

Assignment 1

Title: Installation of Google App Engine and develop web applications.

Objective:

To install Google App Engine and create hello world app and other simple web applications using python/java.

Requirements:

1. Google App Engine
2. Python Interpreter (Python2.7.x)
3. Text Editor
4. Browser

Theory:

A)Google App Engine

1. 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.
2. GAE requires that applications be written in Java or Python, store data in Google Bigtable and use the Google query language.
3. Noncompliant applications require modification to use GAE.
4. GAE provides more infrastructure than other scalable hosting services, such as Amazon Elastic Compute Cloud (EC2).
5. GAE also eliminates some system administration and development tasks to make writing scalable applications easier.
6. Google provides GAE free up to a certain amount of use for resources like CPU, storage, API calls and concurrent requests

B) Google Cloud SDK

1. Google Cloud SDK (Software Development Kit), in simple terms, is a set of tools that are used to manage applications and resources that are hosted on the Google Cloud Platform.
 2. It is composed of the gsutil, gcloud, and bqcommand line tools.
 3. The gcloudtool is automatically downloaded with the Cloud SDK.
 4. Google Cloud SDK run on specific platforms – Windows, Linux, and macOS and requires Python 2.7.x.
 5. SDK might have further necessities like Java tools used for the development of Google App Engine needs Java 1.7 or the later one.
 6. It can be used to locally deploy and test web applications.
-

C) Directory Structure for creating hello world application

1. The web applications to be deployed can be organized in the following directory structure

```

root_directory
|_____templates
| |_____index.html
|_____static
|_____main.py
|_____app.yaml

```

2 The templates directory can be used to store the web templates of the web application (HTML files).

The static directory can be used to store the web static files which contain the styling and the business logic data for the web application (CSS and JS files).

3 The main.py is used to define the routes, rendering logic, data acquisition logic. It provides the WSGI abstraction to the application.

6. The app.yaml file provides the runtime environment, URLs for routes and launch configuration of the application in the form of key value pairs.

4

Procedure:

5 A) Install Google Cloud SDK on Windows or Linux machines:

1. Visit the <https://cloud.google.com/sdk/docs/install> link to download the CLI (Command line interface) tool for the Cloud SDK.

2. Select the appropriate operating system from the installation manual.

3. Follow the provided instructions in the displayed section.

a) For Windows users, the executable downloader is provided for downloading.

b) For Ubuntu and Fedora users, terminal commands for installation are provided using apt and dnf repositories respectively.

B) Creating the new application

1 The application must be initialized using the above-mentioned directory structure. It is a recommended format for organization and readability of code.

2 The app.yaml file should contain the following content:

3 Contents of app.yaml

```

runtime : python2
api_version : 1
threadsafe : true

```

```

handlers
- url : /
script : main.app

```

4. The logic of the application, i.e. the Web server interaction code of the application must be placed in the main.py file.

5. A simple code displaying the hello world on a web page is as follows:

Contents of main.py for Hello World application

```
import webapp2
```

```
class MainPage(webapp2.RequestHandler) :
```

```
def get(self):
```

```
self.response.write("Hello World")
```

```
app = webapp2.WSGIApplication(
```

```
["/", MainPage],
```

```
debug=True
```

```
)
```

6. Finally, after saving the above code, the application can be run on the localhost server

using the

following command. (The command must be run on the Google Cloud Shell or the terminal in case of Ubuntu).

Command:

```
python <path_to_sdk>/bin/devappserver.py <path_to_application_directory>
```

Conclusion: Thus, in this assignment set up and use of GAE alongwith it's use in different application context is demonstrated successfully.

Assignment 2

Title: Use GAE launcher to launch web application.

Objective: To learn design and deployment process involved in creating cloud based application.

Theory:

Introduction:

Google App Engine is a web application hosting service. By “web application,” we mean an application or service accessed over the Web, usually with a web browser: storefronts with shopping carts, social networking sites, multiplayer games, mobile applications, survey applications, project management, collaboration, publishing, and all the other things we’re discovering are good uses for the Web. App Engine can serve traditional website content too, such as documents and images, but the environment is especially designed for real-time dynamic applications. Of course, a web browser is merely one kind of client: web application infrastructure is well suited to mobile applications, as well.

