

Department of Computer Engineering

Lab Manual

310258:Laboratory Practice II (Cloud Computing)

Preface

Cloud Computing is the delivery of computing services such as servers, data storage, databases, networking, software, analytics, and intelligence over the internet (“cloud”) to offer flexible resources, faster innovation, and economies of scale. In simpler terms, instead of owning data centers, organizations can rent access to someone else’s infrastructure like storage, computing servers, and databases from a Cloud Computing service provider and only pay for resources that they use.

You only need to pay for the cloud services that you use, which helps lower your operating costs, run infrastructure more efficiently, and scale your applications as per business needs.

Organizations of different types, sizes, and industries are using the cloud for a wide variety of use cases, such as building customer-facing web applications, data backup, sending email/SMS notifications, virtual desktops, software development and testing, big data analytics, and disaster recovery. For example, Telecom companies are using cloud services to connect with their customers by sending different types of communications. Financial services companies are using the cloud to power real-time fraud detection and prevention.

Course Outcomes, Program Outcomes & Program Specific Outcomes

1. Course Outcome

Course Outcome	Statement
	<i>At the end of the course, a student will be able to (write/install/solve/apply)</i>
310258(C).1	Design a system using different informed search / uninformed search or heuristic approaches
310258(C).2	Apply basic principles of AI in solutions that require problem solving, inference, perception, knowledge representation, and learning
310258(C).3	Design and develop an interactive AI application
310258(C).4	Use tools and techniques in the area of Cloud Computing
310258(C).5	Use cloud computing services for problem solving
310258(C).6	Design and develop applications on cloud

B. CO-PO mapping

Course Outcome	Program outcomes											
	1	2	3	4	5	6	7	8	9	10	11	12
310258(C).1	2	-	2		3			2	2	2	1	2
310258(C).2	1	-	2	2	3	2		2	2	2	1	2
310258(C).3	1	-	2	2	3	2		2	2	2	2	2
310258(C).4	1	-	2	-	3	-	-	2	2	2	2	2
310258(C).5	1	-	2	-	3	-	-	2	2	2	2	2
310258(C).6	1	-	2	-	3	-	-	2	2	2	2	2

C. CO-PSO mapping

Course Outcome	Program Specific Outcomes	
	1	2
310258.1	2	1
310258.2	1	2
310258.3	2	1
310258(C).4	2	2
310258(C).5	2	2
310258(C).6	2	2

A.

CO Target	Average percentage marks of the assessment tool	
Attainment Level	1	50% students scoring more than target
	2	60% students scoring more than target
	3	70% students scoring more than target

INDEX

Sr. No	Title of Assignment
1	Installation and configuration of own CloudCase study on Amazon EC2 and learn about Amazon EC2 web services
2	Installation and Configuration of virtualization using KVM
3	Creating an Application in Salesforce.com using Apex programming Language.
4	Design and develop custom Application (Mini Project) using Sales force Cloud
5	Mini-Project Setup your own cloud for Software as a Service (SaaS) over the existing LAN in your laboratory. In this assignment you have to write your own code for cloud controller using open- source technologies to implement with HDFS. Implement the basic operations may be like to divide the file in segments/blocks and upload/download file on/from cloud in encrypted form.

Software Required:

Operating System :-Operating System recommended :- 64-bit Open source Linux or its derivative

Programming Languages: C++/JAVA/PYTHON

Write-ups must include:

- **Group:**
- **Assignment No.**
- **Title**
- **Problem Statement**
- **Prerequisites**
- **Course Objectives**
- **Course Outcomes**
- **Theory(in brief)**
- **Algorithm**

- **Mathematical Model (Important)**
- **Conclusion:**
- **FAQs:**
- **Output: Printout of program with output**

Assignment No: 1

TITLE: Problem Statement: Case study on Amazon EC2 to learn about Amazon EC2, Amazon Elastic Compute Cloud is a central part of Amazon.com's cloud computing platform, Amazon Web Services. How EC2 allows users to rent virtual computers on which to run their own computer applications.

PREREQUISITES: Networking Basics

COURSE OBJECTIVES: To implement SaaS, PaaS, IaaS on AWS cloud system

COURSE OUTCOMES: Ability to implement SaaS, PaaS, IaaS on AWS cloud system

CO4: Use tools and techniques in the area of Cloud Computing

CO5: Use the knowledge of Cloud Computing for problem solving

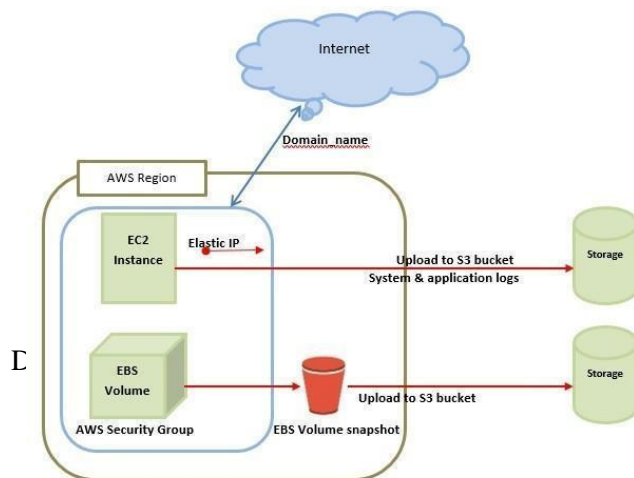
THEORY

Amazon Web Services (AWS) is Amazon's cloud web hosting platform that offers flexible, reliable, scalable, easy-to-use, and cost-effective solutions

Cloud computing is an internet-based computing service in which large groups of remote servers are networked to allow centralized data storage, and online access to computer services or resources. Using cloud computing, organizations can use shared computing and storage resources rather than building, operating, and improving infrastructure on their own.

Cloud computing is a model that enables the following features.

- Users can provision and release resources on-demand.
- Resources can be scaled up or down automatically, depending on the load.
- Resources are accessible over a network with proper security.
- Cloud service providers can enable a pay-as-you-go model, where customers are charged based on the type of resources and per usage.



Types of Clouds

There are three types of clouds – Public, Private, and Hybrid cloud.

Amazon Web Services - Basic Architecture

Note – In the above diagram s3 stands for simple storage service. it allows the users to store and retrieve various types of data using api calls. it doesn't contain any computing element.

AWS - ELASTIC COMPUTE CLOUD



Amazon EC2 (Elastic Compute Cloud) is a web service interface that provides resizable compute capacity in the AWS cloud. It is designed for developers to have complete control over web-scaling and computing resources.

EC2 instances can be resized and the number of instances scaled up or down as per our requirement. These instances can be launched in one or more geographical locations or regions, and **Availability Zones (AZs)**. Each region comprises of several AZs at distinct locations, connected by low latency networks in the same region.

FEATURES OF EC2

RELIABLE –

- Designed for Amazon Web Services –
- Secure
- Flexible Tools
- Inexpensiv

How to Use AWS EC2

Step 1 – Sign-in to AWS account and open IAM console by using the following link <https://console.aws.amazon.com/iam/>.

Step 2 – In the navigation Panel, create/view groups and follow the instructions.

Step 3 – Create IAM user. Choose users in the navigation pane. Then create new users and add users to the groups.

Step 4 – Create a Virtual Private Cloud using the following instructions.

- Open the Amazon VPC console by using the following link – <https://console.aws.amazon.com/vpc/>
- Select VPC from the navigation panel. Then select the same region in which we have created key-pair.
- Select start VPC wizard on VPC dashboard.
- Select VPC configuration page and make sure that VPC with single subnet is selected. The choose Select.
- VPC with a single public subnet page will open. Enter the VPC name in the name field and leave other configurations as default.
- Select create VPC, then select Ok.

Step 5 – Create WebServerSG security groups and add rules using the following instructions.

- On the VPC console, select Security groups in the navigation panel.
- Select create security group and fill the required details like group name, name tag, etc.
- Select your VPC ID from the menu. Then select yes, create button.
- Now a group is created. Select the edit option in the inbound rules tab to create rules.

Step 6 – Launch EC2 instance into VPC using the following instructions.

- Open EC2 console by using the following link – <https://console.aws.amazon.com/ec2/>
- Select launch instance option in the dashboard.
- A new page will open. Choose Instance Type and provide the configuration. Then select Next: Configure Instance Details.
- A new page will open. Select VPC from the network list. Select subnet from the subnet list and leave the other settings as default.
- Click Next until the Tag Instances page appears.

Step 7 – On the Tag Instances page, provide a tag with a name to the instances. Select Next: Configure Security Group.

Step 8 – On the Configure Security Group page, choose the Select an existing security

group option. Select the WebServerSG group that we created previously, and then choose Review and Launch.

Step 9 – Check Instance details on Review Instance Launch page then click the Launch button.

Step 10 – A pop up dialog box will open. Select an existing key pair or create a new key pair. Then select the acknowledgement check box and click the Launch Instances button.

LOAD BALANCING

Load balancing simply means to hardware or software load over web servers, that improves the efficiency of the server as well as the application. Following is the diagrammatic representation of AWS architecture with load balancing.

Elastic load balancing can dynamically grow and shrink the load -balancing capacity to adjust to traffic demands and also support sticky sessions to address more advanced routing needs

HOSTING RDMS ON EC2 INSTANCES

Amazon RDS allows users to install RDBMS (Relational Database Management System) of your choice like MySQL, Oracle, SQL Server, DB2, etc. on an EC2 instance and can manage as required.

