Project Name: ADLI ~ Linux Admin Tool

Problem:

Imagine you are Linux administrator for a company. You are managing 10 machines. You want to write a Python program using modules like fabric or paramiko to find the usage of these machines. The information might include CPU load, memory used, available disk space, number of logged in users etc. The program must run hourly and should save the above information into a csv file, excel file or in a database. You might have to use modules like csv, xlwt, sqlite etc. If a query is made to find out load on a specific machine or the load on all the machines at a specific time, you should return the results.

Modules used:

- Fabric
- Itertools
- Matplotlib.pyplot
- DateTime
- Csv

Description:

ADLI is a software tool which helps linux administrator of a company to find the usage of these machines. The information might include CPU load, memory used, available disk space, number of logged in users etc. The program run hourly and save’s the above information into a csv file. Contain’s four main features:

1. Collect the data for all the listed machine
2. Collect the data for Single machine
3. Collect Data periodically
4. Graphical Representation

Python elements used in this project:

1. List data structure
2. Class
3. Function
4. External Module – Fabric,Matplotlib.pyplot
5. Itertools

6. File input and output

7. Try catch for handling exception

Commands to perform operation:

1. To run for one host (host supplied by user):
   fab per_host_call:hostnm="deepak-Vm"

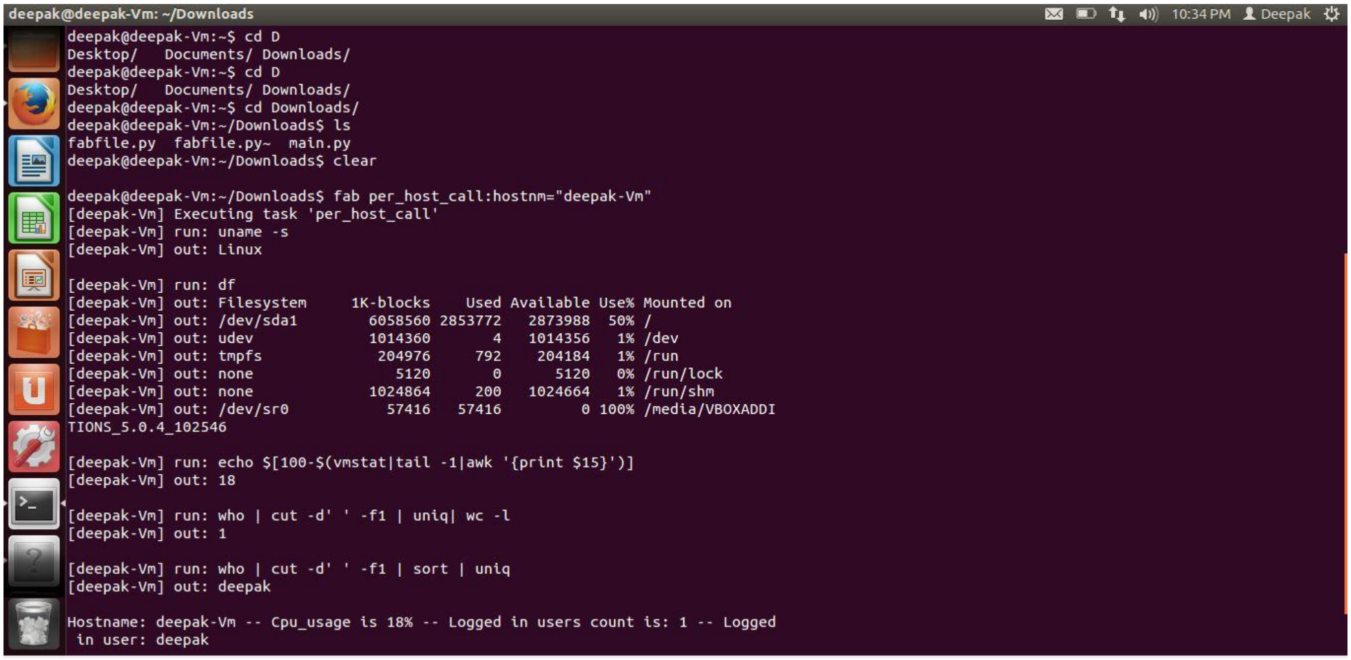
2. For Periodic (for every 1 hour) execution"
   python main.py