Google App Engine:

It is a platform-as-a-service (PaaS) Cloud computing platform that is fully managed and uses inbuilt services to run your apps. You can start development almost instantly after downloading the software development kit (SDK). You can go on to the developer’s guide right away when you click on the language you wish to develop your app in.

Advantages of Google App Engine:

- Infrastructure for Security
- Scalability
- Performance and Reliability
- Cost Savings
- Platform Independence

Installation of Google-cloud-sdk shell

The App Engine SDK allows you to run Google App Engine Applications on your local computer. It simulates the run-time environment of the Google App Engine infrastructure.

Making your First Application

Make a folder for your Google App Engine applications. I am going to make the Folder on my Desktop called "cc-ass2" – the path to this folder is:

C:\Users*\Desktop\cc-ass2

And then make a sub---folder in within cc-ass2 called "www" – the path to this folder would be:

C:\Users*\Desktop\cc-ass2\www

Using a text editor such as Visual Studio Code (<https://code.visualstudio.com/>), create a file called app.yaml in the cc-ass2 folder with the following contents:

```
runtime: python27
api_version: 1
threadsafe: true

handlers:
- url: /
  static_files: www/index.html
  upload: www/index.html

- url: /(.*)
  static_files: www/\1
  upload: www/(.*)
```

Note: Please do not copy and paste these lines into your text editor – you might end up with strange characters – simply type them into your editor.

Then create a file in the www folder called index.html with few lines of code in it:

Then start the Google cloud sdk shell program that can be found under the command and then continue writing the file path as shown below:

```
>py google-cloud-sdk\bin\dev_appserver.py C:\Users\*
\Desktop\cc-ass2
```

Once you have written command in gcloud sdk shell and press Enter. After a few moments your application will start. Then press Browse to open a browser pointing at your application which is running at <http://localhost:8000/>

Click on default to run your application.

When you make a mistake in the app.yaml file – you must the fix the mistake and attempt to start the application again.

If you make a mistake in a file like index.html, you can simply fix the file and press refresh in your browser – there is no need to restart the server.

Conclusion: Thus, in this assignment demonstrated how to design and deploy a web application on Google App Engine.

Assignment 3

Title: Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.

Objective:

To learn cloud scenario using CloudSim.

Requirements:

Java JDK and JRE

CloudSim archives (CloudSim4)

Eclipse IDE

Theory:

A) CloudSim

1. CloudSim is an open-source framework, which is used to simulate cloud computing infrastructure and services.
2. It is developed by the CLOUDS Lab organization and is written entirely in Java.
3. 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.
4. 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.