Amazon EC2 uses Amazon EBS (Elastic Block Storage) similar to network-attached storage. All data and logs running on EC2 instances should be placed on Amazon EBS volumes, which will be available even if the database host fails.

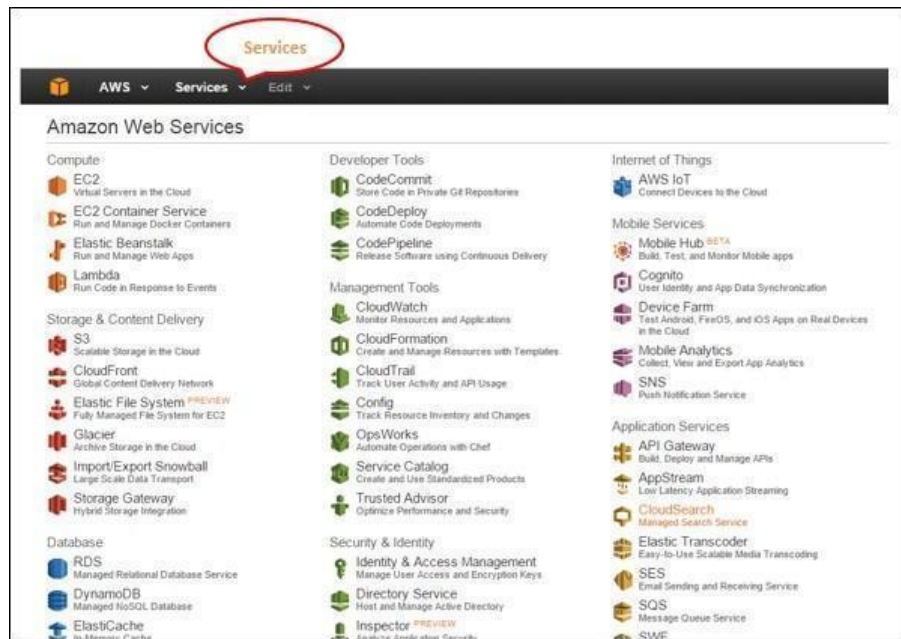
Create an AWS account

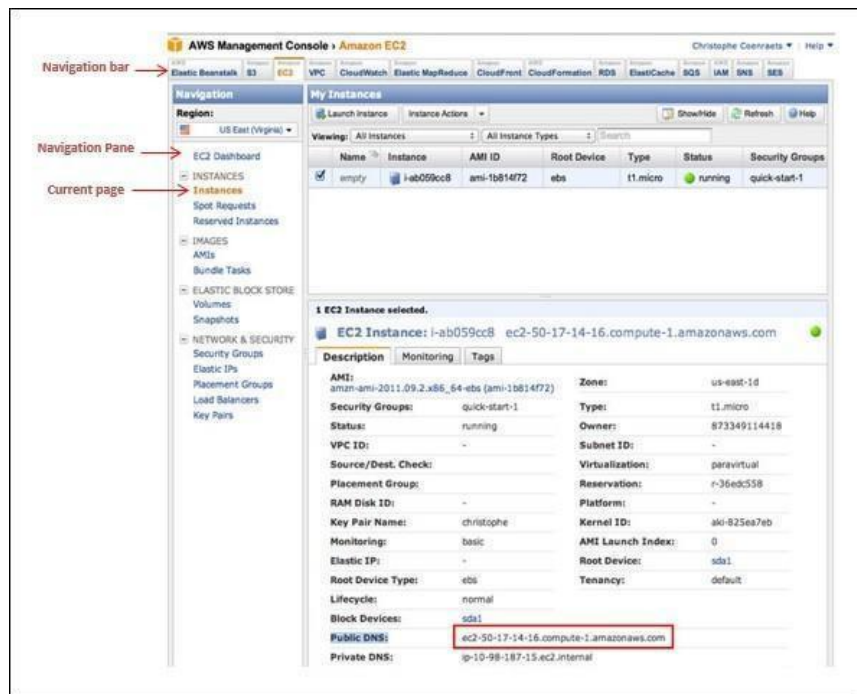
Go to AWS homepage <http://www.aws.amazon.com> . Click the Sign In to Console or Create an AWS account button on the top right corner. This will bring you to the sign in/sign up page.

STEP 1

create your account there with your email and password following screen shot of aws management

console for amazon ec2 service





Step 2 – select the choice from the list of categories and we get their sub-categories such as computer and database category is selected in the following screenshots.



Step 3 – select the service of your choice and the console of that service will open.

COMPUTE

Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable

compute capacity in the cloud. It is designed to make web scale computing easier for developers. The

Amazon EC2 simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon's proven computing environment.

Amazon EC2 reduces the time required to obtain and boot new server instances (called Amazon EC2 instances) to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change. Amazon EC2 changes the economics of computing by allowing you to pay only for capacity that you actually use.

Amazon EC2 provides developers and system administrators the tools to build failure resilient applications and isolate themselves from common failure scenarios.

Launch an Instance

Select Launch Instance to create and configure your virtual machine.

Configure your Instance

FAQ's

1. Compare AWS and OpenStack
2. What is AWS?
3. What is the importance of buffer in Amazon Web Services?
4. Name the several layers of Cloud Computing.
5. Distinguish between scalability and flexibility
6. What is DynamoDB?
7. How is stopping and terminating an instance different from each other?
8. What are the best practices for Security in Amazon EC2?

Assignment No: 2

TITLE: Installation and Configuration of virtualization using KVM

PROBLEM STATEMENT: Installation and Configuration of virtualization using KVM

PREREQUISITES: Networking Basics

COURSE OBJECTIVES: To Study and implement Virtualization in Cloud Computing

COURSE OUTCOMES: Ability to implement Virtualization in Cloud Computing

CO4: Use tools and techniques in the area of Cloud Computing

CO5: Use the knowledge of Cloud Computing for problem solving

WHAT IS VIRTUALIZATION?

Virtualization is the process of creating a software-based, or virtual, representation of something, such as virtual applications, servers, storage and networks. It is the single most effective way to reduce IT expenses while boosting efficiency and agility for all size businesses

BENEFITS OF VIRTUALIZATION

Virtualization can increase IT agility, flexibility and scalability while creating significant cost savings. Greater workload mobility, increased performance and availability of resources, automated operations – they're all benefits of virtualization that make IT simpler to manage and less costly to own and operate. Additional benefits include:

- Reduced capital and operating costs.
- Minimized or eliminated downtime.
- Increased IT productivity, efficiency, agility and responsiveness.
- Faster provisioning of applications and resources.
- Greater business continuity and disaster recovery.
- Simplified data center management.
- Availability of a true Software-Defined Data Center..

Virtualization relies on software to simulate hardware functionality and create a virtual computer system. This enables IT organizations to run more than one virtual system – and multiple operating systems and applications – on a single server

VMS HAVE THE FOLLOWING CHARACTERISTICS, WHICH OFFER SEVERAL BENEFITS.**Partitioning**

- Run multiple operating systems on one physical machine.
- Divide system resources between virtual machines.

Isolation

- Provide fault and security isolation at the hardware level.
- Preserve performance with advanced resource controls.

Encapsulation

- Save the entire state of a virtual machine to files.
- Move and copy virtual machines as easily as moving and copying files.

Hardware Independence

- Provision or migrate any virtual machine to any physical server.

KVM (Kernel-based Virtual Machine) is free and open source virtualisation software. You can create multiple VM (virtual machines) , each VM has its own private virtualised hardware like disk, CPU, RAM etc.

Prerequisites

For installing the KVM you must have following prerequisites.

1. Enable Virtualisation in your system BIOS.
2. Check your system CPU if it supports virtualisation. Run the given below command.

```
egrep -c '(vmx|svm)' /proc/cpuinfo
```

