

SCTR's
PUNE INSTITUTE OF COMPUTER TECHNOLOGY



**THIRD YEAR
INFORMATION TECHNOLOGY
(2019 COURSE)**

**LABORATORY MANUAL
FOR
LABORATORY PRACTICE - II
(CLOUD COMPUTING LABORATORY)**

SEMESTER - VI

[Subject code: 314458]

[Prepared By]

Mr. Sachin Pande
Dr. Shyam Deshmukh

DEPARTMENT OF INFORMATION TECHNOLOGY

INSTITUTE VISION AND MISSION

VISION

Pune Institute of Computer Technology aspires to be the leader in higher technical education and research of international repute.

MISSION

To be leading and most sought-after Institute of education and research in emerging engineering and technology disciplines that attracts, retains and sustains gifted individuals of significant potential.

DEPARTMENT VISION AND MISSION

VISION

The department endeavors to be recognized globally as a center of academic excellence & research in Information Technology.

MISSION

To inculcate research culture among students by imparting information technology related fundamental knowledge, recent technological trends and ethics to get recognized as globally acceptable and socially responsible professionals.

<p style="text-align: center;">Savitribai Phule Pune University, Pune Third Year Information Technology (2019 Course) 314458: Laboratory Practice-II (Cloud Computing)</p>		
Teaching Scheme:	Credit Scheme:	Examination Scheme:
Practical (PR): 04 hrs/week	02 Credit	PR :25 Marks TW : 50Marks
<p>Prerequisites:</p> <ul style="list-style-type: none"> • Basics of Computer Networks • Operating Systems 		
<p>Course Objectives :</p> <ol style="list-style-type: none"> 1. To develop web applications in cloud. 2. To learn the design and development process involved in creating a cloud based application. 		
<p>Course Outcomes :</p> <p>On completion of the course, students will be able to–</p> <p>CO1: To design and develop cloud-based applications. CO2: To Simulate a cloud scenario using CloudSim. CO3: To design and deploy web applications in cloud environment</p>		
<p>List of Laboratory Assignments</p>		
<ol style="list-style-type: none"> 1. Install Google App Engine. Create hello world app and other simple web applications using python/java. 2. Use GAE launcher to launch the web applications. 3. Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim. 4. Find a procedure to transfer the files from one virtual machine to another virtual machine. 5. Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version) 6. Design and deploy a web application in a PaaS environment. 7. Design and develop custom Application (Mini Project) using Salesforce Cloud. 8. Design an Assignment to retrieve, verify, and store user credentials using Firebase Authentication, the Google App Engine standard environment, and Google Cloud Data store. 		
<p>CASE STUDIES</p>		
<ul style="list-style-type: none"> • Data storage security in private cloud • Application of IoT/Ubiqitous based on cloud • Tools for building private cloud 		
<p>Text Books:</p>		
<ol style="list-style-type: none"> 1. Thomas Erl, Zaigham Mahmood and Ricardo Puttini, Cloud Computing: Concepts, Technology & Architecture, Pearson, ISBN :978 9332535923, 9332535922, 1 st Edition 2. Anthony T. Velte Toby J. Velte, Robert Elsenpeter, “Cloud Computing: A Practical Approach”. 		

Reference Books:

1. Rajkumar Buyya, Christian Vecchiola, S. ThamaraiSelvi, Mastering Cloud Computing: Foundations and Applications Programming, McGraw Hill, ISBN: 978 1259029950, 1259029956.
2. Gautam Shrof, "ENTERPRISE CLOUD COMPUTING Technology Architecture, Applications, Cambridge University Press, ISBN: 9780511778476
3. Srinivasan, J. Suresh, Cloud Computing: A practical approach for learning and implementation, Pearson, ISBN :9788131776513.
4. Jack J. Dongarra, Kai Hwang, Geoffrey C. Fox, Distributed and Cloud Computing: From Parallel Processing to the Internet of Things, Elsevier, ISBN :9789381269237, 9381269238, 1st Edition.
5. Brian J.S. Chee and Curtis Franklin, Jr., Cloud Computing: Technologies and Strategies of the Ubiquitous Data Center, CRC Press, ISBN :9781439806128.
6. Kris Jamsa, Cloud Computing: Saas, Paas, Iaas, Virtualization, Business Models, Mobile, Security, and More, Jones and Bartlett, ISBN :9789380853772.
7. John W. Ritting house, James F. Ransome, Cloud Computing Implementation, Management, and Security, CRC Press, ISBN : 978 1439806807, 1439806802.
8. Karl Matthias, Sean P. Kane, Docker: Up and Running, O'Reilly, ISBN:9781491917572,1491917571.
9. Barrie Sosinsky, Cloud Computing Bible, Wiley, ISBN: 978 8126529803.
10. Ronald L. Krutz and Russell D. Vines, Cloud Security: A Comprehensive guide to Secure Cloud Computing, Wiley, ISBN: 9788126528097.
11. Scott Adkins, John Belamaric, Vincent Giersch, Denys Makogon, Jason E. Robinson, OpenStack: Cloud Application Development, Wrox, ISBN :9781119194316.
12. Kailash Jayaswal, Jagannath Kallakurchi, Donald J. Houde, Cloud Computing Black Book ,Wiley Dreamtech,ISBN:9789351194187

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

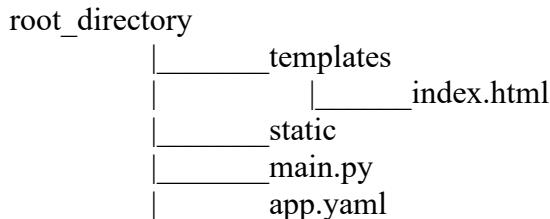
1. Google App Engine (GAE) is a platform-as-a-service product that provides web app developers and enterprises with access to Google's scalable hosting and tier 1 internet service.
2. GAE requires that applications be written in Java or Python, store data in Google Bigtable and use the Google query language.
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



2. The templates directory can be used to store the web templates of the web application (HTML files).
3. 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).
4. The main.py is used to define the routes, rendering logic, data acquisition logic.
5. 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.

Procedure:

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.
2. It is a recommended format for organization and readability of code.
3. The app.yaml file should contain the following content:

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
)
```

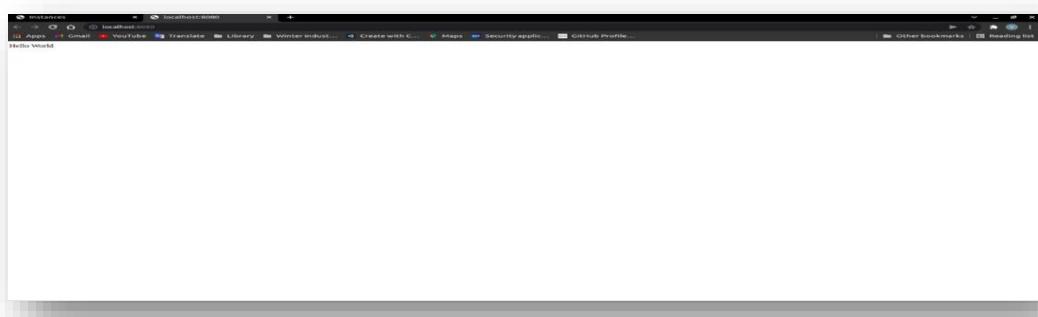
6. Finally, after saving the above code, the application can be run on the localhost server using the following command. (The command must be run on the Google Cloud Shell or the terminal in case of Ubuntu).

Command:

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

Sample Output

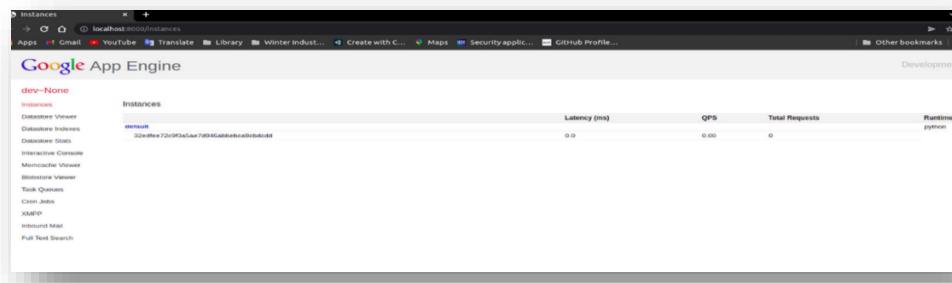
1. The application, if no errors are found, is launched on the port 8080 of the localhost server.
2. The cloud console is visible on port 8000 of the localhost server.
3. The URL of localhost:8080 can be typed in the address bar of the browser to view the application.
4. Screenshots:
 - a) Application launch at port 8080



b) Terminal / Command Line prompts

```
varadnash@varadnash-G3-3590:~$ python3 ./google-cloud-sdk/bin/dev_appserver.py ./cloud_computing_lab/Assignment1/
Updates are available for some Cloud SDK components. To install them,
please run:
$ gcloud components update

WARNING 2022-01-16 07:03:14,121 application_configuration.py:210] The "python" runtime specified in "./cloud_computing_lab/Assignment1/app.yaml" is not supported
tead. A description of the differences between the two can be found here:
https://developers.google.com/appengine/docs/python/python25/diff27
INFO 2022-01-16 07:03:14,123 devappserver2.py:316] Skipping SDK update check.
WARNING 2022-01-16 07:03:15,018 simple_search_stub.py:196] Could not read search indexes from /tmp/appengine.None.varadnash/search_indexes
INFO 2022-01-16 07:03:15,021 <string>:383] Starting API server at: http://localhost:3917
INFO 2022-01-16 07:03:15,155 dispatcher.py:281] Starting module "default" running at: http://localhost:8080
INFO 2022-01-16 07:03:15,157 admin_server.py:150] Starting admin server at: http://localhost:8000
INFO 2022-01-16 07:03:21,172 instance.py:294] Instance PID: 3381
```

c) Cloud Console at port 8000

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

FAQ :

1. What is the use of Google App Engine?
2. What is .yaml file?

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.

Pre--Requisites: Python 3.10.4

If you don't already have Python 3.10.4 installed in your computer, download and Install Python 3.10.4 from:

<https://www.python.org/ftp/python/3.10.4/python-3.10.4-amd64.exe>

Download and Install

You can download the Google App Engine SDK by going to:

<https://dl.google.com/dl/cloudsdk/channels/rapid/GoogleCloudSDKInstaller.exe>

and download the appropriate install package.

The screenshot shows a web browser displaying the Google Cloud SDK documentation at cloud.google.com/sdk/docs/install. The page is titled 'Installing the gcloud CLI'. It includes a sidebar with navigation links for 'gcloud CLI', 'Quickstart', 'How-to guides', and 'Install the Google Cloud CLI'. The main content area contains instructions for installing the Google Cloud CLI, mentioning the gcloud, gsutil, and bq command-line tools. It provides links for 'All features', 'Cloud Client Libraries', and 'Installation instructions'. A note about proxy/firewall is included. Below the note, there are tabs for Linux, Debian/Ubuntu, Red Hat/Fedora/CentOS, macOS, and Windows, with Windows selected. A note states that the Google Cloud CLI works on Windows 8.1 and later and Windows Server 2012 and later. Step 1 of the installation process is listed as 'Download the Google Cloud CLI installer.'

Window of Google cloud sdk shell .

The screenshot shows a terminal window titled 'Google Cloud SDK Shell'. The window displays the following text:

```
Google Cloud SDK Shell
Welcome to the Google Cloud CLI! Run "gcloud -h" to get the list of available commands.
---

C:\Program Files (x86)\Google\Cloud SDK>
```

Making your First Application

Make a folder for your Google App Engine applications. I am going to make theFolder 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 endup 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:

```

app.yaml
index.html
www > index.html > html > style
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>Fitness</title>
9  </head>
10 <link rel="stylesheet" href="css/style.css">
11 > style
135 </style>
136
137 <body>
138 >     <header class="header">...
162 </header>
163 <div class="container">
164     <h1>Welcome to he Fitness gym</h1>
165     <form action="/" method="post" class="basicData">
166         <div class="form-group">
167             <input type="text" name="Name" placeholder="Enter your Name">
168         </div>
169         <div class="form-group">
170             <input type="text" name="Age" placeholder="Enter your Age">
171         </div>
172         <div class="form-group">
173             <input type="text" name="Gender" placeholder="Enter your Gender">
174         </div>
175         <div class="form-group">
176             <input type="number" name="contact" placeholder="Enter your contact no. ">
177         </div>
178         <div class="form-group">
179             <input type="text number" name="Address" placeholder="Enter your Address">
180         </div>
181         <div class="form-group">
182             <input type="text" name="City" placeholder="Enter your City">
183         </div>
184     <button class="btn">submit</button>

```

Then start the **Google cloud sdk shell** program that can be found under **Applications**. Then follow 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
```

```

Google Cloud SDK Shell - py google-cloud-sdk\bin\dev_appserver.py C:\Users\cheta\Desktop\cc-ass2
Welcome to the Google Cloud CLI! Run "gcloud -h" to get the list of available commands.
---

C:\Program Files (x86)\Google\Cloud SDK>py google-cloud-sdk\bin\dev_appserver.py C:\Users\cheta\Desktop\cc-ass2
INFO    2022-04-25 23:54:26,285 devappserver2.py:316] Skipping SDK update check.
WARNING 2022-04-25 23:54:26,503 simple_search_stub.py:1196] Could not read search indexes from c:\users\cheta\appdata\local\temp\appengine.None\search_indexes
INFO    2022-04-25 23:54:26,503 <string>:384] Starting API server at: http://localhost:63138
INFO    2022-04-25 23:54:26,519 dispatcher.py:281] Starting module "default" running at: http://localhost:8080
INFO    2022-04-25 23:54:26,519 admin_server.py:150] Starting admin server at: http://localhost:8000
INFO    2022-04-25 23:54:30,184 instance.py:294] Instance PID: 11380

```

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/>

Paste <http://localhost:8000> into your browser and you should see window as follows:

The screenshot shows the Google App Engine Instances page. On the left, there's a sidebar with links like Datastore Viewer, Datastore Indexes, Datastore Stats, Interactive Console, Memcache Viewer, Blobstore Viewer, Task Queues, Cron Jobs, XMPP, Inbound Mail, and Full Text Search. The main area has a title 'Google App Engine' and a sub-section 'Development SDK 0.0.0'. It shows a table with one row for the 'default' instance. The table columns are 'Instances', 'Latency (ms)', 'QPS', 'Total Requests', and 'Runtime'. The 'Instances' column shows 'a5e16257922bdcb0aaa24dcae4d2bbcd204'. The other columns show '0.0', '0.00', '0', and 'python27' respectively.

Click on default to run your application.

The screenshot shows a web application titled 'he Fitness gym'. The page has a dark background with a large image of a gym floor and equipment. On the left, there's a form with a title 'Welcome to he Fitness gym'. The form contains six input fields: 'Enter your Name', 'Enter your Age', 'Enter your Gender', 'Enter your contact no.', 'Enter your Address', and 'Enter your City'. Below these fields is a 'submit' button. At the top of the page, there's a navigation bar with links for Home, About, Fitness Calculatur, Contact Us, and two buttons for 'Call us' and 'Email Us'. There's also a small icon of a person working out.

Reference: http://en.wikipedia.org/wiki/Stack_trace

When you make a mistake in the **app.yaml** file – you must 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.

FAQ:

1. What is GAE launcher?
2. Which server is used in GA to run web application?
3. What do you mean by PaaS?

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:

1. Java JDK and JRE
2. CloudSim archives (CloudSim4)
3. Eclipse IDE

Theory:

A) CloudSim

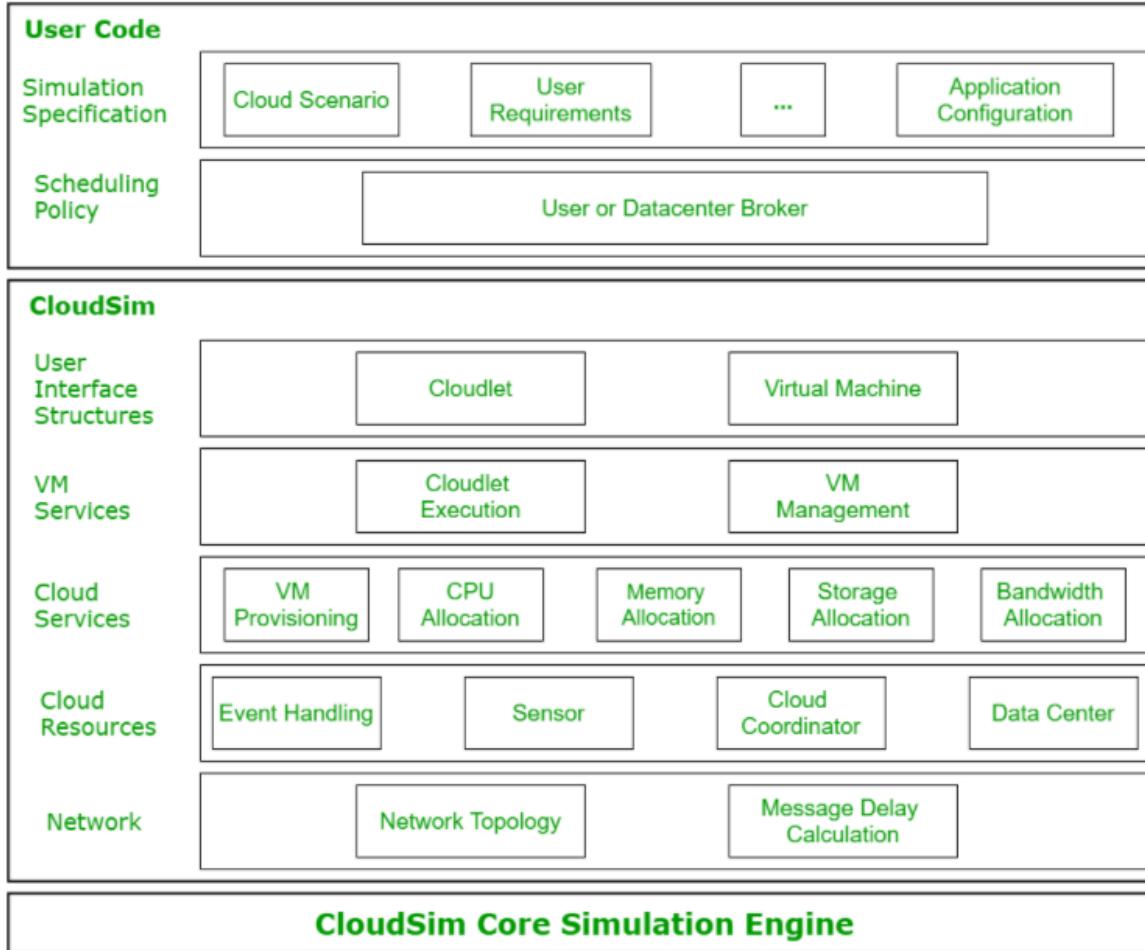
1. CloudSim is an open-source framework, which is used to simulate cloud computing infrastructure and services.
2. It is developed by the CLOUDS Lab organization and is written entirely in Java.
3. It is used for modelling and simulating a cloud computing environment as a means for evaluating a hypothesis 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.
2. Easy to use and scalable.
3. Risks can be evaluated at an earlier stage.
4. No need for try-and-error approaches.

