

Practical No.	Laboratory Assignments
Cloud Computing	
1	Case study on Amazon EC2 and learn about Amazon EC2 web services.
2	Installation and configure Google App Engine.
3	Creating an Application in Salesforce.com using Apex programming Language.
4	Design and develop custom Application (Mini Project) using Salesforce 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.



Study material provided by: Vishwajeet Londhe

Join Community by clicking below links



Telegram Channel



https://t.me/SPPU_TE_BE_COMP

(for all engineering Resources)



WhatsApp Channel

(for all tech updates)



<https://whatsapp.com/channel/0029ValjFriICVfpcV9HFc3b>



Insta Page

(for all engg & tech updates)



@SPPU_ENGINEERING_UPDATE

https://www.instagram.com/sppu_engineering_update

Practical No: 01

Practical Title: Case study on Amazon EC2 and learn about Amazon EC2 web services.

Objectives:

- To learn Amazon EC2 web services
- To study on Amazon EC2 and learn about Amazon EC2 web services.

Hardware Requirements :

- Pentium IV with latest configuration

Software Requirements :

- Ubuntu 20.04

Theory:

An EC2 instance is nothing but a virtual server in Amazon [Web services](#) terminology. It stands for Elastic Compute Cloud. It is a web service where an AWS subscriber can request and provision a compute server in AWS cloud.

An on-demand EC2 instance is an offering from AWS where the subscriber/user can rent the virtual server per hour and use it to deploy his/her own applications.

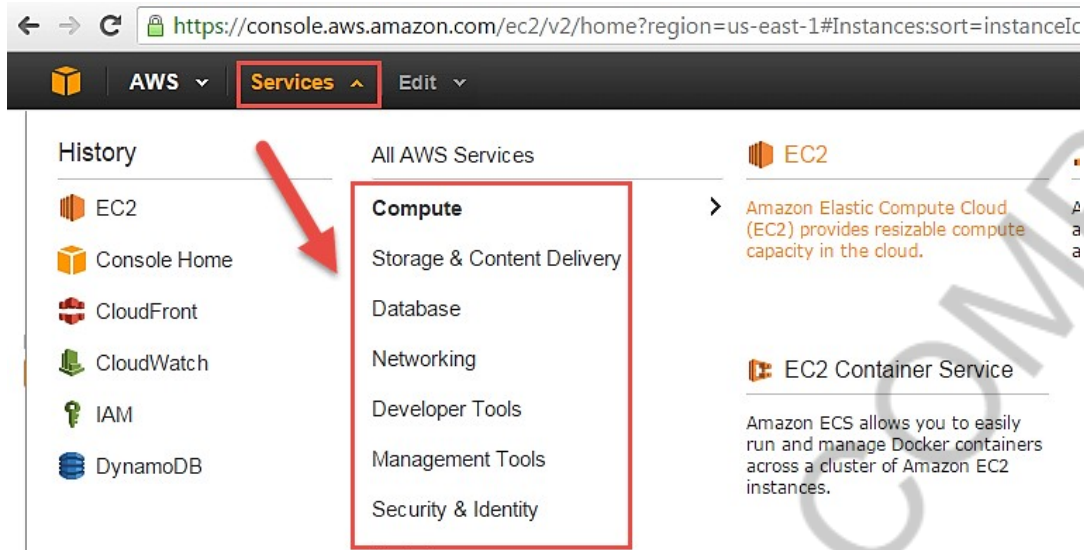
The instance will be charged per hour with different rates based on the type of the instance chosen. AWS provides multiple instance types for the respective business needs of the user.

Thus, you can rent an instance based on your own CPU and memory requirements and use it as long as you want. You can terminate the instance when it's no more used and save on costs. This is the most striking advantage of an on-demand instance- you can drastically save on your CAPEX.

Let us see in detail how to launch an on-demand EC2 instance in AWS Cloud. Login and access to AWS services

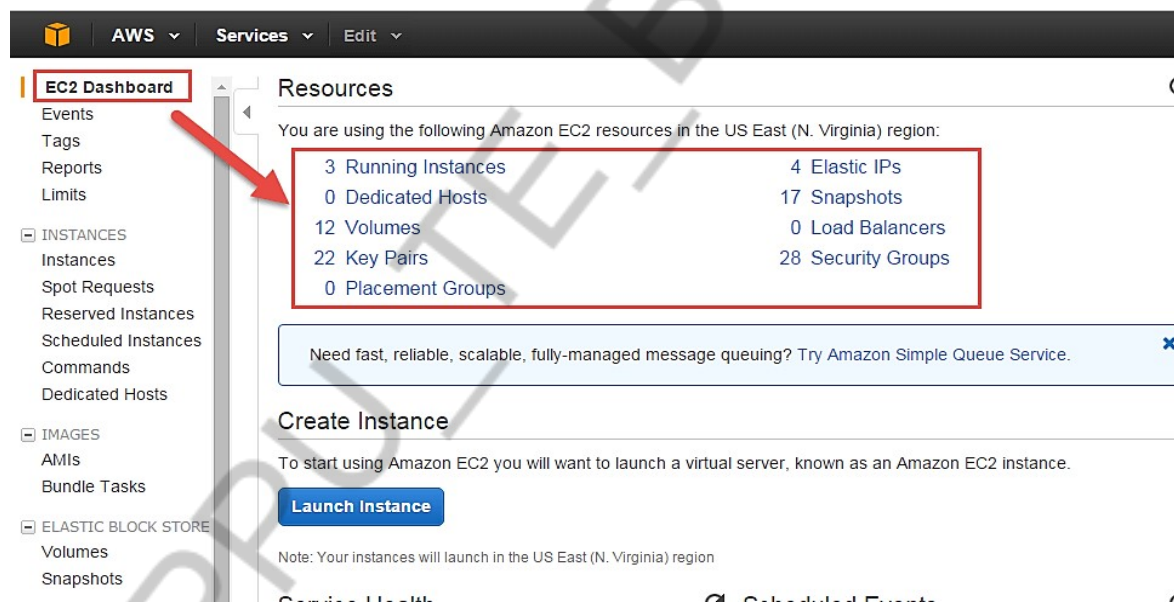
Step 1) In this step,

- Login to your AWS account and go to the AWS Services tab at the top left corner.
- Here, you will see all of the AWS Services categorized as per their area viz. Compute, Storage, Database, etc. For creating an EC2 instance, we have to choose Computeà EC2 as in the next step.



- Open all the services and click on EC2 under Compute services. This will launch the dashboard of EC2.

Here is the EC2 dashboard. Here you will get all the information in gist about the AWS EC2 resources running.

