

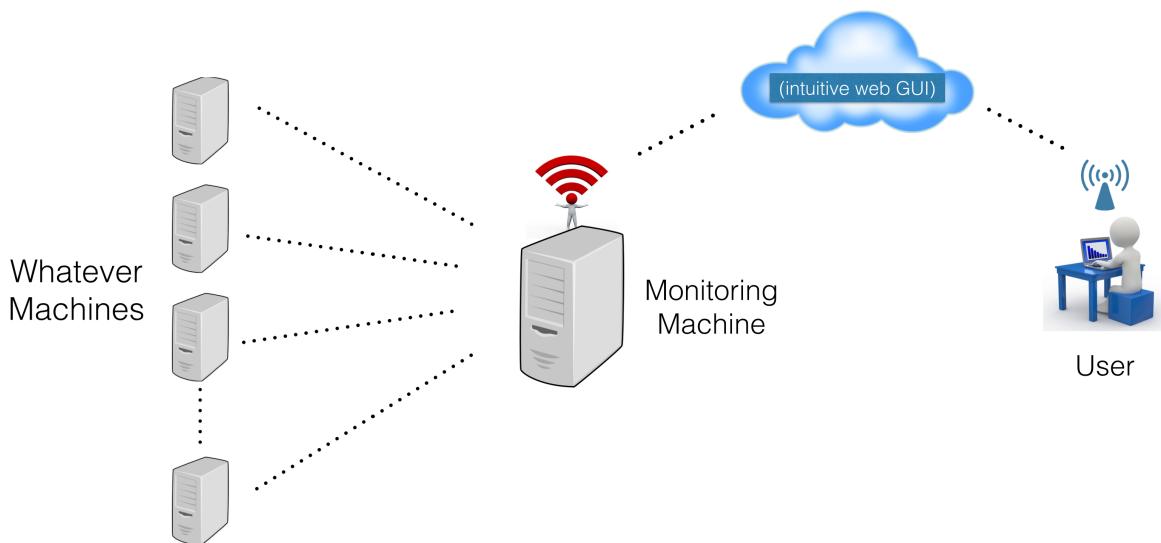
# CNLab Final Project Report

## Workload Distribution

Name	Work	Portion
林于智	環境建置、PHP model & controller、Python CGI scripts、Security(md5 hash + SSH key-gen)、HTML templates、Process Killing	27%
陳瀚浩	Python CGI scripts、User List、Mac OSX support、結報撰寫	23%
周書禾	CSS styling、結報撰寫	17%
黃柏智	CSS styling、結報撰寫	16%
江東峻	Python parsing、demo 簡報製作、動畫大師	12%
林鈺惟	idea 提供、其他資源提供	5%
<b>Total</b>		<b>100%</b>

## Intro

- An integrated service that provides user a more straightforward UI to monitor the status of their machines with intuitive designs.



## Goal

- Construct two normal virtual machines(VMs), say VM1 and VM2.
- Build a web server on another (the 3rd) VM to display all process info of VM1 and VM2.
- Users can kill processes of VM1 and VM2 through the admin page of web server.

## Environment

- For Ubuntu 14.04 LTS
- Configuration of the machines to be monitored:
  - Install SSH server
    - Install the SSH server with the following command:

- \$sudo apt-get install ssh
2. modify the config file for SSH server
    - Edit the file at /etc/ssh/sshd\_config:
      - \$sudo vim /etc/ssh/sshd\_config
    - Make sure the two following settings are set to “yes”:
      - StrictModes yes
      - PubkeyAuthentication yes
  3. run the SSH server
    - To check if SSH server is running:
      - \$sudo service ssh status
    - To start the SSH server:
      - \$sudo service ssh start
    - To stop the SSH server:
      - \$sudo service ssh stop
    - To restart the SSH server:
      - \$sudo service restart
  4. test SSH server
    - Use the following command on another machine to establish connection:
      - \$ssh -l [username] [hostname]
      - (You should be required to enter password and connect successfully.)
- Configuration of the machine to monitor:
    1. Install LAMP environment
      - To install Apache:
        - \$sudo apt-get install apache2
      - To install MySQL:
        - \$sudo apt-get install mysql-server
      - To install PHP:
        - \$sudo apt-get install php5 libapache2-mod-php5 php5-mcrypt
      - Restart Apache server:
        - \$sudo /etc/init.d/apache2 restart
        - (You can now check "<http://localhost/>" to see if your apache server works!)
    2. Install PHPMyAdmin
      - To install PHPMyAdmin:
        - \$sudo apt-get install phpmyadmin
        - (After restarting apache server by "\$sudo service apache2 restart", check out "<http://localhost/phpmyadmin>")
      - If the browser shows “404 phpmyadmin not found”, try:
        - \$sudo ln -s /usr/share/phpmyadmin /var/www/html/
    3. Install Python Package: Chilkat
      - Download Chilkat from the official site:
        - Download the proper version of Chilkat corresponding to python version.
        - Check python version by: "\$python --version".
      - Unpack the .tar.gz file:
        - \$tar zxvf chilkat-9.5.0-python-2.7-x86\_64-linux.tar.gz
      - Get into that directory:
        - \$cd chilkat-9.5.0-python-2.7-x86\_64-linux/
      - Put some files under site\_package directory of python:
        - \$sudo cp \_chilkat.so chilkat.py /usr/lib/python2.7/dist-packages/
        - Find “site\_package” directory by “>>>import site; site.getsitepackages()” in python.
    4. Configure Apache server to run python scripts
      - Enable Apache cgi:
        - \$sudo a2enmod cgi
      - Edit the conf file:
        - \$sudo vim /etc/apache2/sites-enabled/000-default.conf

- Add the following line after “DocumentRoot /var/www/html”:
  - “ScriptAlias /cgi-bin/ var/www/html/cgi-bin/“
  - Now we can put cgi codes inside “/var/www/html/cgi-bin”.

## Implementation

在實作的部分，總共主要分成三個層面分別是CSS Styling, PHP code, Python CGI script  
在最底層的Python CGI script裡面分別有

```
|---ssh_connect.cgi
|---ssh_kill.cgi
|---ssh_maconnect.cgi
|---ssh_userlist.cgi
|---ssh_userlist2.cgi
```

### **ssh\_connect.cgi :**

主要功能是建立一個ssh連線到已經加入的機器並列出機器(linux)的資訊，一開始會先從PHP網頁上使用form.getvalue()拿到使用者所選擇的表單資料主要會拿到三個資訊，有要求的process的形式、排序的方式、回傳的表單數量來讓cgi知道要回傳的方式。接著建立一個ssh連線到一台指定的機器上，在建立連線之後送出command : top -n 1 -b，再將回傳的資料做切割回傳一個整理過的table並以html的格式送回，以使用者要求方式排列，回傳的欄位會有以下幾個：

PID	User	State	CPU	MEM	TIME	Command
-----	------	-------	-----	-----	------	---------

### **ssh\_kill.cgi :**

主要功能是要在已建立的ssh中kill掉使用者擁有權限的process，連線一開始會先從PHP網頁上使用form.getvalue()拿到使用者的資料以及使用者當前所在的機器上想要kill的PID接著就會對該機器送出command : kill -9 PID，在資訊更新之後ssh\_connect.cgi會自動更新以變動後的表單。

### **ssh\_maconnect.cgi :**

主要功能是建立一個ssh連線到已經加入的機器並列出機器(mac os)的資訊，一開始會先從PHP網頁上使用form.getvalue()拿到使用者所選擇的表單資料主要會拿到三個資訊，有要求的process的形式、排序的方式、回傳的表單數量來讓cgi知道要回傳的方式。接著建立一個ssh連線到一台指定的機器上，在建立連線之後送出command : top -l 1 | tr -s ' ' | cut -d ' ' -f 1,2,3,4,8,13,28，再將回傳的資料做切割回傳一個整理過的table並以html的格式送回，以使用者要求方式排列，回傳的欄位會有以下幾個： PID User State CPU MEM TIME Command