3. To run the CPU usage plot graph:
   fab plot_graph:hostnm="deepak-Vm"

4. To run the commands once for all the hosts:
   fab host_type

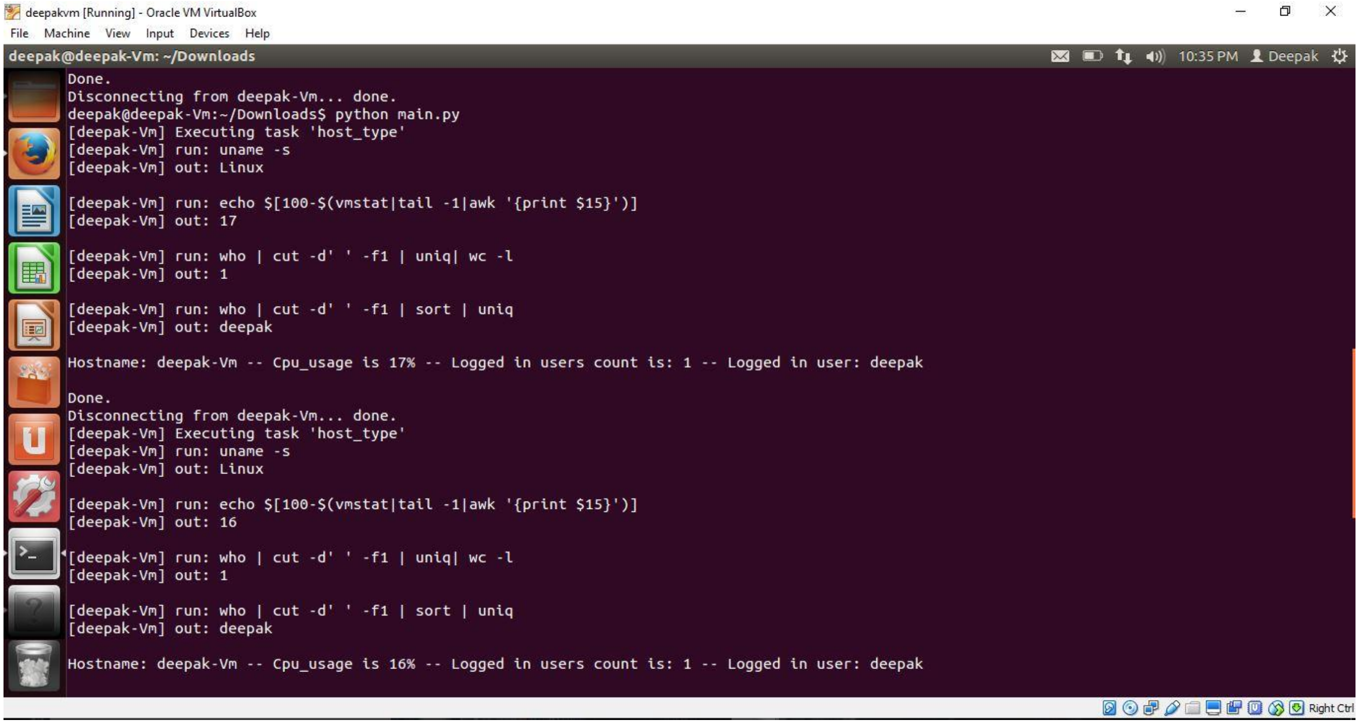
Screenshot:

/*Screenshot1 – Collects the user information and the cpu usage of single Hostname*/