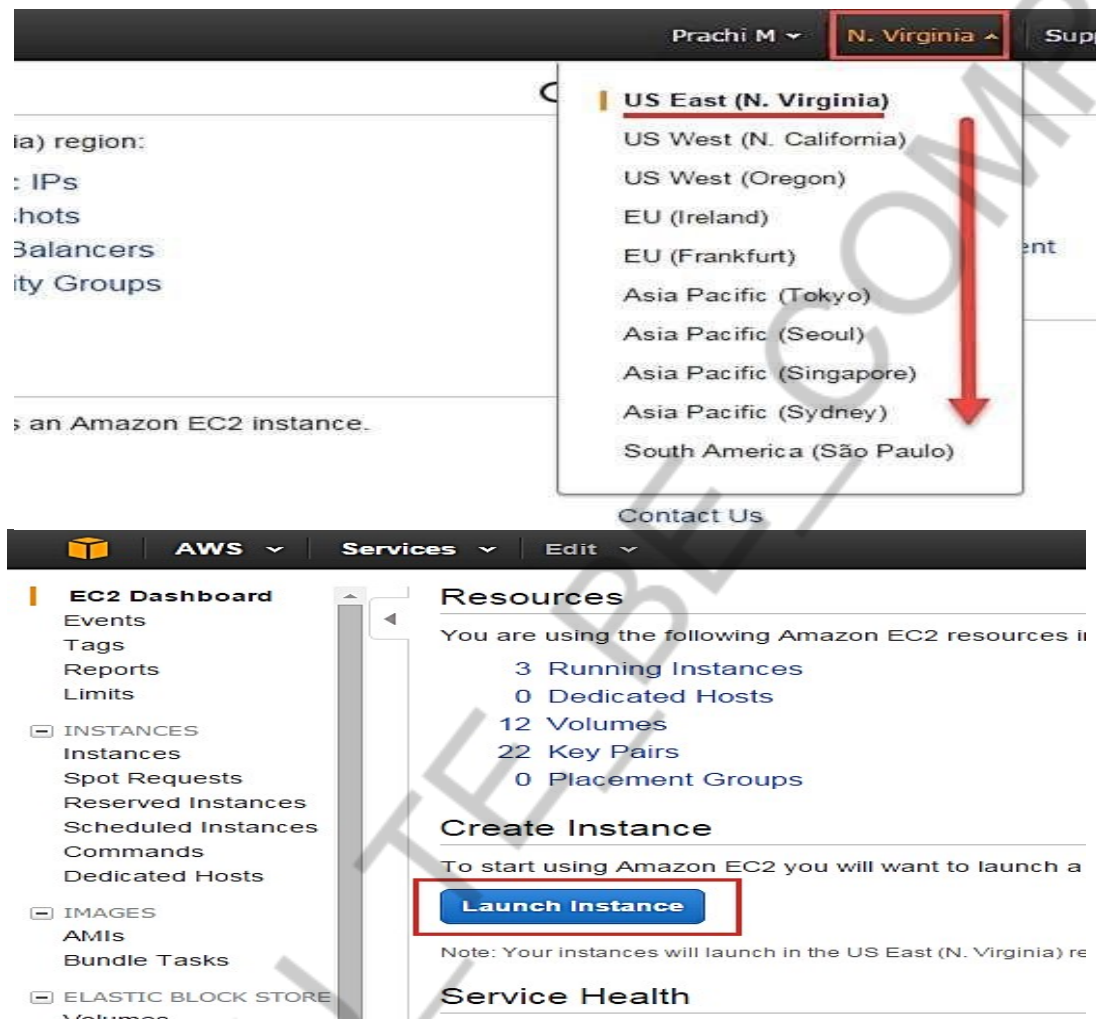


Step 2) On the top right corner of the EC2 dashboard, choose the AWS Region in which you want to provision the EC2 server.

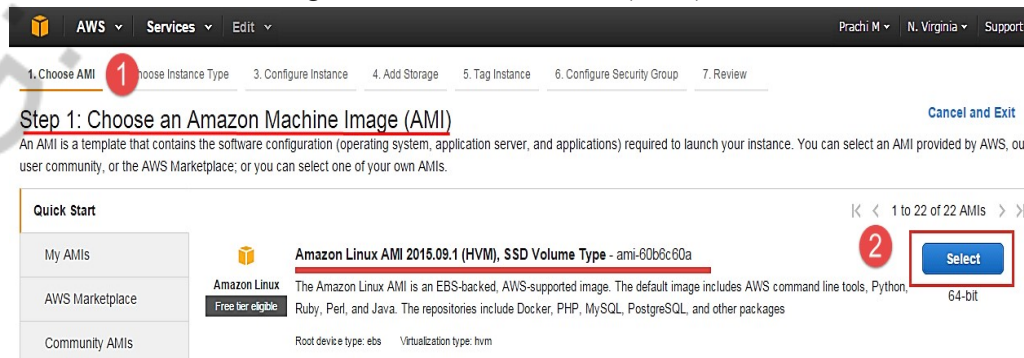
Here we are selecting N. Virginia. AWS provides 10 Regions all over the globe

Step 3) In this step

- Once your desired Region is selected, come back to the EC2 Dashboard.
- Click on 'Launch Instance' button in the section of Create Instance (as shown below).



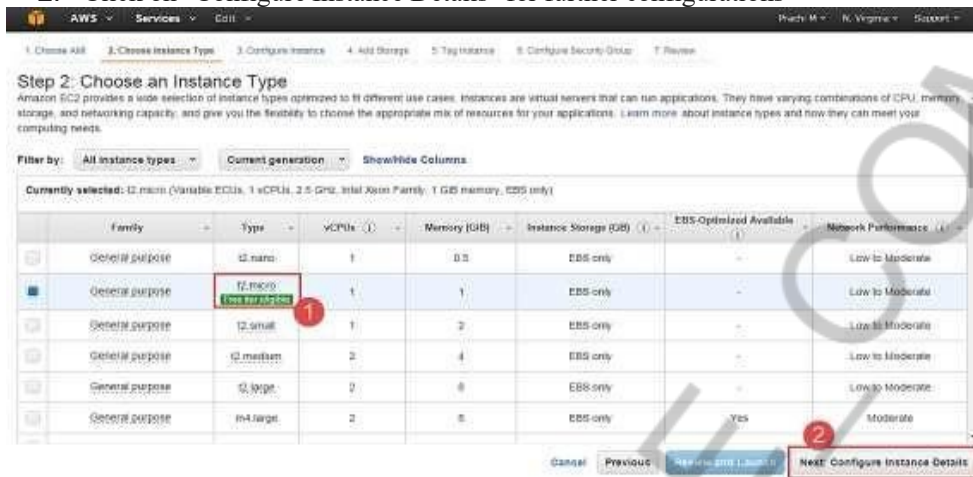
- Instance creation wizard page will open as soon as you click 'Launch Instance'. Choose AMI
- Step 1) In this step we will do,
1. You will be asked to choose an AMI of your choice. (An AMI is an Amazon Machine Image. It is a template basically of an Operating System platform which you can use as a base to create your instance). Once you launch an EC2 instance from your preferred AMI, the instance will automatically be booted with the desired OS. (We will see more about AMIs in the coming part of the tutorial).
 2. Here we are choosing the default Amazon Linux (64 bit) AMI.



Choose EC2 Instance Types

Step 1) In the next step, you have to choose the type of instance you require based on your business needs.

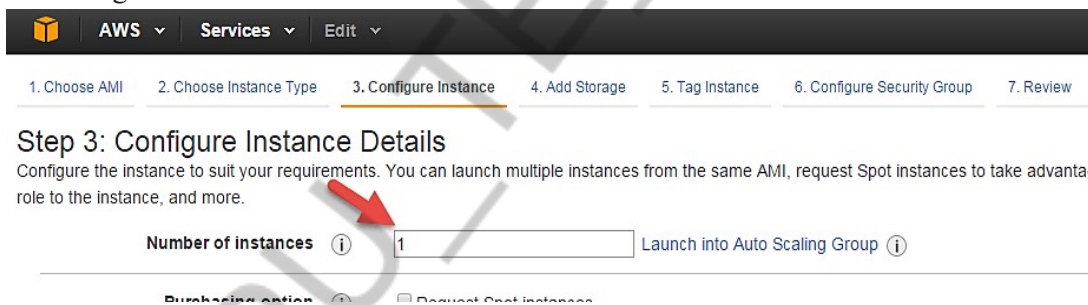
1. We will choose t2.micro instance type, which is a 1vCPU and 1GB memory server offered by AWS.
2. Click on "Configure Instance Details" for further configurations



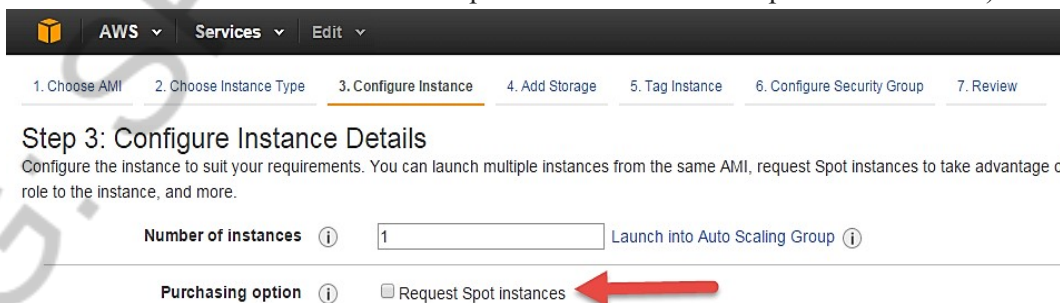
- In the next step of the wizard, enter details like no. of instances you want to launch at a time.
- Here we are launching one

instance. Configure Instance

Step 1) No. of instances- you can provision up to 20 instances at a time. Here we are launching one instance.



Step 2) Under Purchasing Options, keep the option of 'Request Spot Instances' unchecked as of now. (This is done when we wish to launch Spot instances instead of on-demand ones. We will come back to Spot instances in the later part of the tutorial).



Step 3) Next, we have to configure some basic networking details for our EC2 server.

- You have to decide here, in which VPC (Virtual Private Cloud) you want to launch your instance and under which subnets inside your VPC. It is better to determine and plan this prior to launching the instance. Your AWS architecture set-up should include IP ranges for your subnets etc. pre-planned for better management. (We will see how to create a new VPC in Networking section of the tutorial).

- Subnetting should also be pre-planned. E.g.: If it's a web server you should place it in the public subnet and if it's a DB server, you should place it in a private subnet all inside your VPC.

Below,

1. Network section will give a list of VPCs available in our platform.
2. Select an already existing VPC
3. You can also create a new VPC

Here I have selected an already existing VPC where I want to launch my instance.