### **ssh\_userlist2.cgi :**

主要功能是顯示在當前的機器上有哪些使用者以及使用者正在執行的狀態，連線一開始會從PHP網頁上使用form.getvalue()來知道使用者希望顯示出的資訊有哪些。接著建立ssh連線到指定的機器上，在建立連線後送出command : w -h | tr -s ' ' | cut -d ' ' -f 1,3,4,5,6,7,8,9，而在要輸出的格式中比較會有問題的地方是，在處理登入時間的部分，因為在登入時間顯示的格式上較為不一，經過一些處理後我們盡量將表單中的格式趨於統一，回傳的欄位會有以下幾個：

User	IP	Login Time	Idle time	What
------	----	------------	-----------	------

在中間那一層的PHP code裡面有

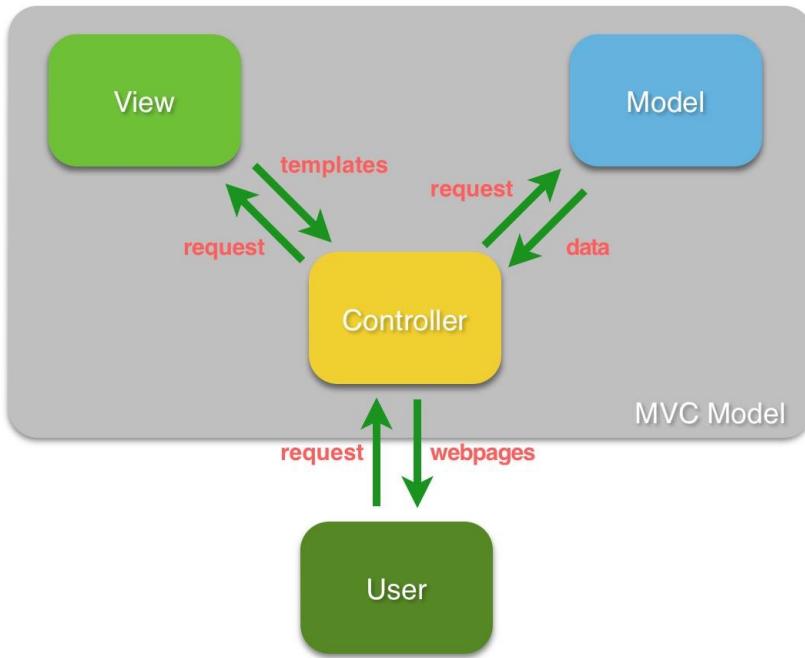
```
|---welcome.php
|---main.php
```

### **welcome.php :**

在這個php網頁中主要有兩個功能，一個是登入一個是註冊，在註冊時使用的要使用的ID必須在系統中是沒有重複的，而在密碼的上設定上，在我們一開始的計畫中是希望一開始會產生一組亂數密碼並且送到使用者註冊時所填寫的mail中來做認證的動作，使用者可以在之後更改成想要的密碼，但是在我們試著使用php中的內建mail函式來幫助我們發送e-mail，要做到這點得更改php.ini中的設定，將他的smtp設定到google以及csie的mail，因為我們的server主要是架設在virtualbox中因此php不讓我們透過這在NAT後方的虛擬IP來發送mail，若今天我們擁有的IP是架設在Himet的LAMP上的話就能夠成功的發送認證信，而要解決的方式可能要自己架設一個mail server，但礙於時間上的問題我們只能先將它列入future work，暫時無法實現，另一個部分則是登入的功能，在完成註冊之後就可以直接透過這個頁面進行登入。

**main.php :**

在main.php中有一個MVC的架構分別負責做不同面向的工作，一開始當User送出一個url的 webpage時會先透過Controller進行邏輯的判斷來先知道現在User在開的是哪一個頁面以及所要求的資訊，接著Controller會分別把資訊送往兩個不同的地方分別是負責data處理的Model以及負責頁面顯示的View，在Model中會對底層的python cgi送出要求，告訴他所需要的資料是什麼並接收回傳的資料，以及會從Data Base中取出當前的使用者有紀錄的資訊，最後回傳給Controller，在View是要提供網頁的模板給controller來render出User所送出的webpage所需的完整資訊，最後回傳給User。