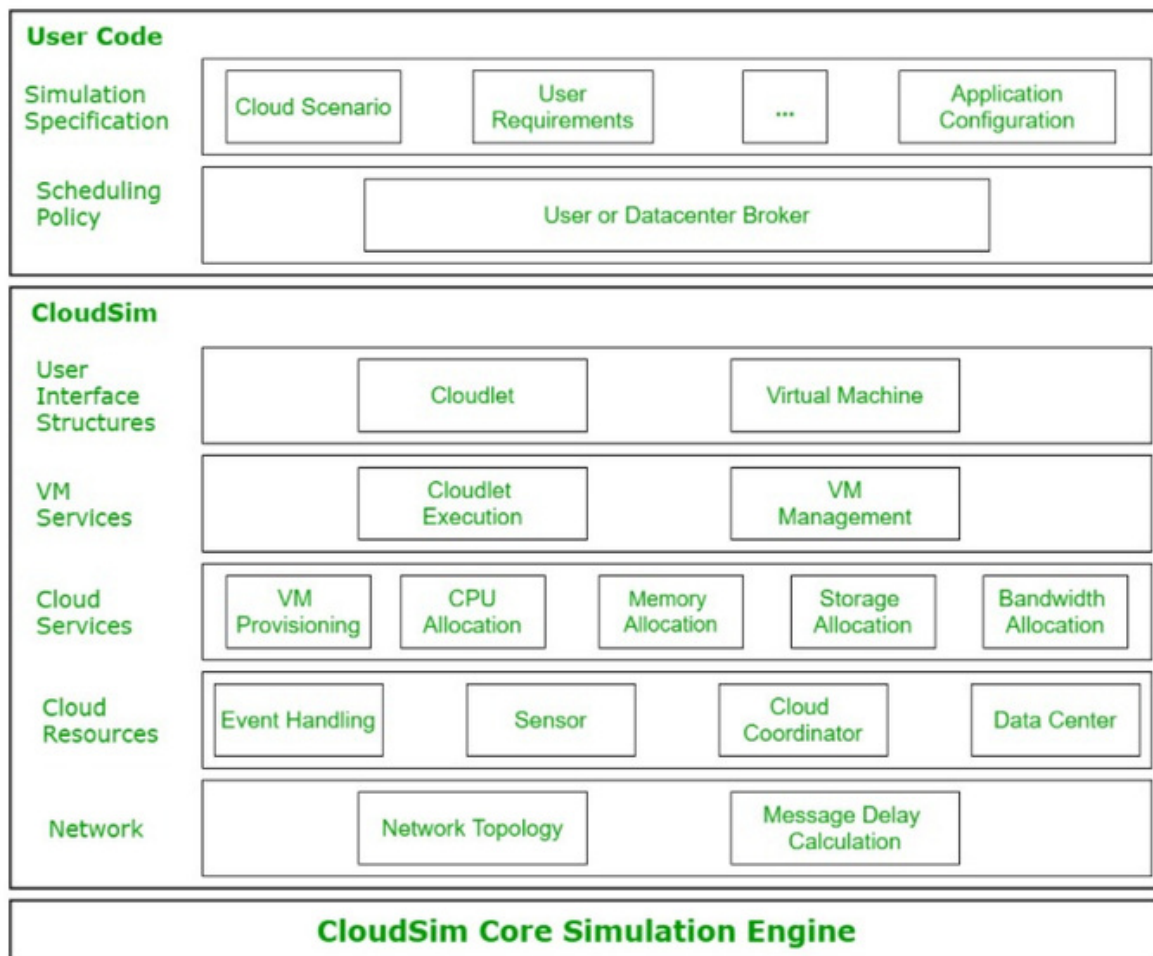
B) Benefits of CloudSim

- 1 No capital investment involved.
 - . Easy to use and scalable.
- 2 Risks can be evaluated at an earlier stage.
 - . No need for try-and-error approaches.
- 3

G) CloudSim Layered Architecture

4

1. CloudSim has a layered architecture which separates the User Code and the simulation environment.
 2. It can be depicted as follows:
-



CloudSim Layered Architecture

D) CloudSim Components

- **Datacenter** : used for modelling the foundational hardware equipment of any cloud environment, that is the Datacenter. This class provides methods to specify the functional requirements of the Datacenter as well as methods to set the allocation policies of the VMs etc.
- **Host**: this class executes actions related to management of virtual machines. It also defines policies for provisioning memory and bandwidth to the virtual machines, as well as allocating CPU cores to the virtual machines.
- **VM**: this class represents a virtual machine by providing data members defining a VM's bandwidth, RAM, mips (million instructions per second), size while also providing setter and getter methods for these parameters.
- **Cloudlet**: a cloudlet class represents any task that is run on a VM, like a processing task, or a memory access task, or a file updating task etc. It stores parameters defining the characteristics of a task such as its length, size, mi (million instructions) and provides methods similarly to VM class while also providing methods that define a task's execution time, status, cost and history.