Step 3: Configure Instance Details
Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the role to the instance, and more.

Number of instances [Launch into Auto Scaling Group](#)

Purchasing option ☐ Request Spot instances

Network [Create new VPC](#)

Subnet [Create new subnet](#)

Auto-assign Public IP ☐ [Create new IAM role](#)

IAM role [Create new IAM role](#)

Step 4) In this step,

- A VPC consists of subnets, which are IP ranges that are separated for restricting access.
- Below,
- 1. Under Subnets, you can choose the subnet where you want to place your instance.
- 2. I have chosen an already existing public subnet.
- 3. You can also create a new subnet in this step.

Step 3: Configure Instance Details
Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the role to the instance, and more.

Number of instances [Launch into Auto Scaling Group](#)

Purchasing option ☐ Request Spot instances

Network [Create new VPC](#)

Subnet [Create new subnet](#)

Auto-assign Public IP ☐ [Create new IAM role](#)

IAM role [Create new IAM role](#)

- Once your instance is launched in a public subnet, AWS will assign a dynamic public IP to it from their pool of IPs.

Step 5) In this step,

- You can choose if you want AWS to assign it an IP automatically, or you want to do it manually later. You can enable/ disable 'Auto assign Public IP' feature here likewise.

- Here we are going to assign this instance a static IP called as EIP (Elastic IP) later. So we keep this feature disabled as of now.

AWS Services Edit

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances 1 Launch into Auto Scaling Group

Purchasing option ☐ Request Spot instances

Network vpc-d5194fb0 (192.168.0.0/16) | Prachi_Test - VPC Create new VPC

Subnet subnet-b3e3d0ea (192.168.2.0/24) | Prachi_Test-Pi Create new subnet
251 IP Addresses available

Auto-assign Public IP Use subnet setting (Disable) Use subnet setting (Disable) Enable Disable

IAM role Create new IAM role

Shutdown behavior Stop

AWS Services Edit

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

Step 3: Configure Instance Details

Auto-assign Public IP Use subnet setting (Disable)

IAM role None Create new IAM role

Shutdown behavior Stop

Enable termination protection ☒ Protect against accidental termination

Monitoring ☐ Enable CloudWatch detailed monitoring
Additional charges apply.

Tenancy Shared - Run a shared hardware instance Shared - Run a shared hardware instance Dedicated - Run a Dedicated instance Dedicated host - Launch this instance on a Dedicated host

Network interfaces

AWS Services Edit

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances 1 Launch into Auto Scaling Group

Purchasing option ☐ Request Spot instances

Network vpc-d5194fb0 (192.168.0.0/16) | Prachi_Test - VPC Create new VPC

Subnet subnet-b3e3d0ea (192.168.2.0/24) | Prachi_Test-Pi Create new subnet
251 IP Addresses available

Auto-assign Public IP Use subnet setting (Disable)

IAM role None Create new IAM role

Shutdown behavior Stop

Enable termination protection ☒ Protect against accidental termination

Monitoring ☐ Enable CloudWatch detailed monitoring
Additional charges apply.

Tenancy Shared - Run a shared hardware instance

Cancel Previous Review and Launch Next: Add Storage

Launch Status



Your instances are now launching

The following instance launches have been initiated: [i-4c2c3cff](#) [Hide launch log](#)

Creating security groups	Successful (sg-62d7d21b)
Authorizing inbound rules	Successful
Initiating launches	Successful
Applying tags	Successful
Launch initiation complete	



Get notified of estimated charges

Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an am

The screenshot shows the AWS Management Console interface. On the left, the navigation pane includes sections for EC2 Dashboard, INSTANCES, IMAGES, ELASTIC BLOCK STORE, and NETWORK & SECURITY. The 'INSTANCES' section is expanded, showing a list of instances. The instance 'Dev_Web Server 01' with ID 'i-4c2c3cff' is selected. The instance details are displayed in the main pane, showing it is in the 'running' state. A red arrow points to the 'Private IP: 192.168.2.167' field. The details are organized into two columns: Description and Status Checks, Monitoring, Tags, and Public DNS.

Description		Status Checks	Monitoring	Tags	Public DNS
Instance ID	i-4c2c3cff	Instance state	running		
Instance type	t2.micro	Instance type	t2.micro		
Private DNS	ip-192-168-2-167.ec2.internal	Private DNS	192.168.2.167		
Private IPs	192.168.2.167	Secondary private IPs			
VPC ID	vpc-45194fb0	Subnet ID	subnet-b3e3d4aa		
Network interfaces	eth0	Source/dest. check	True		
ClassicLink		EBS-optimized	False		
Availability zone	us-east-1b	Security groups	Web Server SG - view rules		
Scheduled events	No scheduled events	AMI ID	ami-ami-hvm-2015.09.1.x86_64-gp2 (ami-60b6c60a)		
Platform		IAM role			
Key pair name	Dev Key	Owner	018611290429		
Launch time	February 3, 2016 at 7:52:22 PM UTC+5:30 (less than one hour)				

Conclusion:

Thus, we saw in detail how to create an on-demand EC2 instance in this tutorial. Because it is an on-demand server, you can keep it running when in use and 'Stop' it when it's unused to save on your costs

Practical No : 2

Practical Title: Installation and configure Google App Engine.

Objectives:

- To learn basic of Google App Engine.
- To install and configure Google App Engine.

Hardware Requirements :

- Pentium IV with latest configuration

Software Requirements :

- Ubuntu 20.04, Web application i.e. Google App Engine

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.

In particular, Google App Engine is designed to host applications with many simultaneous users. When an application can serve many simultaneous users without degrading performance, we say it scales. Applications written for App Engine scale automatically. As more people use the application, App Engine allocates more resources for the application and manages the use of those resources. The application itself does not need to know anything about the resources it is using.

The app engine is a Cloud-based platform, is quite comprehensive and combines infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS). The app engine supports the delivery, testing and development of software on demand in a Cloud computing environment that supports millions of users and is highly scalable.

The company extends its platform and infrastructure to the Cloud through its app engine. It presents the platform to those who want to develop SaaS solutions at competitive costs. Have you ever wondered as to who stands to benefit the most from the Google app engine? If you are a business SME or enterprise which owns any web-based application that needs to be scaled

without any compromise on the performance then Google App Engine is a good fit. Companies like Best Buy and Khan Academy have chosen Google App Engine for their apps.

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.

As soon as you have signed up for a Cloud account, you can build your app:

- ☐ With the template/HTML package in Go
- ☐ With Jinja2 and webapp2 in Python
- ☐ With Cloud SQL in PHP
- ☐ With Maven in Java

Generally Available Features

These are covered by the depreciation policy and the service-level agreement of the app engine. Any changes made to such a feature are backward-compatible and implementation of such a feature is usually stable. These include data storage, retrieval, and search; communications; process management; computation; app configuration and management.

- ☐ Data storage, retrieval, and search include features such as HRD migration tool, Google Cloud SQL, logs, datastore, dedicated Memcache, blobstore, Memcache and search.
- ☐ Communications include features such as XMPP. channel, URL fetch, mail, and Google Cloud Endpoints.
- ☐ Process management includes features like scheduled tasks and task queue
- ☐ Computation includes images.
- ☐ App management and configuration cover app identity, users, capabilities, traffic splitting, modules, SSL for custom domains, modules, remote access, and multitenancy.

Advantages of Google App Engine:

☐ **Infrastructure for Security**

Around the world, the Internet infrastructure that Google has is probably the most secure. There is rarely any type of unauthorized access till date as the application data and code are stored in highly secure servers. You can be sure that your app will be available to users worldwide at all times since Google has several hundred servers globally. Google's security and privacy policies are applicable to the apps developed using Google's infrastructure.

☐ **Scalability**

For any app's success, this is among the deciding factors. Google creates its own apps using GFS, Big Table and other such technologies, which are available to you when you

utilize the Google app engine to create apps. You only have to write the code for the app and Google looks after the testing on account of the automatic scaling feature that the app engine has. Regardless of the amount of data or number of users that your app stores, the app engine can meet your needs by scaling up or down as required.

☐ **Performance and Reliability**

Google is among the leaders worldwide among global brands. So, when you discuss performance and reliability you have to keep that in mind. In the past 15 years, the company has created new benchmarks based on its services' and products' performance. The app engine provides the same reliability and performance as any other Google product.

☐ **Cost Savings**

You don't have to hire engineers to manage your servers or to do that yourself. You can invest the money saved into other parts of your business.

☐ **Platform Independence**

You can move all your data to another environment without any difficulty as there is not many dependencies on the app engine platform.

Conclusion :

Thus, We have installed and Configured Google App Engine.

