

Configuration Manual

MSc Research Project Cloud Computing

Siddhant Padmakar Kadam Student ID: x19219156

School of Computing National College of Ireland

Supervisor: Akeel Kazmi

National College of Ireland Project Submission Sheet School of Computing



Student Name:	Siddhant Padmakar Kadam	
Student ID:	x19219156	
Programme:	Cloud Computing	
Year:	2021	
Module:	MSc Research Project	
Supervisor:	Akeel Kazmi	
Submission Due Date:	31/01/2022	
Project Title:	Configuration Manual	
Word Count:	XXX	
Page Count:	8	

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

<u>ALL</u> internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	31st January 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

 Attach a completed copy of this sheet to each project (including multiple copies).
 □

 Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).
 □

 You must ensure that you retain a HARD COPY of the project, both for
 □

your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only			
Signature:			
Date:			
Penalty Applied (if applicable):			

Configuration Manual

Siddhant Padmakar Kadam x19219156

1 Introduction

This paper includes details as well as standard operating procedures, which are a collection of specific instructions that illustrate the step-by-step approach that must be followed in order to correctly execute the code and obtain the results.

1.1 Project Brief

With the use of several machine learning techniques, the study was conducted to address three research questions on green computing, resource pooling, and load balancing. The research's end objectives were achieved, and the evaluation of the simulations is reported in the research project.

2 General Requirements

- **OS** Microsoft Windows 10 Home
- Version 21H2
- **Processor** Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.11 GHz, 4 Core(s), 8 logical processor
- **RAM** 8Gb
- Storage 512Gb

2.1 Utilized Platforms

- Python- Version 3.8.10
- **flask** Flask framework is a Python API which is used to construct web apps.
- Oracle VirtualBox Version 6.1.22 r144080 (Qt5.6.2)
- Ubuntu Version 20.04
- Three Ubuntu VM's -1/1.5/1 CPU, Variable RAM size.

2.2 Libraries and Packages

- **NumPy** It's is among the most popular Python tools for numerical computation. The NumPy module handles numeric values and provides a strong object termed Array.
- Matplotlib- Matplotlib is among the most famous and earliest Python charting libraries being used Machine Learning. It aids in the understanding of a large quantity of data in machine learning by using various visual representations.

2.3 Pre-requisite

1. To use this document user should have some hand on experience with Unix command line, Oracle VirtualBox and bit of networking. With Current setup Nat Network is used for VM's to communicate with each other. Virtual box is installed on windows 10 machine. Ubuntu 20.04 is installed on three VM's.

Oracle VM VirtualBox Manager File Machine Help		- σ
Tools	New Settings Discard Show	
wbuntu 🕹 Running	B General Name: ubuntu Operating System: Ubuntu (64-bit)	E Presiex
Para Contraction C	System Base Memory: 2048 MB Boot Order: Floppy, Optical, Hard Disk Arreferation: (71-4000). Netter Basing, KVM P	
View Contraction C	Display Video Memory: 16 MB	
	Remote Deskako Server : Deskako Remote Deskako Server : Deskako Recording: Disabled	
	Controler: IDE IDE Secondary Device 0: [Optical Drive] V8oxGue Controler: SATA SATA Port 0: ubuntu.vdi (Normal, 40	stakdama ao (38, 12 MI) .00 GD
	(D Audio Host Driver: Windows DirectSound Controller: ICH AC97	
	Retwork Adapter 1: Intel PRO/1000 MT Desktop (NAT Netri Adapter 2: Intel PRO/1000 MT Desktop (Nat' Netri	vark, Nathelsnork) Adoptor, Virtualitov (Host-Orly Ethernet Adoptor)
	USB Controller: CHCI Device Filters: 0 (0 active)	
	Shared folders None	
	Description None	

Figure 1: Virtual box with Ubuntu VM's

ubuntu - Settings		? >
General	Network	
System	Adapter 1 Adapter 2 Adapter 3 Adapter 4	
Display	Enable Network Adapter	
Storage	Attached to: NAT Network	•
Audio	Name: NatNetwork	•
Network		
Serial Ports		
USB		

Figure 2: Virtual box with Ubuntu VM's Network

2. Install OpenSSh server

siddhant@siddhant-V3:~\$ sudo apt install openssh-server
[sudo] password for siddhant:
Reading package lists... Done
Building dependency tree

Figure 3: Install openSSH

3. Update /etc/hosts entries on each server.

```
siddhant@siddhant-V3:~$ cat /etc/hosts|tail -3
172.25.1.4 testserver
172.25.1.5 siddhant
172.25.1.6 siddhant-V3
```