◀main.php架構圖

在最上層的css styling中有

- |---welcome.css
- |---main.css

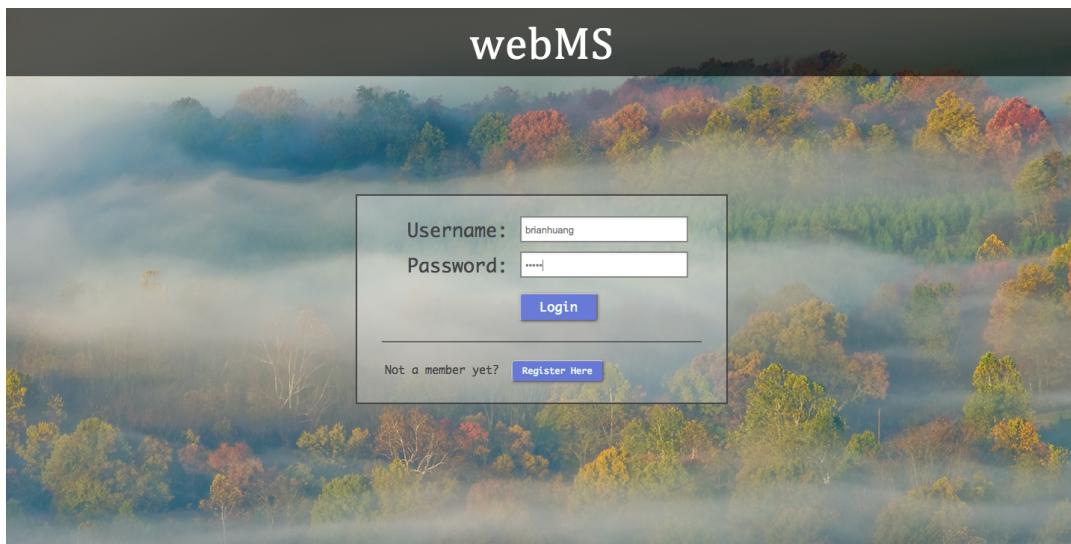
**設計理念：**

我們想在美觀的前提下，提供使用者最直觀的使用介面。首先，是背景的選擇。作為整個系統的門面，我們想帶給使用者一種放鬆的感覺，並且讓使用者印象深刻。因此，背景不能太花俏，也不能太單調，自然系的主題便十分合適。

接著，是字型的挑選。在顯示標題的文字上，我們挑選線條俐落的字體，用來凸顯其重要性，也讓使用者易於注目(包含Cambria, Athelas, Constantia)。在顯示重要資訊的文字上，我們選了一些優雅且清晰的字體，讓使用者在閱讀data時較為舒適(包含Courier, Monaco)。至於在顯示使用者姓名的字體上，我們挑了一個生動的字體，讓整個畫面嚴謹中不失活潑(我們使用Bradley Hand)。而錯誤訊息的字體同樣選用剛性字體，讓文字更有力道，更能吸引使用者的視線(使用Constantia)

再來，我們使用一些圖片來解釋功能鍵的設計。在每個頁面中，標題的下方會有使用者姓名，以及一組顯示所有頁面的button，其中所在頁面的button會以淺橘紅色表示，其餘的頁面以淺藍色代表。至於顯示主要內容的部份，不論是checkbox、select、input text、table還是button，最主要的目標都是清楚有條理，並盡量不讓文字太過擁擠。特別提一下table的部份，可以發現，滑鼠移動到的那行會變色，此設計能讓使用者更明瞭游標所在位置。