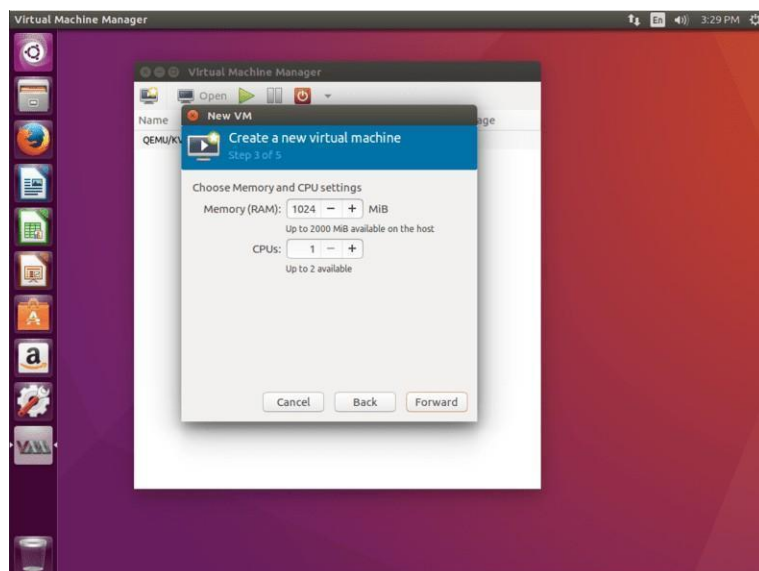
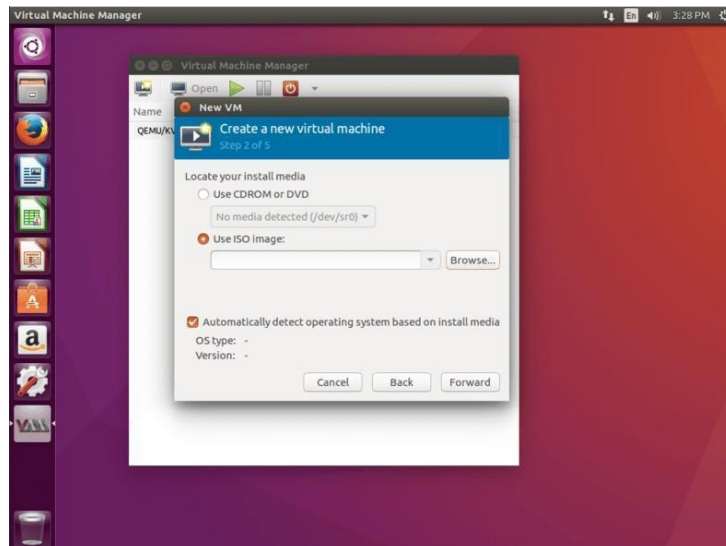
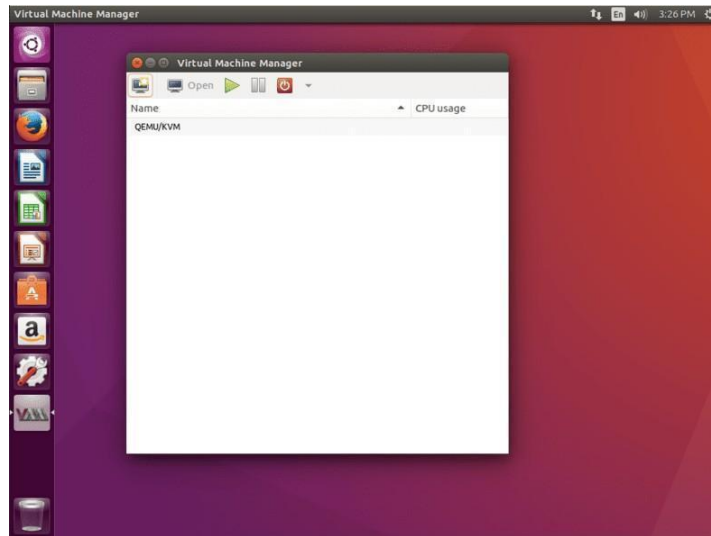
When you get the output from above command either 1 or more it means CPU supports the virtualisation else 0 or less means it does not support.

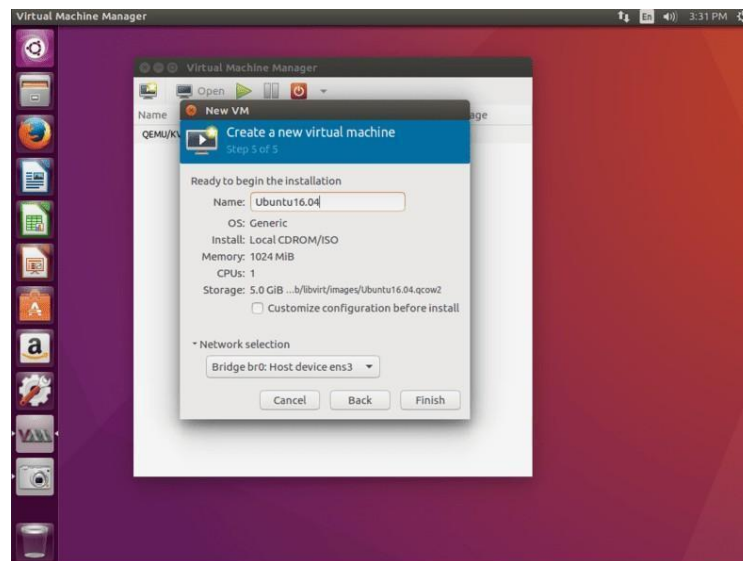
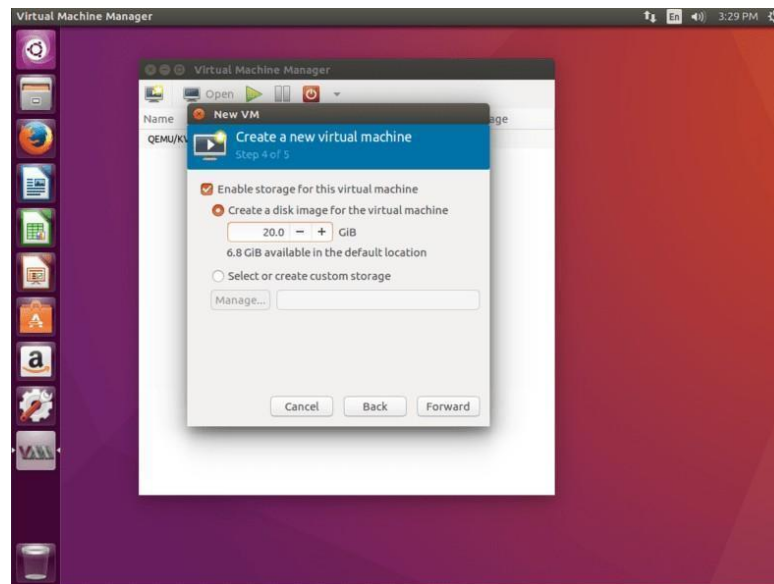
1. Install KVM and other dependencies packages

In Ubuntu 16.04 LTS, you can use the command apt or apt-get both. There will be no differences in packages installed via apt or apt-get command so you are good here.

```
sudo apt update
```

```
sudo apt install qemu-kvm libvirt-bin bridge-utils
```





Your new virtual machine will start up as though it were booting into the install ISO that you selected. Just run through the install process and reboot when prompted at the end. The virtual machine will reboot, and you will have a working install on the new VM.

FAQ's

1. What is owncloud? What are its features?
2. Define Saas
3. Describe Paas
4. What is Iaas
5. What allows users to create their own cloud applications using supplier-specific tools and languages?
6. What allows users to rent processing, storage, networking and other fundamental computing resources for all purposes?
7. What allows users to run existing online applications?

Assignment No: 3

TITLE: Creating an Application in Salesforce.com using Apex programming Language.

PROBLEM STATEMENT: Creating an Application in Salesforce.com using Apex programming Language.

PREREQUISITES: Networking Basics

COURSE OBJECTIVES:

COURSE OUTCOMES:

CO4: Use tools and techniques in the area of Cloud Computing

CO5: Use the knowledge of Cloud Computing for problem solving

CO6: Apply the concepts Cloud Computing to design and develop applications

THEORY:

Apex is a strongly typed, object-oriented programming language that allows developers to execute flow and transaction control statements on the lightning platform server in conjunction with calls to the lightning platform api. using syntax that looks like java and acts like database stored procedures, apex enables developers to add business logic to most system events, including button clicks, related record updates, and visualforce pages. apex code can be initiated by web service requests and from triggers on objects.

apex triggers:

apex code can be invoked by using triggers. apex triggers can be configured to perform custom actions before or after changes to salesforce records, such as insertions, updates, or deletions

to define a trigger, use the following syntax:

```
trigger triggername on objectname (trigger_events) {  
  
    code_block  
  
}
```

where *trigger_events* can be a comma-separated list of one or more of the following events:

- before insert
- before update

-
- before delete

- after insert
- after update
- after delete
- after undelete

This below given steps shows how to create a simple apex class and trigger, and how to deploy these components to a college..

apex is very similar to java, you may recognize much of the functionality.

here we create a custom object called book that is created in the first step. this custom object is updated through a trigger.

- [create a custom object](#)

in this step, you create a custom object called book with one custom field called price.

- [adding an apex class](#)

in this step, you add an apex class that contains a method for updating the book price. this method is called by the trigger that you will be adding in the next step.

- [add an apex trigger](#)

in this step, you create a trigger for the book_c custom object that calls the applydiscount method of the myhelloworld class that you created in the previous step.

- [add a test class](#)

in this step, you add a test class with one test method. you also run the test and verify code coverage. the test method exercises and validates the code in the trigger and class. also, it enables you to reach 100% code coverage for the trigger and class.

- [deploying components to production](#)

in this step, you deploy the apex code and the custom object you created previously to your production organization using change

EXAMPLE: STUDENT & SUBJECT/COURSE

CREATE TWO OBJECTS EX. TEACHER AND SUBJECT OR
STUDENT & COURSE

DETAILS OF STUDENT
RECORDS IN STUDENT

INSERTING A NEW RECORD
SUBJECT/COURSE AUTOMATICALLY ASSIGNED
TRIGGER ON TEACHER/STUDENT OBJECT TO AUTOMATICALLY
ASSIGN Subject/course

ref: https://developer.salesforce.com/docs/atlas.en.us.apexcode.meta/apexcode/apex_qs_deploy.htm

FAQs

1. What is CRM?
2. What are the characteristics of Apex?
3. What are apex Triggers
4. What are apex classes and Objects

Assignment 4

TITLE: Design and develop custom Application (Mini Project) using Sales force Cloud.