Practical No: 3

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

Objectives:

- To learn salesforce cloud administration
- To create application in Salesforce.com using Apex programming

Hardware Requirements :

- Pentium IV with latest configuration

Software Requirements :

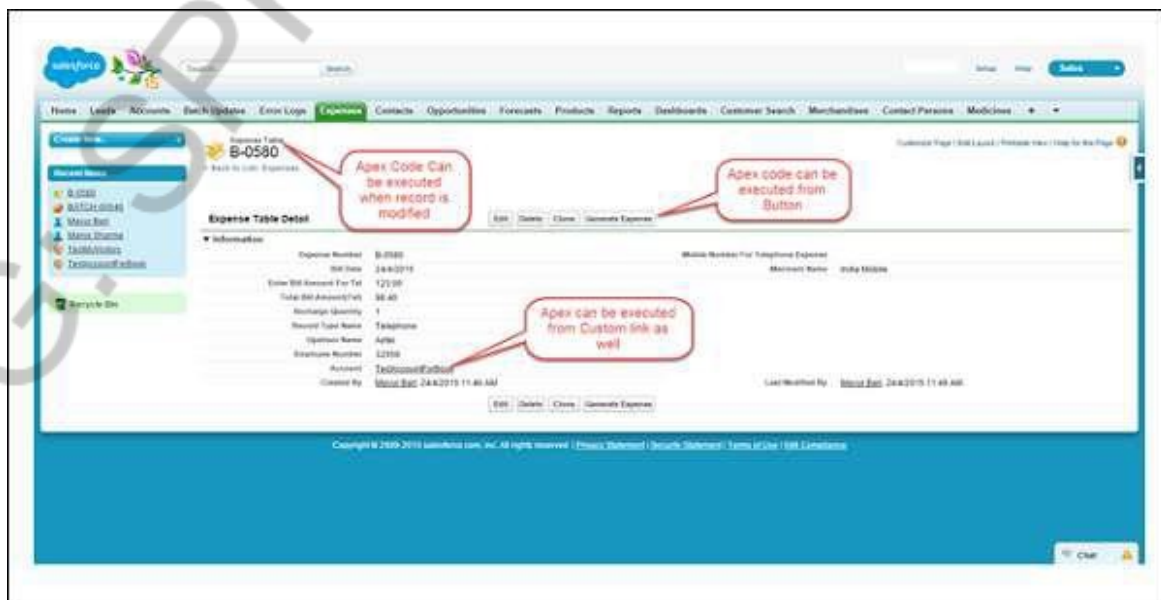
- Ubuntu 20.04, Web application i.e. salesforce.com

Theory:

What is Apex?

Apex is a proprietary language developed by the Salesforce.com. As per the official definition, Apex is a strongly typed, object-oriented programming language that allows developers to execute the flow and transaction control statements on the Force.com platform server in conjunction with calls to the Force.com API.

It has a Java-like syntax and acts like database stored procedures. It enables the developers to add business logic to most system events, including button clicks, related record updates, and Visual force pages. Apex code can be initiated by Web service requests and from triggers on objects. Apex is included in Performance Edition, Unlimited Edition, Enterprise Edition, and Developer edition.



Features of Apex as a Language

Let us now discuss the features of Apex as a Language – Integrated

Apex has built in support for DML operations like INSERT, UPDATE, DELETE and also DML Exception handling. It has support for inline SOQL and SOSL query handling which returns the set of sObject records. We will study the sObject, SOQL, SOSL in detail in future chapters.

☐ **Java like syntax and easy to use**

Apex is easy to use as it uses the syntax like Java. For example, variable declaration, loop syntax and conditional statements.

☐ **Strongly Integrated With Data**

Apex is data focused and designed to execute multiple queries and DML statements together. It issues multiple transaction statements on Database.

☐ **Strongly Typed**

Apex is a strongly typed language. It uses direct reference to schema objects like Object and any invalid reference quickly fails if it is deleted or if is of wrong data type.

☐ **Multitenant Environment**

Apex runs in a multitenant environment. Consequently, the Apex runtime engine is designed to guard closely against runaway code, preventing it from monopolizing shared resources. Any code that violates limits fails with easy-to-understand error messages.

☐ **Upgrades Automatically**

Apex is upgraded as part of Salesforce releases. We don't have to upgrade it manually.

☐ **Easy Testing**

Apex provides built-in support for unit test creation and execution, including test results that indicate how much code is covered, and which parts of your code can be more efficient.

When Should Developer Choose Apex?

Apex should be used when we are not able to implement the complex business functionality using the pre-built and existing out of the box functionalities. Below are the cases where we need to use apex over Salesforce configuration.

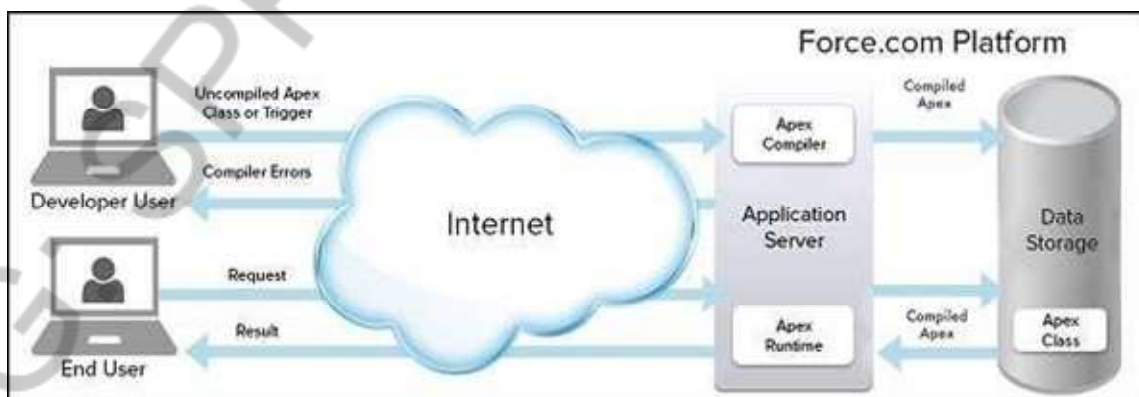
Apex Applications

We can use Apex when we want to –

- ☐ Create Web services with integrating other systems. Create email services for email blast or email setup.
- ☐ Perform complex validation over multiple objects at the same time and also custom validation implementation.
- ☐ Create complex business processes that are not supported by existing workflow functionality or flows.
- ☐ Create custom transactional logic (logic that occurs over the entire transaction, not just with a single record or object) like using the Database methods for updating the records.
- ☐ Perform some logic when a record is modified or modify the related object's record when there is some event which has caused the trigger to fire.

Working Structure of Apex

As shown in the diagram below (Reference: Salesforce Developer Documentation), Apex runs entirely on demand Force.com Platform.



Flow of Actions

There are two sequence of actions when the developer saves the code and when an end user performs some action which invokes the Apex code as shown below –

Developer Action

When a developer writes and saves Apex code to the platform, the platform application server first compiles the code into a set of instructions that can be understood by the Apex runtime interpreter, and then saves those instructions as metadata.

End User Action

When an end-user triggers the execution of Apex, by clicking a button or accessing a Visual force page, the platform application server retrieves the compiled instructions from the metadata and sends them through the runtime interpreter before returning the result. The end user observes no differences in execution time as compared to the standard application platform request.

Since Apex is the proprietary language of Salesforce.com, it does not support some features which a general programming language does. Following are a few features which Apex does not support –

- ☐ It cannot show the elements in User Interface.
- ☐ You cannot change the standard SFDC provided functionality and also it is not possible to prevent the standard functionality execution.
- ☐ You cannot change the standard SFDC provided functionality and also it is not possible to prevent the standard functionality execution.
- ☐ Creating multiple threads is also not possible as we can do it in other languages.

Understanding the Apex Syntax

Apex code typically contains many things that we might be familiar with from other programming languages.

Variable Declaration

As strongly typed language, you must declare every variable with data type in Apex. As seen in the code below (screenshot below), lstAcc is declared with data type as List of Accounts.

SOQL Query

This will be used to fetch the data from Salesforce database. The query shown in screenshot below is fetching data from Account object.

Loop Statement

This loop statement is used for iterating over a list or iterating over a piece of code for a specified number of times. In the code shown in the screenshot below, iteration will be same as the number of records we have.

Flow Control Statement

The If statement is used for flow control in this code. Based on certain condition, it is decided whether to go for execution or to stop the execution of the particular piece of code. For example, in the code shown below, it is checking whether the list is empty or it contains records.

DML Statement

Performs the records insert, update, upsert, delete operation on the records in database. For example, the code given below helps in updating Accounts with new field value.