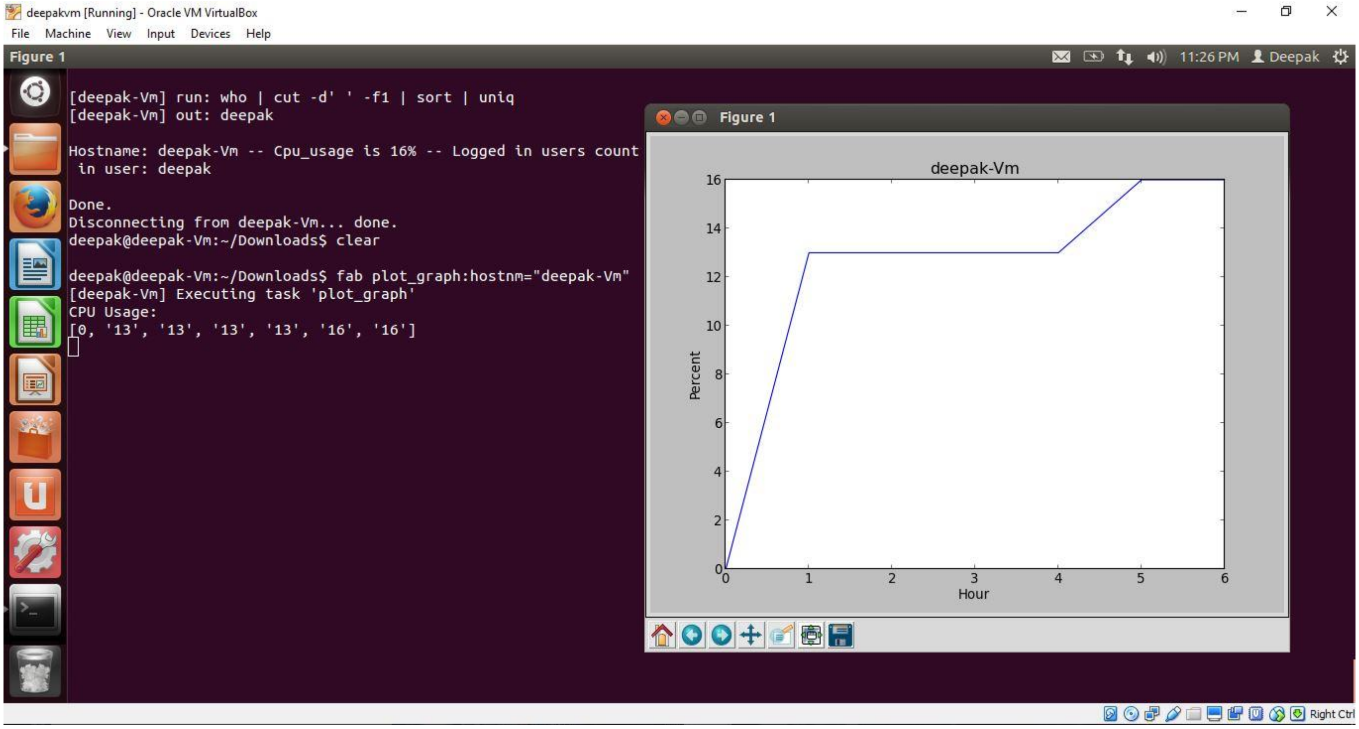
```
deepak@deepak-Vm:~$ cd D
Desktop/   Documents/ Downloads/
deepak@deepak-Vm:~$ cd D
Desktop/   Documents/ Downloads/
deepak@deepak-Vm:~$ cd Downloads/
deepak@deepak-Vm:~/Downloads$ ls
fabfile.py  fabfile.py~  main.py
deepak@deepak-Vm:~/Downloads$ clear

deepak@deepak-Vm:~/Downloads$ fab per_host_call:hostnm="deepak-Vm"
[deepak-Vm] Executing task 'per_host_call'
[deepak-Vm] run: uname -s
[deepak-Vm] out: Linux

[deepak-Vm] run: df
[deepak-Vm] out: Filesystem     1K-blocks    Used Available Use% Mounted on
[deepak-Vm] out: /dev/sda1        6058560 2853772   2873988  50% /
[deepak-Vm] out: udev             1014360       4   1014356   1% /dev
[deepak-Vm] out: tmpfs             204976     792    204184   1% /run
[deepak-Vm] out: none                5120       0      5120   0% /run/lock
[deepak-Vm] out: none             1024864     200   1024664   1% /run/shm
[deepak-Vm] out: /dev/sr0           57416   57416         0 100% /media/VBOXADDI
TIONS_5.0.4_102546

[deepak-Vm] run: echo $[100-$(vmstat|tail -1|awk '{print $15}')]
[deepak-Vm] out: 18

[deepak-Vm] run: who | cut -d' ' -f1 | uniq| wc -l
[deepak-Vm] out: 1

[deepak-Vm] run: who | cut -d' ' -f1 | sort | uniq
[deepak-Vm] out: deepak

Hostname: deepak-Vm -- Cpu_usage is 18% -- Logged in users count is: 1 -- Logged
 in user: deepak
```