C) CloudSim Layered Architecture

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
 - JDK 1.6 or above <http://tinyurl.com/JNU-JAVA>
 - Eclipse 4.2 or above <http://tinyurl.com/JNU-Eclipse>
 - Alternatively, NetBeans <https://netbeans.org/downloads>
 - Up & Running with cloudsim guide: <https://goo.gl/TPL7Zh>
 - Cloudsim-Directory structure
 - cloudsim/ -- top level CloudSim directory
 - docs/ -- CloudSim API Documentation
 - examples/ -- CloudSim examples
 - jars/ -- CloudSim jar archives
 - sources/ -- CloudSim source code

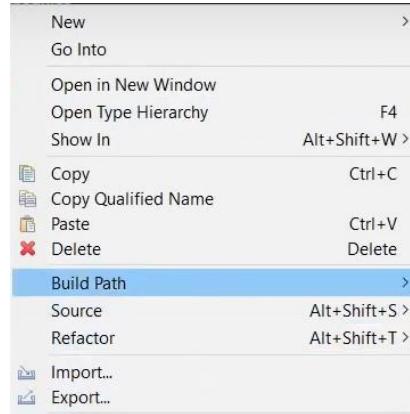
E) SJF algorithm

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.

Procedure:

A) Installation of CloudSim and creation of simulation environment

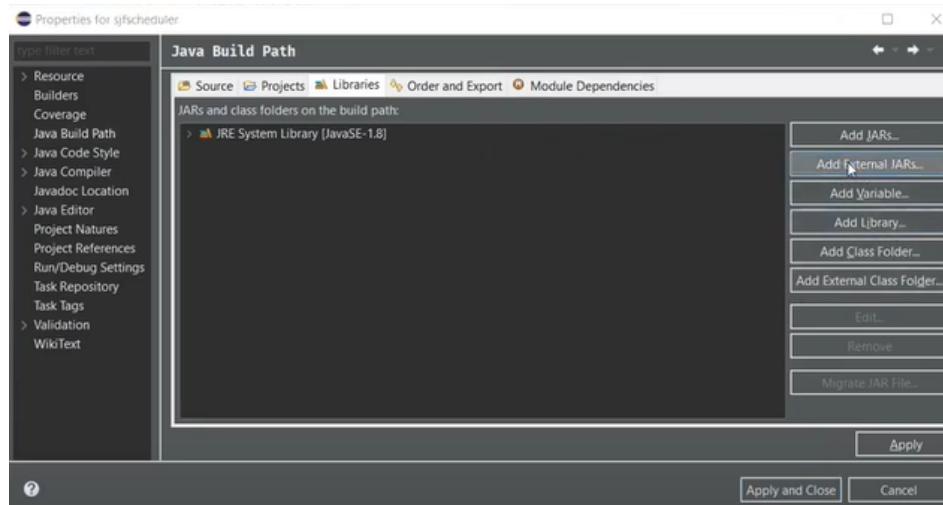
1. Visit <https://github.com/Cloudslab/cloudsim/releases> to download the CloudSim archives for CloudSim 4.
2. Extract the archive.
3. The jars folder of the extracted archive should contain the following files:
 - a) cloudsim-4.0.jar
 - b) cloudsim-examples.jar
4. Create a new Java Project using the Eclipse IDE.
5. Right click on the project root and select the Build Path option from the dropdown.



6. Select the Configure Build Path section from the extended dropdown.



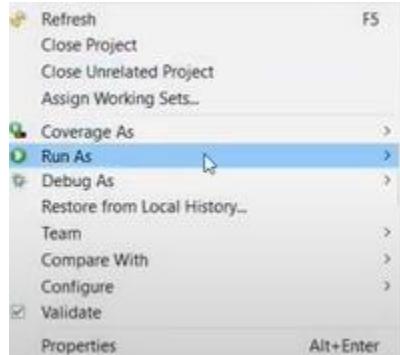
7. Select the Libraries section and click on Add External JARs field on the pop up.



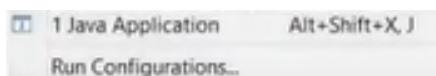
8. Navigate to the jars directory of the CloudSim archive and include the 2 jars in the project.
9. Create a new package in the src directory of the project.
10. Copy the code files for the constants, Data Center Creator, Data Center Broker, Matrix Generator and the SJF scheduler files from the <https://github.com/suyash-more/Cloud-Computing-Projects/tree/master/Scheduling-Algorithm-in-CloudSim/src> link.
11. Make sure that the package name provided in each file is the same as the previously created package.

B) Execution of code

1. Right click on the project



2. Run the project as a Java Application (option in extended dropdown)



3. The result is displayed on the console

```
Starting SJF Scheduler...
Initialising new Matrices ...
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Datacenter_1 is starting...
Datacenter_2 is starting...
Datacenter_3 is starting...
Datacenter_4 is starting...
Broker_0 is starting...
Entities started.
0.0: Broker_0: Cloud Resource List received with 5 resource(s)
0.0: Broker_0: Trying to Create VM #2 in Datacenter_0
0.0: Broker_0: Trying to Create VM #3 in Datacenter_1
0.0: Broker_0: Trying to Create VM #4 in Datacenter_2
0.0: Broker_0: Trying to Create VM #5 in Datacenter_3
0.0: Broker_0: Trying to Create VM #6 in Datacenter_4
0.1: Broker_0: VM #2 has been created in Datacenter #2, Host #0
0.1: Broker_0: VM #3 has been created in Datacenter #3, Host #0
0.1: Broker_0: VM #4 has been created in Datacenter #4, Host #0
0.1: Broker_0: VM #5 has been created in Datacenter #5, Host #0
0.1: Broker_0: VM #6 has been created in Datacenter #6, Host #0
0.1: Broker_0: Sending cloudlet 0 to VM #5
0.1: Broker_0: Sending cloudlet 1 to VM #6
0.1: Broker_0: Sending cloudlet 2 to VM #4
0.1: Broker_0: Sending cloudlet 3 to VM #3
0.1: Broker_0: Sending cloudlet 4 to VM #5
0.1: Broker_0: Sending cloudlet 5 to VM #2
0.1: Broker_0: Sending cloudlet 6 to VM #6
0.1: Broker_0: Sending cloudlet 7 to VM #3
0.1: Broker_0: Sending cloudlet 8 to VM #4
0.1: Broker_0: Sending cloudlet 9 to VM #4
0.1: Broker_0: Sending cloudlet 10 to VM #3
0.1: Broker_0: Sending cloudlet 11 to VM #2
0.1: Broker_0: Sending cloudlet 12 to VM #6
0.1: Broker_0: Sending cloudlet 13 to VM #4
```

```

1292.724: Broker_0: Cloudlet 0 received
1907.232: Broker_0: Cloudlet 5 received
2260.772: Broker_0: Cloudlet 1 received
2784.16: Broker_0: Cloudlet 3 received
2903.4: Broker_0: Cloudlet 2 received
3065.932: Broker_0: Cloudlet 11 received
4113.036: Broker_0: Cloudlet 4 received I
4485.576: Broker_0: Cloudlet 18 received
4837.776: Broker_0: Cloudlet 20 received
4956.164: Broker_0: Cloudlet 6 received
5643.272: Broker_0: Cloudlet 8 received
5807.608: Broker_0: Cloudlet 21 received
6354.656: Broker_0: Cloudlet 7 received
6905.915999999999: Broker_0: Cloudlet 23 received
7719.535999999999: Broker_0: Cloudlet 24 received
8614.368: Broker_0: Cloudlet 12 received
8752.444: Broker_0: Cloudlet 10 received
8986.24: Broker_0: Cloudlet 9 received
10703.216: Broker_0: Cloudlet 15 received
10857.967999999999: Broker_0: Cloudlet 14 received
11948.996: Broker_0: Cloudlet 25 received
13310.556: Broker_0: Cloudlet 13 received
13635.776: Broker_0: Cloudlet 16 received
15582.328: Broker_0: Cloudlet 17 received
16230.772: Broker_0: Cloudlet 19 received
17007.956: Broker_0: Cloudlet 27 received
19003.152000000002: Broker_0: Cloudlet 22 received
19533.66: Broker_0: Cloudlet 28 received
22878.644: Broker_0: Cloudlet 26 received
25918.7: Broker_0: Cloudlet 29 received
25918.7: Broker_0: All Cloudlets executed. Finishing...

```

OUTPUT							
Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time	Waiting Time
00	SUCCESS	05	05	1292.62	00.1	1292.72	00
05	SUCCESS	02	02	1907.13	00.1	1907.23	00
01	SUCCESS	06	06	2260.67	00.1	2260.77	00
03	SUCCESS	03	03	2784.06	00.1	2784.16	00
02	SUCCESS	04	04	2903.3	00.1	2903.4	00
11	SUCCESS	02	02	1158.7	1907.23	3065.93	1907.13
04	SUCCESS	05	05	2820.31	1292.72	4113.04	1292.62
18	SUCCESS	02	02	1419.64	3065.93	4485.58	3065.83
20	SUCCESS	02	02	352.2	4485.58	4837.78	4485.48
06	SUCCESS	06	06	2695.39	2260.77	4956.16	2260.67
08	SUCCESS	04	04	2739.87	2903.4	5643.27	2903.3
21	SUCCESS	05	05	1694.57	4113.04	5807.61	4112.94
07	SUCCESS	03	03	3570.5	2784.16	6354.66	2784.06
23	SUCCESS	02	02	2068.14	4837.78	6905.92	4837.68
24	SUCCESS	02	02	813.62	6905.92	7719.54	6905.82
12	SUCCESS	06	06	3658.2	4956.16	8614.37	4956.06
10	SUCCESS	03	03	2397.79	6354.66	8752.44	6354.56
09	SUCCESS	04	04	3342.97	5643.27	8986.24	5643.17
15	SUCCESS	06	06	2088.85	8614.37	10703.22	8614.27
14	SUCCESS	03	03	2105.52	8752.44	10857.97	8752.34
25	SUCCESS	03	03	1091.03	10857.97	11949	10857.87
13	SUCCESS	04	04	4324.32	8986.24	13310.56	8986.14
16	SUCCESS	06	06	2932.56	10703.22	13635.78	10703.12
17	SUCCESS	06	06	1946.55	13635.78	15582.33	13635.68
19	SUCCESS	04	04	2920.22	13310.56	16230.77	13310.46
27	SUCCESS	06	06	1425.63	15582.33	17007.96	15582.23
22	SUCCESS	04	04	2772.38	16230.77	19003.15	16230.67
28	SUCCESS	06	06	2525.7	17007.96	19533.66	17007.86
26	SUCCESS	04	04	3875.49	19003.15	22878.64	19003.05
29	SUCCESS	04	04	3040.06	22878.64	25918.7	22878.54

Makespan using SJF: 4396.012266367984

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

FAQ:

1. What is CloudSim?
2. What are the default scheduling policies in CloudSim?
3. What is Datacenter Broker in CloudSim?

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):

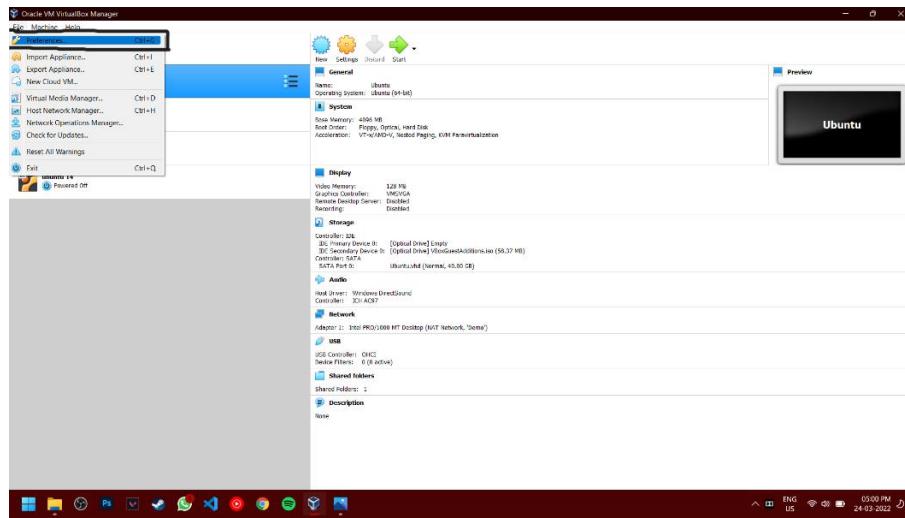
- Network Address Translation is a technique used in IP networking that modify IP packet header, by changing IP address, while the routing device processes the packet.
- Only the IP address of the routing device (in case of a Virtual Machine, the host or router for the guest) is seen from outside. Usually the routed device (in our case, the guest machine) has an IP address that belongs to a completely different IP subnet.
- NAT makes it easy to use a virtual machine because:
 - it does not require to configure guest network parameters (the router acts as DHCP server and gives 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:

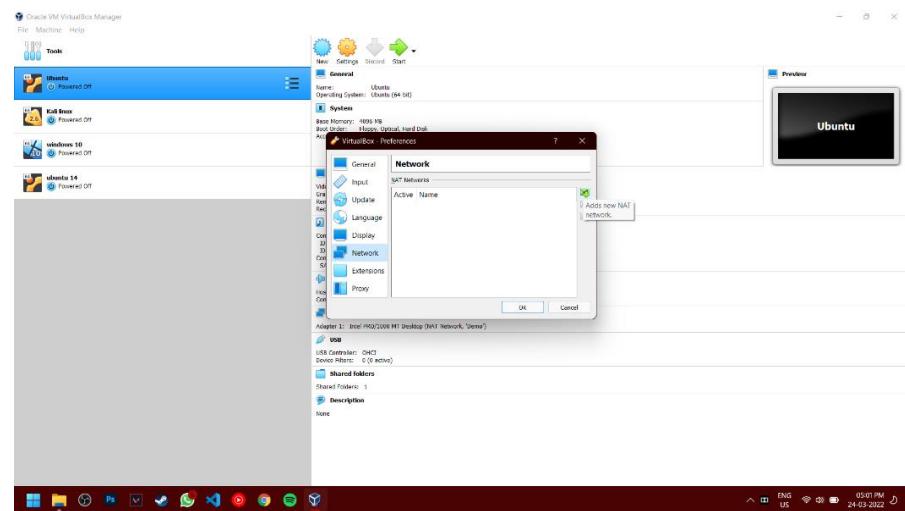
Requirement: Two Virtual machines installed, for this case we have Ubuntu 21 and Kali Linux.

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

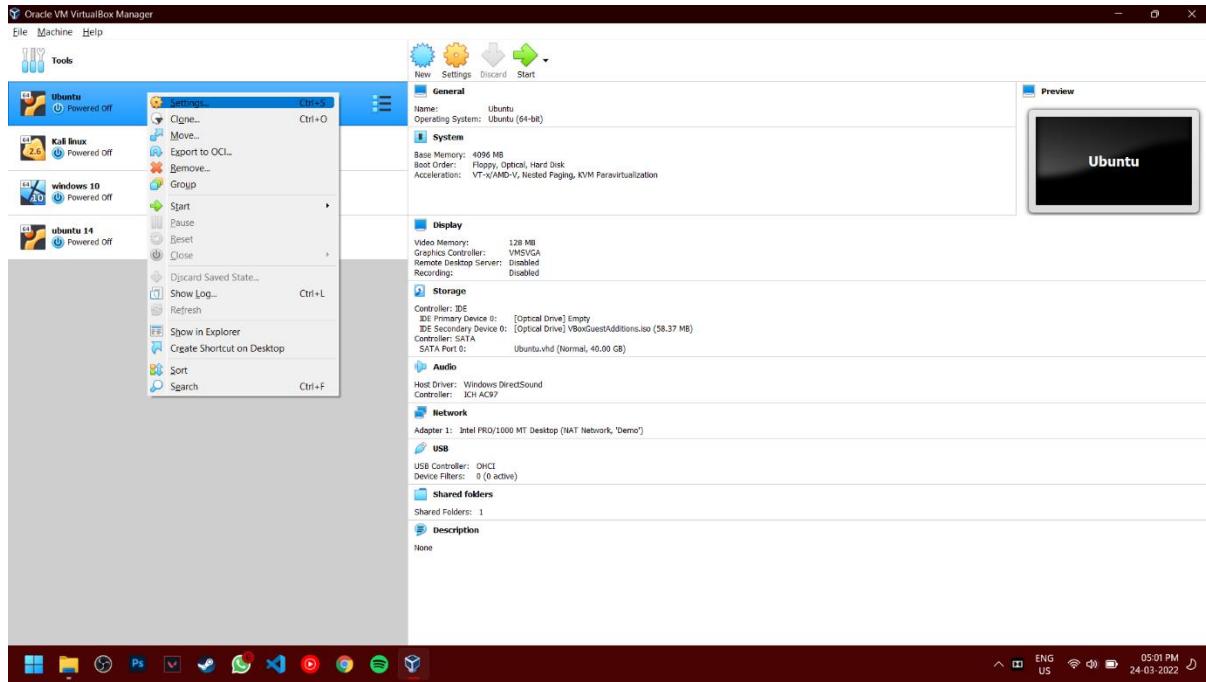
- Go to preferences by clicking File option in Top.



