee.tk:/home/guest/myDir
guest@192.168.1.73's password:
meme.png                                      100% 7370     7.5MB/s   00:00
```





Thank You for Coming!

Feel free to hang out, play around, and ask questions!