Apex Code Development Tools

In all the editions, we can use any of the following three tools to develop the code –

- ☐ Force.com Developer Console
- ☐ Force.com IDE
- ☐ Code Editor in the Salesforce User Interface

Conclusion:

Thus, We have created an Application in SalesForce.com using Apex programming Language.

Reference: https://www.tutorialspoint.com/apex/apex_overview.html

Practical No : 04

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

Objectives:

- To learn salesforce cloud administration
- To install and configure the salesforce cloud administrative features

Hardware Requirements :

- Pentium IV with latest configuration

Software Requirements :

- Ubuntu 20.04, Web application i.e. salesforce.com

Theory:

Introduction

Salesforce.com Inc. is an American cloud-based software company headquartered in San Francisco, California. Though the bulk of its revenue comes from a customer relationship management (CRM) product, Salesforce also sells a complementary suite of enterprise applications focused on customer service, marketing automation, analytics and application development.

Salesforce is the primary enterprise offering within the Salesforce platform. It provides companies with an interface for case management and task management, and a system for automatically routing and escalating important events. The Salesforce customer portal provides customers the ability to track their own cases, includes a social networking plug-in that enables the user to join the conversation about their company on social networking websites, provides analytical tools and other services including email alert, Google search, and access to customers' entitlement and contracts.

Lightning Platform

Lightning Platform (also known as Force.com) is a platform as a service (PaaS) that allows developers to create add-on applications that integrate into the main Salesforce.com application. These third-party applications are hosted on Salesforce.com's infrastructure. Force.com applications are built using declarative tools, backed by Lightning and Apex (a proprietary Java-like programming language for Force.com) and Lightning and Visual force (a framework that includes an XML syntax typically used to generate HTML). The Force.com platform typically receives three complete releases a year. As the platform is provided as a service to its developers, every single development instance also receives all these updates.

Community Cloud

Community Cloud provides Salesforce customers the ability to create online web properties for external collaboration, customer service, channel sales, and other custom portals in their instance of Salesforce. Tightly integrated to Sales Cloud, Service Cloud, and App Cloud, Community Cloud can be quickly customized to provide a wide variety of web properties. Salesforce Sales Cloud is a customer relationship management (CRM) platform designed to support sales, marketing and customer support in both business-to-business (B2B) and business-to-customer (B2C) contexts. Sales Cloud is a fully customizable product that brings all the customer information together in an integrated platform that incorporates marketing, lead generation, sales, customer service and business analytics and provides access to thousands of applications through the AppExchange. The platform is provided as Software as a Service (SaaS) for browser-based access; a mobile app is also available. A realtime social feed for collaboration allows users to share information or ask questions of the user community. Salesforce.com offers five versions of Sales Cloud on a per-user, per month basis, from lowest to highest: Group, Professional, Enterprise, Unlimited and Performance. The company offers three levels of support contracts: Standard Success Plan, Premier Success Plan and Premier+ Success Plan.

Create Custom Apps for Salesforce Classic

Create custom apps to give your Salesforce Classic users' access to everything they need all in one place.

If you're new to custom apps, we recommend using Lightning Platform quick start to create an app. With this tool, you can generate a basic working app in just one step.

If you've already created the objects, tabs, and fields you need for your app, follow these steps. With this option, you create an app label and logo, add items to the app, and assign the app to profiles.

1. From Setup, enter Apps in the Quick Find box, then select Apps.
2. Click New.
3. If the Salesforce console is available, select whether you want to define a custom app or a Salesforce console.
4. Give the app a name and description.
An app name can have a maximum of 40 characters, including spaces.
5. Optionally, brand your app by giving it a custom logo.
6. Select which items to include in the app.
7. Optionally, set the default landing tab for your new app using the Default Landing Tab drop-down menu below the list of selected tabs. This determines the first tab a user sees when logging into this app.

8. Choose which profiles the app will be visible to.
9. Check the Default box to set the app as that profile's default app, meaning that new users with the profile see this app the first time they log in. Profiles with limits are excluded from this list.
10. Click Save

What is the difference between custom application and console application in sales force?

A custom application is a collection of tabs, objects etc that function together to solve a particular problem.

A console application uses a specific Salesforce UI - the console. Console applications are intended to enhance productivity by allowing everything to be done from a single, tabbed, screen.

Conclusion:

Thus, We have designed and developed custom application using salesforce cloud.

Practical No : 05

Practical Title: 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.

Objectives:

- To set your own cloud for SaaS over existing LAN
- To implement the basic operations may be like to divide the file in segments/blocks

Hardware Requirements :

- Pentium IV with latest configuration

Software Requirements :

- Ubuntu 20.04, VMware ESXi cloud

Theory:

Here we are installing VMware ESXi cloud

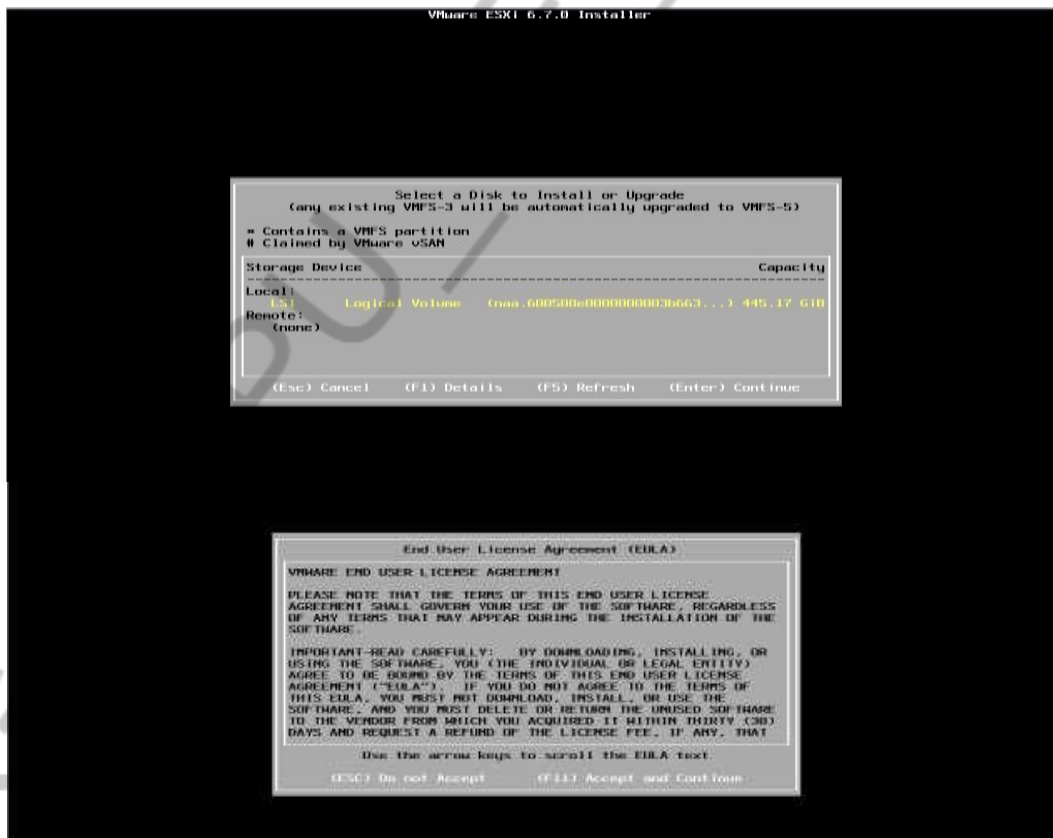
- Host/Node ESXi installation:-
- **ESXi Hardware Requirements:-**
 - ESXi 6.7 requires a host machine with at least two CPU cores.
 - ESXi 6.7 supports 64-bit x86 processors
 - ESXi 6.7 requires the NX/XD bit to be enabled for the CPU in the BIOS.
 - ESXi 6.7 requires a minimum of 4 GB of physical RAM. It is recommended to provide at least 8 GB of RAM to run virtual machines in typical production environments.
 - To support 64-bit virtual machines, support for hardware virtualization (Intel VT-x or AMD RVI) must be enabled on x64 CPUs.
 - One or more Gigabit or faster Ethernet controllers. For a list of supported network adapter models.
 - SCSI disk or local, non-network, RAID LUN with unpartitioned space for the virtual machines.

ForSerialATA(SATA), a disk connected through supported SAS controller or supported on board SATA controllers. SATA disks are considered remote not local. These disks are not used as a scratch partition by default because they are seen as remote.