3 Implementation Process

- 1. Run sudo apt update and upgrade commands on each server/VM.
- 2. Install pip on each server pip is just a Python package-management tool for installing as well as managing software packages.

```
siddhant@siddhant-V3:~$ sudo apt install python3-pip
[sudo] password for siddhant:
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

Figure 5: pip Installation

3. Install required libraries or packages

siddhant@siddhant-V3:~\$ pip install -U flask matplotlib numpy
Collecting flask
Downloading Flask-2.0.2-py3-none-any.whl (95 kB)
95 kB 877 kB/s
Collecting matplotlib
Downloading matplotlib-3.5.1-cp38-cp38-manylinux_2_5_x86_64.manylinux1_x86_64.whl (11.3 MB)
11.3 MB 11 kB/s
Collecting numpy
Downloading numpy-1.21.4-cp38-cp38-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (15.7 MB)
15.7 MB 35 kB/s
Collecting click>=7.1.2

Figure 6: Installation of additional packages/Libraries

4. Navigate using cd command to the location where applications are stored.



Figure 7: Navigate to the code location

5. Copy all the files from one server to other by using scp command as shown below.

<pre>siddhant@testserver:~/Downloads\$ scp - f:/beme/ciddhant/Decktop</pre>	pr 'Final se	em proje	ect' siddhau	nt@172.25.	.1.
siddhant0172 25 1 6's password					
olot utility py	100%	1722	174 9KB/s	00.00	
	100%	2826	1 1MR/s	00:00	
delayed plot.png	100%	14KB	5.9MB/S	00:00	
idle plot.png	100%	12KB	10.2MB/S	00:00	
capacity plot.png	100%	19KB	3.4MB/s	00:00	
lead alst and	100%	2010	4 040 /0	00.00	

Figure 8: SCP command syntax

6. Run the Cloud Server with round robin algorithm on one of the VM , a number of machines can connect to it and contribute their computing power, and it is also linked to the resource monitoring component. Same process need to be followed for ESCE algorithm after stopping running server/applications.

sic *	ddhant@siddhant-VirtualBox:~/Downloads/Final sem project\$ python3 server_app.py 7000 RR Serving Flask app 'server_app' (lazy loading)
	Environment: production
	Use a production WSGT server instead
-	
*	Debug mode: off
*	Running on all addresses.
	MARNING: This is a dayalopment corver. Do not use it is a production deployment
1000	wakaind. This is a development server, bo not use it in a production deployment.
*	Running on http://172.25.1.5:7000/ (Press CTRL+C to quit)

Figure 9: Cloud Server with RR

7. Run the resource nodes on the next VMs, these resource nodes contributes to resource pool through Cloud server. To run multiple resource we need to follow the same process on next VM for demo purpose we can use new terminal and run it on different port as well. Follow the same process to create desired number of resource nodes.

si	ddhant@testserver:~/Downloads/Final sem project\$ python3 processor_app.py 9000
*	Serving Flask app 'processor_app' (lazy loading)
*	Environment: production
	Use a production WSGI server instead.
*	Debug mode: off
*	Running on all addresses.
	WARNING: This is a development server. Do not use it in a production deployment.
*	Running on http://172.25.1.4:9000/ (Press CTRL+C to quit)

Figure 10: Run Resource Node

8. Run the client server; this machine will generate load, which will be governed by the load balancing method chosen when the cloud server was launched. The servers are linked by the local LAN network.



Figure 11: Run Client Node

- 9. Registration of resource nodes on cloud server To register resources open the url for every processor in browser, and enter the url http://IPofCludServer: portnumber/processor/register/
- 10. In this case the url is http://172.25.1.5:7000/processor/register/ , follow the same process to register resource nodes on cloud server, enter the capacity and click on Connect/Update.

$\leftarrow \ \rightarrow \ G$	0 👌 172.25.1.4:9000	
Processor A	Арр	
URL:		
'2.25.1.5:7000/processor/	r/register/	
Charles and the second second		
capacity:		

Figure 12: Resource Registration

11. Now we can check the servers connected to the cloud server by using url http: //cloudserverIP:portNumber/info/ , in this scenario url is http://172.25.1. 5/info/

172.25.1.4:8000/ ×	172.25.1.4:9000/	×	172.25.1.6:10000/		
