

Assignment 1:

Assignment 1 and 2 (refer same theory)

What is Google App Engine?

Google App Engine (GAE) is a platform-as-a-service product that provides web app developers and enterprises with access to Google's scalable hosting and tier 1 internet service. GAE requires that applications be written in Java or Python, store data in Google Bigtable and use the Google query language. Noncompliant applications require modification to use GAE. GAE provides more infrastructure than other scalable hosting services, such as Amazon Elastic Compute Cloud (EC2). GAE also eliminates some system administration and development tasks to make writing scalable applications easier

What are GAE's key features?

Key features of GAE include the following:

API selection: GAE has several built-in APIs

- 1) Managed infrastructure: Google manages the back-end infrastructure for users. This approach makes GAE a serverless platform and simplifies API management.
- 2) Several programming languages: GAE supports a number of languages, including GO, PHP, Java, Python, NodeJS, .NET and Ruby. It also supports custom runtimes.
- 3) Security features: GAE enables users to define access policies with the GAE firewall and managed Secure Sockets Layer/Transport Layer Security certificates for free.
- 4) Traffic splitting: GAE lets users route requests to different application versions

-> hey, How are you?

ctrl + X and save

Some Commands for Linux Based Distro: **Installation of Google App Engine –**

The Google Cloud CLI works on Windows 8.1 and later and Windows Server 2012 and later.

1. download google cloud sdk installer
2. install it
3. open cloud sdk
4. run 'gcloud init'
5. create new configuration
6. (test.py file)
7. (app.yaml file)
8. save test.py and app.yaml file in same folder
9. run, 'py google-cloud-sdk\bin\dev_appserver.py <folder path where test and app.yaml files are save

Assignment 3:

CloudSim is an open-source framework, which is used to simulate cloud computing infrastructure and services. It is developed by the CLOUDS Lab organization and is written entirely in Java. It is used for modelling and simulating a cloud computing environment as a means for evaluating a hypothesis prior to software development in order to reproduce tests and results.

For example, if you were to deploy an application or a website on the cloud and wanted to test the services and load that your product can handle and also tune its performance to overcome bottlenecks before risking deployment, then such evaluations could be performed by simply coding a simulation of that environment with the help of various flexible and scalable classes provided by the CloudSim package, free of cost.

Following are the benefits of CloudSim:

- No capital investment involved. With a simulation tool like CloudSim there is no installation or maintenance cost.
- Easy to use and Scalable. You can change the requirements such as adding or deleting resources by changing just a few lines of code.

Scheduling in Cloud

As cloud computing is the virtualized operating environment, and the virtual machines are the primary computing component which is responsible for the execution of the workloads(tasks). The virtual machine(s) are powered by a physical server host machine (i.e.) hardware. Depending on the requirement of the Virtual Machine(VM) there could be 'one to one' or 'many to one' mapping between the VM and host machine. That means in cloud computing the scheduling is done at both the mapping levels that are:

- Virtual Machine to Host Machines
- Tasks to Virtual Machines

Scheduling in Cloudsim

The Cloudsim simulation toolkit framework has effectively addressed the Scheduling scenario and implemented it as a set of the programmable class hierarchies with parent class as:

1. VmScheduler
2. CloudletScheduler

Installation:

The basic requirement to configure CloudSim as follows:

1. Download

a) CloudSim(3.0.3 or 4.0) installer from

<https://github.com/Cloudslab/cloudsim/releases>

b) Eclipse IDE installer for java developers from

<http://www.eclipse.org/downloads>

c) Java Development Kit(1.6 or 1.7) if it is not available on your desktop or laptop from

<https://www.java.com>

2. Extract

a) CloudSim Installer

b) Eclipse installer

3. Open Eclipse and create a new Java Project:

File -> New-> Java Project -> Write Project Name -> Use execution environment JRE ->

Next-> JRE System Library -> Add External file where you extracted the of cloudsim 3.0.3 or 4.0 -> Finish

4. Import CloudSim project into new java project.

Assignment 4:

What is a virtual machine?

Virtual machine is an application providing a platform independent programming runtime that allows applications to execute in the same manner on different platforms. The virtual machine acts as a bridge to the real environment, hiding the details of operating system. Do not confuse this term with system virtual machines, such as VMware, Virtual server, Xen which enables one to run multiple OS on a single piece of hardware.

Types of virtualization techniques:

1. Guest Operating system virtualization
2. Shared Kernel Virtualization
3. Kernel Level Virtualization
4. Hypervisor Virtualization

Steps :

1. Download and install Oracle's Virtual Box. (Reboot needed after installation)
2. Download Ubuntu VMDK Image.
3. Launch Virtualbox and create a new VM.

4. Click on new and mention the Name and the machine folder along with the Type and Version of the Machine to be created.
5. Assign memory size for our VM (1024 MB sufficient for now).
6. Select the option ****Use an existing virtual hard disk file**** and locate the downloaded VMDK image below and create VM.
7. Now we have to create a NAT Network so go to ****File -> Preferences -> Network -> Add a New NAT Network (Click on ****+****)****
8. Right click and edit the Network name and CIDR if needed.

Example :

Name - My VMbox Network

CIDR - 172.168.2.0/24 and save the changes.

9. Repeat the process of launching the VM for 2 instances.
10. Now go to the setting, go to the network setting and change the adapter to NAT Network and select the NAT Network you made (in our case : ****My VMbox Network****) and click ok.
11. Launch the VM now.
12. Install the net-tools to know the IP's of the instance.

```
```sh
```

```
$ sudo apt install net-tools
```

```
$ sudo apt update
```

---

13. To know the IP address

```
```sh
```

```
$ ifconfig
```

Now the IP will be in the range of __172.168.2.*__

* - any number in the range of 1 to 254 (total 256 addresses)

14. Now create a file and write something into it.

```
```sh
```

```
$ touch tranfer.txt
```

```
$ nano transfer.txt
```

ls - list all the files and directories

cat - show the content inside a file

scp - it will help us to copy files from one vm to other

cd - change directory

mkdir - make a new directory

touch - it makes a new file

nano - nano is a editor inside linux os

15. If your file is on the VM with IP **\*\*172.168.2.4\*\*** and the second VM's IP is **\*\*172.168.2.5\*\***.

16. Tranfer the file using **\*\*SCP\*\***

```
```sh
```

```
$ scp tranfer.txt vagrant@172.168.2.5:/home/vagrant
```

Put in the password of the 2nd VM and done.

17. Check for the file in the Second VM under the ****/home/vagrant**** directory.

18. Done..!!!!