- Datacenter Broker: is an entity acting on behalf of the user/customer. It is responsible for functioning of VMs, including VM creation, management, destruction and submission of cloudlets to the VM.
- CloudSim: this is the class responsible for initializing and starting the simulation environment after all the necessary cloud entities have been defined and later stopping after all the entities have been destroyed.
- Cloudsim – Essentials
 - o JDK 1.6 or above <http://tinyurl.com/JNU-JAVA>
 - o Eclipse 4.2 or above <http://tinyurl.com/JNU-Eclipse>
 - o Alternatively, NetBeans <https://netbeans.org/downloads>
 - o Up & Running with cloudsim guide: <https://goo.gl/TPL7Zh>
 - o Cloudsim-Directory structure
 - o cloudsim/ -- top level CloudSim directory
 - o docs/ -- CloudSim API Documentation
 - o examples/ -- CloudSim examples
 - o jars/ -- CloudSim jar archives
 - o sources/ -- CloudSim source code

E) SJF algorithm

1. SJF stands for Shortest Job First
2. Shortest Job first has the advantage of having a minimum average waiting time among all scheduling algorithms.
3. It is a Greedy Algorithm.
4. It may cause starvation if shorter processes keep coming. This problem can be solved using the concept of ageing.
5. It is practically infeasible as Operating System may not know burst time and therefore may not sort them. While it is not possible to predict execution time, several methods can be used to estimate the execution time for a job, such as a weighted average of previous execution times. SJF can be used in specialized environments where accurate estimates of running time are available.

Conclusion: Thus, in this assignment the cloudsim using Eclipse Environment is configured and different scheduling algorithm is used successfully.

Assignment 4

Title: Find a procedure to transfer the files from one virtual machine to another virtual machine

Objective: To learn procedure File Transfer in Client & Server using virtual machine

Theory:

- You can copy files using network utilities as you would between physical computers on your network. To do this between two virtual machine:
 - Both virtual machines must be configured to allow access to your network. Any of the networking methods (host-only, bridged and NAT) are appropriate.
 - With host-only networking, you copy files from the virtual machines to the host and vice-versa, since host-only networking only allows the virtual machines see your host computer.
 - With bridged networking or NAT enabled, you can copy files across your network between the virtual machines.
- You can create a shared drive, either a virtual disk or a raw partition, and mount the drive in each of the virtual machines.

Network Address Translation (NAT):

- Network Address Translation is a technique used in IP networking that modify IP packet header, by changing IP address, while the routing device processes the packet.
- Only the IP address of the routing device (in case of a Virtual Machine, the host act as router for the guest) is seen from outside. Usually the routed device (in our case, the guest machine) has an IP address that belongs to a completely different IP subnet.
- NAT make easy to use a virtual machine because:
 - it does not require to configure guest network parameters (the router act as DHCP server and give all the needed information to the guest)
 - it guarantees that the VM can be executed in any network environment without breaking the existing network.
- However it's really hard to configure NAT to let the VM to act as a server (e.g. as NFS or TFTP server), for this reason the developer usually choose a bridged configuration, with proper IP parameter assignments.
- The Network Address Translation (NAT) service works in a similar way to a home router, grouping the systems using it into a network and preventing systems outside of this network from directly accessing systems inside it, but letting systems inside communicate with each other and with systems outside using TCP and UDP over IPv4 and IPv6.

Procedure:

Requirement: Two Virtual machines installed, for this case we have Ubuntu 21 and Kali Linux.

1. Create a NAT network in which 2 virtual machine can communicate.

- Go to preferences by clicking File option in Top.
- Now select the network option and create a new NAT network
- After creating a NAT network, now go to virtual machine setting by right clicking on the preferred machine.

Now go to Network and change the attached option to “NAT network” and select the network we created earlier.

Now repeat the same process for another machine.

Launch both virtual machines

Now install “Net-tools” on both machine which will help to identify i/p address of the machine. Command: `sudo apt install net-tools`

Now create a file in Home folder using any text editor. Here we have used “Test_file.txt” and will transfer from Kali Linux (Right Machine) to Ubuntu (left Machine).