$\leftrightarrow \rightarrow C$	0 👌 172.25.1.5:70	000/ir	nfo/		
Server App					
http://172.25.1.4:8000/ad	ld/load/ 10.0				
<u>create graphs</u>					
http://172.25.1.4:9000/ad	ld/load/ 20.0				
<u>create graphs</u>					
http://172.25.1.6:10000/a	dd/load/ 30.0				
create graphs					
http://172.25.1.6:11000/add/load/ 40.0					
create graphs					
http://172.25.1.5:12000/add/load/ 50.0					
<u>create graphs</u>					

Figure 13: Connected Resource Information

12. To check performance of algorithm we need to create a load on system for which open link of client server http://ClientServerIP:port/ in browser in this case it is go to link of client http://172.25.1.5:13000/, set the add url to http:

//CloudServerIP:port/add/load/ in this case it is set to http://172.25.1.5: 7000/add/load/

172.25.1.4:8000/	×	172.25.1.4:9000/	×	172.25.1.6:10000/	×	172.25.1.6:11000/	×
$\leftarrow \ \rightarrow \ G$		0 172.25.1.5	:13000				

http://172.25.1.5:700	0/add/load/
min load:	0,000,000
145	\$

Connect/Update stop client connected to the server

Figure 14: Load Generation

13. Plot the graph by clicking on create graphs 13 which will get saved on path where code is saved on system.

siddhant@si	.do	dhant-Vir	tualBox:~/	Downlo	pads,	/Fi	nal sem	project\$ ls -lrt
total 156								
drwxrwxr-x	5	siddhant	siddhant	4096	0ct	28	15:44	templates
- rw- rw- r	1	siddhant	siddhant	2075	0ct	31	15:31	client_app.py
- FW- FW- F	1	siddhant	siddhant	4330	0ct	31	15:33	server_app.py
- FW- FW- F	1	siddhant	siddhant	1722	0ct	31	15:37	plot_utility.py
drwxrwxr-x	2	siddhant	siddhant	4096	0ct	31	15:39	round_robin_graphs
- rw- rw- r	1	siddhant	siddhant	2826	0ct	31	15:41	processor_app.py
drwxrwxr-x	2	siddhant	siddhant	4096	0ct	31	15:43	esce_algorithm
drwxrwxr-x	2	siddhant	siddhant	4096	Dec	8	14:42	pycache
drwxrwxr-x	2	siddhant	siddhant	4096	Dec	9	01:29	'RR plots'
drwxrwxr-x	2	siddhant	siddhant	4096	Dec	10	11:25	'ES graph'
- FW	1	siddhant	siddhant	527	Dec	14	22:43	nohup.out
- FW- FW- F	1	siddhant	siddhant	957	Dec	14	23:28	readme.txt
- FW- FW- F	1	siddhant	siddhant	27862	Dec	15	00:59	capacity_plot.png
- FW- FW- F	1	siddhant	siddhant	26861	Dec	15	00:59	load_plot.png
- FW- FW- F	1	siddhant	siddhant	16180	Dec	15	00:59	idle_plot.png
- FW- FW- F	1	siddhant	siddhant	15941	Dec	15	00:59	delayed_plot.png
- FW- FW- F	1	siddhant	siddhant	14201	Dec	15	00:59	processed_plot.png
siddhant@si	.do	dhant-Virt	tualBox:~/	Downlo	oads,	/Fi	nal sem	project\$

Figure 15: location of Graphs created

- 14. Stop all the applications on servers.
- 15. Run Cloud server with ESCE algorithm (change RR to ESCE in command) and follow the steps 6 to 13.
- 16. generate and compare the graphs to evaluate performance.

17. Use WinSCP to get graphs copied on windows system, Its a tool used to connect to server and local machine using SFTP protocol. Login to the server using credentials, navigate to the directories on local machine and VM where graphs are supposed to be copied and from the location. Else if Ubuntu desktop on VM installed we can view and compare them on VM itself.

WinSCP	-	
Local Mark Files Commands Ses	ssion Options Remote Help	
🗰 😤 🖨 Synchronize 🔄 🖉	Transfer Settings Default	
New Sersion		
		Files
C:\Users\91883\Desktop\3rd sem\	Session	
Name	Mew Size File restoral	
t	Local ubuntu Gerro	
finl project	ubuntu@ec2-63-33-198-38.eu-west-1.comp	
Manual	Ubuntu@ec2-63-35-227-136.eu-west-1.com Host name: Port number:	
project	192.168.56.200 22	
Sid final sem projectreport		
22786 5.ipg	User name: Password:	
Cover page CA.docx		
A DG.pdf	Edit Advanced w	
DG1JPG (1).ipg		
DG1.JPG (2).ipg		
DG1.IPG.ing		
evaluation docx		
finl project zin		
A Optimizing the application pe		
PPT Siddhant.pptx		
Reprojectkey.ppk		
Research Project Configurat	g Tools V Manage V Login V Close Help	
Sid final sem project zip	Show Login dialog on startup and when the last session is closed	
Sid.pem	2 KB PFM File 14-12-2021 21	
at sid.ppk	2 KB PuTTY Priv., 14-12-2021 21	
Siddhant Load Balancing usin	148. Adobe Acr., 02-12-2021 14	
SidfinalPro.pem	2 KB PEM File 14-12-2021 17	
a cidEinalaro ank	2 KR DITTV Drin 1/4-12-2021 17	

Figure 16: WinSCP interface