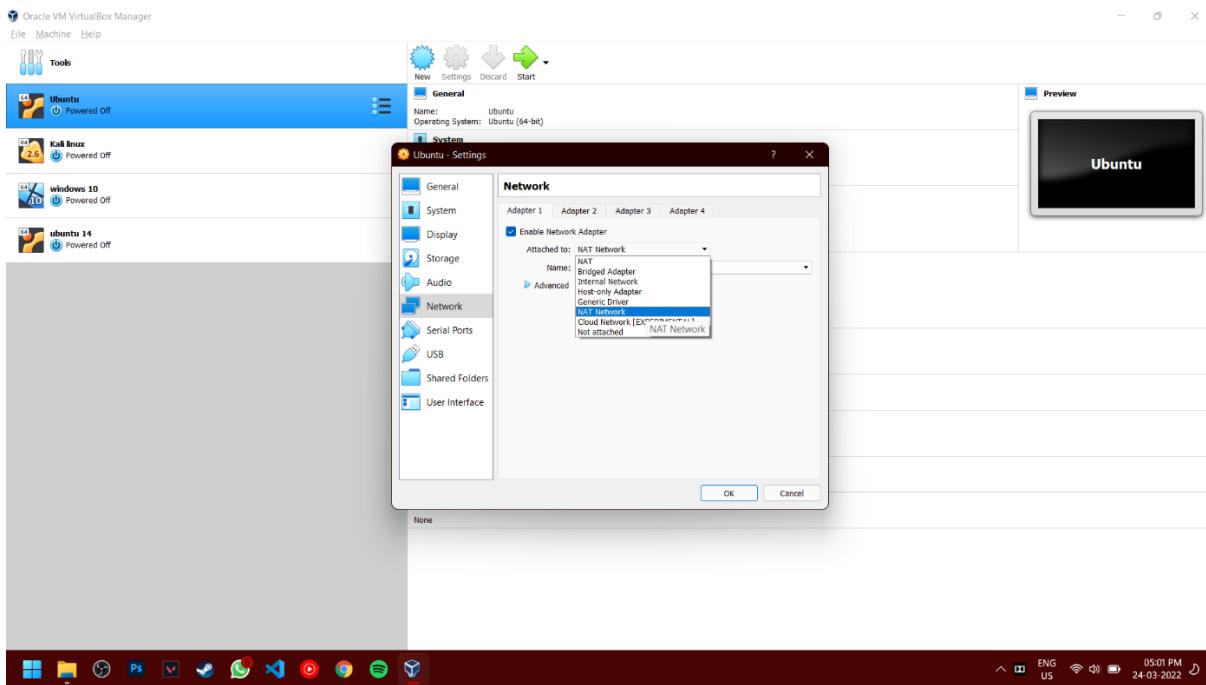
- Now select the network option and create a new NAT network

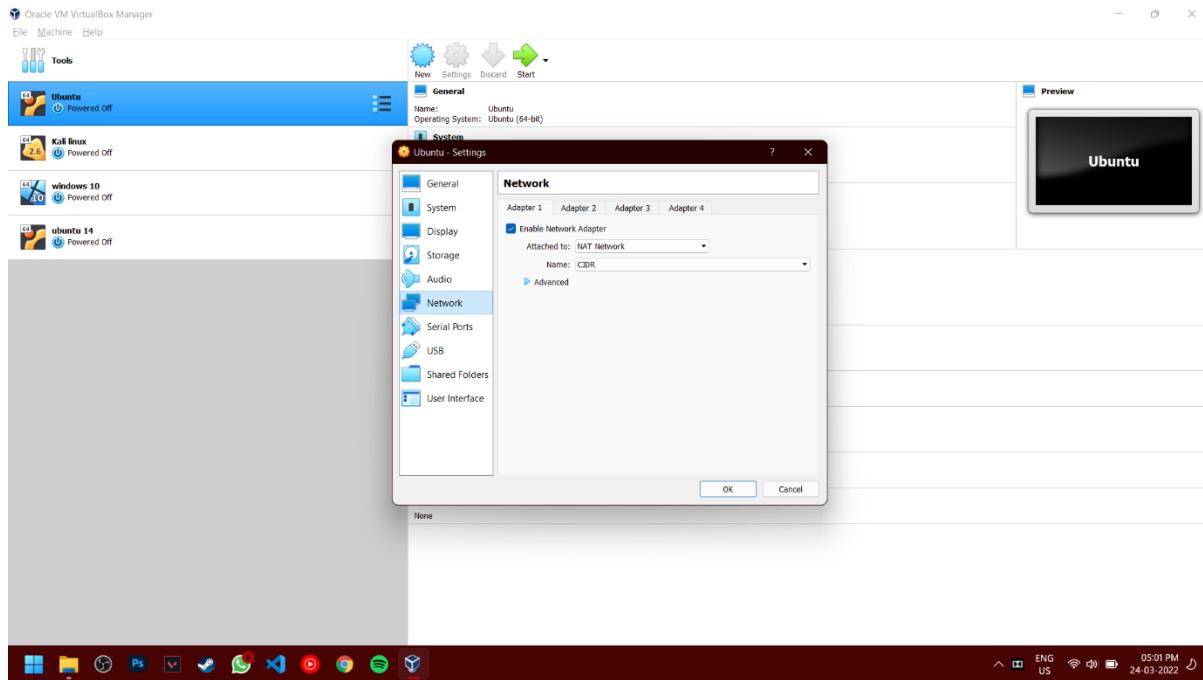


- After creating a NAT network, now go to virtual machine setting by right clicking on the preferred machine.



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

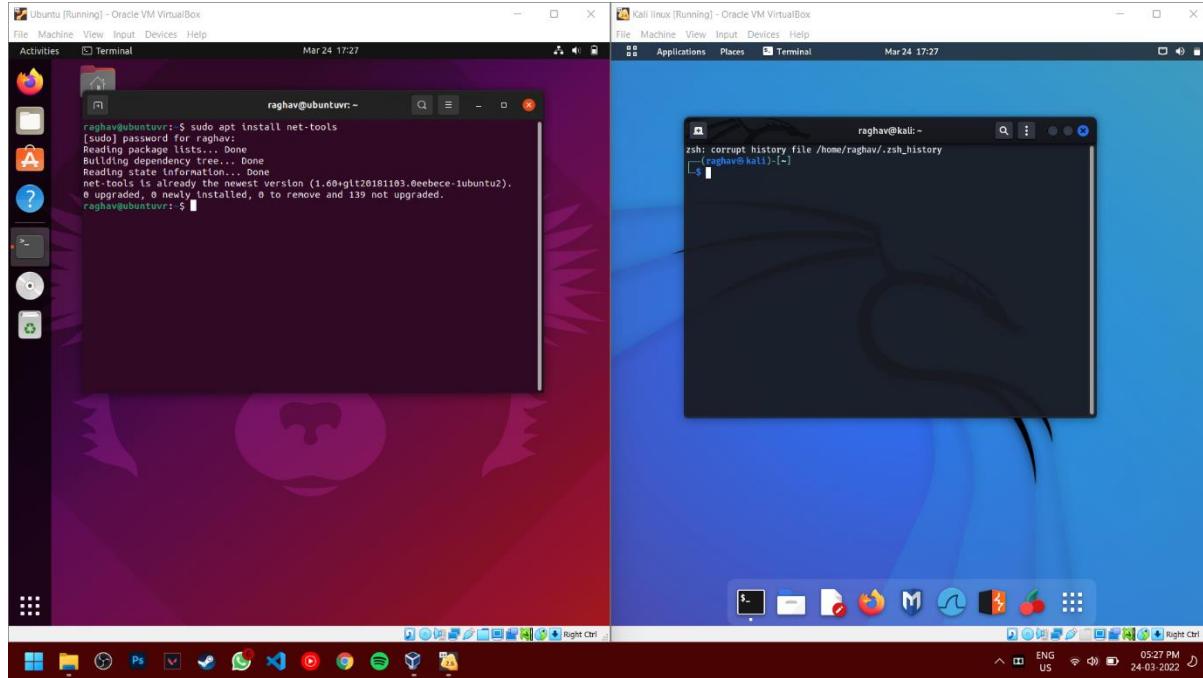




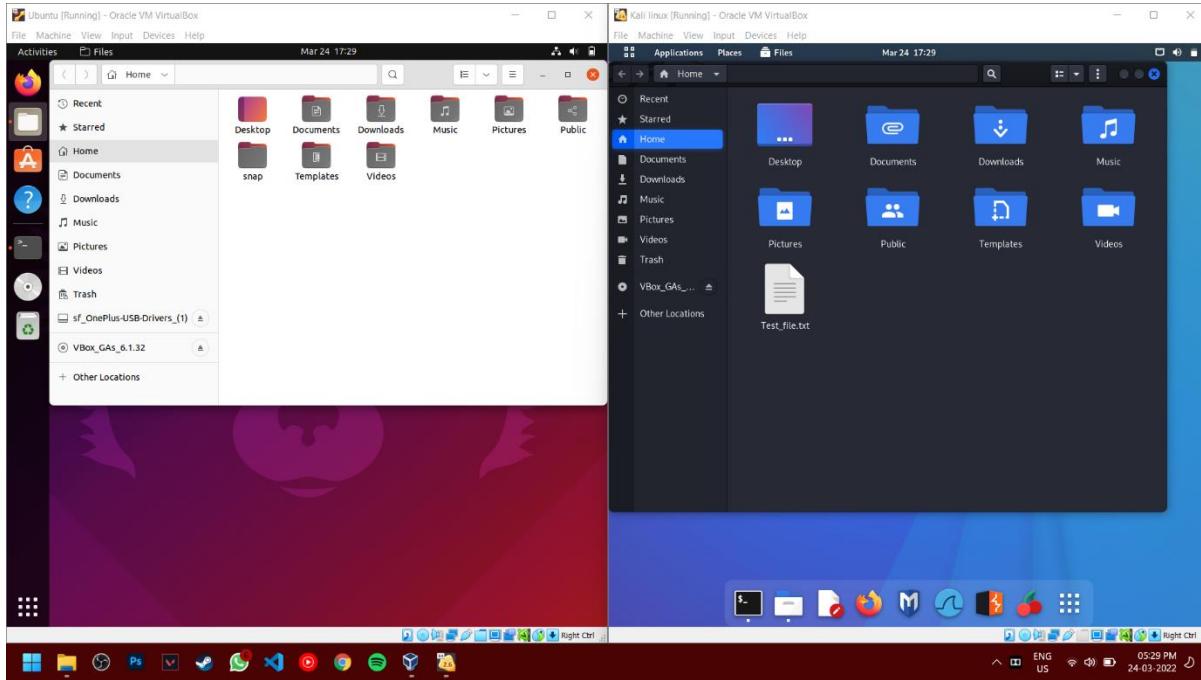
- Now repeat the same process for another machine.

2. Launch both virtual machines

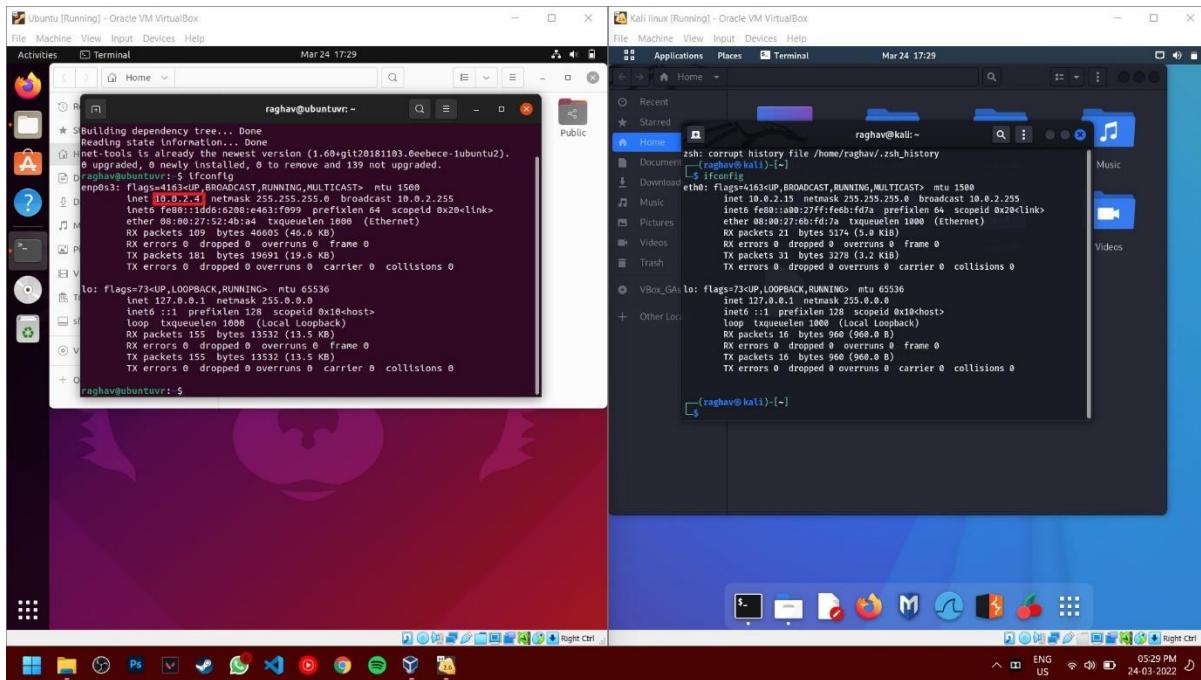
- Now install “Net-tools” on both machine which will help to identify i/p address of the machine.
Command: sudo apt install net-tools



- 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.

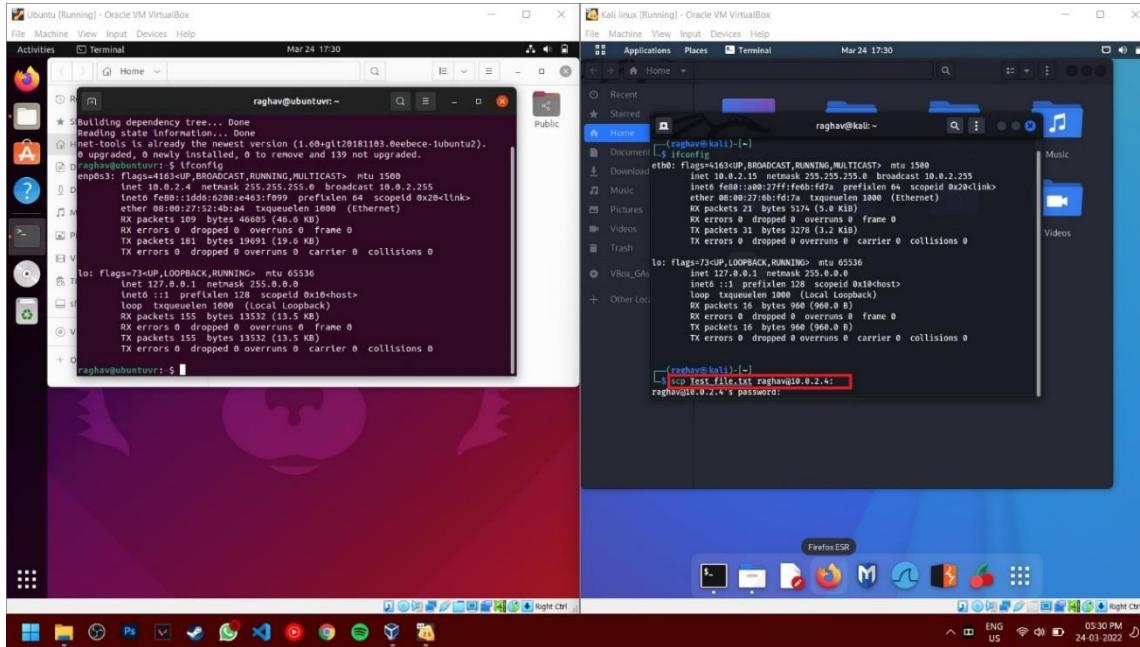


- Transfer the file using command- scp Test_file.txt raghav@10.0.2.4:
Where Test_file.txt is our file
raghav is username of Ubuntu

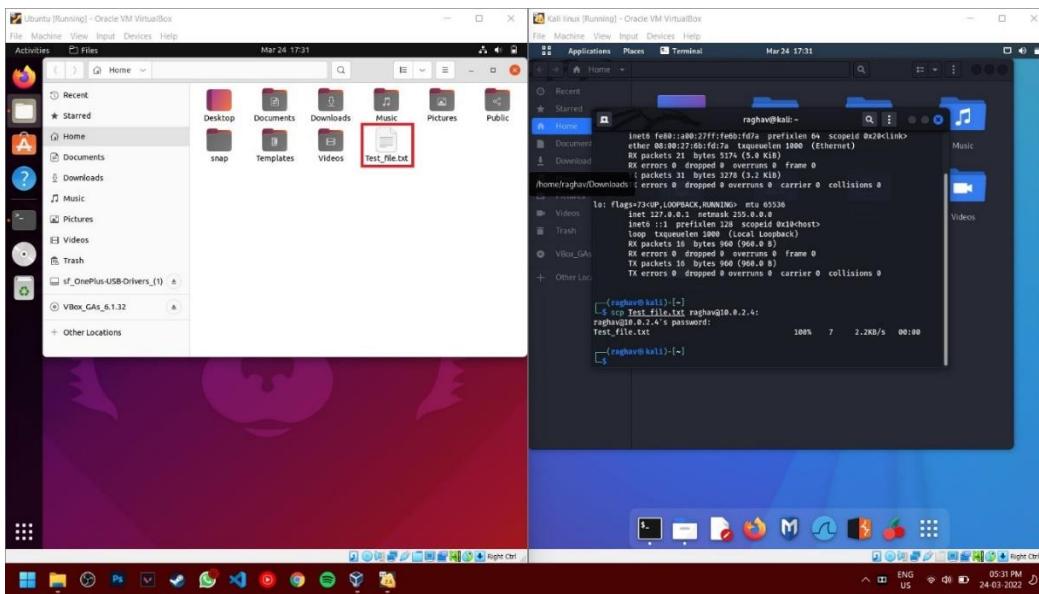
10.0.2.4 is ip address of Ubuntu (left machine)

Optional: if scp is not installed then install by using command:

`sudo apt install openssh-server`

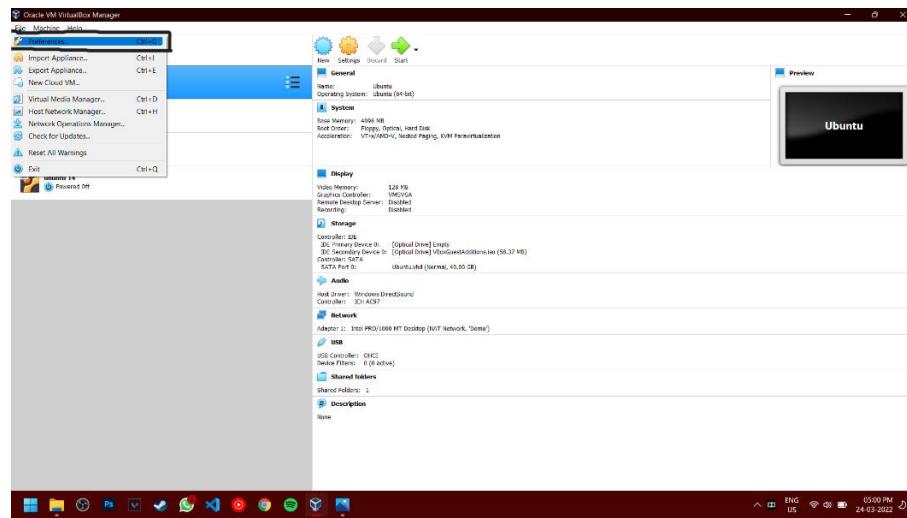


- Now enter the password for Ubuntu(left machine) admin, after enter the password, the file will be sent from Kali Linux(Right machine) to Ubuntu (Left machine)