/*Screenshot2- Scheduler which run's and collect the system usage and logged in user details for every 1 hour of all the defined systems*/


deepakvm [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
deepak@deepak-Vm: ~/Downloads
10:35 PM
Deepak
Right Ctrl

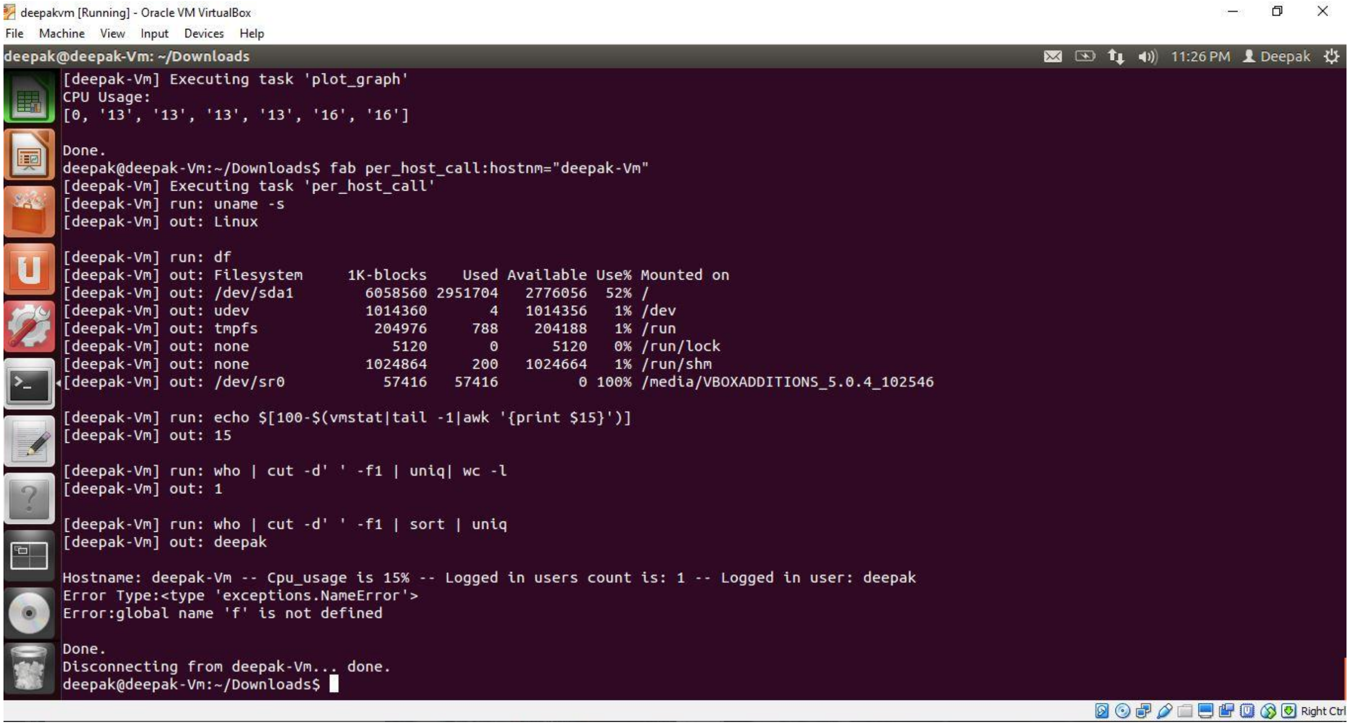
```
Done.
Disconnecting from deepak-Vm... done.
deepak@deepak-Vm:~/Downloads$ python main.py
[deepak-Vm] Executing task 'host_type'
[deepak-Vm] run: uname -s
[deepak-Vm] out: Linux

[deepak-Vm] run: echo $[100-$(vmstat|tail -1|awk '{print $15}')]
[deepak-Vm] out: 17

[deepak-Vm] run: who | cut -d' ' -f1 | uniq| wc -l
[deepak-Vm] out: 1

[deepak-Vm] run: who | cut -d' ' -f1 | sort | uniq
[deepak-Vm] out: deepak

Hostname: deepak-Vm -- Cpu_usage is 17% -- Logged in users count is: 1 -- Logged in user: deepak

Done.
Disconnecting from deepak-Vm... done.
[deepak-Vm] Executing task 'host_type'
[deepak-Vm] run: uname -s
[deepak-Vm] out: Linux

[deepak-Vm] run: echo $[100-$(vmstat|tail -1|awk '{print $15}')]
[deepak-Vm] out: 16

[deepak-Vm] run: who | cut -d' ' -f1 | uniq| wc -l
[deepak-Vm] out: 1

[deepak-Vm] run: who | cut -d' ' -f1 | sort | uniq
[deepak-Vm] out: deepak

Hostname: deepak-Vm -- Cpu_usage is 16% -- Logged in users count is: 1 -- Logged in user: deepak
```