PREREQUISITES: Networking Basics

COURSE OBJECTIVES: To Design and develop custom Application (Mini Project) using Sales force Cloud.

CO4: Use tools and techniques in the area of Cloud Computing

CO5: Use the knowledge of Cloud Computing for problem solving

CO6: Apply the concepts Cloud Computing to design and develop applications

THEORY:

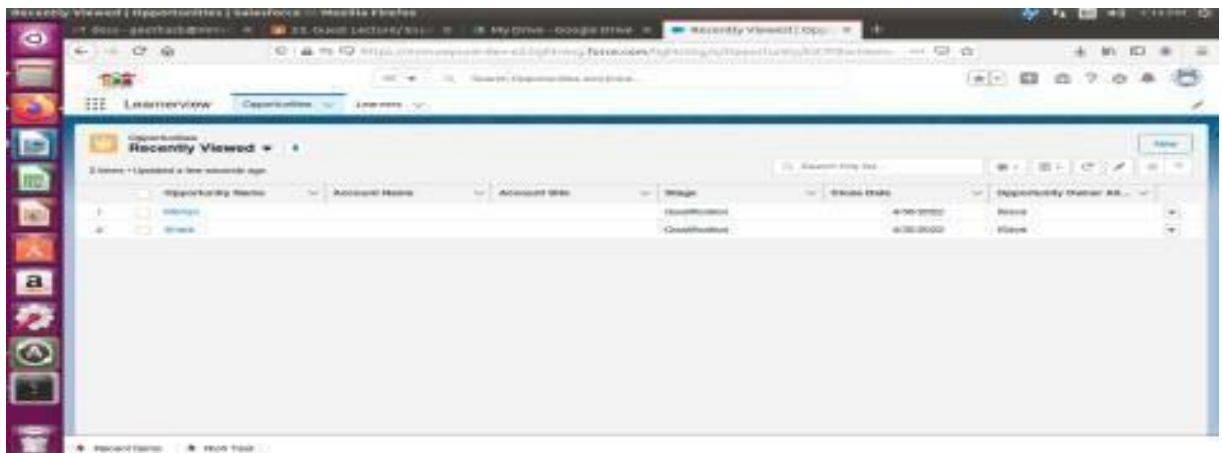
Salesforce Org

The cloud computing space offered to you or your organization by Force.com is called Sales force org. It is also called Salesforce environment. Developers can create custom Salesforce Apps, objects, workflows, data sharing rules, Visualforce pages and Apex coding on top of Sales force Org.

Salesforce Apps

The primary function of a Salesforce app is to manage customer data. Salesforce apps provide a simple UI to access customer records stored in objects (tables). Apps also help in establishing relationship between objects by linking fields.

Apps contain a set of related tabs and objects which are visible to the end user. The below screenshot shows, how the *StudentForce* app looks like.

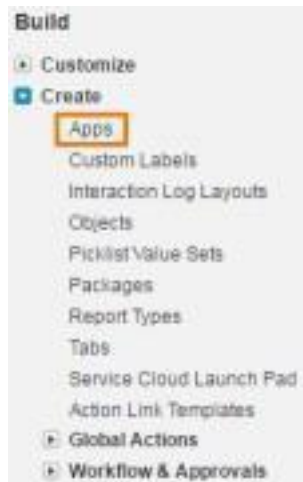


The highlighted portion in the top right corner of the screenshot displays the app name: *Student Force*. The text highlighted next to the profile pic is my username: *Vardhan NS*.

Before you create an object and enter records, you need to set up the skeleton of the app. You can follow the below instructions to set up the app.

Steps To Setup The App

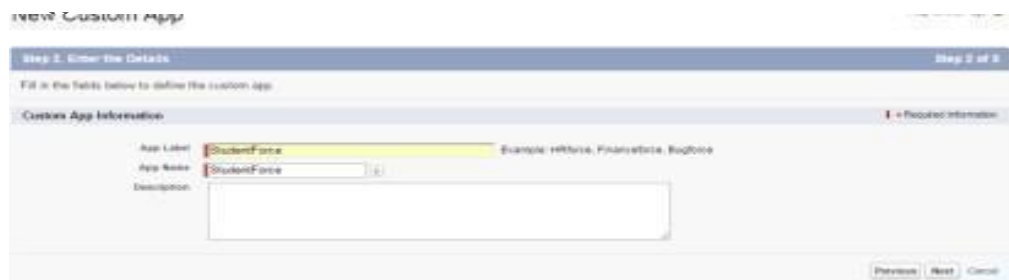
- Click on *Setup* button next to app name in top right corner.
 - In the bar which is on the left side, go to *Build* → select *Create* → select *Apps* from the drop down menu.



- Click on *New* as shown in the below screenshot.



- Choose *Custom App*.
- Enter the *App Label*. *StudentForce* is the label of my app. Click on



- Choose a profile picture for your app. Click *Next*.
- Choose the tabs you deem necessary. Click *Next*.
- Select the different profiles you want the *app* to be assigned

to Click *Save*.

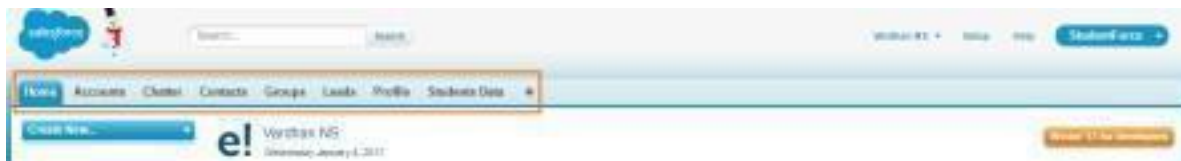
In steps 7 and 8, you were asked to choose the relevant tabs and profiles. Tabs and profiles

are an integral part of Salesforce Apps because they help you to manage objects and records in Sales force.

In this salesforce tutorial, I will give you a detailed explanation of Tabs, Profiles and then show you how to create objects and add records to it.

Salesforce Tabs

Tabs are used to access objects (tables) in the Salesforce App. They appear on top of the screen and are similar to a toolbar. It contains shortcut links to multiple objects. On clicking the object name in a tab, records in that object will be displayed. Tabs also contain links to external web content, custom pages and other URLs. The highlighted portion in the below screenshot is that of Salesforce tabs.



All applications will have a *Home* tab by default. Standard tabs can be chosen by clicking on '+' in the Tab menu. Accounts, Contacts, Groups, Leads, Profile are the standard tabs offered by Salesforce. For example, *Accounts* tab will show you the list of accounts in the SFDC org and *Contacts* tab will show you the list of contacts in the SFDC org.

Steps To Add Tabs

- Click on '+' in the tab menu.
- Click on *Customize tabs*, which is present on the right side.
- Choose the tabs of your choice and click on *Save*.

Besides standard tabs, you can also create custom tabs. *Students* tab that you see in the above screenshot is a custom tab that I have created. This is a shortcut to reach the custom object: *Students*.

Steps To Create Custom Tabs

1. Navigate to Setup → Build → Create → Tabs.
2. Click on *New*.
3. Select the object name for which you are creating a tab. In my case, it is *Students Data*. This is a custom object which I have created (the instructions to create this object is covered later in this blog).
4. Choose a tab style of your preference and enter a description.
5. Click on Next → Save. The new *Students Data* tab will appear as shown below.

Salesforce Profiles

Every user who needs to access the data or SFDC org will be linked to a profile. A profile is a collection of settings and permissions which controls what a user can view, access and

modify in Salesforce.

A profile controls user permissions, object permissions, field permissions, app settings, tab settings, apex class access, Visualforce page access, page layouts, record types, login hour and login IP addresses.

You can define profiles based on the background of the user. For example, different levels of access can be set for different users like system administrator, developer and sales representative.

Similar to tabs, we can use any standard profile or create a custom profile. By default, the available standard profiles are: read only, standard user, marketing user, contract manager, solution manager and system administrator. If you want to create custom profiles, you have to first clone standard profiles and then edit that profile. Do note that one profile can be assigned to many users, but one user cannot be assigned many profiles.

Steps To Create A Profile

- Click on Setup → Administer → Manage users → Profiles
- You can then clone any of the existing profiles by clicking on *Edit*.

Once the tabs and profiles are set up for your App, you can load data into it. The next section of this Salesforce tutorial will thus cover how data is added to objects in the form of records and fields.

Objects, Fields And Records In Salesforce

Objects, Fields and Records are the building blocks of Salesforce. So, it is important to know what they are and what role they play in building Apps.

Objects are the database tables in Salesforce where data is stored. There are two types of objects in Salesforce:

1. **Standard objects:** The objects provided by Salesforce are called standard objects. For example, Accounts, Contacts, Leads, Opportunities, Campaigns, Products, Reports, Dash board etc.
2. **Custom objects:** The objects created by users are called custom

objects. Objects are a collection of records and records are a collection of fields.

Every row in an object consists of many fields. Thus a record in an object is a combination of related fields. Look at the below excel for illustration.

I will create an object called *Students Data* which will contain personal details of students.

Steps to create a custom object:

1. Navigate to Setup → object manager → Create → custom Object

2. Click on *New Custom Object*.
3. Fill in the *Object Name* and *Description*. As you can see from the below image, the object name is *Students Data*.
4. Click on *Save*.