Now we can check i/p address of Ubuntu where we want to transfer the file using “ifconfig” command. Here Ubuntu has i/p address 10.0.2.4.

Transfer the file using command- `scp Test_file.txt raghav@10.0.2.4:`

Where Test_file.txt is our file

raghav is username of Ubuntu

10.0.2.4 is ip address of Ubuntu (left machine)

Now enter the password for Ubuntu(left machine) admin, after enter the password, the file will be sent from Kali Linux(Right machine) to Ubuntu (Left machine)

Conclusion:

Thus, in this assignment the transfer of file between 2 VM's is demonstrated through the setting up the Virtual Environment using VirtualBox with virtual networking (NAT).

Assignment 05

Title: Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version)

Objective: To learn procedure to launch virtual machine using trystack.

Theory:

What is Virtual Machine?

Virtual Machine is like fake computer system operating on your hardware. It partially uses the hardware of your system (like CPU, RAM, disk space, etc.) but its space is completely separated from your main system. Two virtual machines don't interrupt in each other's working and functioning nor they can excess each other's space which gives an illusion that we are using totally different hardware system.

Types of Virtual Machines:

You can classify virtual machines into two types:

1. System Virtual Machine:

These types of virtual machines gives us complete system platform and gives the execution of the complete virtual operating system. Just like virtual box,

system virtual machine is providing an environment for an OS to be installed completely. We

can see in

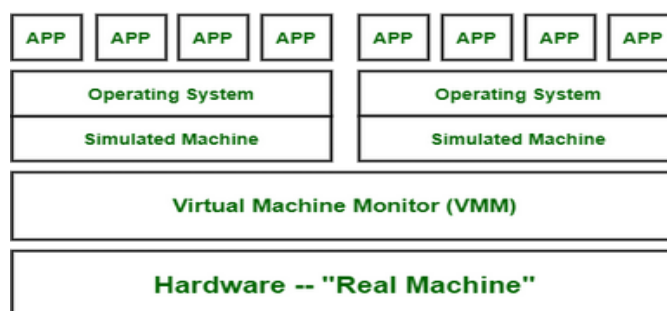
below image that our hardware of Real Machine is being distributed

between two simulated operating systems by Virtual machine monitor. And then some

programs,

processes are going on in that distributed hardware of simulated machines separately.

System Virtual Machine



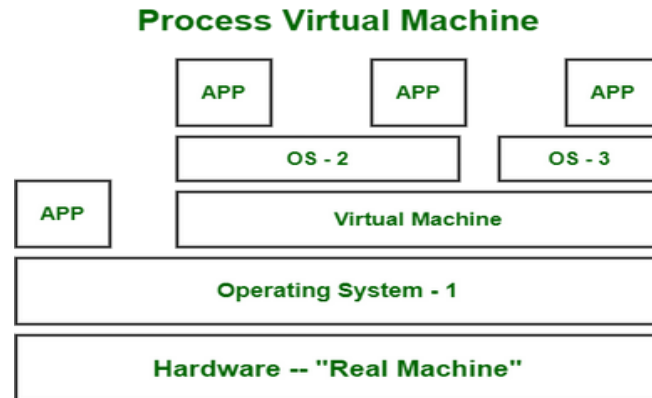
2. Process Virtual Machine:

While process virtual machines, unlike system virtual machine, does not provide us with the facility to install the virtual operating system completely.

Rather it creates virtual environment of that OS while using some app or program and this environment

will be destroyed as soon as we exit from that app.

Like in below image, there are some apps running on main OS as well some virtual machines are created to run other apps. This shows that as those programs required different OS, process virtual machine provided them with that for the time being those programs are running.



How do Virtual Machine work?

Virtualisation Technology allows you to share a system with many virtual environments. The hypervisor manages the hardware and separates the physical resources from the virtual environments.

What is Amazon EC2?