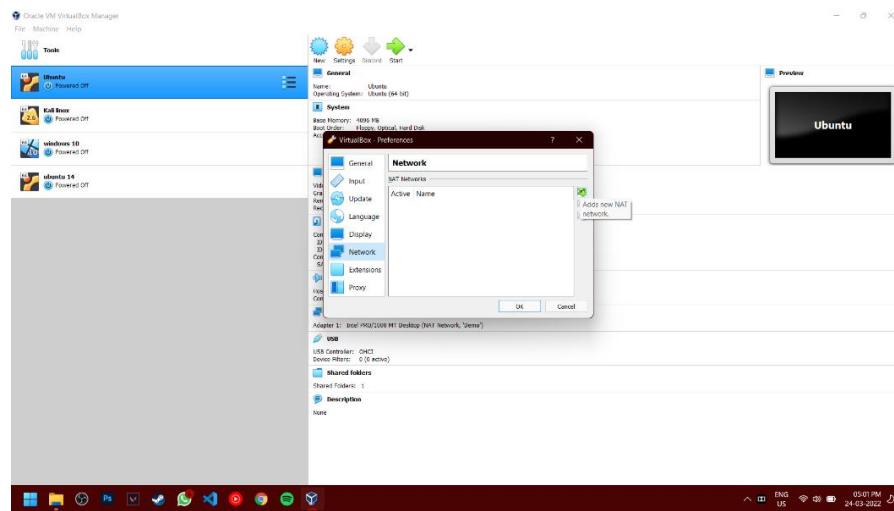


3. Create a NAT network in which 2 virtual machine can communicate.

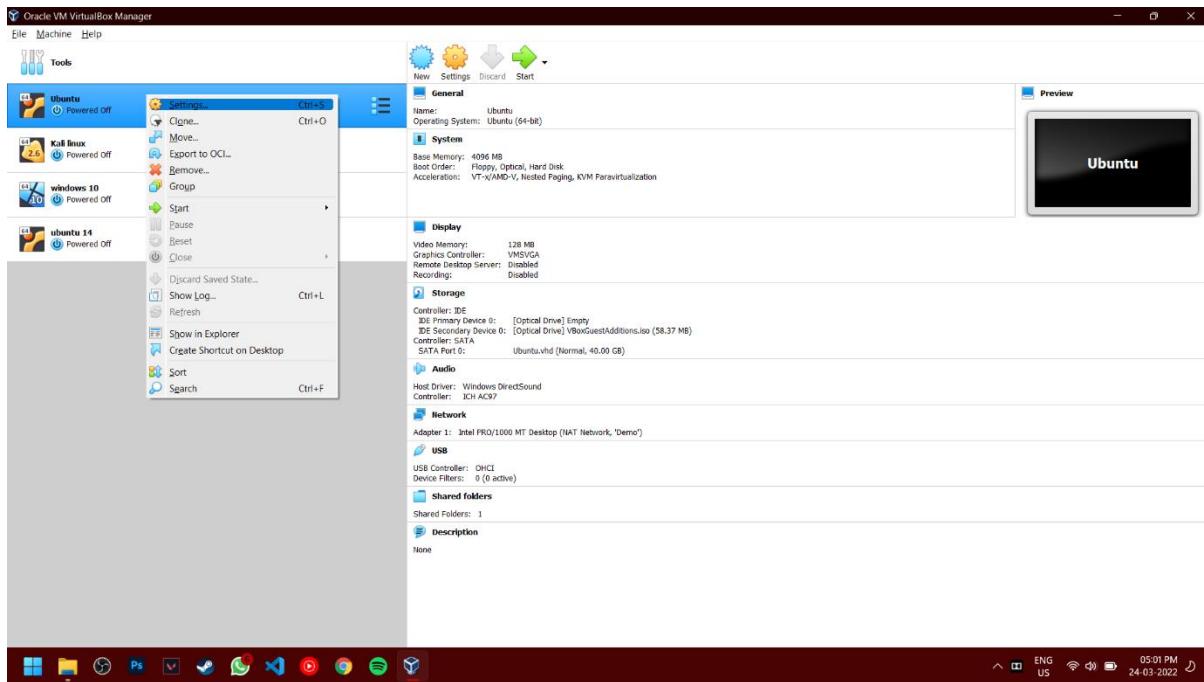
- Go to preferences by clicking File option in Top.



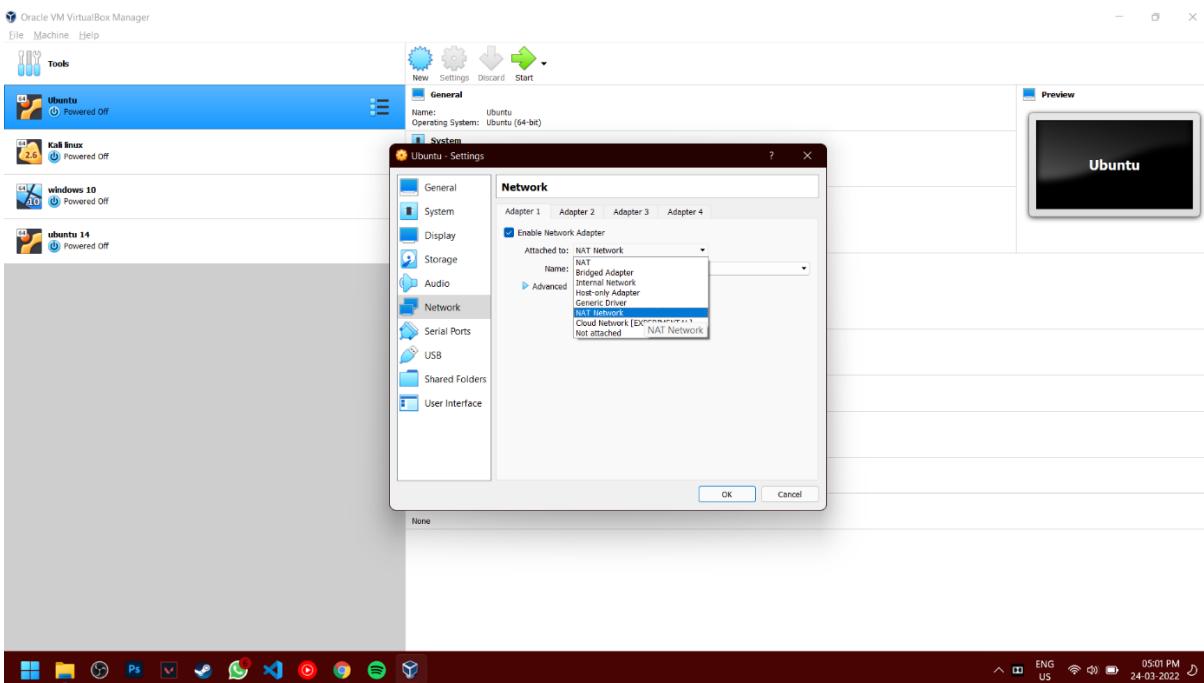
- Now select the network option and create a new NAT network

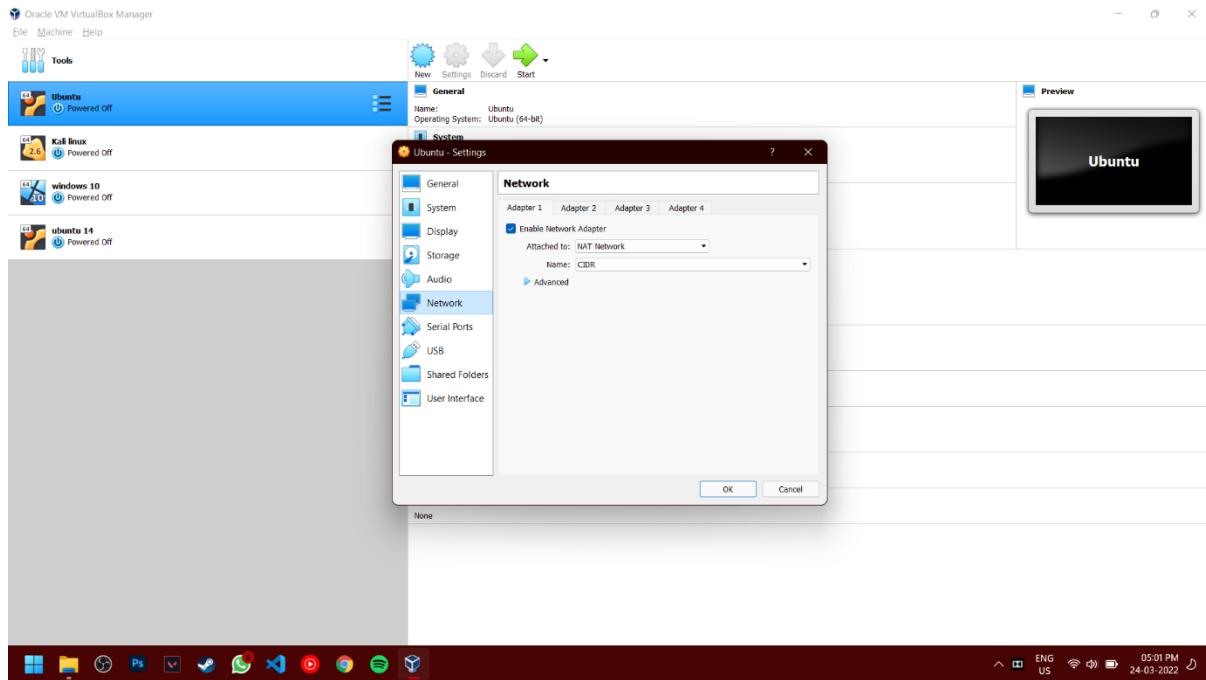


- After creating a NAT network, now go to virtual machine setting by right clicking on the preferred machine.



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

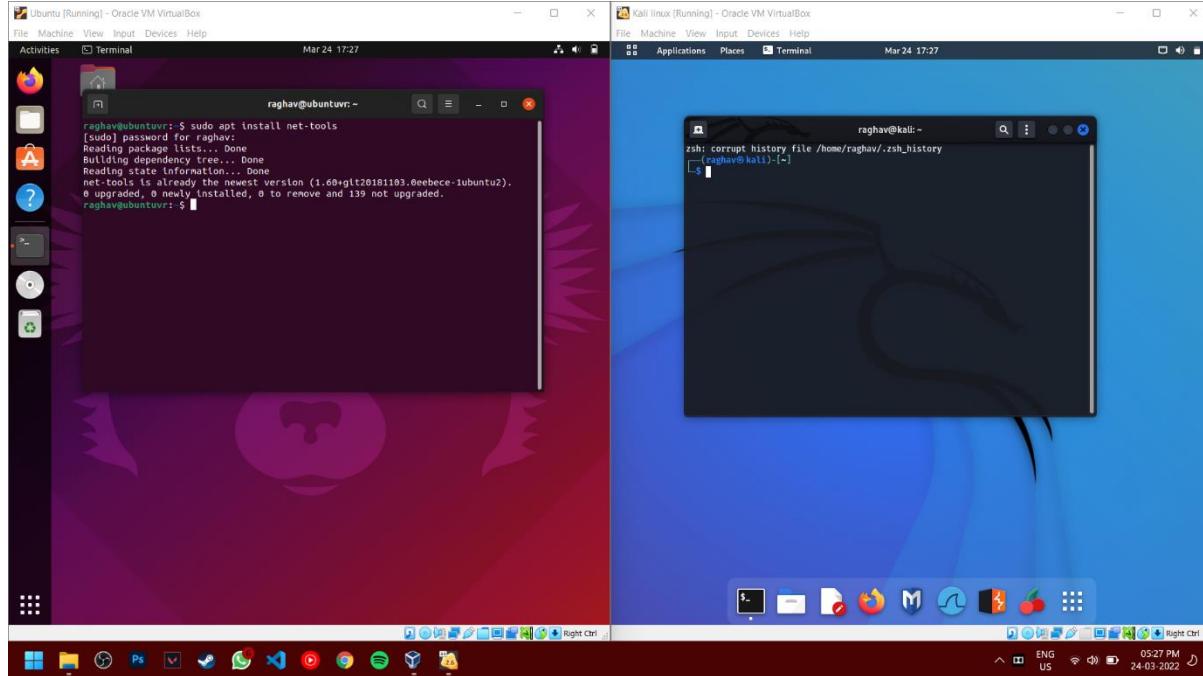




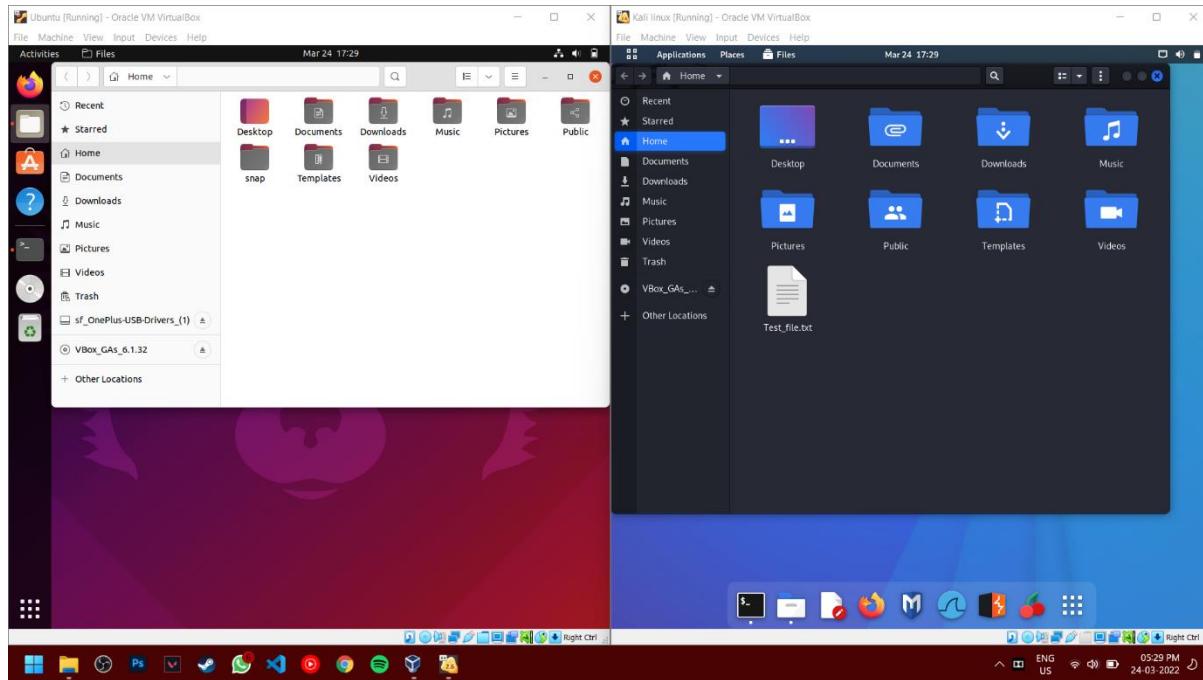
- Now repeat the same process for another machine.

4. Launch both virtual machines

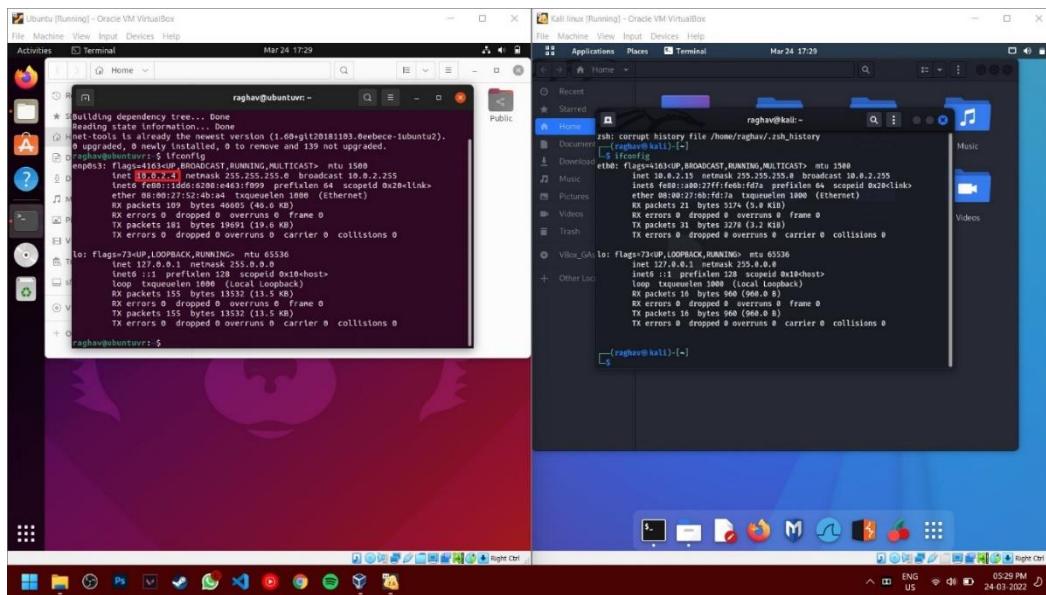
- Now install “Net-tools” on both machine which will help to identify i/p address of the machine.
Command: sudo apt install net-tools



- 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.



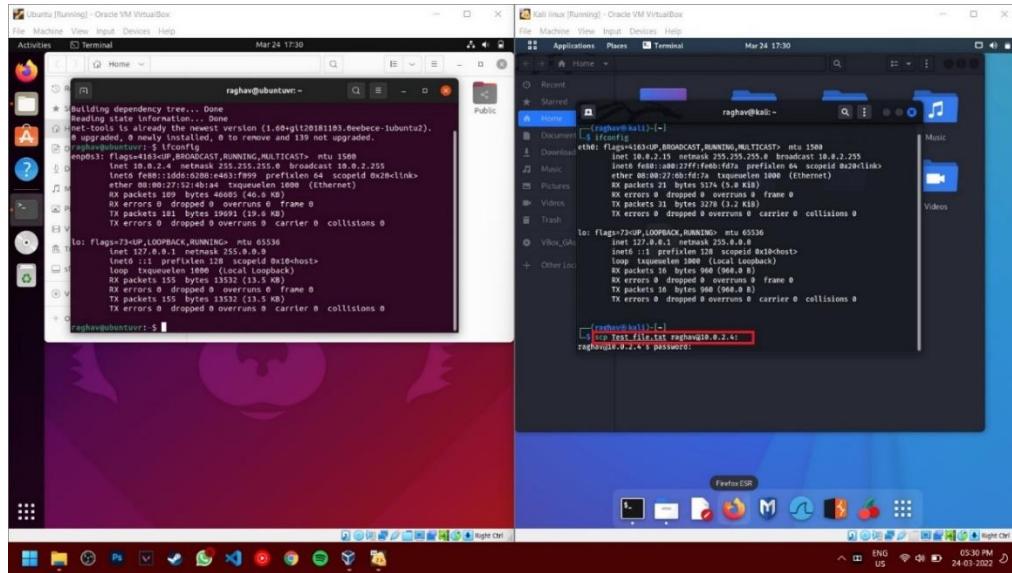
- Transfer the file using command- `scp Test_file.txt raghav@10.0.2.4`:

Where Test_file.txt is our file

raghav is username of Ubuntu

10.0.2.4 is ip address of Ubuntu (left machine)

Optional: if scp is not installed then install by using command:
 sudo apt install openssh-server



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).