ESXiInstaller:

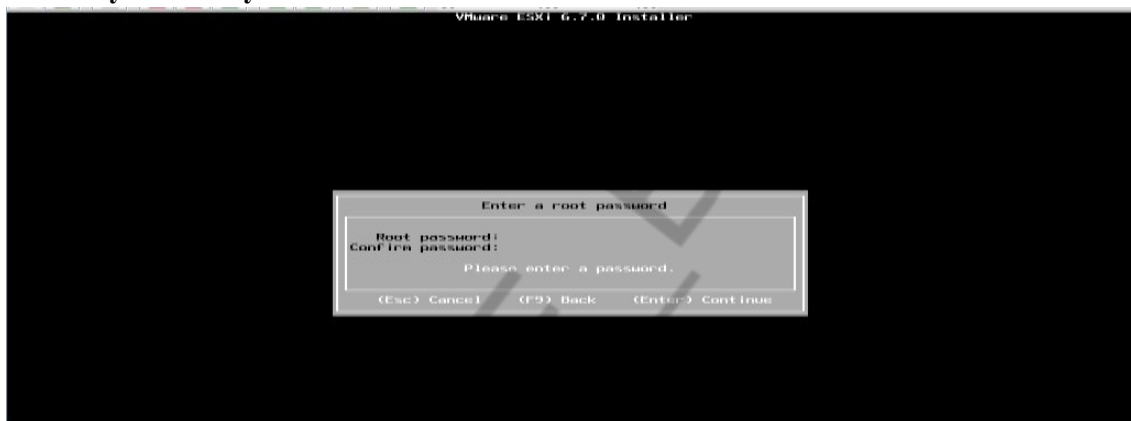
Accept Agreement:



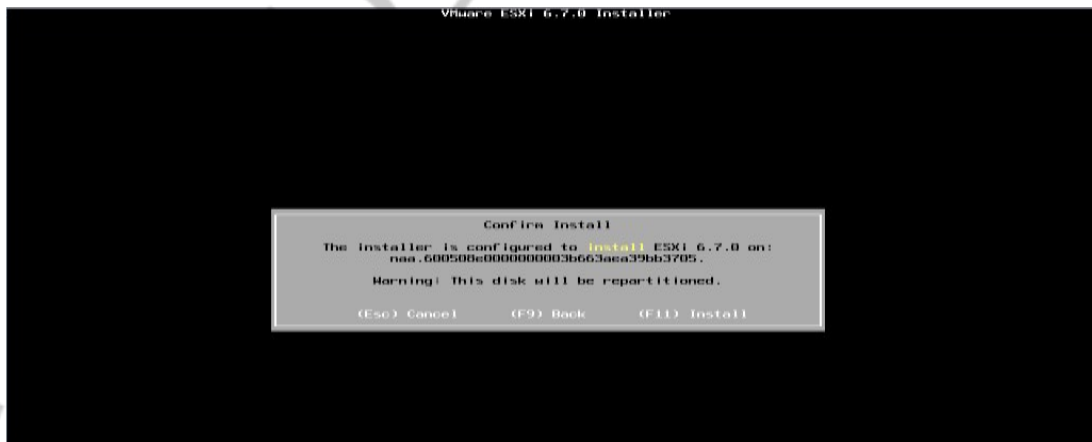
Select storage :



Select Keyboard Layout :



Set NodeESXi Root Password :



Installation complete (Reboot) CLI interface to configuration



CLI Interface to Configuration:



Configure Management Network

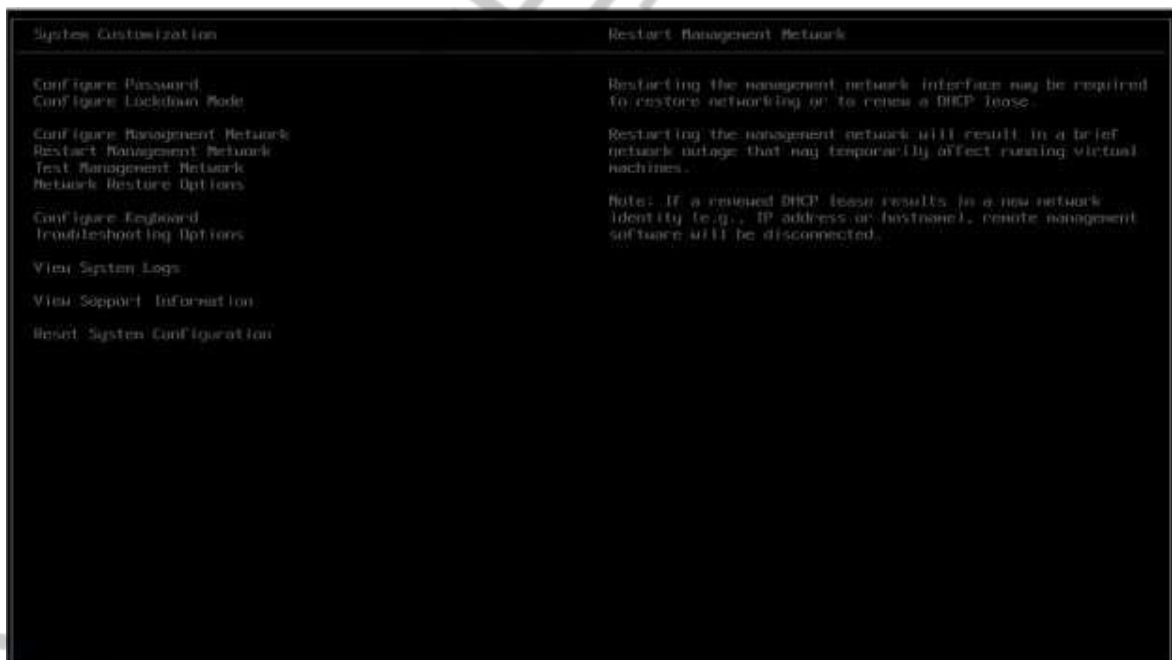


Set IPV4

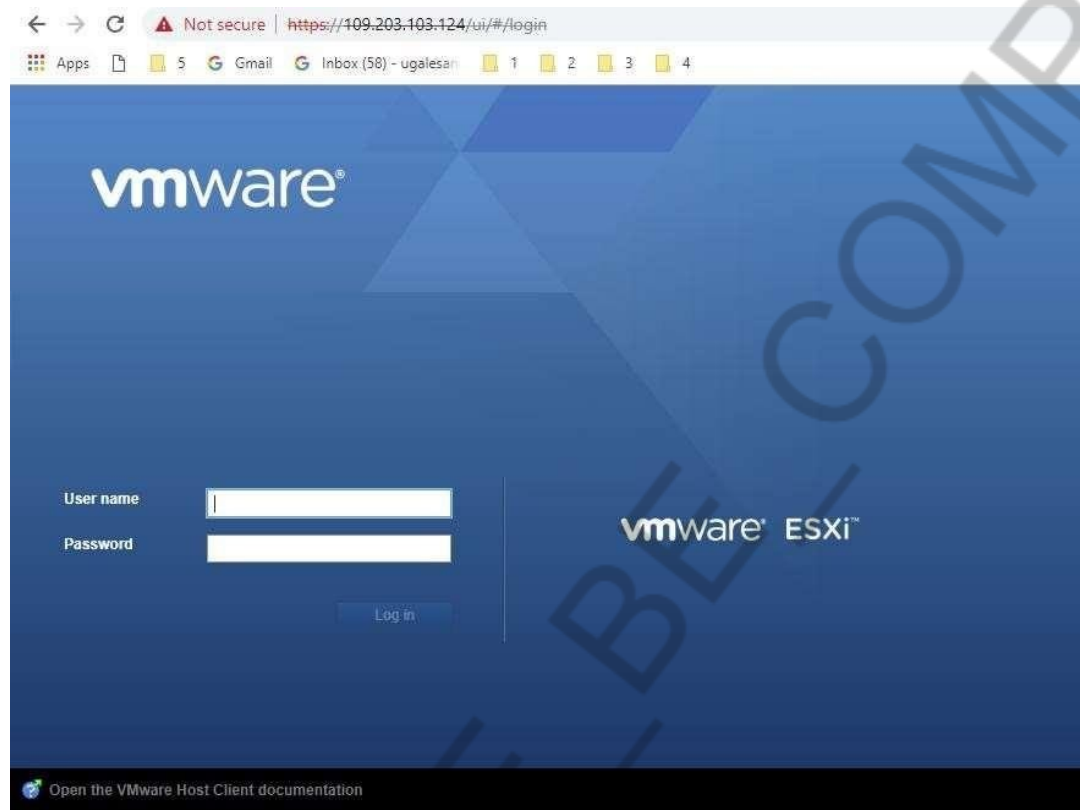


Set DNServer :