/* Screenshot3 – Graphical Output about the CPU usage of single system*/



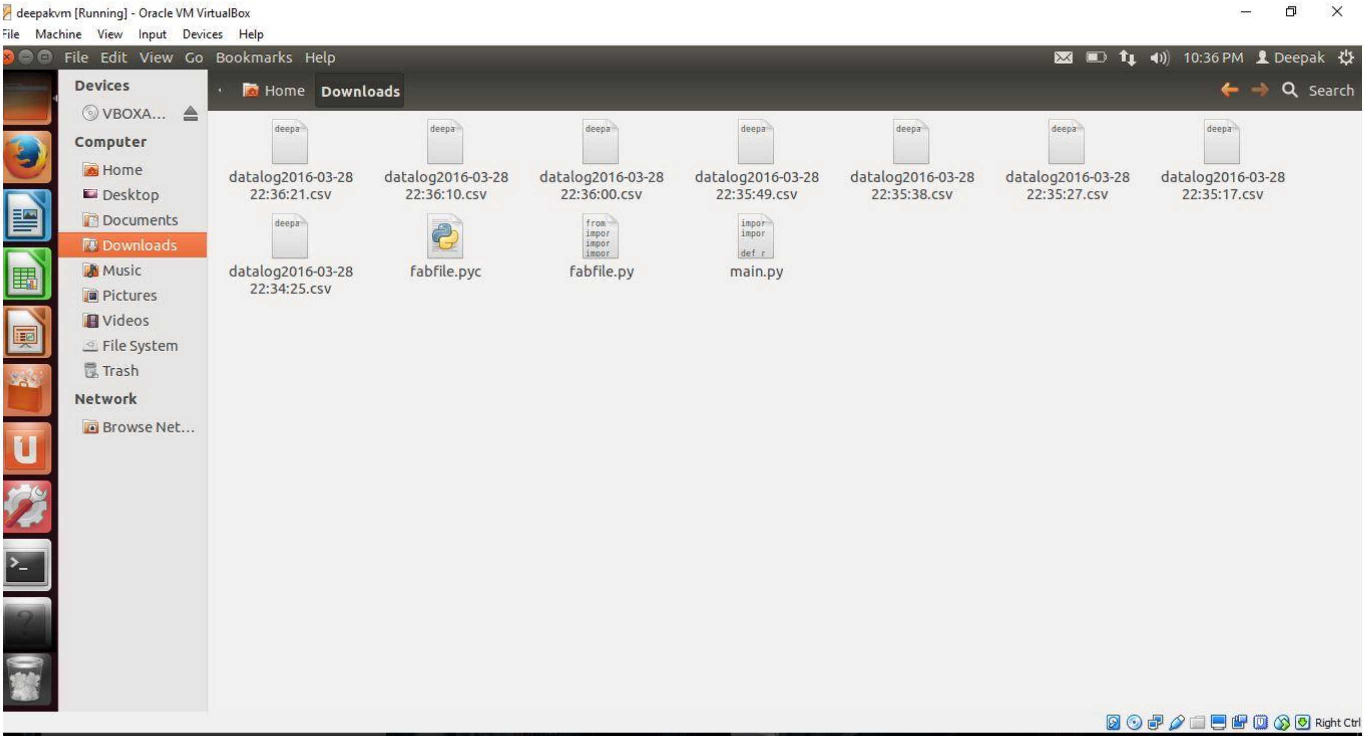
```
[deepak-Vm] run: who | cut -d' ' -f1 | sort | uniq
[deepak-Vm] out: deepak

Hostname: deepak-Vm -- Cpu_usage is 16% -- Logged in users count
 in user: deepak

Done.
Disconnecting from deepak-Vm... done.
deepak@deepak-Vm:~/Downloads$ clear

deepak@deepak-Vm:~/Downloads$ fab plot_graph:hostnm="deepak-Vm"
[deepak-Vm] Executing task 'plot_graph'
CPU Usage:
[0, '13', '13', '13', '13', '16', '16']
```