FAQ:

1. What is Virtual Machine?
2. What are the different types of virtualization?
3. What is SCP command?

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:

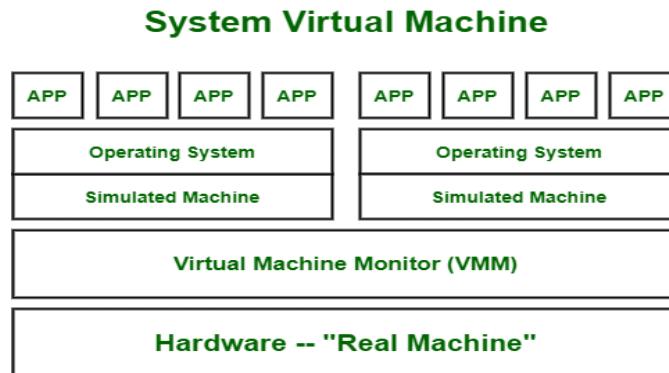
You can classify virtual machines into two types:

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.



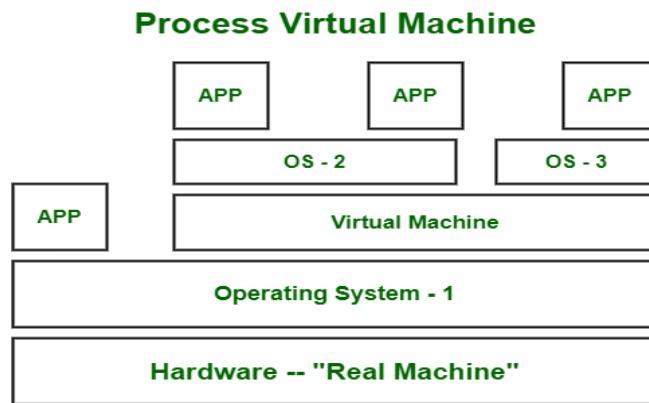
2. Process Virtual Machine:

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.

Deploying Openstack (private cloud) on AWS (public cloud)

Probably everyone with OpenStack hands-on experience would agree that sometimes it could be hard and frustrating to install it and test it. Especially if you do not own a small Data Center or the proper physical infrastructure to support its installation needs... However, wouldn't it be great to use a public cloud provider and its infrastructure to experiment and create POC environments?

This assignment provide the basic steps of how to spin up a Redhat OpenStack POC environment on AWS and the things you should modify in order to get OpenStack up and running probably on every cloud provider.

Step 0: Prerequisites

Hardware

Instance with at least 8GB RAM and 50GB storage, with hardware virtualization enabled AMI



RHEL-7.6_HVM_GA-20190128-x86_64-0-Hourly2-GP2 - ami-000db10762d0c4c05
Provided by Red Hat, Inc.
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
64-bit (x86)
General purpose t2.xlarge 4 16 EBS only - Moderate Yes

Software

As we are installing RedHat Openstack, We will need RHOSP iso file. Generally RedHat asks for a subscription to download this file. [COPY LINK DRIVE](#)

Network

We need to disable NetworkManager and firewall as they manage network interfaces and the firewall, and Neutron wants to manage them as well. When there are two managers that don't know about each other, you can perhaps imagine that the result is chaos.

```
$ sudo systemctl disable NetworkManager$ sudo systemctl stop NetworkManager
$ systemctl disable firewalld$ systemctl stop firewalld
$ su
```

Step 1: Installation

I used *gdown* command to download the required iso files inside the instance.

```
sudo yum install python3 -y
sudo pip3 install gdown#gdown needs the file ID to download files from gdrive
the ID can be found in the Gdrive link after file/d/ till /viewgdown --id ID --output rhosp.iso
```

Now, after downloading the Iso file, lets copy the contents to another folder. For this, we have to mount the iso file first

```
mkdir temp
mount rhosp.iso temp
cp -rvf temp/ openstack/
openstack temp
```

After this, we have to create a repo inside the folder

```
sudo yum install createrepo
cd openstack
createrepo -v .
```

then we have to write a repo file for the newly created repo

```
vi /etc/yum.repos.d/local.repo[openstack]
baseurl=file:///home/ec2-user/openstack
gpgcheck=0
```

We can check the created repo using *yum repolist*, If the repo isn't shown use *yum clean all*.

```
[root@ip-172-31-51-195 ec2-user]# yum repolist
Loaded plugins: amazon-id, rhui-lb, search-disabled-repos
Repository 'openstack' is missing name in configuration, using id
repo id          repo name      status
openstack        openstack      2,126
rhui-REGION-client-config-server-7/x86_64   Red Hat Update Infrastru 10
rhui-REGION-rhel-server-releases/7Server/x86_64 Red Hat Enterprise Linux 29,219
rhui-REGION-rhel-server-rh-common/7Server/x86_64 Red Hat Enterprise Linux 243
repolist: 31,598
```

Install Openstack using yum install opensack-packstack

Generate	answer	file	using
packstack --gen-answer-file=openstack.txt	After generating,	Run	answer file using
packstack --answer-file=openstack.txt			

```
root@ip-172-31-51-195:/home/ec2-user - □ ×
Preparing Nova VNC Proxy entries [ DONE ]
Preparing OpenStack Network-related Nova entries [ DONE ]
Preparing Nova Common entries [ DONE ]
Preparing Neutron LBaaS Agent entries [ DONE ]
Preparing Neutron API entries [ DONE ]
Preparing Neutron L3 entries [ DONE ]
Preparing Neutron L2 Agent entries [ DONE ]
Preparing Neutron DHCP Agent entries [ DONE ]
Preparing Neutron Metering Agent entries [ DONE ]
Checking if NetworkManager is enabled and running [ DONE ]
Preparing OpenStack Client entries [ DONE ]
Preparing Horizon entries [ DONE ]
Preparing Swift builder entries [ DONE ]
Preparing Swift proxy entries [ DONE ]
Preparing Swift storage entries [ DONE ]
Preparing Gnocchi entries [ DONE ]
Preparing Redis entries [ DONE ]
Preparing Ceilometer entries [ DONE ]
Preparing Aodh entries [ DONE ]
Preparing Puppet manifests [ DONE ]
Copying Puppet modules and manifests [ DONE ]
Applying 172.31.51.195_controller.pp
Testing if puppet apply is finished: 172.31.51.195_controller.pp [ | ]
Testing if puppet apply is finished: 172.31.51.195_controller.pp [ | ]
```

We have successfully installed Openstack on AWS

Connecting to Horizon

As Horizon runs on apache web server , We need to configure Apache to use public IP of the instance for accessing openstack.

```

# *****
# Vhost template in module puppetlabs-apache
# Managed by Puppet
# *****

<VirtualHost *:80>
  ServerName ip-172-31-51-195.ec2.internal

  ## Vhost docroot
  DocumentRoot "/var/www/"
  ## Alias declarations for resources outside the DocumentRoot
  Alias /dashboard/static "/usr/share/openstack-dashboard/static"

  ## Directories, there should at least be a declaration for /var/www/
  <Directory "/var/www/">
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Require all granted
  </Directory>

  ## Logging
  ErrorLog "/var/log/httpd/horizon_error.log"
  ServerSignature Off
  CustomLog "/var/log/httpd/horizon_access.log" combined

  ## RedirectMatch rules
  RedirectMatch permanent ^/$ /dashboard

  ## Server aliases
  ServerAlias 54.89.193.220
  ServerAlias ip-172-31-51-195.ec2.internal
  ServerAlias localhost
  WSGIApplicationGroup %(GLOBAL)
  WSGIDaemonProcess apache display-name=horizon group=apache processes=3 threads=10 user=apache
  WSGIProcessGroup apache
  WSGIScriptAlias /dashboard "/usr/share/openstack-dashboard/openstack_dashboard/wsgi/django.wsgi"
</VirtualHost>

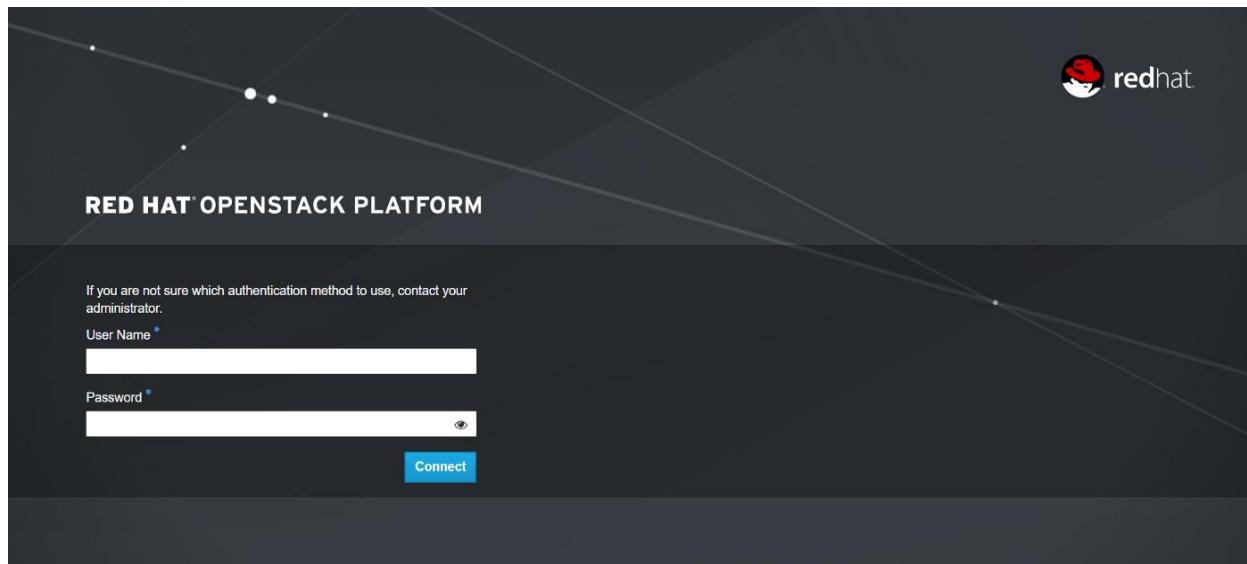
```

Here, we have to change ServerAlias from private IP to public IP.

After that, just restart the webserver using

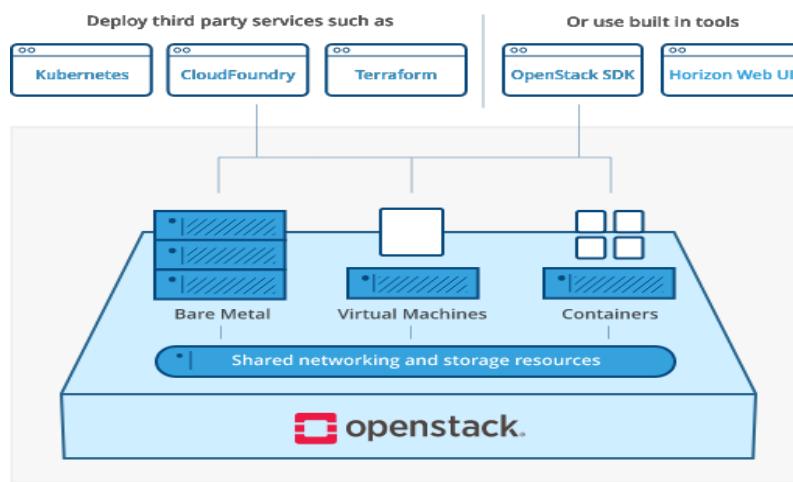
`systemctl restart httpd`

Now, We can access Openstack in our web browser using public IP of instance



What is Open Stack?

OpenStack is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a datacenter, all managed and provisioned through APIs with common authentication mechanisms.



Vision for OpenStack Clouds :

- Purpose : This is a living document. Its purpose is to document the OpenStack community's vision for the output of the OpenStack project as a whole, as it evolves over time.
- Scope : The scope of this document is limited to the cloud services that an end-user interacts with. This corresponds to the main 'OpenStack' bucket and parts of the 'OpenStack Operations' bucket in the OpenStack project map.
- The Pillars of Cloud : There are at least as many different opinions of what 'cloud' means as there are software developers. However, we can all agree that cloud does mean something. Cloud computing promotes more efficient utilization of resources by reducing the transaction costs involved in provisioning and deprovisioning infrastructure to near zero, and it is able to do so because it differs in qualitative ways from previous models of computing (including virtualization).
- Self-service : Clouds are self-service. They provide users with the ability to deploy applications on demand without having to wait for human action or review in the loop. The cloud has no ticket trackers.

Application Control : Clouds allow control of an application's infrastructure to be vested in the application itself. Just as clouds eliminate the need for a human approver in the loop, they also eliminate the need for a human user to be in the loop. While a cloud may have a user interface (graphical or otherwise), it must have an application programming interface.

What is Trystack?

TryStack is a free and easy way for users to try out OpenStack, and set up their own cloud with networking, storage, and computer instances. If you haven't tried it yet, go check it out now—we'll be here when you get back. TryStack is an OpenStack Foundation project; they oversee the political side of it. There are different vendors and companies that have donated resources, such as rack and power and network and servers. At the time we got involved, there wasn't much maintenance being done on the software side of it. I think it was running Ubuntu at the time, and it was running Essex I think. And, so Red Hat came in and donated RHEL subscriptions to run the operating system underneath it, and said they'd help us run RDO on top of it. So, we have a

RPM packaging of OpenStack running on Red Hat Enterprise Linux, and we're donating my team's time to manage that. We went through and installed the whole cluster with RHEL, and put the initial release of RDO, which was Folsom, onto that cluster and got it up and running. At that point we had RDO available to the community to use, so there's a block of IP addresses and a block of servers in a datacenter out in San Jose, and Red Hat is managing those servers and running our software on it, to showcase our community-based OpenStack offering.

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

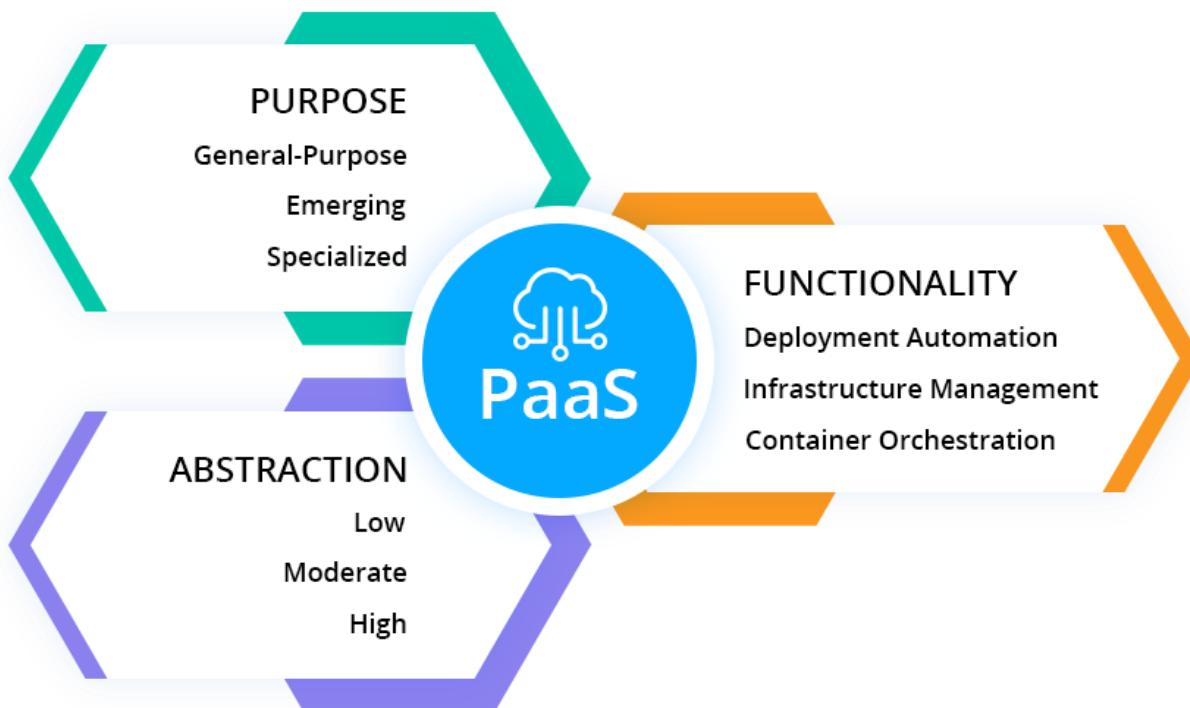
Connect to your Instance (either via SSH) Terminate Your Instance (go to the console and select the VM under actions and terminate).

Conclusion: Thus, 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

Steps

1. <https://www.awseducate.com/registration/s/>

2. Register yourself in "Learn Cloud Skills" by entering email id

3.

The screenshot shows the AWS Educate homepage with the 'Courses & Labs' tab selected. Under 'Most popular courses and labs', there are four cards:

- Introduction to the AWS Management Console (Foundational | 1 hour, Cloud Computing)
- Builder Labs (Intermediate | 3 hours, Development)
- Introduction to Cloud 101 (Labs) (Foundational | 8 hours, Cloud Computing) - This card is circled in red.
- Machine Learning Foundation (Lab) (Foundational | 3 hours, Machine Learning & AI)

4.

The screenshot shows the 'Module 4 AWS Core Services' page. On the left sidebar, 'Courses' is selected. The main content area lists several lab modules:

- AWS Core Services
- AWS Core Services Labs
- Lab 1 - Introduction to Amazon S3 (100 pts | Score at least 70.0)
- Lab 2 - Introduction to Amazon EC2 (100 pts | Score at least 70.0) - This link is circled in red.
- Lab 3 - Introduction to Amazon DynamoDB (100 pts | Score at least 70.0)
- Lab 4 - Introduction to Amazon Relational Database Service (Amazon RDS) (100 pts | Score at least 70.0)

5.

6. Follow all instructions as given by them, don't change default settings

7. After following the steps at the end your instance will be in running state

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

FAQ's:

- 1). What are benefits of PAAS.
- 2). What is cost of PAAS.

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.

