LP-II Lab Manual

Third Year Computer Engineering

310258: Laboratory Practice II

Teaching Scheme	Credit	Examination Scheme
PR: 04 Hours/Week	02	TW : 50 Marks PR : 25 Marks

Guidelines for Instructor's Manual

The instructor‘s manual is to be developed as a hands-on resource and reference. The instructor's manual need to include prologue (about University/program/ institute/ department/foreword/ preface etc), University syllabus, conduction & Assessment guidelines, topics under consideration-concept, objectives, outcomes, set of typical applications/practicals/ guidelines, and references.

Guidelines for Student Journal

The laboratory practical are to be submitted by student in the form of journal. Journal consists of prologue, Certificate, table of contents, and handwritten write-up of each practical (Title, Objectives, Problem Statement, Outcomes, software & Hardware requirements, Date of Completion, Assessment grade/marks and assessor's sign, Theory- Concept in brief, algorithm, flowchart, test cases, conclusion/analysis. Program codes with sample output of all perform practical’s are to be submitted as softcopy.

As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal may be avoided. Use of DVD containing students programs maintained by lab In-charge is highly encouraged. For reference one or two journals may be maintained with program prints at Laboratory.

Guidelines for Assessment

Continuous assessment of laboratory work is done based on overall performance and lab practicals performance of student. Each lab practical assessment will assign grade/marks based on parameters with appropriate weightage. Suggested parameters for overall assessment as well as each lab practical assessment include- timely completion, performance, innovation, efficient codes, punctuality and neatness.

Guidelines for Practical Examination

Both internal and external examiners should jointly set problem statements. During practical assessment, the expert evaluator should give the maximum weightage to the satisfactory implementation of the problem statement. The supplementary and relevant questions may be asked at the time of evaluation to test the student‘s for advanced learning, understanding of the fundamentals, effective and efficient implementation. So encouraging efforts, transparent evaluation and fair approach of the evaluator will not create any uncertainty or doubt in the minds of the students. So adhering to these principles will consummate our team efforts to the promising start of the student's academics.

Guidelines for Laboratory Conduction

The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The assignment framing policy need to address the average students and inclusive of an element to attract and promote the intelligent students. Use of open source software is encouraged. Based on the concepts learned. Instructor may also set one assignment or mini-project that is suitable to respective branch beyond the scope of syllabus.

Operating System recommended :- 64-bit Windows OS and Linux

Programming tools recommended: - Information Security : - C/C++/Java

Augmented and Virtual Reality :- Unity, C#, Blender, VRTK, ARTK, Vuforia VR Devices: HTC Vive, Google Daydream and Samsung gear VR.

Cloud Computing :- NA

Software Modeling and Architectures: Front end:HTML5, Bootstrap, jQuery, JS etc. Backend: MySQL/MongoDB/NodeJS

Cloud Computing

	Cloud Computing **(All assignments are compulsory)**
1.	Case study on Microsoft azure to learn about Microsoft Azure is a cloud computing platform and infrastructure, created by Microsoft, for building, deploying and managing applications and services through a global network of Microsoft-managed data centers. OR Case study on Amazon EC2 and learn about Amazon EC2 web services.
2.	Installation and configure Google App Engine. OR 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.

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 itas 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.