/Screenshot 4 – Error Handling*/


deepakvm [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
deepak@deepak-Vm: ~/Downloads
11:26 PM Deepak
Right Ctrl

```
[deepak-Vm] Executing task 'plot_graph'
CPU Usage:
[0, '13', '13', '13', '13', '16', '16']

Done.
deepak@deepak-Vm:~/Downloads$ fab per_host_call:hostnm="deepak-Vm"
[deepak-Vm] Executing task 'per_host_call'
[deepak-Vm] run: uname -s
[deepak-Vm] out: Linux

[deepak-Vm] run: df
[deepak-Vm] out: Filesystem     1K-blocks    Used Available Use% Mounted on
[deepak-Vm] out: /dev/sda1        6058560 2951704   2776056  52% /
[deepak-Vm] out: udev             1014360       4   1014356   1% /dev
[deepak-Vm] out: tmpfs             204976     788    204188   1% /run
[deepak-Vm] out: none                5120       0      5120   0% /run/lock
[deepak-Vm] out: none             1024864     200   1024664   1% /run/shm
[deepak-Vm] out: /dev/sr0           57416   57416         0 100% /media/VBOXADDITIONS_5.0.4_102546

[deepak-Vm] run: echo $[100-$(vmstat|tail -1|awk '{print $15}')]
[deepak-Vm] out: 15

[deepak-Vm] run: who | cut -d' ' -f1 | uniq| wc -l
[deepak-Vm] out: 1

[deepak-Vm] run: who | cut -d' ' -f1 | sort | uniq
[deepak-Vm] out: deepak

Hostname: deepak-Vm -- Cpu_usage is 15% -- Logged in users count is: 1 -- Logged in user: deepak
Error Type:<type 'exceptions.NameError'>
Error:global name 'f' is not defined

Done.
Disconnecting from deepak-Vm... done.
deepak@deepak-Vm:~/Downloads$
```

/Screenshot5- Datalog which get created when the Scheduler or on demand service is done*/


deepakvm [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
File Edit View Go Bookmarks Help
10:36 PM Deepak
Home
Downloads
Search
Devices
VBOXA...
Computer
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
File System
Trash
Network
Browse Net...
datalog2016-03-28 22:36:21.csv
datalog2016-03-28 22:36:10.csv
datalog2016-03-28 22:36:00.csv
datalog2016-03-28 22:35:49.csv
datalog2016-03-28 22:35:38.csv
datalog2016-03-28 22:35:27.csv
datalog2016-03-28 22:35:17.csv
datalog2016-03-28 22:34:25.csv
fabfile.pyc
fabfile.py
main.py
Right Ctrl

Conclusion:

Using fabric module able to successfully do ssh on all the listed machine and collect the user details and machine performance metrics. Plot functionality allows the user to plot the graph based on the result.