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) Cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

Steps to launch an Amazon EC2 Instance :

Step 1: Choose an Amazon Machine Image (AMI)

Open Amazon EC2 console and then click Launch Instance to create and configure your virtual machine. Configure your Instance In this screen, you are shown options to choose an Amazon Machine Image (AMI). AMIs are preconfigured server templates you can use to launch an instance. Each AMI includes an operating system, and can also include applications and application servers.

Step 2: Choose an Instance Type

Instance types comprise of varying combinations of CPU, memory, storage, and networking capacity so you can choose the appropriate mix for your applications.

Step 3: Configure Instance Details

You can review the configuration, storage, tagging, and security settings that have been selected for your instance.

Step 4: Add Storage

On the next screen you will be asked to choose an existing key pair or create a new key pair. A key pair is used to securely access your Linux instance using SSH. Amazon Web Services stores the public part of the key pair which is just like a house lock. You download and use the private part of the key pair which is just like a house key.

Step 5: Add Tags

Select Create a new key pair and give it the name MyKeyPair. Next click the Download Key Pair button.

Step 6: Configure Security Group

After you download the MyKeyPair key, you will want to store your key in a secure location. If you lose your key, you won't be able to access your instance. If someone else gets access to your key, they will be able to access your instance.

Step 7: Review Instance Launch

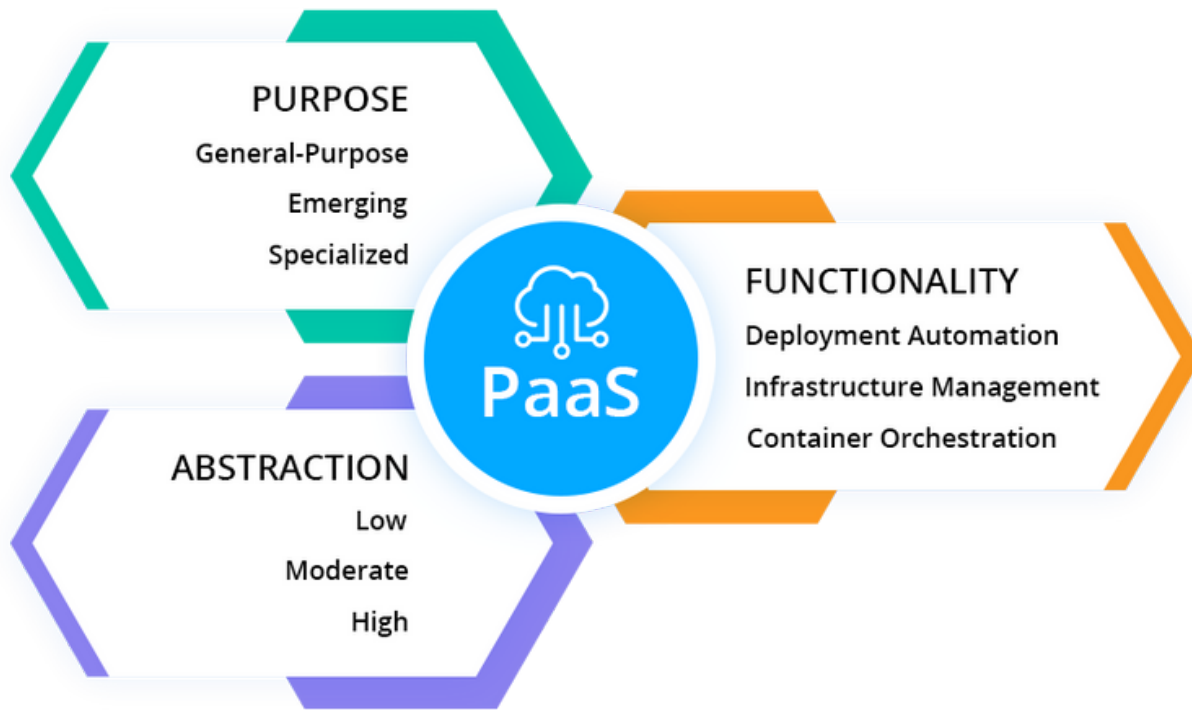
Connect to your Instance (either via SSH) Terminate Your Instance (go to the console and select the VM under actions and terminate).