Restart Management Network



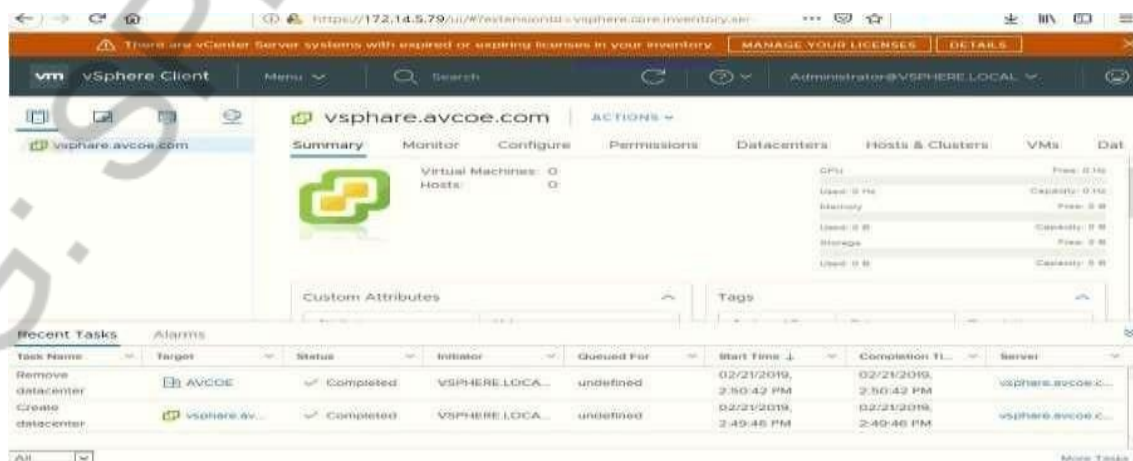
GUIAccess :



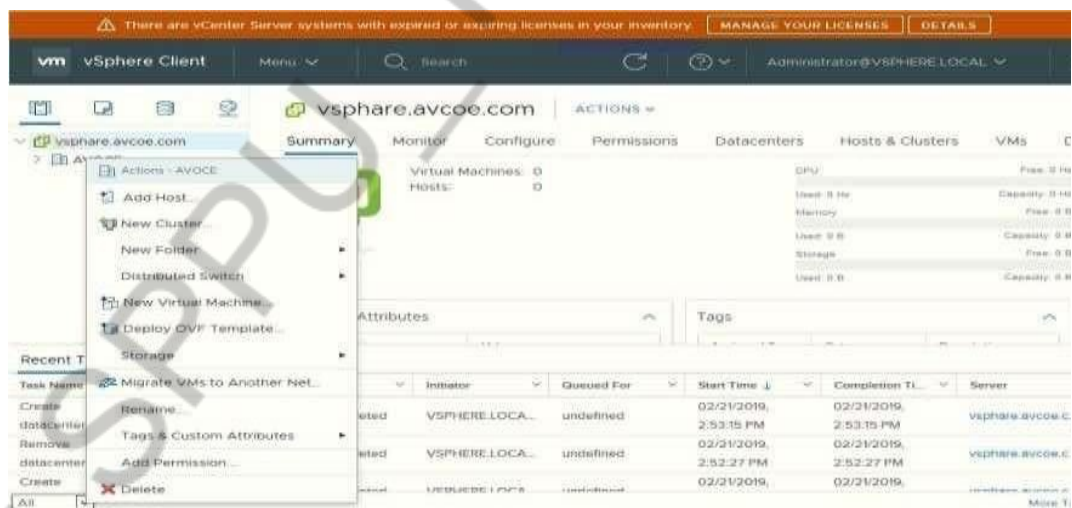
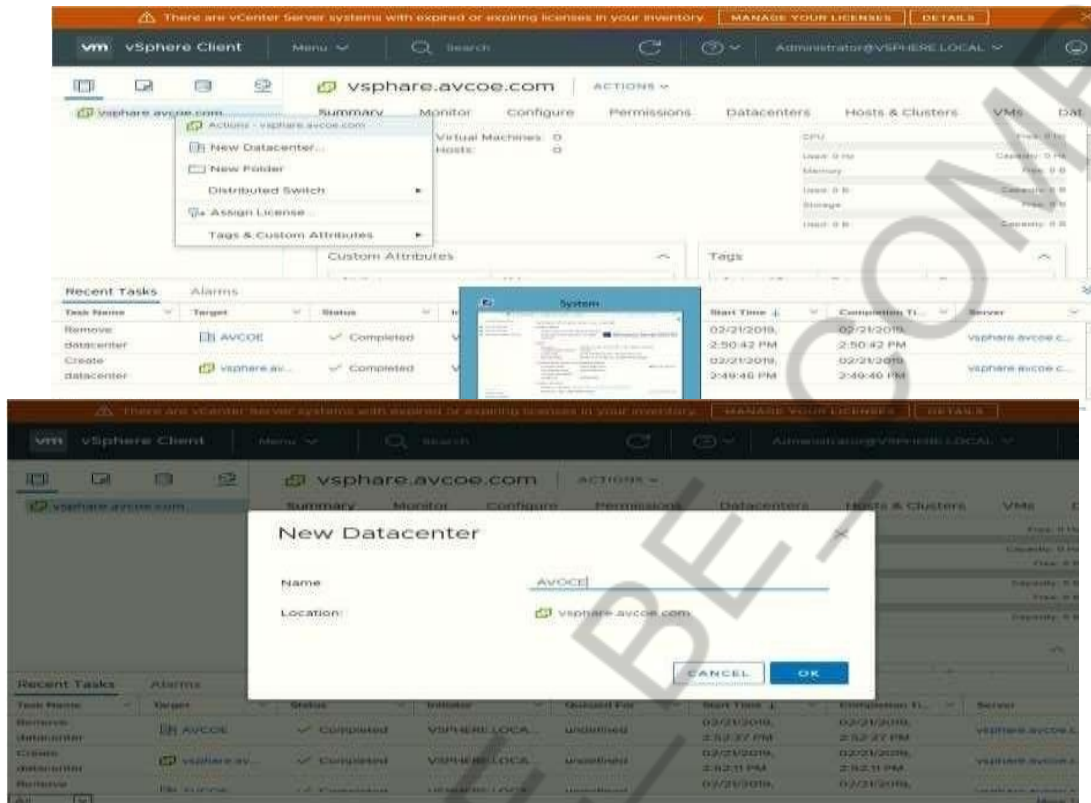
ClusterSetup

- CreatingDatacenter
- CreatingCluster
- Adding Hosts incluster
- Resourcesafteraddingcluster.
- DRS
- Failover

VCenter Access:



Create Datacenter:

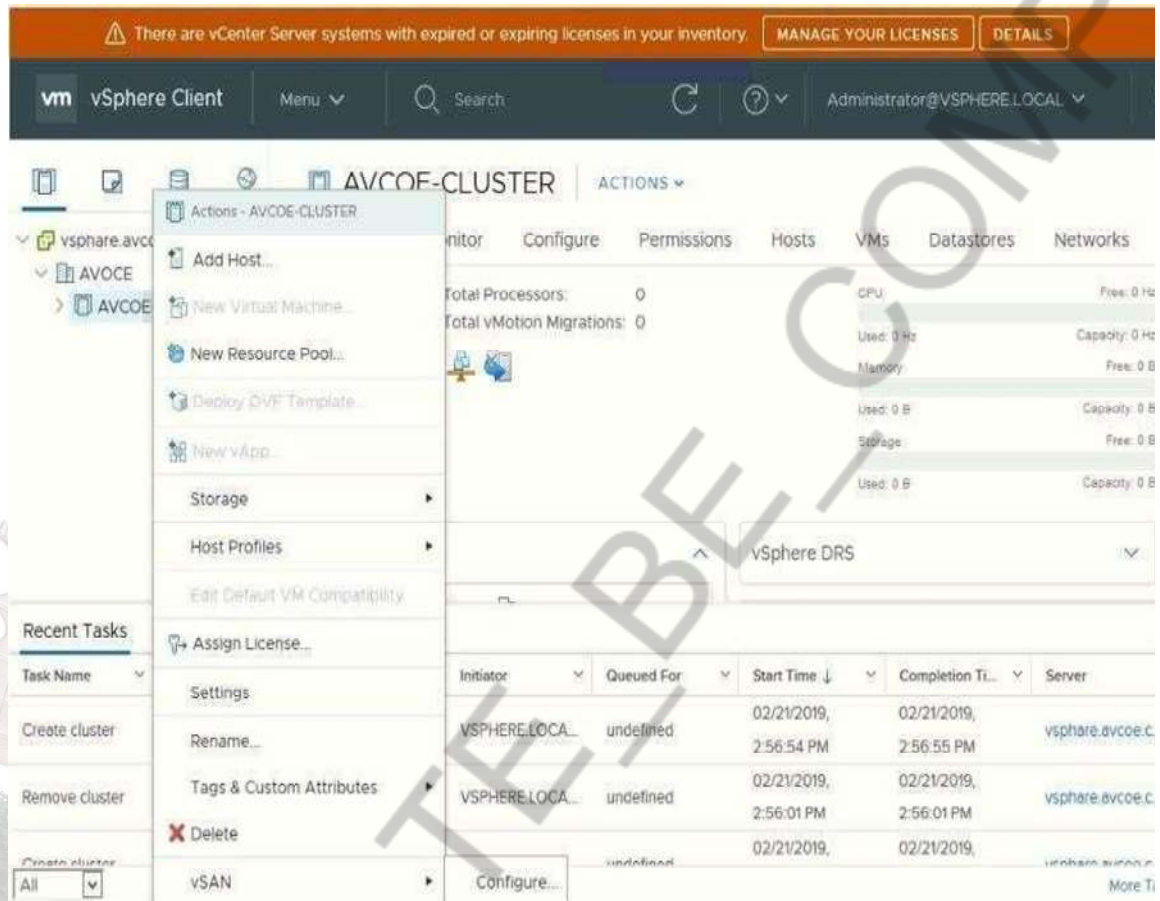


Create cluster :