After creating the object, you need to define various fields in that object. e.g. the fields in a student's record will be student name, student phone number, student email ID, the department a student belongs to and his native city.

You can add records to objects only after defining the fields.

Steps To Add Custom Fields

- Navigate to Setup → Build → Create → Objects
- Select the object to which you want to add fields. In my case, it is *Students Data*.
- Scroll down to Custom Fields & Relationships for that object and click on *New* as shown in the below screenshot.
- You need to choose the data type of that particular field and then click *Next*. I have chosen *text* format because I will be storing letters in this field.
The different data types of fields have been explained in detail in the next section of this blog.
- You will then be prompted to enter the name of the field, maximum length of that field and description.
- You can also make it an optional/ mandatory field and allow/ disallow duplicate values for different records by checking on the check boxes. See the below screenshot to get a better understanding.
- Click on *Next*.
- Select the various profiles who can edit that text field at a later point of time. Click *Next*.
- Select the page layouts that should include this field.
- Click *Save*.

As you can see from the below screenshot, there are two types of fields. Standard fields created for every object by default and Custom fields created by myself. The four fields which I have created for *Students Data* are City, Department, Email ID and Phone No. You will notice that all custom fields are suffixed with '_C' which indicates that you have the power to edit and delete those fields. Whereas some standard fields can be edited, but not deleted.

You can now add student records (complete row) to your object.

Steps To Add A Record

1. Go to the object table from the tab menu. *Students Data* is the object to which I will add records.
2. As you can see from the below image, there are no existing records. Click on *New* to add new student records.
3. Add student details into different fields as shown in the below screenshot. Click on *Save*.
4. You can create any number of student records. I have created 4 student records as shown in the below screenshot.

5. In case you want to edit the student details, you can click on *Edit* as shown in the below screenshot.

Data Types Of Fields

Data type controls which type of data can be stored in a field. Fields within a record can have different data types. For example:

1. If it is a phone number field, you can choose *Phone*.
2. If it is a name or a text field, you can choose *Text*.
3. If it is a date/ time field, you can choose *Date/Time*.
4. By choosing *Picklist* as data type for a field, you can write predefined values in that field and create a drop-down.

You can choose any one of the data types for custom fields. Below is a screenshot listing the different data types.

Data types like *Lookup Relationship*, *Master-Detail Relationship* and *External Lookup Relationship* are used to create links/ relationships between one or more objects. Relationships between objects is the next topic of discussion in this Salesforce tutorial blog.

Object Relationship In Salesforce

As the name suggests, object relationship is used in Salesforce to create a link between two objects. The question on your mind would be, why is it needed? Let me talk about the need with an example.

In my *StudentForce* app, there is a *Students Data* object, which contains personal information of students. Details regarding student's marks and their previous college are present in different objects. We can use relationships to link these objects using related fields. The marks of the students and colleges can be linked with the *Student Name* field of *Student Data* object.

Relationships can be defined while choosing the data type. They are always defined in the child object and are referenced to the common field in master object. Creating such links will help you to search and query data easily when the required data is present in different objects. There are three different types of relationships that can exist between objects. They are:

1. Master-Detail
2. Lookup
3. Junction

Master-Detail Relationship (1:n)

Master-Detail relationship is a parent-child relationship in which the master object controls the behaviour of the dependent object. It is a 1:n relationship, in which there can be only one parent, but many children. In my example, *Students Data* is the master object and *Marks* is the child object.

Let me give you an example of a Master-Detail relationship. The *Students Data* object contains student records. Each record contains personal information about a student. However, the marks obtained by students are present in another record called *Marks*. Look at the screenshot of *Marks* object below.

I have created a link between these two objects by using the student's name. Below are the points you have to keep in mind when setting up a Master-Detail relationship.

- Being the controlling object, the master field cannot be empty.
- If a record/ field in master object is deleted, the corresponding fields in the dependent object are also deleted. This is called a cascade delete.
- Dependent fields will inherit the owner, sharing and security settings from its master.

You can define master-detail relationships between two custom objects, or between a custom object and standard object as long as the standard object is the master in the relationship.

Lookup Relationship (1:n)

Lookup relationships are used when you want to create a link between two objects, but without the dependency on the parent object. You can think of this as a form of parent-child relationship where there is only one parent, but many children i.e. 1:n relationship. Below are the points you have to keep in mind when setting up a Lookup relationship.

- The lookup field on the child object is not necessarily required.
 - The fields/ records in a child object cannot be deleted by deleting a record in the parent object. Thus the records in the child object will not be affected.
- The child fields will not inherit the owner, sharing and security settings of its parent.

An example of a lookup relationship in my case would be that of a *College* object. You can see the child object: *Students Data* in the below screenshot. You will notice that there is an empty *College* field for the first record. This indicates that the dependency is not a necessity.

Below is a screenshot of the schema diagram of both the relationships. *College – Student Data* forms the Lookup relationship and *Student Data – Marks* forms the Master-Detail relationship.

Self-Relationship

This is a form of lookup relationship where instead of two tables/ objects, the relationship is within the same table/ object. Hence the name self-relationship. Here, the lookup is referenced to the same table. This relationship is also called Hierarchical relationship.

Junction Relationship (Many-To-Many)

This kind of a relationship can exist when there is a need to create two master-detail relationships. Two master-detail relationships can be created by linking 3 custom objects. Here, two objects will be master objects and the third object will be dependent on both the objects. In simpler words, it will be a child object for both the master objects.

To give you an example of this relationship, I have created two new objects.

- A master object called *Professor*. It contains the list of professors.
- A child object called *Courses*. It contains the list of courses available.
- I will use the *Students Data* object as another master object.

I have created a many-to-many relationship such that every record in the *Courses* object must have at least one student and at least one professor. This is because every course is a combination

of students and professors. In fact, a course, can have one or more number of students and professors associated with them.

The dependency on *Student* and *Professor* objects makes *Courses* as the child object. *Student* and *Professor* are thus the master objects. Below is a screenshot of *Courses* object.

You will notice that there are different combinations of professors and students for these subjects. For example, Kate is associated with two courses and has two different professors for each of those two courses. Mike is associated with only one course, but, has two different professors for that course. Both Joe and Kate are associated with the same course and same professor. In the below screenshot, you will find the schematic diagram of this relationship.

FAQs

1. What are the features of Salesforce?
2. How many types of objects are available in Salesforce?
3. Differentiate standard objects with custom objects
4. What is workflow in Salesforce?
5. what is the difference between lookup and master relationship?

Assignment 5

TITLE: Setup your own cloud for Software as a Service (SaaS)

PROBLEM STATEMENT: Setup your own cloud for Software as a Service (SaaS) over the existing LAN in your laboratory. In this assignment you have to write your own code for cloud controller using open-source technologies to implement with HDFS. Implement the basic operations may be like to divide the file in segments/blocks and upload/ download file on/from cloud in encrypted form

PREREQUISITES: Networking Basics

COURSE OBJECTIVES:

COURSE OUTCOMES:

CO4: Use tools and techniques in the area of Cloud Computing

CO5: Use the knowledge of Cloud Computing for problem solving

CO6: Apply the concepts Cloud Computing to design and develop applications

THEORY:

Opensource cloud platforms:

Apache CloudStack

Apache CloudStack is an open source, multi-hypervisor, multi-tenant, high-availability Infrastructure-as-a-Service CMP, which facilitates creating, deploying and managing cloud services by providing a complete stack of features and components for cloud environments. It uses existing hypervisors such as KVM, VMware vSphere, VMware ESXi, VMware vCenter and XenServer/XCP for virtualisation. CloudStack can also orchestrate the non-technical elements of service delivery such as billing and metering. It presents a range of APIs, allowing it to be integrated with any other platform.

OpenStack

OpenStack consists of a set of software tools for building and managing cloud computing platforms for public and private clouds using pooled virtual resources. The tools comprising the OpenStack platform are called projects. They handle core cloud computing services of compute, networking, storage, identity and image services. OpenStack software controls large pools of compute, storage and networking resources throughout a data centre, and is managed through a dashboard or via the OpenStack API.

Eucalyptus

Eucalyptus is an acronym for Elastic Utility Computing Architecture for Linking Your Programs to Useful Systems. It is an open source software framework that provides the platform for private cloud computing implementation on computer clusters. Eucalyptus implements Infrastructure as a Service (IaaS) methodology for solutions in private and hybrid clouds.

CASE 1

Install OpenNebula Front-end on Ubuntu 20.04 | 18.04 | 16.04

Here we'll walk through all steps to get OpenNebula Front-end engine up and running on Ubuntu 20.04/18.04 /16.04 server. I assume you already have a running instance of one of this operating systems, this can be a virtual machine or a physical server.

Step 1: Add OpenNebula and Debian repositories

Run the following commands to add epel and OpenNebula repositories on Ubuntu.

Import Repository Key:

Add the repository to the system:

Ubuntu 20.04:

For Ubuntu 18.04:

Please check the recent version of OpenNebula as you install.

Step 2: Install and Configure MySQL database

Since we're going to configure our OpenNebula to use MySQL instead of SQLite, let's configure it.

