

## Assignment No. 1

### **Title-**

Install Google App Engine. Create hello world app and other simple web applications using python/java

### **Steps-**

Download python from- <https://www.python.org/downloads/>

Download Google Cloud SDK from-  
<https://cloud.google.com/sdk/docs/install#windows>

Launch the installer and follow the prompts

Perform initial setup by running *gcloud init*

Grant authorization to Cloud SDK tools to access Google Cloud

- Write python file with hello world statement
- Write app.yaml configuration file
- Open the shell
- Run the application with the following command in shell:
- *cmd> py google-cloud-sdk\bin\dev\_appserver.py <path to the directory where application reside>*
- Open the web browser and type *http://localhost:8080*

## Assignment No. 2

- **Title-**
  - Use GAE launcher to launch the web applications
- **Steps-**
  - Already you have installed google cloud SDK and python
  - Write the configuration file
  - Write the web application file
  - Deploy and run it

# Assignment No. 3

- **Title-**

- Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim

- **Steps-**

- Download the cloudsim jar file-  
<https://github.com/Cloudslab/cloudsim/releases/tag/cloudsim-4.0>
- Download and install java sdk from-  
<https://www.oracle.com/java/technologies/downloads/#java11>
- Download and install eclipse from-  
<https://www.eclipse.org/downloads/packages/release/kepler/sr1/eclipse-ide-java-developers>
- Write the java files for the any task scheduling algorithm
- Add cloudsim jar file as external jar file in the build configuration
- Run application

- **Steps :**
- Install any of the IDE for running JAVA applications (eclipse recommended)
- Install JDK and JRE for the same
- Add the jdk\bin path to the environment variables Open environment variables window, add the following to the path variable
- Do include your bin path wherever you have installed JDK Mine is as following : > C:\Program Files\Java\jdk-14.0.1\bin
- Open eclipse in your confined workspace
- Click on new and open a new **JAVA Project**, give it a name
- Create a package inside the src folder.
- Dump in all the files [here](#) inside the package.
- Now right click on the project and choose **configure build path**.
- Click on the libraries section and add external jars
- Extract the Cloudsim.tar file you downloaded
- Now import the jar files in that Cloudsim.tar into the external jars.
- Do remember to change the name of the package in all the source files.
- Now right click on the project and run the project as **JAVA Application**.
- Select the SJF\_Scheduler.java file present in the list.

## Assignment No. 4

- **Title-**
  - Find a procedure to transfer the files from one virtual machine to another virtual machine
- **Steps-**
  - Download and install Oracle's Virtual Box-  
<https://www.virtualbox.org/wiki/Downloads>
  - Download Ubuntu VMDK Image-  
<https://app.vagrantup.com/bento/boxes/ubuntu-18.04>
  - Launch Virtualbox and create a new VM

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

- Repeat the process of launching the VM for 2 instances
- Now go to the setting, go to the network setting and change the adapter to NAT Network and select the NAT Network you made
- Launch the VM now
- Install the net-tools to know the IP's of the instance
- create a file and write something into it
- If your file is on the VM with IP **\*\*172.168.2.4\*\*** and the second VM's IP is **\*\*172.168.2.5\*\***.
- Transfer the file using **\*\*SCP\*\***  
\$ scp tranfer.txt [vagrant@172.168.2.5:/home/vagrant](#)
- Check for the file in the Second VM under the **\*\*/home/vagrant\*\*** directory



# Assignment No. 5

- **Title-**

- Find a procedure to launch virtual machine using try stack (Online Open stack Demo Version)

- ▶ **References-**

- <https://www.amazonaws.cn/en/getting-started/tutorials/launch-a-virtual-machine/>
- <https://docs.microsoft.com/en-us/azure/virtual-machines/windows/quick-create-portal>
- <https://cloud.google.com/compute/docs/instances/create-start-instance>

- **Steps-**

- Launch an Amazon EC2 Instance
- Configure your Instance
- Connect to your Instance
- Terminate Your Instance

# Assignment No. 6

- **Title-**
  - Design and deploy a web application in a PaaS environment
- **Steps-**
  - Login to the AWS console
  - Find for AWS Amplify in the services
  - Get Started with Amplify service
  - Click on Host a Web App
  - Then choose to launch it with Github and authenticate your GitHub account for the same
  - After that choose the Repository containing your source code
  - Then Launch the application with the default configurations provided by AWS Amplify

# Assignment No. 7

- **Title-**

- Design and develop custom Application (Mini Project) using Salesforce Cloud

- **Steps-**

- Start your 30-day free trial of the world's leading CRM

<https://www.salesforce.com/eu/form/signup/freetrial-sales-pe/>

- Follow the steps to create application using salesforce.

<https://www.youtube.com/watch?app=desktop&v=XL0MSkNSl8E&feature=youtu.be>

# Assignment No. 8

- **Title-**
  - Design an Assignment to retrieve, verify, and store user credentials using Firebase Authentication, the Google App Engine standard environment, and Google Cloud Data store
- **Steps-**
  - Install required software's as per the requirement
  - Install all the packages which are needed for firebase (firebase-admin , express etc)
  - And follow the steps as per the references-  
<https://firebase.google.com/docs/reference/admin>  
<https://firebase.google.com/docs/auth/admin>  
<https://firebase.google.com/docs/admin/setup>  
<https://cloud.google.com/appengine/docs/standard/python/configuration-files>  
<https://livebook.manning.com/book/google-cloud-platform-in-action/chapter-11/>