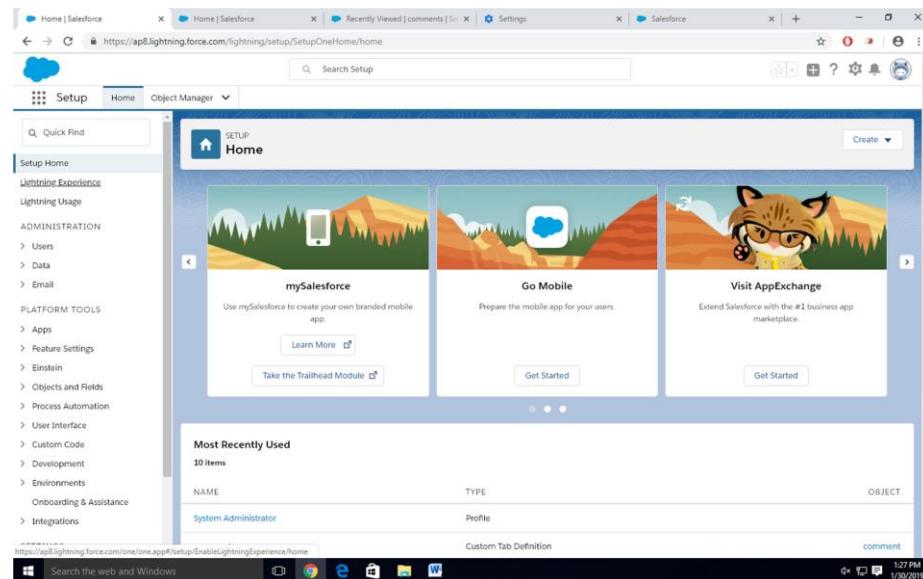
Salesforce cloud CRM software and applications are used for Sales, Service, Finance, Marketing, Business Development, Recruiting, HR, Supply Chain Management, and other lines of business that manage external and internal relationships. It allows storing Existing Customers and Potential Customer's information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

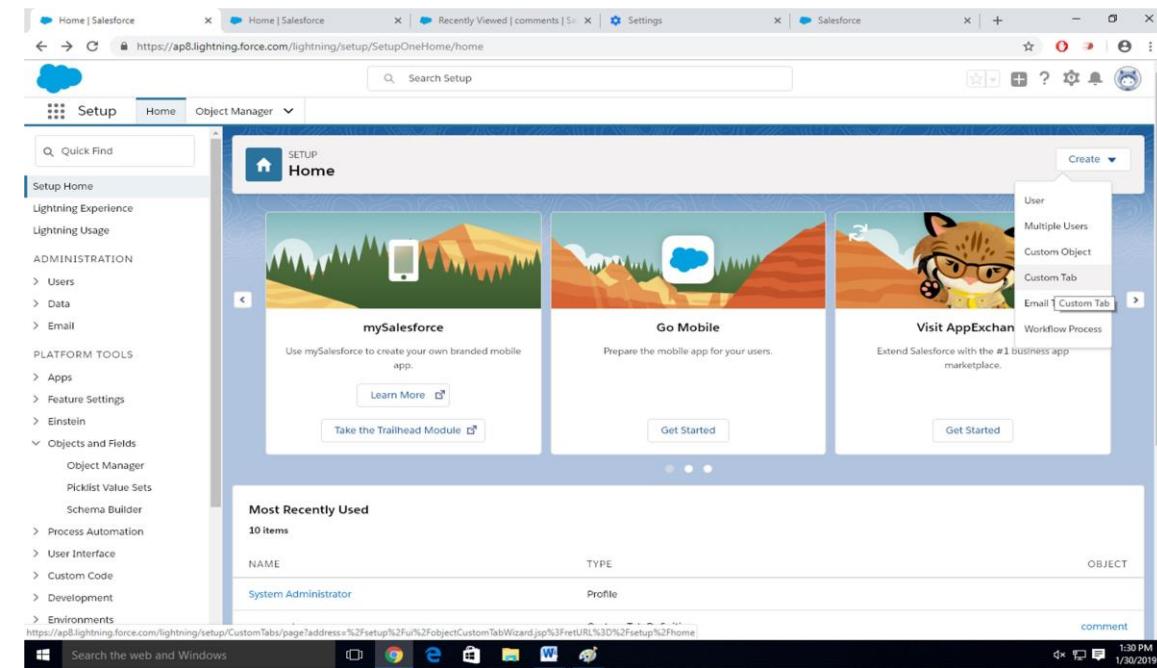
Step-2: Click on Setup and select Setup for current App.

Step-3:

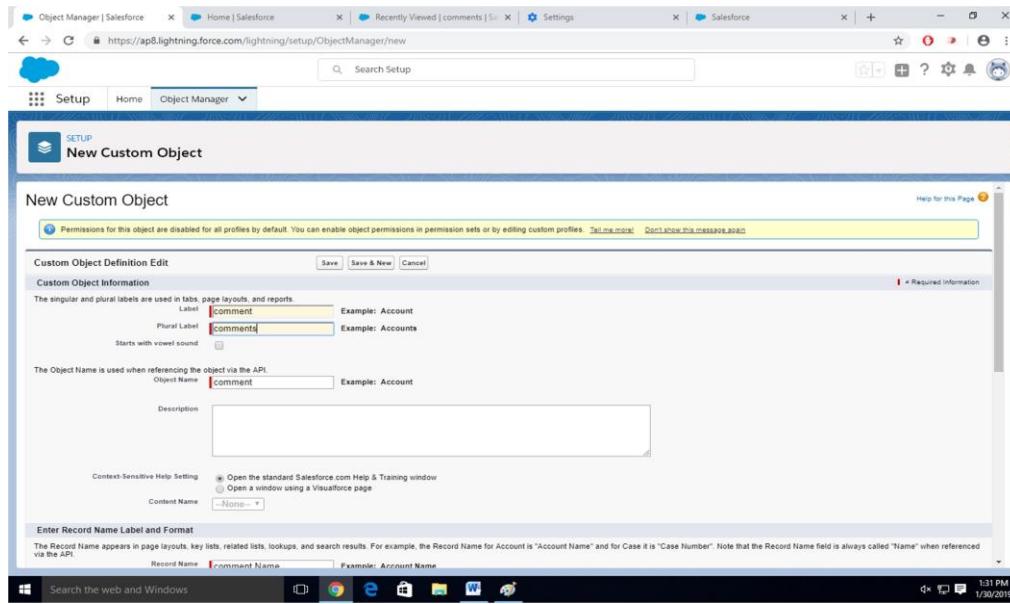


Click on Create an Object

So Click on Object Manager Tab next to Home Tab



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

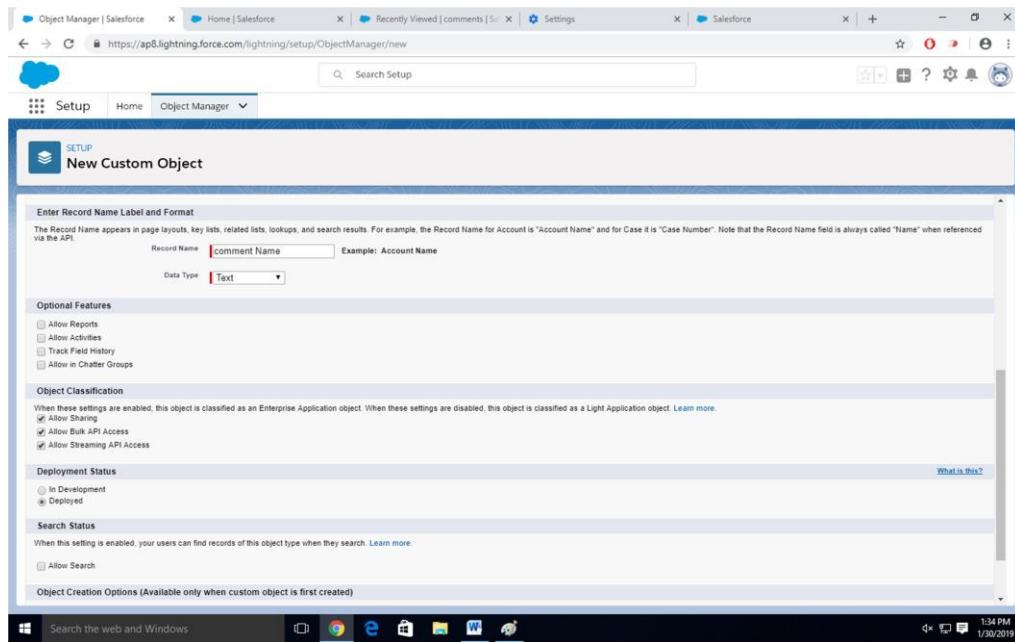


Give Record Name as —comment name

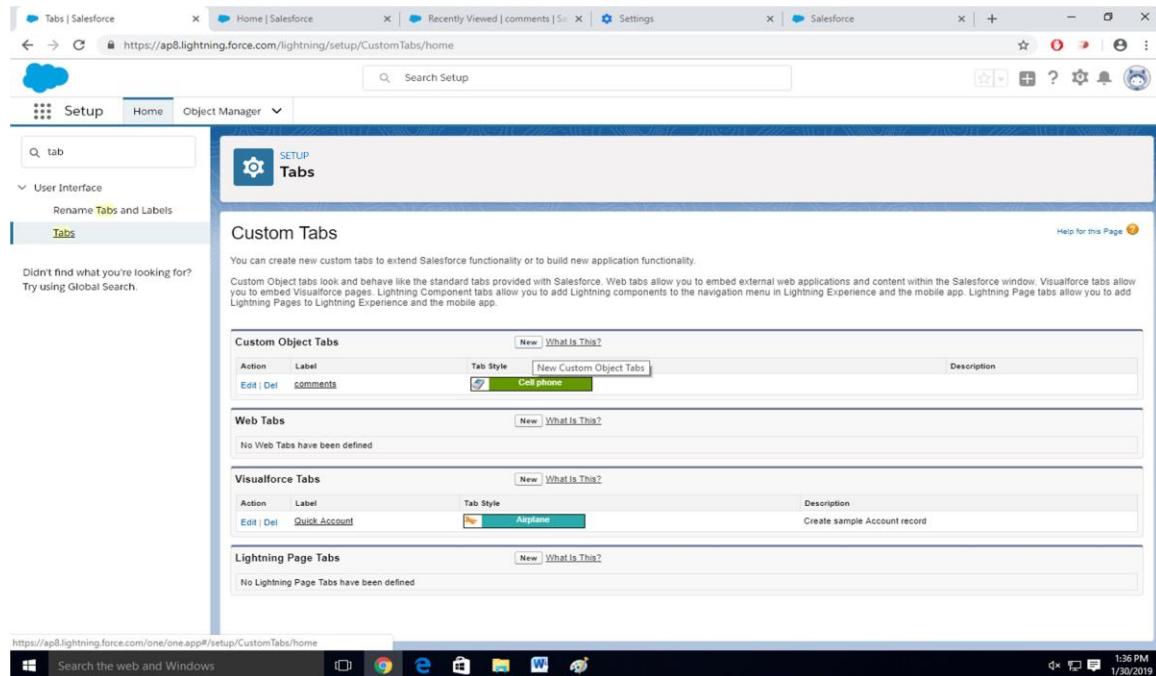
Data type- text

Select Allow Reports Check Box

Click on Save

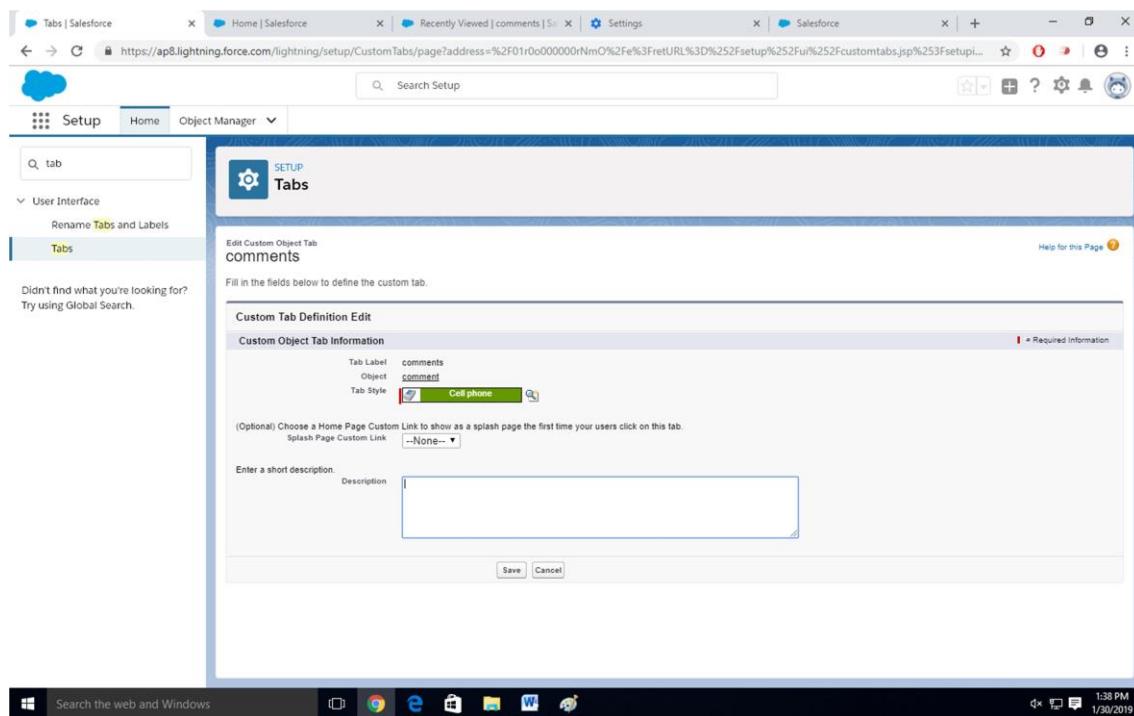


Step-5: Click on Home-Search Tabs in Quick search Select Custom Object-Click on New

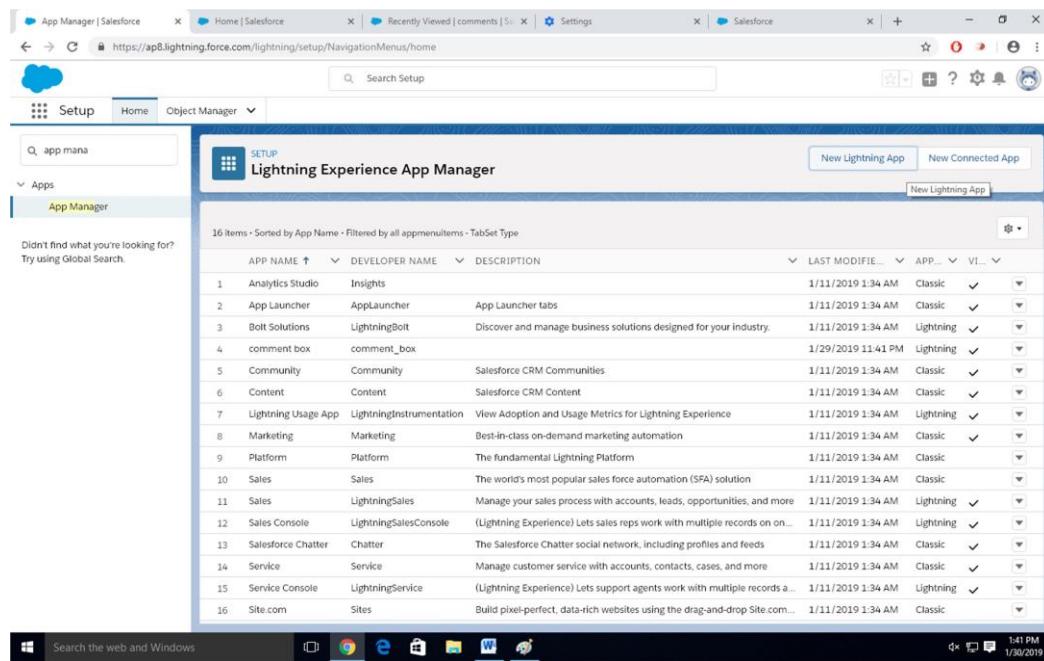


Step-6: For Object Select Comment For Tab Style Select Any Icon

Click-Next-Next-Save



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



Enter name to app name

New Lightning App

App Details & Branding

Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.

App Details

* App Name Complete this field.

* Developer Name

Description

App Branding

Image Upload

Primary Color Hex Value #0070D2

Org Theme Options Use the app's image and color instead of the org's custom theme

App Launcher Preview

Next

Click on Next-Next-Next.

Select Items (Contacts, Comment)

Click on Next

New Lightning App

Available Items

Selected Items

Accounts

Back Next

15 Service Console LightningService (Lightning Experience) Lets support agents work with multiple records a... 1/11/2019 1:34 AM Lightning ✓

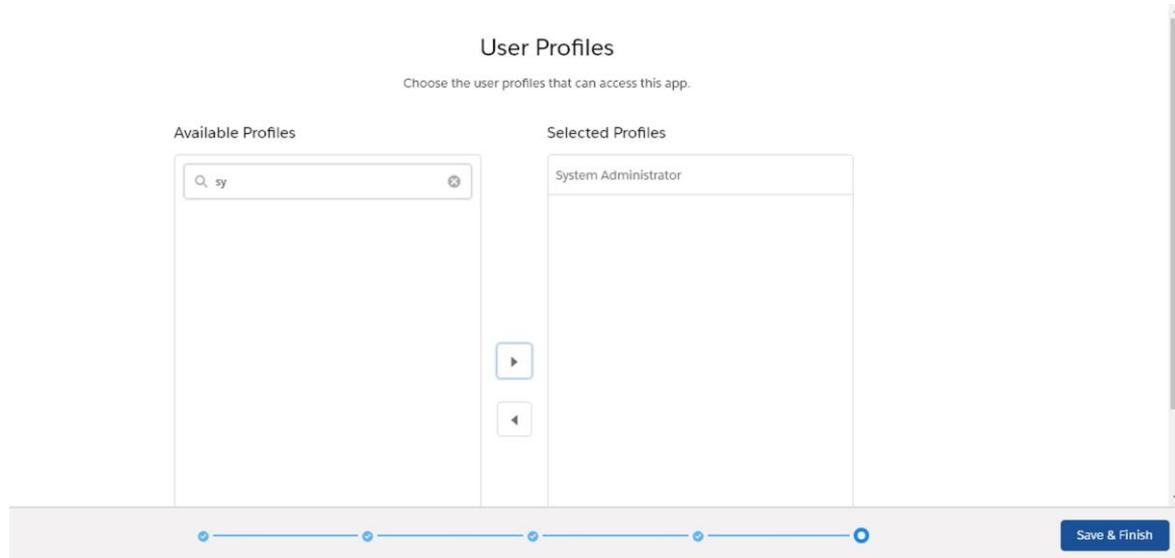
16 Site.com Sites Build pixel-perfect, data-rich websites using the drag-and-drop Site.com... 1/11/2019 1:34 AM Classic