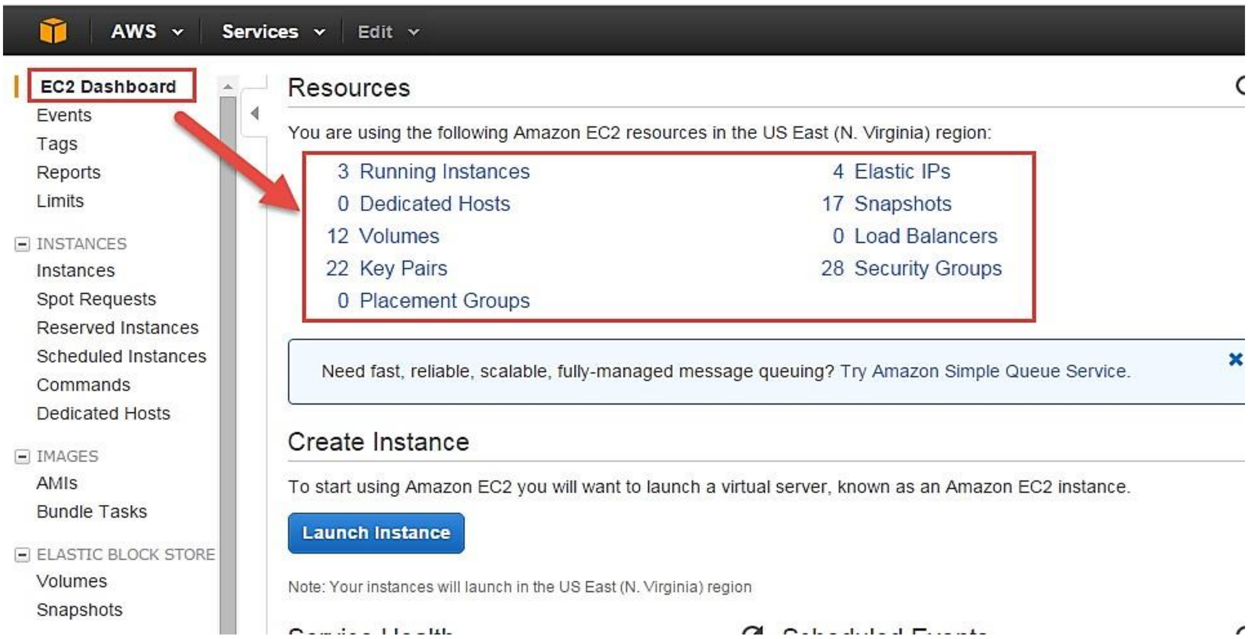
Let us see in detail how to launch an on-demand EC2 instance in AWS Cloud.Login and access to AWS services

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.



Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) region:

3	Running Instances	4	Elastic IPs
0	Dedicated Hosts	17	Snapshots
12	Volumes	0	Load Balancers
22	Key Pairs	28	Security Groups
0	Placement Groups		

Need fast, reliable, scalable, fully-managed message queuing? Try Amazon Simple Queue Service.

Create Instance

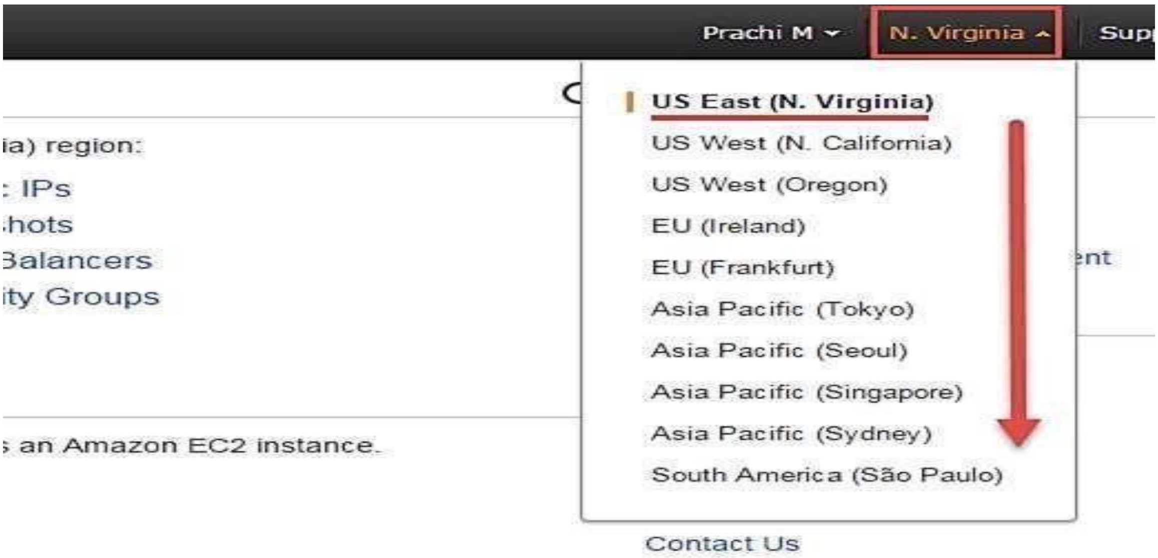
To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Note: Your instances will launch in the US East (N. Virginia) region

Step 2) On the top right corner of the EC2 dashboard, choose the AWS Region in which youwant 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).