Program:

/*fabfile.py*/

```
from fabric.api import *
from itertools import *

import matplotlib.pyplot as plt
import datetime
import time
import csv
import sys

class MachineStats:
    cpu_load = 0
    num_logged = 0

env.user = 'deepak'
env.password = 'HelloWorld123'
#list of Hosts
department_qa=['deepak-Vm']
department_engineering=['deepak-Vm']
env.hosts = chain(department_qa,department_engineering)
def host_type():
    try:
        host_name= env.host
        print host_name
        run('uname -s')
        #CPU Usage
        cpu_usage = run("echo $[100-$(vmstat|tail -1|awk '{print $15}')]")
        cpu_usage
        #Number of Logged in Users
        logged_in_users_count = run ("who | cut -d' ' -f1 | uniq| wc -l")
        logged_in_users = run("who | cut -d' ' -f1 | sort | uniq")
        print "Hostname: " + host_name + " -- Cpu_usage is " + cpu_usage + "% -- Logged in users
count is: " + logged_in_users_count + " -- Logged in user: "+logged_in_users
        ts=time.time()
        time_stamp= datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d %H:%M:%S')
        with open('datalog'+time_stamp+'.csv', 'ab') as csvfile:
          spamwriter = csv.writer(csvfile, delimiter=',', quotechar='|',
quoting=csv.QUOTE_MINIMAL)
          data=[env.host,cpu_usage,logged_in_users_count,logged_in_users]
          spamwriter.writerow(data)
    except:
       error_type=sys.exc_info()[0]
       error=sys.exc_info()[1]
       print "Error Type:"+str(error_type)
       print "Error:"+str(error)

def per_host_call(hostnm):
    try:
       env.host = hostnm
       run('uname -s')
       #Disk Space
       disk_space = run ('df')
       #CPU Usage
       cpu_usage = run("echo $[100-$(vmstat|tail -1|awk '{print $15}')]")
       #Number of Logged in Users
       logged_in_users_count = run ("who | cut -d' ' -f1 | uniq| wc -l")
       logged_in_users = run("who | cut -d' ' -f1 | sort | uniq")
       print "Hostname: " + hostnm + " -- Cpu_usage is " + cpu_usage + "% -- Logged in users
count is: " + logged_in_users_count + " -- Logged in user: "+logged_in_users
       ts=time.time()
       time_stamp= datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d %H:%M:%S')
       with open(hostnm+'.csv', 'ab') as csvfile:
            spamwriter = csv.writer(csvfile, delimiter=',', quotechar='|',
quoting=csv.QUOTE_MINIMAL)
            data=[hostnm,cpu_usage,logged_in_users_count,logged_in_users]
            spamwriter.writerow(data)
    except:
       error_type=sys.exc_info()[0]
       error=sys.exc_info()[1]
       print "Error Type:"+str(error_type)
       print "Error:"+str(error)

def plot_graph(hostnm):
    try:
       cpu_usage=[0]
       with open(hostnm+'.csv', 'rb') as csvfile:
        spamreader = csv.reader(csvfile, delimiter=',', quotechar='|')
        for row in spamreader:
         cpu_usage.append(row[1])
        print "CPU Usage:"
        print cpu_usage
        plt.title(hostnm)
        plt.plot(cpu_usage)
        plt.ylabel('Percent')
        plt.xlabel('Hour')
        plt.show()
    except:
       error_type=sys.exc_info()[0]
       error=sys.exc_info()[1]
       print "Error Type:"+str(error_type)
       print "Error:"+str(error)
```

/*Main.py*/

```
import time, threading
import os
def run_periodically():
    os.system("fab host_type")
    threading.Timer(60*60, run_periodically).start()
run_periodically()
```