Search the web and Windows

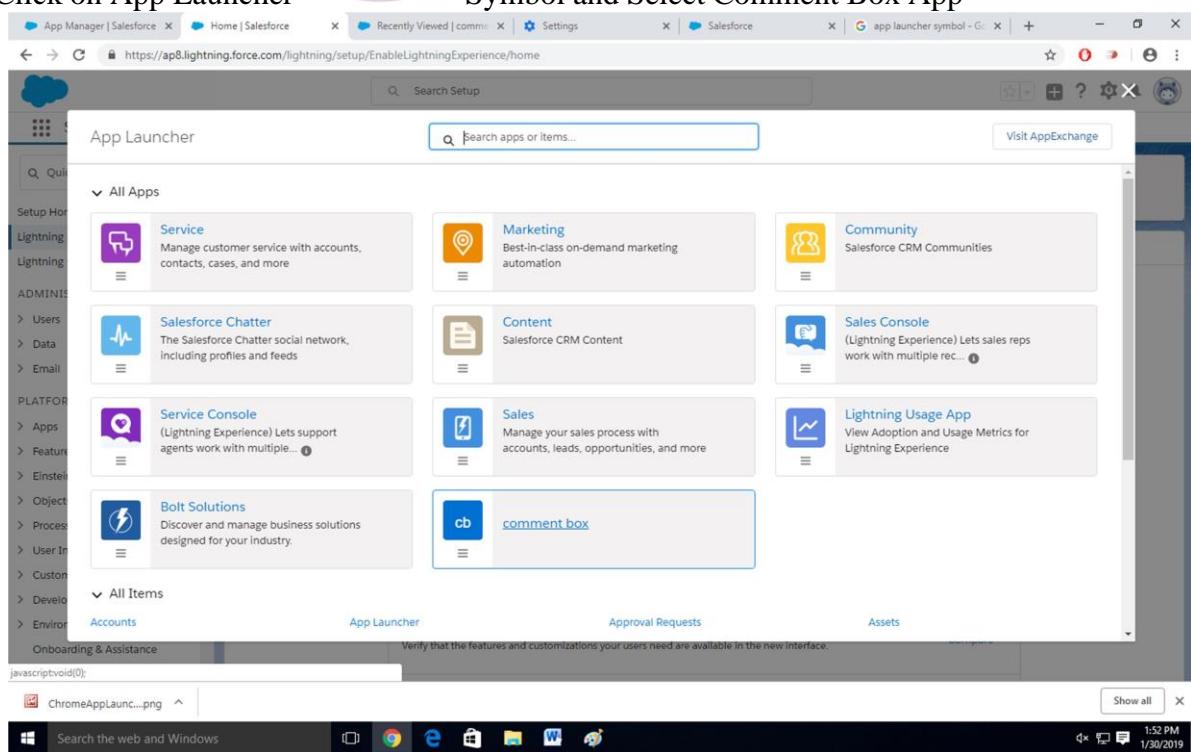
1:44 PM 1/30/2019

Step-8

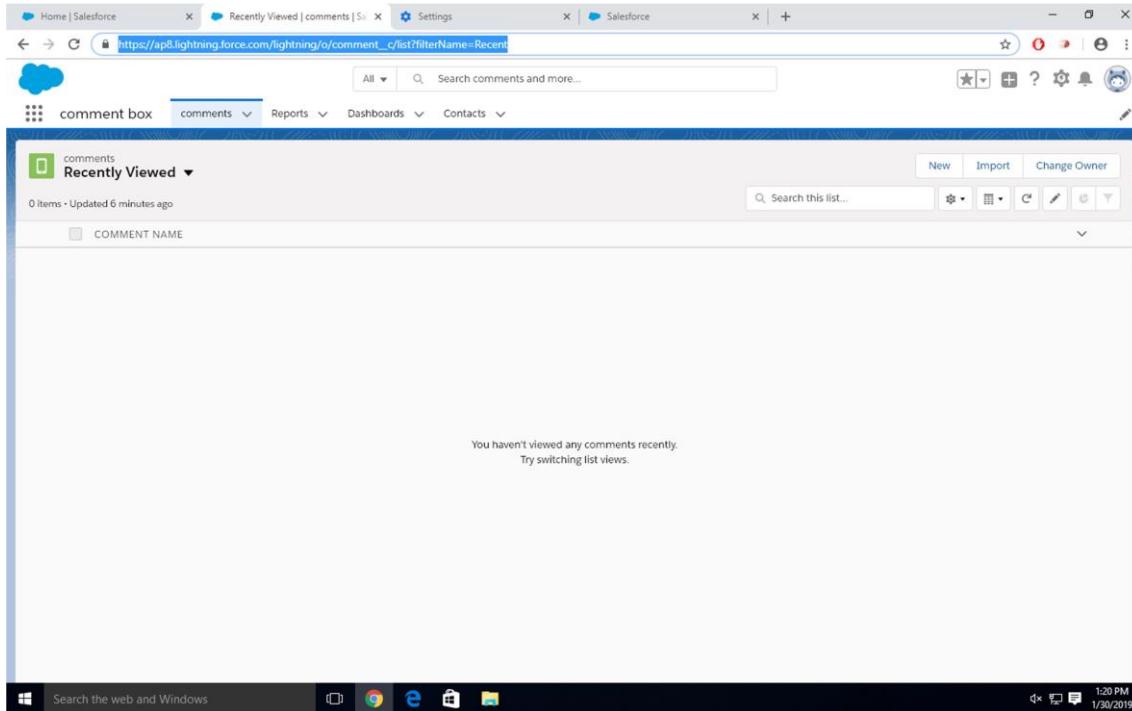
Select Profiles (System Administrator) and move to selected profile. Click on Save and Finish.



Step-9: Click on App Launcher

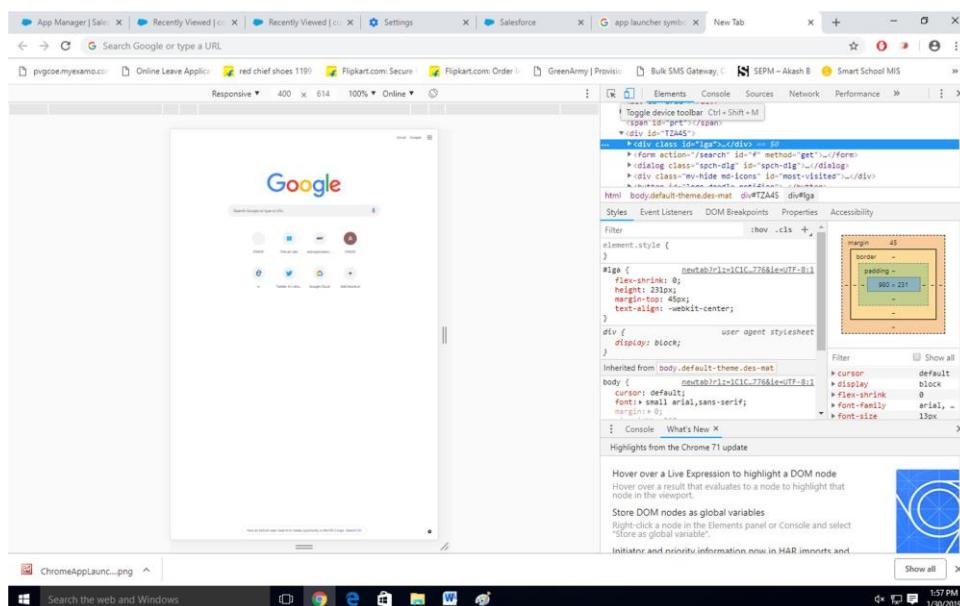


Step-10: Tour the app



Step-11: Try out mobile app

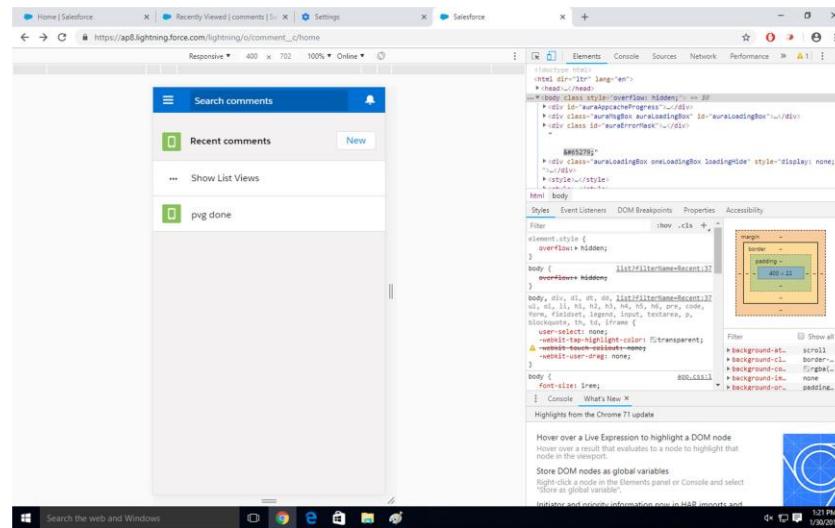
- Select Chrome developer tools
- Open Chrome-Right Click on Chrome page- Select Inspect
- Click Toggle Device Mode Button to simulate your browser as a mobile device



Step-12

To simulate the sales force mobile app in your browser, copy and paste in urlfrom previous tab. Delete the part of the url immediately.

- Click on Left navigation bar
- Find comment object under recent and click on it
- Click new to create a comment



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

FAQ's:

- 1). Is it required to move entire org at once?
- 2). Can anyone stay on SalesForce temporarily?

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.firebaseio/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.firebaseio/firenotes/frontend/main.js](#)

[appengine/standard.firebaseio.firenotes/frontend/main.js](#)