最後是錯誤訊息的設置，我們以紅色為主色，並把錯誤訊息顯示在畫面的中央地帶，讓使用者能立刻警覺，做出調整。



(圖一) 背景圖

**User Profile:**

Hello, brianhuang

**Navigation:**

- Status
- User List
- Settings
- Logout

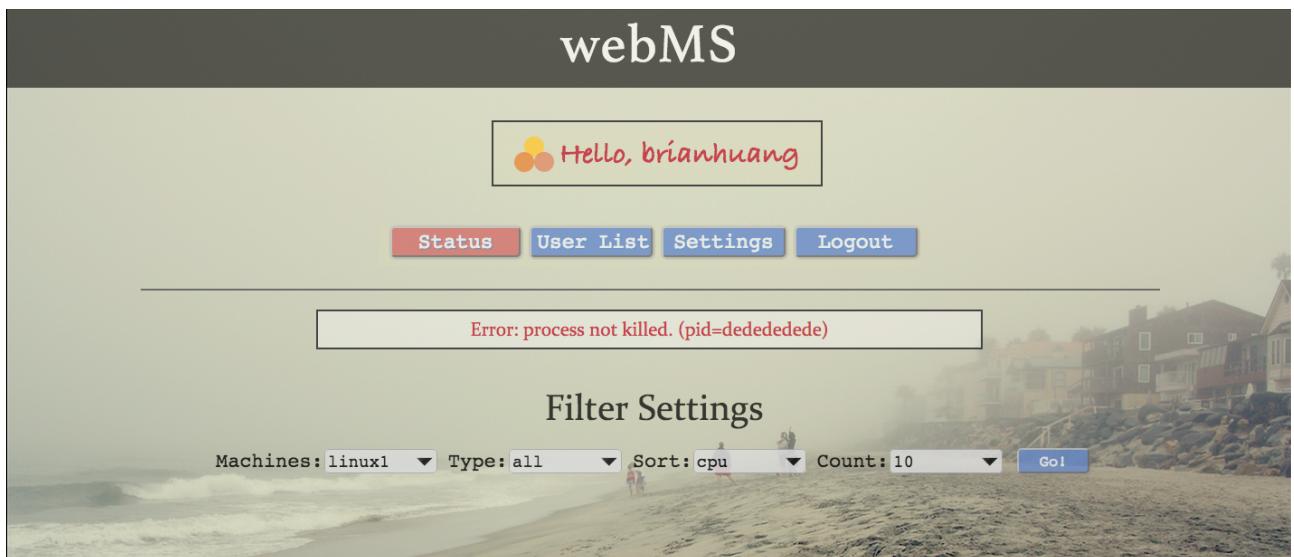
**Filter Settings:**

Machines: linux1 ▾ Type: all ▾ Sort: cpu ▾ Count: 10 ▾ Go!

(圖二) 功能鍵的設定、排版

PID	User	State	CPU	MEM	TIME	Command
24149	b93118	S	27.6	0.0	709.42	htop
30602	b019020+	R	22.1	0.0	0.06	top
8422	p039220+	S	5.5	0.1	6125.52	java
13903	msar	S	5.5	0.1	24276.37	MATLAB
19247	msar	S	5.5	0.1	37495.34	MATLAB
25097	r039221+	S	5.5	0.0	4472.92	MATLAB
1	root	S	0.0	0.0	747.43	systemd
2	root	S	0.0	0.0	4.19	kthreadd
3	root	S	0.0	0.0	1194.36	ksoftirqd/0
5	root	S	0.0	0.0	0.0	kworker/0+0

(圖三) 表格的設計



(圖四) 錯誤訊息

## Future work

---

### 1. MVC improvement :

在前面的main.php中可以看到我們是將MVC所有的東西凌亂的放在整個php中，之後希望可以做到分別將Model、View、Controller分別包成一個class方便往後的控制也能讓code變得比較易讀在debug上也會容易許多。

---

### 2. NodeJS & AngularJS :

我們現在的架構主要是由三個層級所構成分別是最上層的Css styling，中層的PHP code，以及最下層的Python CGI，但這方法在建構一個網頁上算是一個較為粗略的做法，往後可以透過NodeJS來取代掉PHP code、Python CGI、Server的建構，而最上層的Css styling則可以透過AngularJS來取代掉，這一來便可以用較為統一以及簡易的方式來完成建構。

---

### 3. mail server:

如同前面所提及的，我們的信箱認證系統並沒有被完整的實現出來，但這可以在我們改使用NodeJS的建構時來達到這目地，利用NodeJS來自建mail function這一來就不會被PHP中的內建mail function所阻擋，因此我們同樣可以將server架構在virtualbox上並同時達到認證信箱的目的。