First, install MariaDB server:

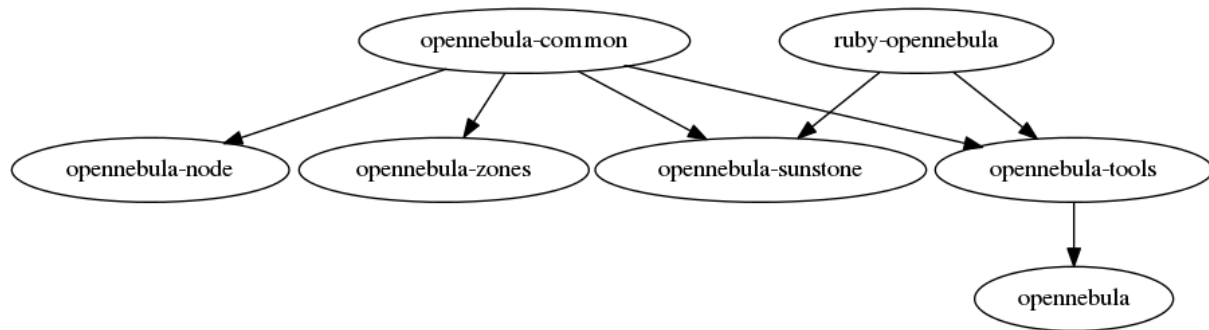
Initiate security settings of MariaDB database server:

Once MariaDB is installed and running, create a database and user for OpenNebula.

Step 3: Install OpenNebula Front-end packages

These are the packages available on OpenNebula CentOS repository:

- **opennebula-common:** Provides the user and common files.
- **ruby-opennebula:** Ruby API.
- **libopennebula-java:** Java API.
- **libopennebula-java-doc:** Java API Documentation.
- **opennebula-node:** Prepares a node as an opennebula-node.
- **opennebula-sunstone:** [Sunstone](#) (the GUI).
- **opennebula-tools:** Command Line interface.
- **opennebula-gate:** [OneGate](#) server that enables communication between VMs and OpenNebula.
- **opennebula-flow:** [OneFlow](#) manages services and elasticity.
- **opennebula:** OpenNebula Daemon.



Install all OpenNebula packages by running the commands:

Besides `/etc/one`, the following files are marked as configuration files:

- `/var/lib/one/remotes/etc/datastore/ceph/ceph.conf`
- `/var/lib/one/remotes/etc/vnm/OpenNebulaNetwork.conf`

Step 4: Ruby Runtime Installation.

Some OpenNebula components need Ruby libraries. OpenNebula provides a script that installs the required gems as well as some development libraries packages needed.

You'll get a prompt asking you to select OS and confirm.

Press **enter** to start the installation.

Step 5: Configure OpenNebula DB

Then open the fileoned.conf and edit to add MySQL db settings.

Uncomment the line:

Then add:

Replace **StrongPassword** with the password you specified earlier for the DB. Confirm that the user can log in to DB.

Step 6: Configure oneadmin credentials

A randomly generated file is usually placed on the file `/var/lib/one/.one/one_auth`.

If you need to change the password, do it here.

This will set the **oneadmin** password on the first boot. From that point, you must use the command below to change oneadmin's password.

Step 7: Configure Firewall (UFW)

We're close to the final step. If you have an active UFW firewall, you need to allow specific ports required by OpenNebula.

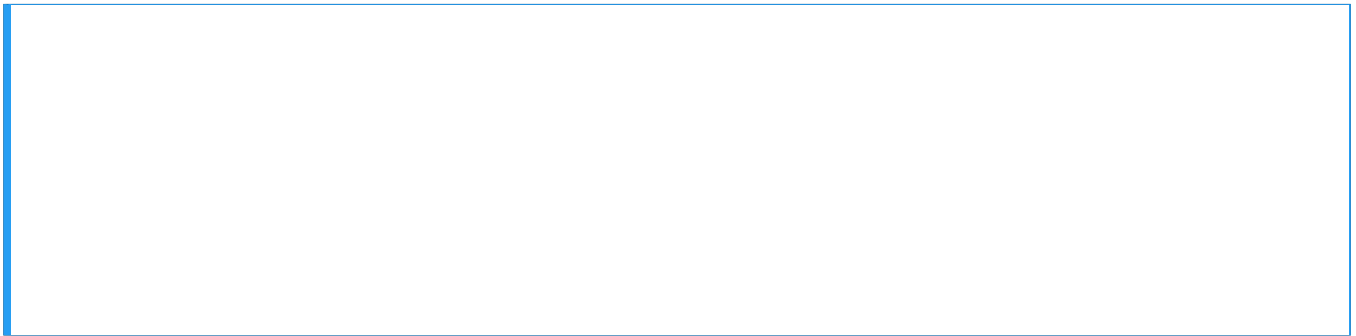
If running, open the port required to access Sunstone web interface.

Step 8: Start OpenNebula daemons.

Used systemd service management command to start OpenNebula services.

Step 10: Verifying the Installation and Accessing Sunstone UI

After OpenNebula is started for the first time, you should check that the commands can connect to the OpenNebula daemon. You can do this in the Linux CLI or in the graphical user interface: Sunstone. From CLI in the Front-end, run the following command as oneadmin:



If you get an error message, then the OpenNebula daemon could not be started properly.

The OpenNebula logs are located in **/var/log/one**, namely:

- **oned.log** – Core logs. Error messages will be located here. prefixed with **[E]**
- **sched.log** – Scheduler logs are located here.

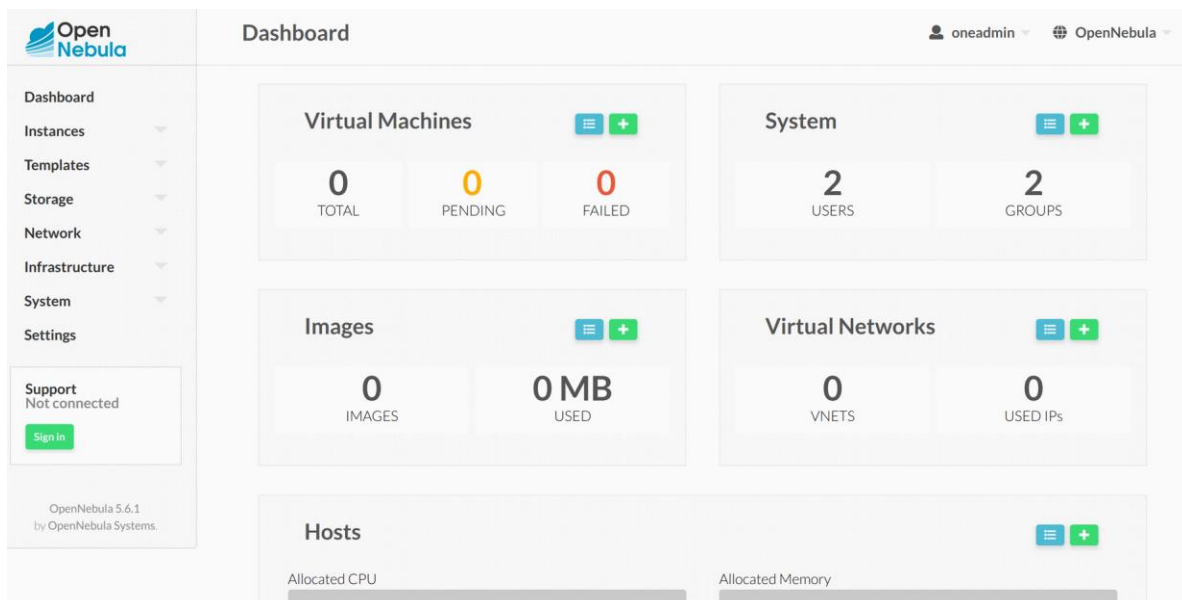
Accessing Sunstone UI

Now you can try to log in to Sunstone web interface. To do this point your browser to:

If everything is OK you will be greeted with a login page.



The user is **oneadmin** and the password is the one in the file **/var/lib/one/.one/one_auth** in your Front-end. On logging in, you should get a dashboard like below.



CASE 2:

TITLE: Study and implementation of infrastructure as Service using Open Stack.

PROBLEM STATEMENT: Study and implementation of infrastructure as Service using Open Stack.

PREREQUISITES: Networking Basics

COURSE OBJECTIVES: To Study and implement infrastructure as Service using Open Stack.

COURSE OUTCOMES: Ability to implement infrastructure as Service using Open Stack.

CO1: Ability to install and configure cloud computing environments

THEORY:

OpenStack is a free, open source platform developed specifically for cloud computing. Through this cloud-centric platform, virtual machines and other resources can be made available to clients as Infrastructure-as-a-Service. There are a number of ways to get OpenStack installed.