[VIEW ON GITHUB](#)

```
// 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.firebaseio/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.firebaseio/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.

Provider	Status
Email/Password	Disabled
Google	Disabled
Facebook	Disabled
Twitter	Disabled
Github	Disabled
Anonymous	Disabled

- 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**.

Facebook

Enable

App ID

App secret

To complete set up, add this OAuth redirect URL to your Facebook app configuration. [Learn more](#)

https://test-84e93.firebaseio.com/_/auth/handler

CANCEL

- In the Firebase console, under **Authorized Domains**, click **Add Domain** and enter the domain of your app on App Engine in the following format:
[PROJECT_ID].appspot.com

Do not include http:// before the domain name.

Reference:

<https://cloud.google.com/appengine/docs/standard/python/authenticating-users-firebase-appengine>

Case Study-1

1. What is data storage in private cloud?

Private cloud storage is a type of storage mechanism that stores an organization's data at in-house storage servers by implementing cloud computing and storage technology. Private cloud storage is similar to public cloud storage in that it provides the usability, scalability and flexibility of the storage architecture. But unlike public cloud storage, it is not publicly accessible and is owned by a single organization and its authorized external partners.

Private cloud storage is also known as internal cloud storage.

Private cloud storage works much like public cloud storage and implements storage virtualization across an organization, providing a centralized storage infrastructure that can only be accessed by the authorized nodes.

Private cloud storage operates by installing a data center, which houses a series of storage clusters that are integrated with a storage virtualization application. Administrative policies and a management console provide access to the different storage nodes and applications within the organization's network. The applications or nodes access the private storage through file access and data retrieving protocols, while the automated storage administrator application allocates storage capacity to them on run time.

Private cloud storage has a multitenant architecture, where a single storage array can house storage space to multiple applications, nodes or departments.

2. Examples of data stored on cloud (success stories)

a. IT Priorities 2020: Covid-19 consolidates storage push to cloud

The emphasis on home and remote working has been felt in storage and backup, as in all areas of IT, with continued evolution towards the cloud and as-a-service storage and backup

b. Big storage meets cloud in the datacentre 'as-a-service' revolution

Cloud is mainstream but the datacentre is here to stay. This has resulted in a trend towards as-a-service in the datacentre where big storage array makers and the cloud giants meet.

3. Do you use cloud storage for these use cases yet?

We look at the use cases most suited to a quick transition to the cloud: backup, archiving, disaster recovery, file storage and cloud bursting – cloud storage's low-hanging fruit.

4. Cloud storage costs: How to get cloud storage bills under control?

Many organizations look to the cloud to cut storage budgets, but the potential costs are many and varied. So, what are the key ways to cut cloud storage costs?

5. Cloud storage 101: File, block and object storage in the cloud

We run the rule over file, block and object storage from the big three public cloud providers: AWS, Azure and GCP. We look at what's on offer and the use cases they are aimed at

6. Procedures/ ways to upload data to cloud (are there any applications designed)?

Here's how to build a cloud-ready application, and the application architecture concepts you'll need to succeed. The process presented below represents a step-by-step approach that mixes traditional software development concepts and reviews what's new with the cloud. It is a collection of best practices, concepts, and procedures for success.

Design the application as a collection of services

Cloud applications are best deployed as a collection of cloud services, or APIs. You build up from the data to the services and then combine those services into composite services or complete composite applications.

This is service-based or service-oriented architecture, at its essence. While many understand the concepts, developers still have a tendency to create tightly coupled applications that focus on the user interface, rather than expose the underlying functions as services they can leverage independently.

When developing an application architecture for the cloud, you deal with complex distributed systems that can take advantage of loosely coupled applications built on many services that can also be decoupled from the data (see "Decouple the data" below). You can separate the application services physically, executing on the proper machine instances, and service/API managers and governance technology that provide services directories can help track the many services that make up your application.

Additional benefits may include service reuse from other applications or more coarse-grained services. You can break up applications into hundreds of underlying services that have value when used by other applications. In this way, you're not reinventing the wheel each time you build an application. Consider the example of a credit-check service that many applications use. Combine these into a single service and the application becomes much more efficient.

Decouple the data

If you tightly couple the data to the application, it won't find a good home in the cloud. Private and public clouds are complex distributed systems that work best with application architectures that break out processing and data into separate components.

You decouple the data for the same reason you want to build the application out of services. Once decoupled, you have the option to store and process the data on any public or private cloud instance. For example, many enterprises insist that their data remain on local servers but want to take advantage of the commodity virtual machine instances within a public cloud.

You must consider performance. Database reads and writes across the open Internet can cause latency, and database communications may determine how close your data sits to the services and applications that need to leverage it.

Consider using caching systems. These provide additional database performance by locally storing commonly accessed data, thereby reducing all database read requests back to the physical database. Those are best built into the application, however, and they should be tested with the application data to determine how efficient the cache will be. Systems that are constantly reading new data don't benefit as much from database caches.

Consider communications between application components

Decoupling applications, both data and services, doesn't mean your application is properly architected for the cloud. Chatty application components that constantly communicate with each other will lower the performance of the overall application, given that they're typically distributed over a network or the open Internet, where tolerance for high latency is desirable.

Focus on designing applications that optimize communications between application components. For example, combine communications into a single stream of data or a group of

messages, rather than constantly communicating as if the application components reside on a single platform.

Model and design for performance and scaling

Extend considerations around how application components communicate to include overall performance as well. This includes understanding how the application will scale under an increasing load.

Designing for performance means first building a model that represents how the application behaves under an increasing load. If 1,000 or more users log on at the same time, how will the application handle the increased traffic on the network, the increased load on the application servers, and the load placed on the back-end databases? You need to understand how application components handle the load as the number of users increases to 1,000 or more users.

This example might increase the load on the application servers by 80 percent, the load on the network by 10 percent, and the load on the database by 40 percent. Given that, adding 1,000 more users will likely saturate the application servers you've provisioned, and you'll need to spin up more application server instances. The network capacity might remain the same, but the number of database instances may have to increase to handle any additional load.

Armed with this model, you can figure out how best to scale the application by automatically spinning up resource instances that are needed. In some cases, cloud service providers offer auto-scaling capabilities, where provisioning occurs automatically. The most efficient path, however, lies in understanding the application's workload profile and defining the path to scaling the application, as well as putting mechanisms in place to ensure that it will, indeed, scale.

Finally, monitor overall application performance using application-aware performance monitoring tools, and create interfaces within the application to better enable performance monitoring. How the application provisions and de-provisions resources should be innate to the application as well.

Make security systemic within the application

For most people who build applications, security is typically an afterthought. When hosting an application in the cloud, however, security should be a high priority. Your cloud-based application architecture should make security systemic to the application—it should be designed and built into the application architecture.

Pick a security approach and technology prior to building your application that will be effective for the type of application you're running and that will address any compliance or other data-level security issues. If you're in healthcare, for example, you must consider personally identifiable information as well as the Health Insurance Portability and Accountability Act (HIPAA) in the US. You'll need to store data in a certain way, on clouds that are HIPAA-compliant. Moreover, the application will need to handle sensitive data in specified ways, with required levels of security, such as encryption.

Generally speaking, cloud-based applications should leverage identity and access management (IAM). Enterprises that develop mature IAM capabilities can reduce their security costs and, more importantly, become significantly more agile at configuring security for cloud-based applications. Indeed, IAM will be a part of more than 50 percent of existing applications that migrate to the public cloud and nearly 90 percent of new applications built on clouds.

What's more, the use of IAM within cloud application deployments will backfill into the enterprise, as these organizations modernize security approaches and technologies to align with

the use of public clouds. In many cases, IAM will be provided as a service to the enterprise. This concept of cloud-delivered IAM quickly leads to the concept of centralized identity management. As you build more cloud-based applications using IAM, each application should become significantly more secure and more cost effective.

Your core objective is to design security into the application and take advantage of the native features of both the cloud and the IAM system you use. However, each application has its own requirements based upon the needs of the business, and security always differs from one enterprise to another.

Building a cloud-ready application architecture requires that you pay attention to a few new things, but many of the traditional concepts are still important, such as sound design, testing, and learning from your mistakes. Most developers who deploy applications on private or public cloud platforms will make some blunders, but as long as they recognize, correct, and learn from those mistakes, they'll be well on their way to finding a more effective path to building applications in the cloud.

Understand that approaches such as service orientation should be given priority, even if it means longer initial application development lifecycles and bigger budgets. Even though you'll pay more for application development in the cloud than you did for traditional application development, the investment in services pays huge dividends year in and year out. It's a smart investment.

7. Security algorithm in place for data security in cloud (list , brief differences , latest standard followed)

Cloud computing security or, more simply, cloud security refers to a broad set of policies, technologies, and controls deployed to protect data, applications, and the associated infrastructure of cloud computing. It is a sub-domain of computer security, network security, and, more broadly, information security.

Cloud computing security is the set of control-based technologies and policies designed to adhere to regulatory compliance rules and protect information, data applications and infrastructure associated with cloud computing use.

Because of the cloud's very nature as a shared resource, identity management, privacy and access control are of particular concern. With more organizations using cloud computing and associated cloud providers for data operations, proper security in these and other potentially vulnerable areas have become a priority for organizations contracting with a cloud computing provider.

Cloud computing security processes should address the security controls the cloud provider will incorporate to maintain the customer's data security, privacy and compliance with necessary regulations. The processes will also likely include a business continuity and data backup plan in the case of a cloud security breach.

The following are 10 security-as-a-service categories are:

1. Identity and Access Management should provide controls for assured identities and access management. Identity and access management includes people, processes and systems that are used to manage access to enterprise resources by assuring the identity of an entity is verified and is granted the correct level of access based on this assured identity. Audit logs of activity such as successful and failed authentication and access attempts should be kept by the application/solution.
2. Data Loss Prevention is the monitoring, protecting and verifying the security of data at rest, in motion and in use in the cloud and on-premises. Data loss prevention services offer protection of data usually by running as some sort of client on desktops/servers and running rules around what can be done. Within the cloud, data loss prevention services could be offered as something that is provided as part of the build, such that all servers built for that client get the data loss prevention software installed with an agreed set of rules deployed.
3. Web Security is real-time protection offered either on-premise through software/appliance installation or via the cloud by proxying or redirecting web traffic to the cloud provider. This provides an added layer of protection on top of things like AV to prevent malware from entering the enterprise via activities such as web browsing. Policy rules around the types of web access and the times this is acceptable also can be enforced via these web security technologies.
4. E-mail Security should provide control over inbound and outbound e-mail, thereby protecting the organization from phishing and malicious attachments, enforcing corporate policies such as acceptable use and spam and providing business continuity options. The solution should allow for policy-based encryption of e-mails as well as integrating with various e-mail server offerings. Digital signatures enabling identification and non-repudiation are features of many cloud e-mail security solutions.
5. Security Assessments are third-party audits of cloud services or assessments of on-premises systems based on industry standards. Traditional security assessments for infrastructure and applications and compliance audits are well defined and supported by multiple standards such as NIST, ISO and CIS. A relatively mature toolset exists, and a number of tools have been implemented using the SaaS delivery model. In the SaaS delivery model, subscribers get the typical benefits of this cloud computing variant elasticity, negligible setup time, low administration overhead and pay-per-use with low initial investments.
6. Intrusion Management is the process of using pattern recognition to detect and react to statistically unusual events. This may include reconfiguring system components in real

time to stop/prevent an intrusion. The methods of intrusion detection, prevention and response in physical environments are mature; however, the growth of virtualization and massive multi-tenancy is creating new targets for intrusion and raises many questions about the implementation of the same protection in cloud environments.

7. Security Information and Event Management systems accept log and event information. This information is then correlated and analyzed to provide real-time reporting and alerting on incidents/events that may require intervention. The logs are likely to be kept in a manner that prevents tampering to enable their use as evidence in any investigations.

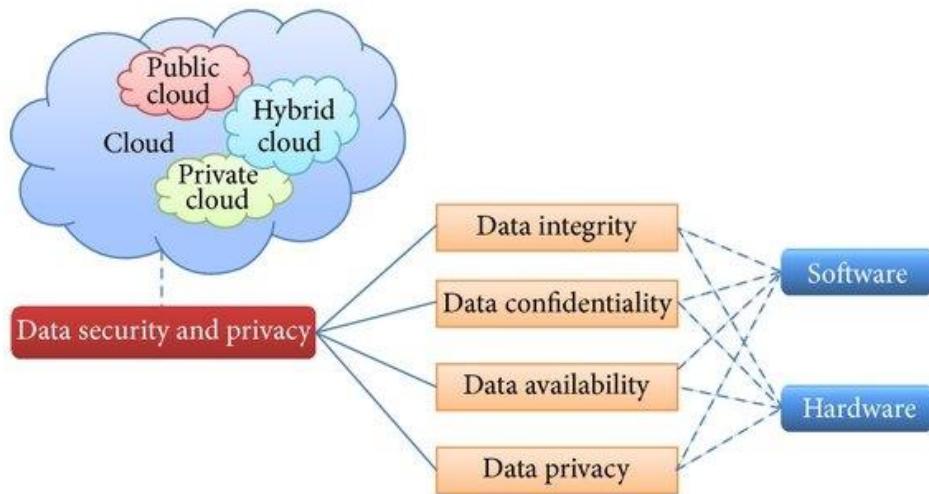
8. Encryption systems typically consist of algorithms that are computationally difficult or infeasible to break, along with the processes and procedures to manage encryption and decryption, hashing, digital signatures, certificate generation and renewal and key exchange.

9. Business Continuity and Disaster Recovery are the measures designed and implemented to ensure operational resiliency in the event of any service interruptions. Business continuity and disaster recovery provides flexible and reliable failover for required services in the event of any service interruptions, including those caused by natural or man-made disasters or disruptions. Cloud-centric business continuity and disaster recovery makes use of the cloud's flexibility to minimize cost and maximize benefits.

10. Network Security consists of security services that allocate access, distribute, monitor and protect the underlying resource services. Architecturally, network security provides services that address security controls at the network in aggregate or specifically addressed at the individual network of each underlying resource.

In a cloud/virtual environment, network security is likely to be provided by virtual devices alongside traditional physical devices.

8. Diagrammatic representation of Data storage security in private cloud

**Conclusion:**

1. Study about Data Storage related to private Cloud is successfully done.
2. Learned about the techniques used to store data in the private cloud.
3. Security Algorithm and diagrammatic representation of security implemented in private cloud.

FAQ:

- 1). What extra benefits does a private cloud get ?
- 2). Are all Operating Systems supported in private cloud.
- 3). How does client get access to updates.

Case Study-2

1) What is IoT?

The internet of things, or IoT, is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

A thing in the internet of things can be a person with a heart monitor implant, a farm animal with a biochip transponder, an automobile that has built-in sensors to alert the driver when tire pressure is low or any other natural or man-made object that can be assigned an Internet Protocol (IP) address and is able to transfer data over a network.

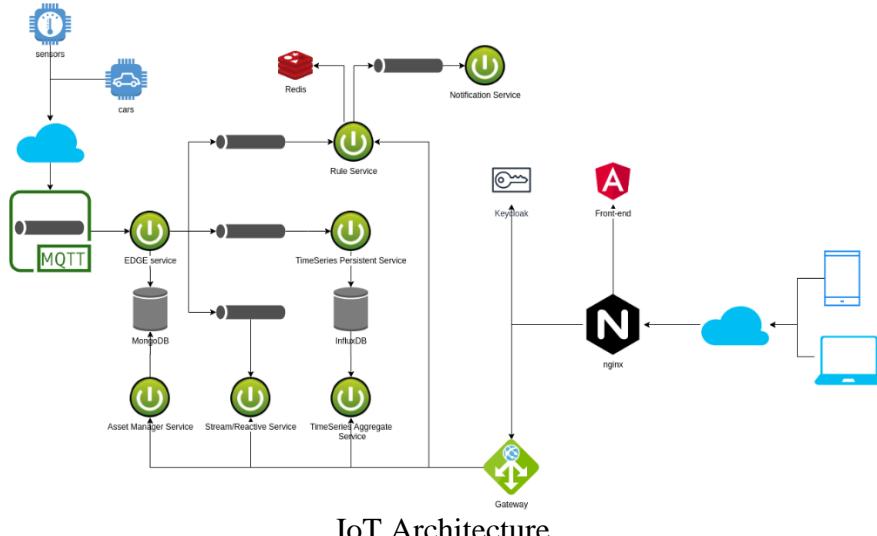
2) What is Ubiquitous system?

Ubiquitous Computing is also known as ambient computing or pervasive computing, everywhere and ambient intelligence. It aims to fully tap into the potential for connected devices and their benefits to become widespread such that they are available everywhere, i.e ubiquitous. The idea is to saturate the work, living, and transportation spaces with devices that can seamlessly communicate with each other thus making our lives more enjoyable and convenient.

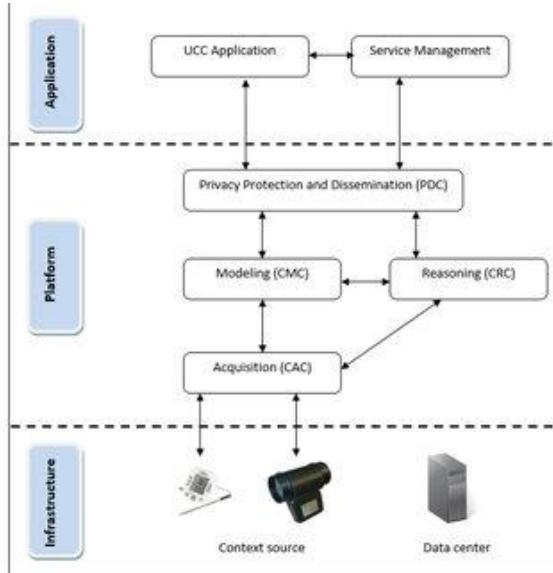
Ubiquitous computing thus relies on contextual data aggregation and application, seamless, intuitive access points, and flexible payment systems. A concept in software engineering, hardware engineering computer science, it lies at the intersection of distributed computing, mobile computing, location computing, mobile networking, sensor networks, human-computer interaction, context-aware smart home technologies, and Artificial Intelligence (AI).

It is, effectively speaking, a paradigm that involves connecting electronic devices, including embedding microprocessors to communicate information. Devices that use ubiquitous computing are constantly available and are fully connected. Thus the main focus of ubiquitous computing is producing smart products that are connected, making communication and the exchange of data easier and less obtrusive. For instance, smartphones, smart speakers, smart watches and smart home gadgets are typically the devices that form a core part of the ubiquitous computing network. Basically, a user communicates with the computing device - this could be a laptop computer, tablet, a mobile device, or a terminal in everyday objects such as thermostat or even a pair of glasses.

3) Draw necessary diagrams for both IoT and Ubiquitous systems.



IoT Architecture.



Ubiquitous Computing

4) What applications can be built using the ubiquitous and IoT based cloud?

IoT Cloud Computing provides many connectivity options, implying large network access. People use a wide range of devices to gain access to cloud computing resources: mobile devices, tablets, laptops. This is convenient for users but creates the problem of the need for network access points.

Ubiquitous computing is all about having computational capability in many different (perhaps all) objects in our environment. Things like stoves Fridges, TV, Cars etc. all having some computational power. The Internet of Things (IoT) is about having these objects in our environment all connected to an internet.

5) Success stories as example for IoT and Ubiquitous based on cloud application- put details.

a) Amazon Dash Button

- Amazon Dash Button is basically a device that gets connected over internet Wi-Fi and makes sure that the user does not lack important household items like soft drinks, grocery material, medical and personal care, kids, and any pet items ever again.

- It allows the user to order products quickly and there is no need to recall the message again and it also helps to reduce the time frame for searching the required product by the user.
 - Amazon Dash Button also allows the user to reorder from popular brands – like Bounty, Tide, Cottonelle, Glad, Clorox etc.
 - It does not accept fresh order if the prior order is not complete unless the user allows multiple orders.
 - It is a good and reliable IoT product that is developed for making the user's lifestyle simple and easy.
- b) August Doorbell Cam
- August Doorbell Cam is an effective IoT innovation. August Doorbell Cam allows you to answer your door from anywhere or remote location. It constantly checks your doors and also captures motion changes in your doorstep.
 - Doorbell Cam pairs with all August Smart Locks to easily let guests into your home.
 - The integrated floodlight delivers clear, full-color HD video even full-color.
 - It constantly monitors your doorstep and will click the moments leading up to a motion alert.
 - Free 24hour video recording.
 - It comes with a speedy and hassle-free installation process.

Conclusion:

1. Here, study of IOT and Ubiquitous Computing Architecture is successfully completed.
2. Applications based on it.
3. Recent inventions related to IOT.

FAQ's:

- 1). How is security maintained in IOT devices.
- 2) Is maintenance of IOT sensors difficult.

Case Study-3

1) What is private cloud?

Private cloud (also known as an internal cloud or corporate cloud) is a cloud computing environment in which all hardware and software resources are dedicated exclusively to, and accessible only by, a single customer. Private cloud combines many of the benefits of cloud computing—including elasticity, scalability, and ease of service delivery—with the access control, security, and resource customization of on-premises infrastructure.

Many companies choose private cloud over public cloud (cloud computing services delivered over infrastructure shared by multiple customers) because private cloud is an easier way (or the only way) to meet their regulatory compliance requirements. Others choose private cloud because their workloads deal with confidential documents, intellectual property, personally identifiable information (PII), medical records, financial data, or other sensitive data.

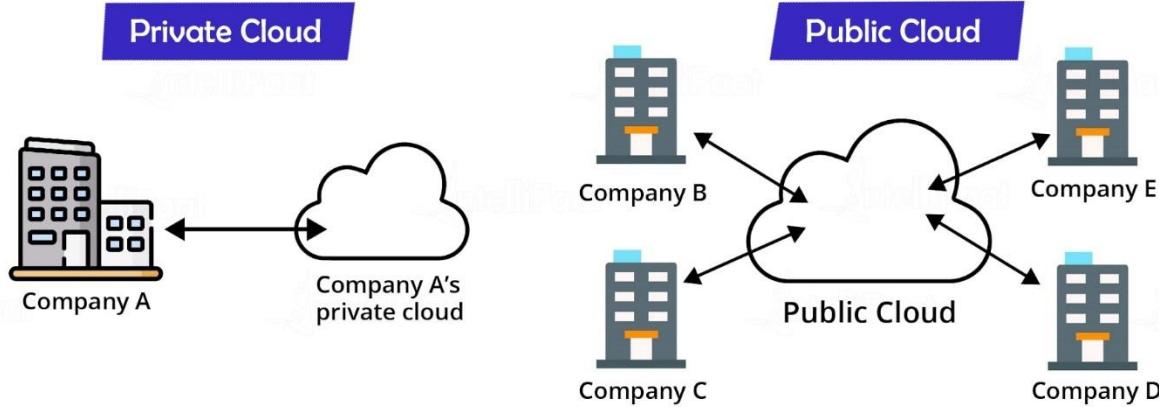
By building private cloud architecture according to cloud native principles, an organization gives itself the flexibility to easily move workloads to public cloud or run them within a hybrid cloud (mixed public and private cloud) environment whenever they're ready.

2) Comparative between private cloud , public could.

Public Cloud	Private Cloud
Cloud Computing infrastructure shared to public by service provider over the internet. It supports multiple customers i.e., enterprises.	Cloud Computing infrastructure shared to private organisation by service provider over the internet. It supports one enterprise.
Multi-Tenancy i.e., Data of many enterprise are stored in shared environment but are isolated. Data is shared as per rule, permission and security	Single Tenancy i.e., Data of single enterprise is stored
Cloud service provider provides all the possible services and hardware as the user-base is world. Different people and organization may need different services and hardware. Services provided must be versatile.	Specific hardware and hardware as per need of enterprise are available in private cloud.
It is hosted at Service Provider site.	It is hosted at Service Provider site or enterprise.
It is connected to the public internet	It only supports connectivity over the private network.
Scalability is very high, and reliability is moderate.	Scalability is limited, and reliability is very high.
Cloud service provider manages cloud and customers use them.	Managed and used by single enterprise.

3) Draw necessary diagrams for both private cloud and public cloud.

Cloud Architecture



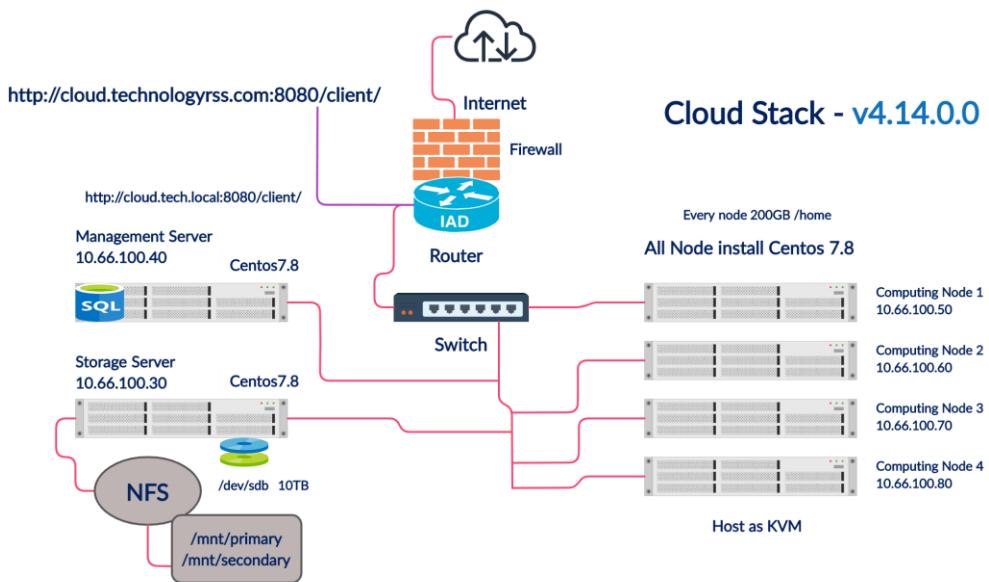
4) Explain any Tools for building private cloud with necessary diagram and explanation.

1. CloudStack:

Apache CloudStack is designed to deploy and manage large networks of virtual machines. This Apache Project offers a turnkey Infrastructure as a Service (IaaS) cloud computing platform. It's used both by public cloud computing vendors and by organizations running their own private clouds.

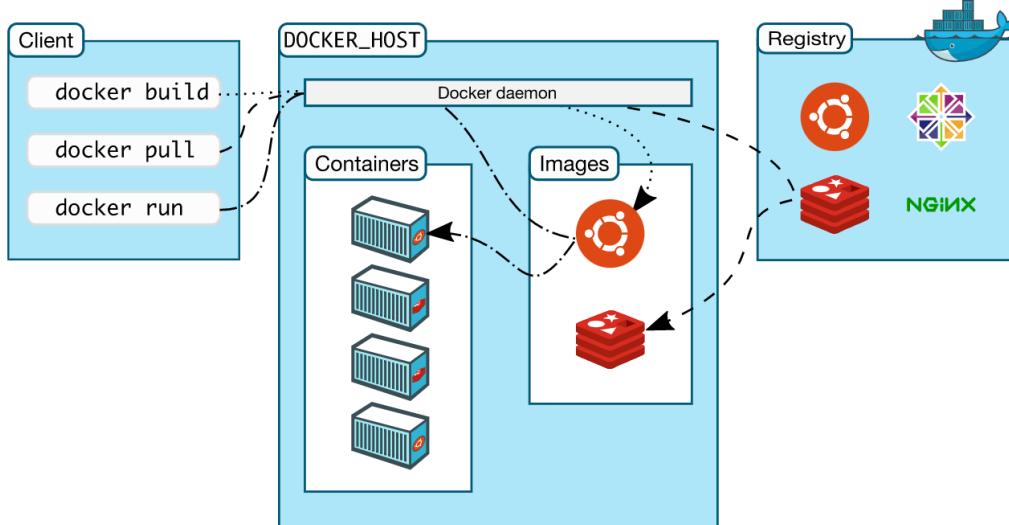
2. FOSS-Cloud

The FOSS-Cloud is a Software that enables you to create your own Private or Public Cloud. It is an integrated infrastructure to provide cloud-Services, Windows or Linux based SaaS. FOSS-Cloud covers all of the aspects of an Open-Source IT environment. This multi-faced cloud computing solution includes virtualization, cloud desktop, IaaS, PaaS and SaaS capabilities.



3. Docker

Docker provides a highly reliable, low-cost way to quickly build, ship, and run distributed applications at scale. It gives developers the freedom to define environments and create apps faster and easier and flexibility for IT ops to quickly respond to change.



Conclusion:

1. Learning about private cloud and its architecture is successfully done.
2. Applications based on it.
3. Tools required to build private cloud.

FAQ's:

Is data lost when the container exits?

Is it possible to run more than one process in a Docker Container.