Prachi M ▾ N. Virginia ▴ Sup

ia) region:

: IPs

hots

Balancers

ity Groups

; an Amazon EC2 instance.

US East (N. Virginia)

US West (N. California)

US West (Oregon)

EU (Ireland)

EU (Frankfurt)

Asia Pacific (Tokyo)

Asia Pacific (Seoul)

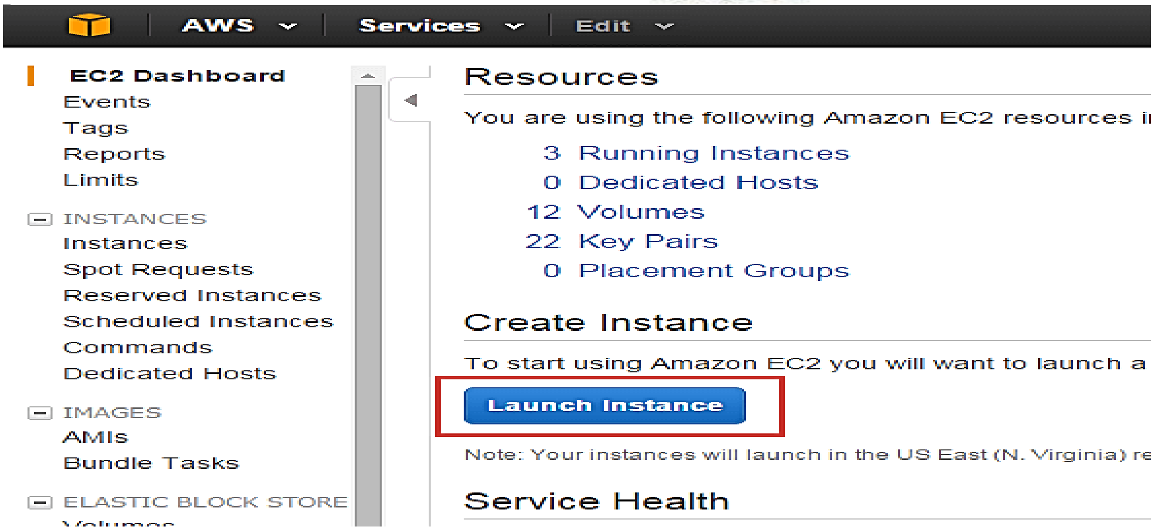
Asia Pacific (Singapore)

Asia Pacific (Sydney)

South America (São Paulo)

ent

Contact Us



AWS ▾ Services ▾ Edit ▾

EC2 Dashboard

Events

Tags

Reports

Limits

⊟ INSTANCES

Instances

Spot Requests

Reserved Instances

Scheduled Instances

Commands

Dedicated Hosts

⊟ IMAGES

AMIs

Bundle Tasks

⊟ ELASTIC BLOCK STORE

Resources

You are using the following Amazon EC2 resources i

3 Running Instances

0 Dedicated Hosts

12 Volumes

22 Key Pairs

0 Placement Groups

Create Instance

To start using Amazon EC2 you will want to launch a

Launch Instance

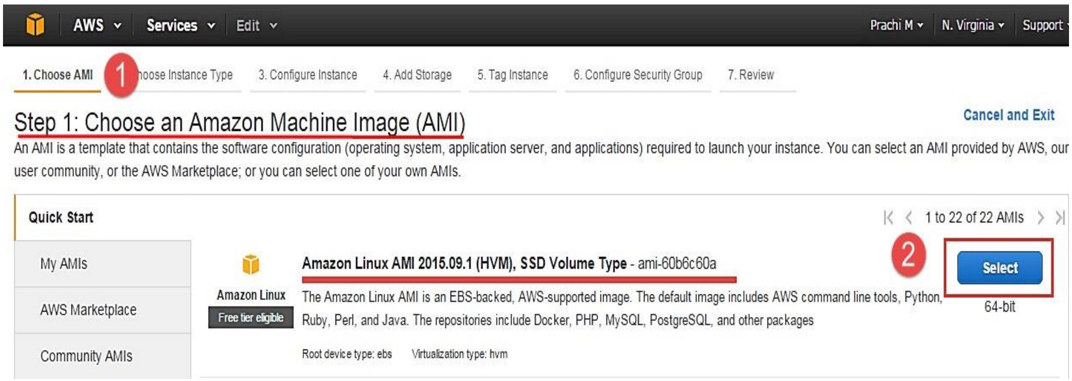
Note: Your instances will launch in the US East (N. Virginia) re

Service Health

- Instance creation wizard page will open as soon as you click 'LaunchInstance'.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.