One such means of installing OpenStack is through DevStack. DevStack is a series of scripts that are used to bring up a complete OpenStack environment based on the latest version. Although installing with DevStack might be one of the easiest methods of installing OpenStack

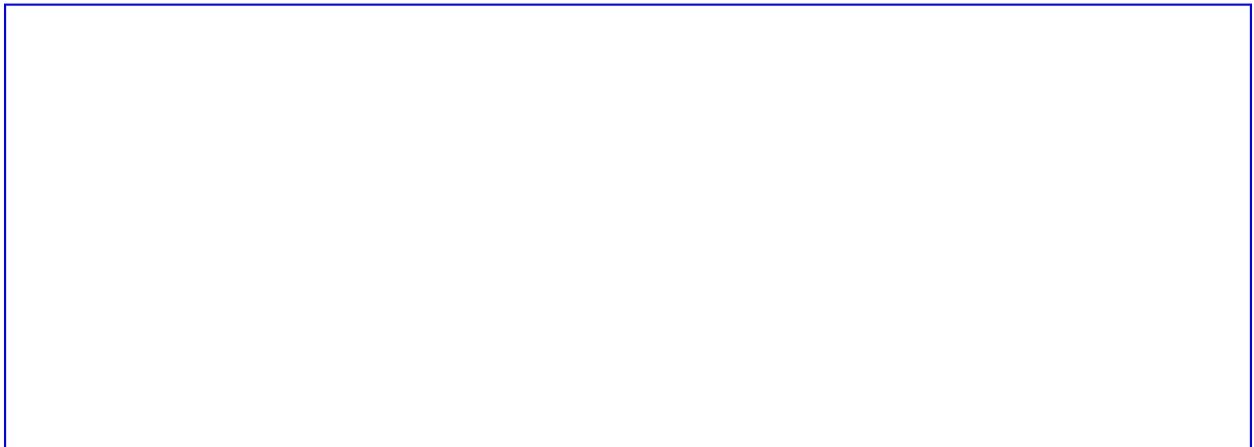
Reference Link: <https://www.tecmint.com/create-deploy-and-launch-virtual-machines-in-openstack/>

Step 1: Initial System Configurations

1. Before you begin preparing the node in order to deploy your own virtual cloud infrastructure, first login with root account and assure that the system is up to date.

2. Next, issue the ss -tulpn command to list all running services.

ss -tulpn



List All Running Linux Services

3. Next, identify, [stop, disable and remove unneeded services](#), mainly postfix, NetworkManager and firewalld. At the end the only daemon that would be running on your machine should be **sshd**.

systemctl stop postfix firewalld NetworkManager

systemctl disable postfix firewalld NetworkManager

systemctl mask NetworkManager

yum remove postfix NetworkManager NetworkManager-libnm

4. Completely disable Selinux policy on the machine by issuing the below commands. Also edit /etc/selinux/config file and set SELINUX line from **enforcing** to **disabled** as illustrated on the below screenshot.

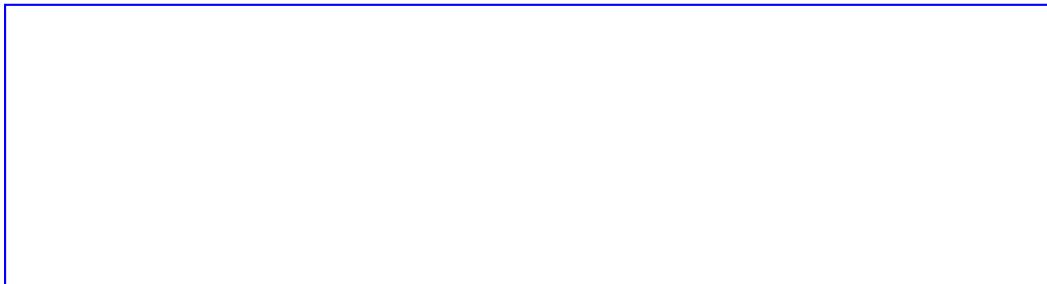
```
# setenforce 0
# getenforce
# vi /etc/selinux/config
```



Disable SELinux

5. On the next step using the **hostnamectl** command to [set your Linux system hostname](#). Replace the **FQDN** variable accordingly.

```
# hostnamectl set-hostname cloud.centos.lan
```



Set Linux System Hostname

6. Finally, install ntpdate command in order to [synchronize time with a NTP server](#) on your premises near your physical proximity.

```
# yum install ntpdate
```

Step 2: Install OpenStack in CentOS and RHEL

7. **OpenStack** will be deployed on your Node with the help of **PackStack** package provided by **rdo** repository (**RPM Distribution of OpenStack**).

In order to enable **rdo** repositories on **RHEL 7** run the below command.

```
# yum install https://www.rdoproject.org/repos/rdo-release.rpm
```

On **CentOS 7**, the **Extras** repository includes the RPM that activates the OpenStack repository. **Extras** is already enabled, so you can easily install the RPM to setup the OpenStack repository:

```
# yum install -y centos-release-openstack-mitaka
# yum update -y
```

8. Now it's time to install **PackStack** package. **Packstack** represents a utility which facilitates the deployment on multiple nodes for different components of **OpenStack** via **SSH** connections and **Puppet** modules.

Install Packstat package in Linux with the following command:

```
# yum install openstack-packstack
```

9. On the next step generate an answer file for **Packstack** with the default configurations which will be later edited with the required parameters in order to deploy a standalone installation of Openstack (single node).

The file will be named after the current day timestamp when generated (day, month and year).

```
# packstack --gen-answer-file='date +%d.%m.%y'.conf
# ls
```



Generate Packstack Answer Configuration File

10. Now edit the generated answer configuration file with a text

editor. # vi 13.04.16.conf

and replace the following parameters to match the below values. In order to be safe replace the passwords fields accordingly.

CONFIG_NTP_SERVERS=0.ro.pool.ntp.org

Please consult <http://www.pool.ntp.org/en/> server list in order to use a public NTP server near your physical location.



Add NTP Server in Packstack

CONFIG_PROVISION_DEMO=n

Add Provision in Packstack

CONFIG_KEYSTONE_ADMIN_PW=your_password for Admin user

Add Admin Account in Packstack

Access OpenStack dashboard via HTTP with SSL

enabled. CONFIG_HORIZON_SSL=y

Enable HTTPS for OpenStack

The root password for MySQL server.

CONFIG_MARIADB_PW=mypassword1234

Set MySQL Root Password in OpenStack

Setup a password for **nagiosadmin** user in order to access Nagios web panel.

CONFIG_NAGIOS_PW=nagios1234

Set Nagios Admin Password

11. After you finished editing save and close the file. Also, open SSH server configuration file and uncomment **PermitRootLogin** line by removing the front hashtag as illustrated on the below screenshot.

```
# vi /etc/ssh/sshd_config
```



Enable SSH Root Login

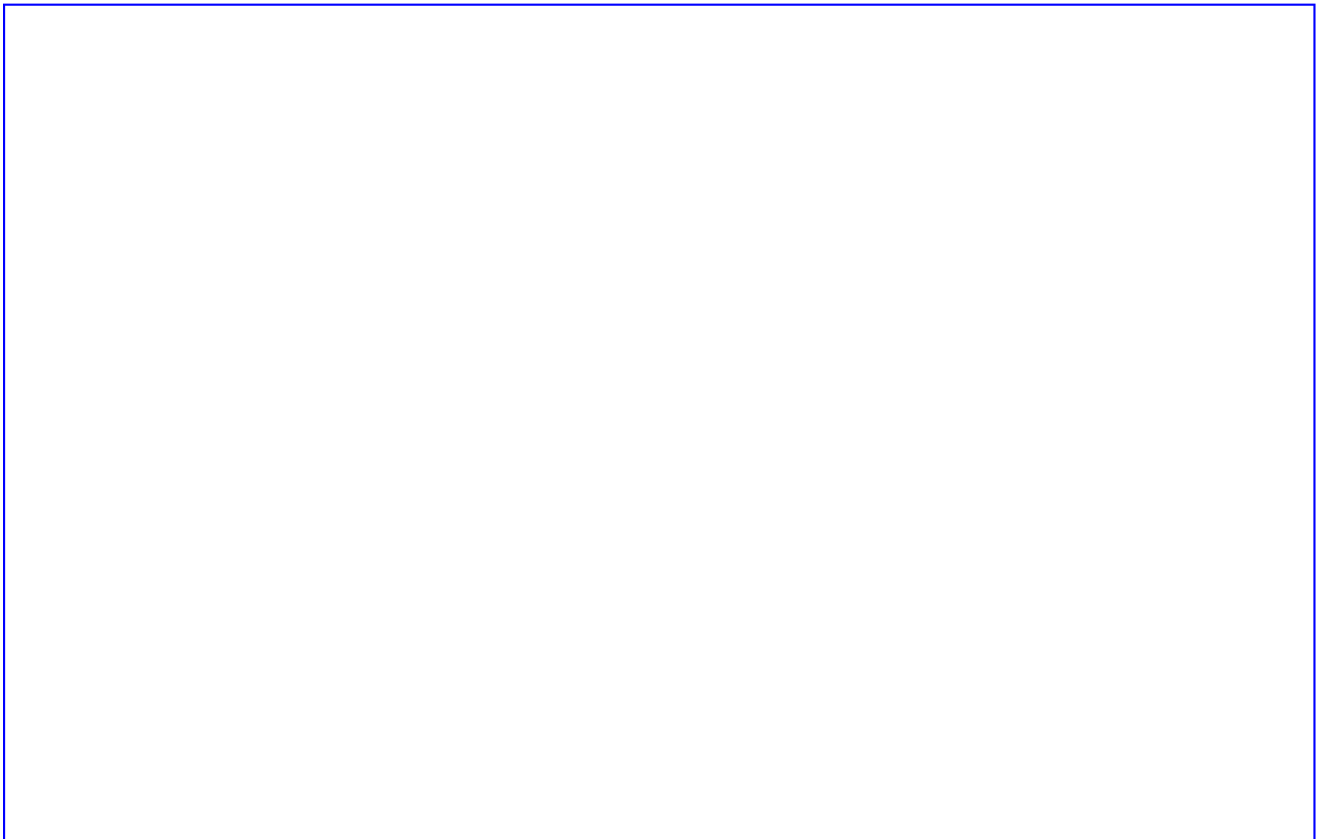
Then restart SSH service to reflect changes.

```
# systemctl restart sshd
```

Step 3: Start Openstack Installation Using Packstack Answer File

12. Finally start **Openstack** installation process via the answer file edited above by running the below command syntax:

```
# packstack --answer-file 13.04.16.conf
```



Openstack Installation in Linux

13. Once the installation of OpenStack components is successfully completed, the installer will display a few lines with the local dashboard links for **OpenStack** and **Nagios** and the required credentials already configured above in order to login on both panels.