Assign cluster name :



Add host .:

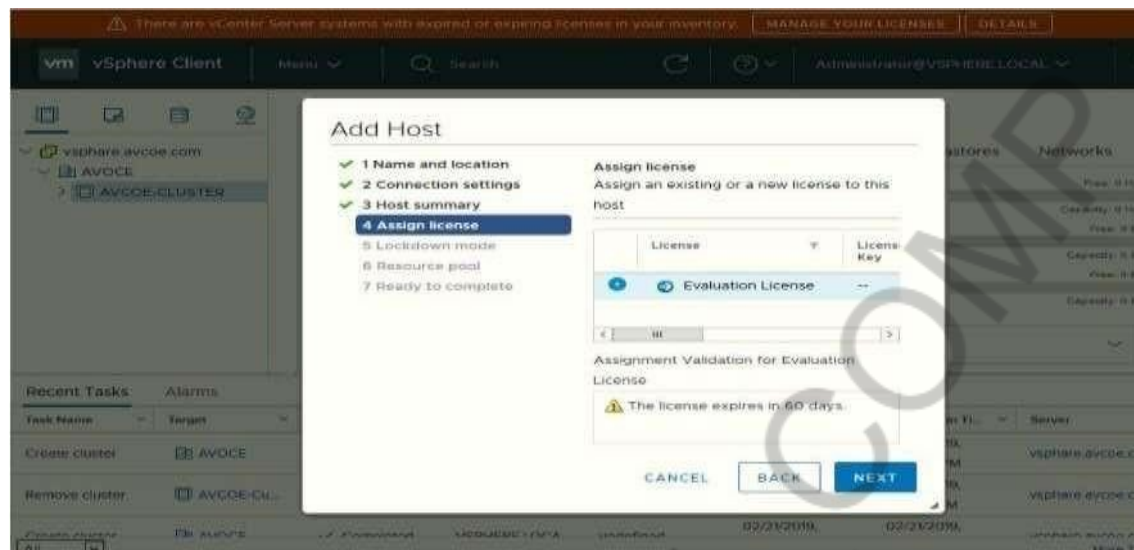


Add host IP :



Enter host credential :

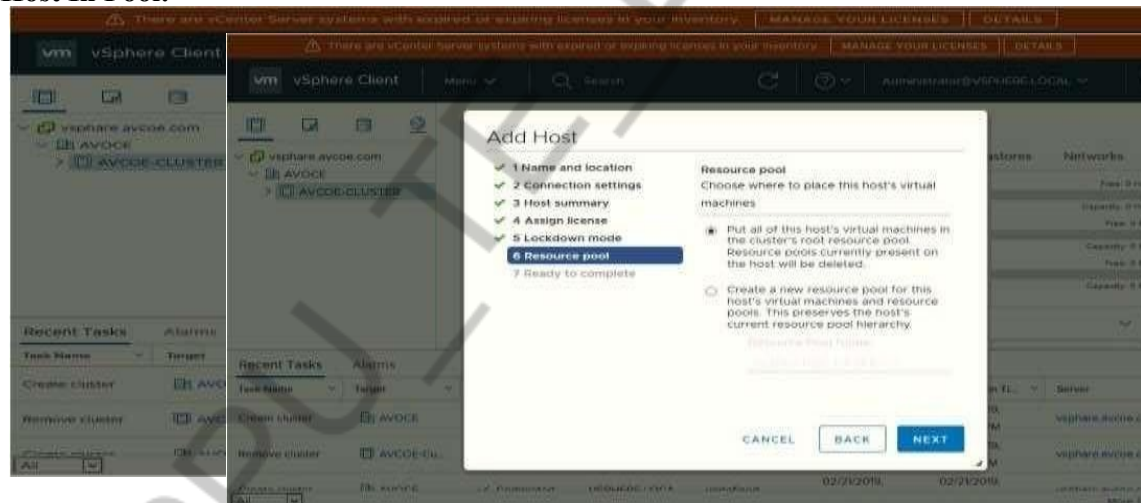




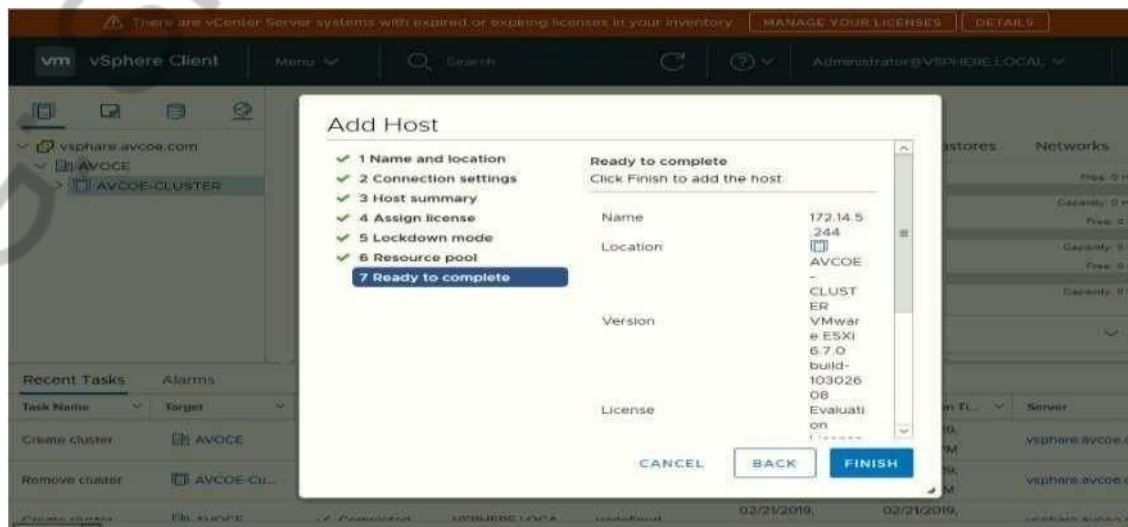
Hot summary :

Lock Down mode:

Add Host In Pool:



Finish:



Host View and View Config:

Cluster View and Configuration:

Task Name	Target	Status	Initiator	Queued For	Start Time	Completion Time	Server
Configuring vSphere HA	172.14.5.245	6%	System	156 ms	02/21/2019, 3:04:54 PM		vsphere.avcoe.c...
Add host	AVCOE-CL...	✓ Completed	VSPHERE.LOCA...	undefined	02/21/2019, 3:04:48 PM	02/21/2019, 3:04:54 PM	vsphere.avcoe.c...
Configuring	172.14.5.245	✓ Completed	System	156 ms	02/21/2019, 3:04:54 PM	02/21/2019, 3:04:54 PM	vsphere.avcoe.c...

Task Name	Target	Status	Initiator	Queued For	Start Time	Completion Time	Server
Configuring vSphere HA	172.14.5.245	✓ Completed	System	156 ms	02/21/2019, 3:04:54 PM	02/21/2019, 3:05:34 PM	vsphere.avcoe.c...
Add host	AVCOE-CL...	✓ Completed	VSPHERE.LOCA...	undefined	02/21/2019, 3:04:48 PM	02/21/2019, 3:04:54 PM	vsphere.avcoe.c...
Configuring	172.14.5.245	✓ Completed	System	156 ms	02/21/2019, 3:04:54 PM	02/21/2019, 3:04:54 PM	vsphere.avcoe.c...

Conclusion: Like this we have configure VSphere Private Cloud