Conclusions In this assignment Virtual Machine using trystack is successfully launched and learning about OpenStack and Trystack is completed.

Assignment 6

Title: Web application in a PaaS

Problem statement: To design and deploy a web application in a PaaS environment



What AWS Amplify is?

AWS Amplify is a set of purpose-built tools and features that lets frontend web and mobile developers quickly and easily build full-stack applications on AWS, with the flexibility to leverage the breadth of AWS services as your use cases evolve. With Amplify, you can configure a web or mobile app backend, connect your app in minutes, visually build a web frontend UI, and easily manage app content outside the AWS console. Ship faster and scale effortlessly—with no cloud expertise needed.

Procedure:

1. Login to the AWS console
2. Find for AWS Amplify in the services.
3. Get Started with Amplify service.
4. Click on Host a Web App.
5. Then choose to launch it with Github and authenticate your GitHub account for the same.
6. After that choose the Repository containing your source code (subfolder if needed)
7. Then Launch the application with the default configurations provided by [AWS Amplify](#)
8. Configurations may be different on type of framework / technology you are launching your application

Conclusion: Thus, in this assignment design and deployment of web application in PaaS environment is successfully completed.

Assignment 7

Title: Design and develop custom Application (Mini Project) using Salesforce Cloud.

Problem Statement: To design and develop custom Application (Mini Project) using Salesforce Cloud.

Theory:

Salesforce is a Cloud-Based Enterprise platform. It provides easy-to-use business applications, that can generate relevant customer experience. Salesforce is a tool that allows staying connected with Customers, Prospects, Partners, Sales, and Market Services globally.

Salesforce offers SaaS, PaaS, and IaaS tools. It also provides its cloud services for running applications.

The data is secure, upgraded, and scaled automatically. Salesforce Cloud is known to be very dependable and offers benefits like Adaptability and Multitenancy over others. It also has a CRM platform.

Salesforce CRM is a product that manages a company's relationship with Prospect Customers. Salesforce cloud CRM software and applications are used for Sales, Service, Finance, Marketing, Business Development, Recruiting, HR, Supply Chain Management, and other lines of business that manage external and internal relationships. It allows storing Existing Customers and Potential Customer's

information under a single platform, which would allow plotting a chart for Personalized Customer

Approach. It also allows to plan and monitor Sales and Marketing Campaigns, and manage services with

insightful and valuable data available.

Procedure:

Step-1: Click on Lightning Experience

Step-2: Click on Setup and select Setup for current App.

Step-3:

Click on Create an Object

So Click on Object Manager Tab next to Home Tab Click on Create Custom Object

Step-4 New custom object page Open Label as a-Comment Plural label- comments

Give Record Name as –comment name

Data type- text

Select Allow Reports Check Box

Step-5: Click on Home-Search Tabs in Quick search Select Custom Object-Click on New

Step-6: For Object Select Comment For Tab Style Select Any Icon

Click-Next-Next-Save

Step-7: Search App Manager in Quick Search and select app manager
Enter name to app name

Step-8

Select Profiles (System Administrator) and move to selected profile.Click on Save and Finish.

Step-9:Click on App Launcher Symbol and Select Comment Box App

Conclusion: Thus, in this assignment design and development of Custom Application using Salesforce Cloud is successfully completed.

Assignment 8

Title: Design an assignment using FireBase, Google App Engine and Data Store.

Problem Statement: Design an Assignment to retrieve, verify, and store user credentials using Firebase Authentication, the Google App Engine standard environment, and Google Cloud Data store.

Procedure:

Step-I: Install Git

```
sudo apt install git-all
```

Step-II: Install Python 2.7

Step-III: Sign into your Google Account.

If you don't already have one, sign up for a new account.

Step-IV: Select or create a GCP project.

Note: If you don't plan to keep the resources you create in this tutorial, create a new project instead of selecting an existing project. After you finish, you can delete the project, removing all resources associated with the project and tutorial.