OpenStack Installation Completed

The credentials are also stored under your home directory in keystone_admin file.

14. If for some reasons the installation process ends with an error regarding httpd service, open **/etc/httpd/conf.d/ssl.conf** file and make sure you comment the following line as illustrated below.

#Listen 443 https



Disable HTTPS SSL Port

Then restart Apache daemon to apply

changes. # systemctl restart httpd.service

Note: In case you still can't browse Openstack web panel on port **443** restart the installation process from beginning with the same command issued for the initial deployment.

```
# packstack --answer-file /root/13.04.16.conf
```

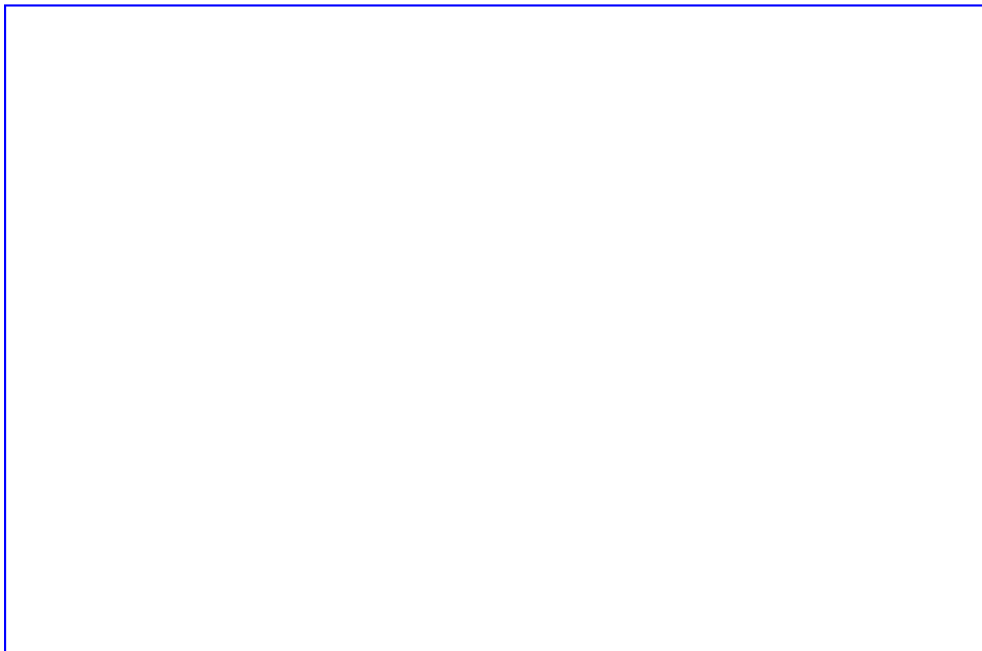
Step 4: Remotely Access OpenStack Dashboard

15. In order to access **OpenStack** web panel from a remote host in your LAN navigate to your machine IP Address or FQDN/dashboard via HTTPS protocol.

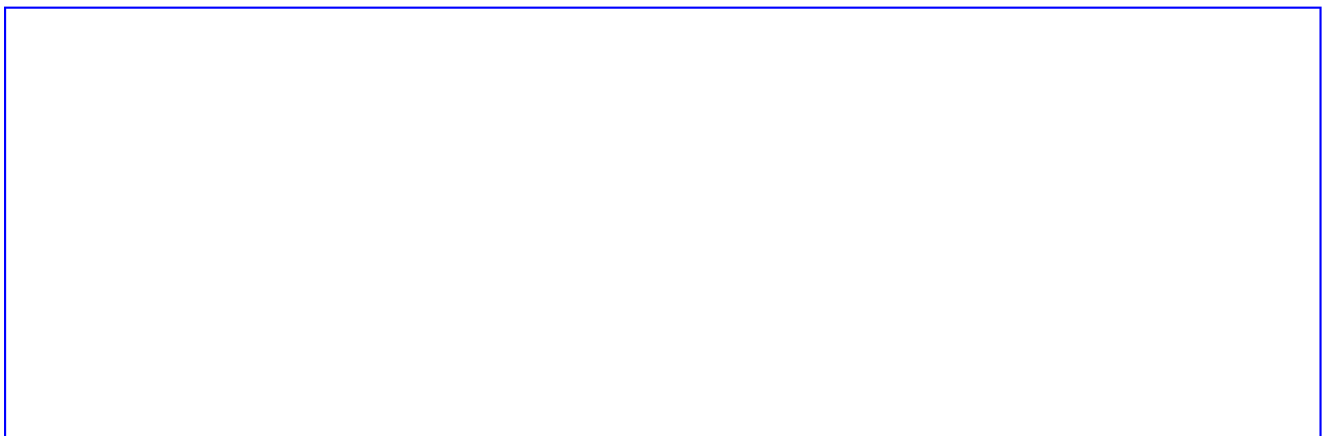
Due to the fact that you're using a **Self-Signed Certificate** issued by an untrusted **Certificate Authority** an error should be displayed on your browser.

Accept the error and login to the dashboard with the user admin and the password set on CONFIG_KEYSTONE_ADMIN_PW parameter from answer file set above.

<https://192.168.1.40/dashboard>



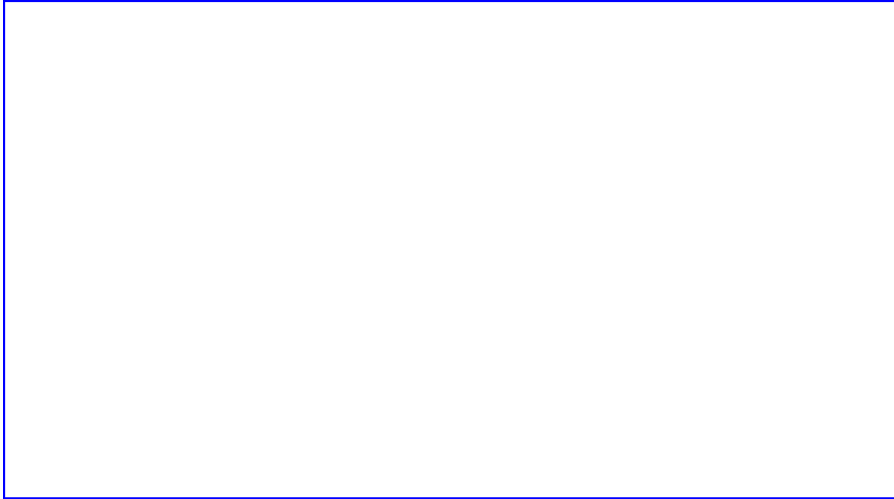
OpenStack Login Dashboard



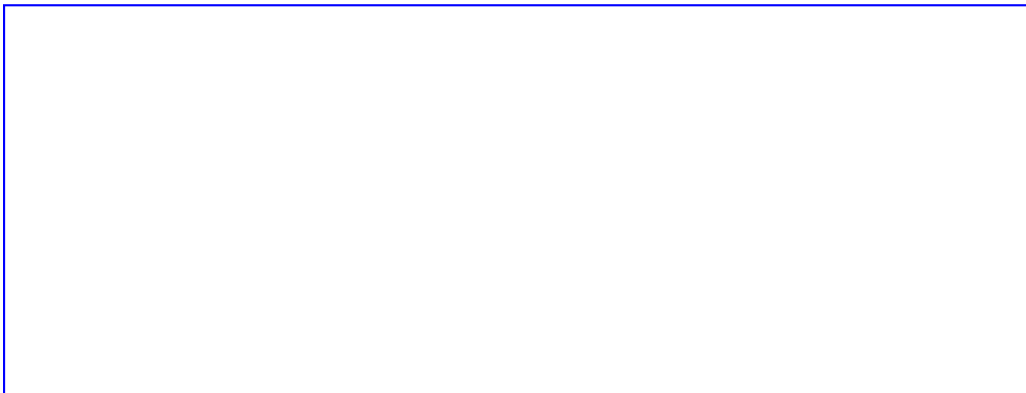
Openstack Projects

16. Alternatively, if you opted to install **Nagios** component for **OpenStack**, you can browse **Nagios** web panel at the following URI and login with the credentials setup in answer file.

<https://192.168.1.40/nagios>



Nagios Login Dashboard



Nagios Linux Monitoring Interface

Now you can start setup your own internal cloud environment.

Reference : <https://www.tecmint.com/openstack-networking-guide/>
to Configure OpenStack Network to Enable Access to OpenStack Instances

CONCLUSION: Thus, we have studied and implemented openstack