GO TO THE MANAGE RESOURCES PAGE

Step-V: Install and initialize the Cloud SDK.

If you have already installed and initialized the SDK to a different project, set the gcloud project to the App Engine project ID you're using for Firenotes. See Managing Cloud SDK Configurations for specific commands to update a project with the gcloud tool.

Step-VI: Cloning the sample app

To download the sample to your local machine:

Clone the sample application repository to your local machine:

```
git clone https://github.com/GoogleCloudPlatform/python-docs-samples.git
```

Alternatively, you can download the sample as a zip file and extract it. Navigate to the

directory that contains the sample code:

```
cd python-docs-samples/appengine/standard/firebase/firenotes
```

Step-VII: Adding the Firebase Authentication user interface

To configure FirebaseUI and enable identity providers:

1. Add Firebase to your app by following these steps:

a. Create a Firebase project in the [Firebase console](#).

- If you don't have an existing Firebase project, click Add project and enter either an existing Google Cloud Platform project name or a new project name.
- If you have an existing Firebase project that you'd like to use, select that project from the console.

b. From the project overview page, click Add Firebase to your web app. If your project already has an app, select Add App from the project overview page.

c. Use the Initialize Firebase section of your project's customized code snippet to fill out the following section of the frontend/main.js file:

Go to given Path `appengine/standard/firebase/firenotes/frontend/main.js`

`appengine/standard/firebase/firenotes/frontend/main.js`

```
// Obtain the following from the "Add Firebase to your web app" dialogue
// Initialize Firebase
var config = {
  apiKey: "<API_KEY>",
  authDomain: "<PROJECT_ID>.firebaseapp.com",
  databaseURL: "https://<DATABASE_NAME>.firebaseio.com",
  projectId: "<PROJECT_ID>",
  storageBucket: "<BUCKET>.appspot.com",
  messagingSenderId: "<MESSAGING_SENDER_ID>"
};
```

2. Edit the `backend/app.yaml` file and enter your Firebase project ID in the environment variables:

[appengine/standard/firebase/firenotes/backend/app.yaml](#)[VIEW ON GITHUB](#)

```
runtime: python27
api_version: 1
threadsafe: true
service: backend

handlers:
- url: /*
```

```
script: main.app

env_variables:
  # Replace with your Firebase project ID.
  FIREBASE_PROJECT_ID: '<PROJECT_ID>'
```

3. In the **frontend/main.js** file, configure the [FirebaseUI login widget](#) by selecting which providers you want to offer your users.

[appengine/standard/firebase/firenotes/frontend/main.js](#)[VIEW ON GITHUB](#)

```
// Firebase log-in widget
function configureFirebaseLoginWidget() {var
uiConfig = {
'signInSuccessUrl': '/',
'signInOptions': [
// Leave the lines as is for the providers you want to offer your
users.
firebase.auth.GoogleAuthProvider.PROVIDER_ID,
firebase.auth.FacebookAuthProvider.PROVIDER_ID,
firebase.auth.TwitterAuthProvider.PROVIDER_ID,
firebase.auth.GithubAuthProvider.PROVIDER_ID,
firebase.auth.EmailAuthProvider.PROVIDER_ID
],
```

```
// Terms of service url
'tosUrl': '<your-tos-url>',
};
var ui = new firebaseui.auth.AuthUI(firebase.auth());
ui.start('#firebaseui-auth-container', uiConfig);
}
```

4. Enable the providers you have chosen to keep in the [Firebase console](#) by clicking **Authentication Sign-in method**. Then, under Sign-in providers, hover the cursor over a provider and click the pencil icon.

Toggle the Enable button and, for third-party identity providers, enter the provider ID and secret from the provider's developer site. The Firebase docs give specific instructions in the "Before you begin" sections of the Facebook, Twitter, and GitHub guides. After enabling a provider, click Save.

- ☐ In the Firebase console, under Authorized Domains, click Add Domain enter the domain of your app on App Engine in the following format:

[PROJECT_ID].appspot.com

☐

and