AWS ▾ Services ▾ Edit ▾ Prachi M ▾ N. Virginia ▾ Support

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

Step 1: Choose an Amazon Machine Image (AMI)

Cancel and Exit

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Quick Start

My AMIs

AWS Marketplace

Community AMIs

1 to 22 of 22 AMIs

Amazon Linux

Free tier eligible

Amazon Linux AMI 2015.09.1 (HVM), SSD Volume Type - ami-60b6c60a

The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages

Root device type: ebs Virtualization type: hvm

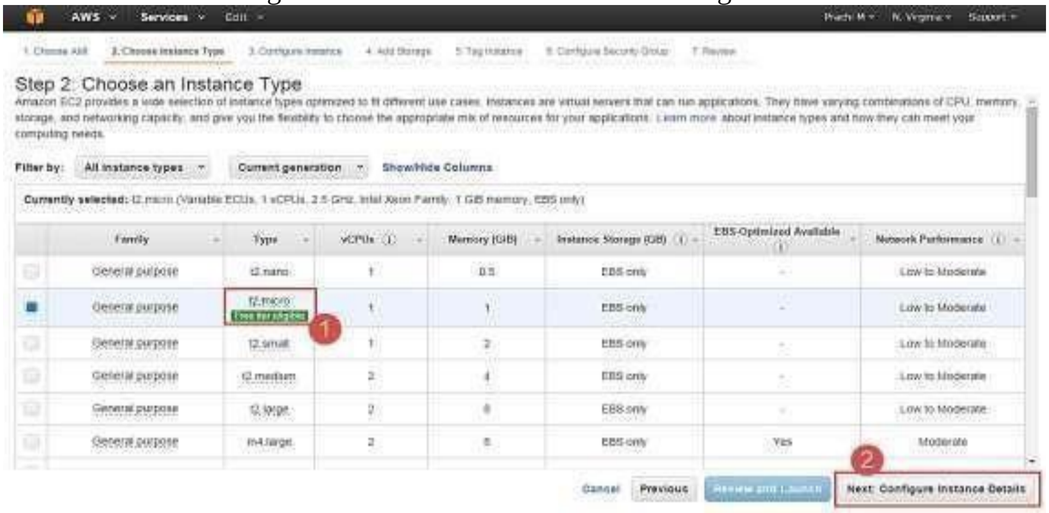
2 Select

64-bit

Choose EC2 Instance Types

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

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



Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Filter by: All instance types · Current generation · Show/Hide Columns

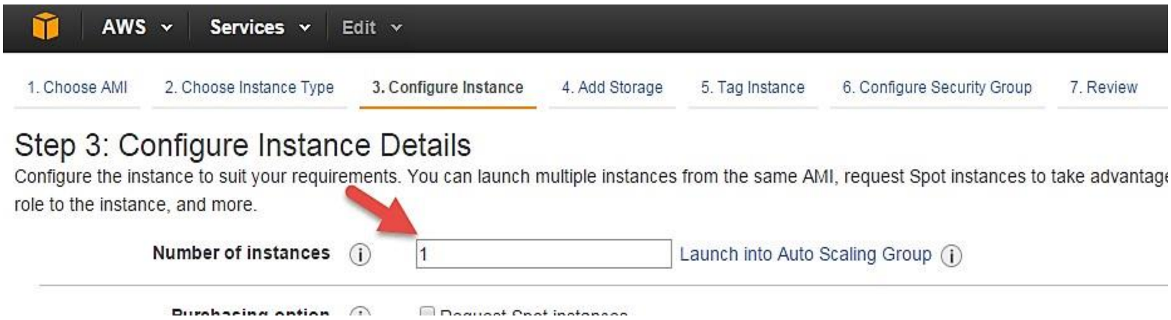
Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
□	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate
■	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate
□	General purpose	t2.small	1	2	EBS only	-	Low to Moderate
□	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
□	General purpose	t2.large	2	8	EBS only	-	Low to Moderate
□	General purpose	m4.large	2	8	EBS only	Yes	Moderate

- 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 launchingone instance.



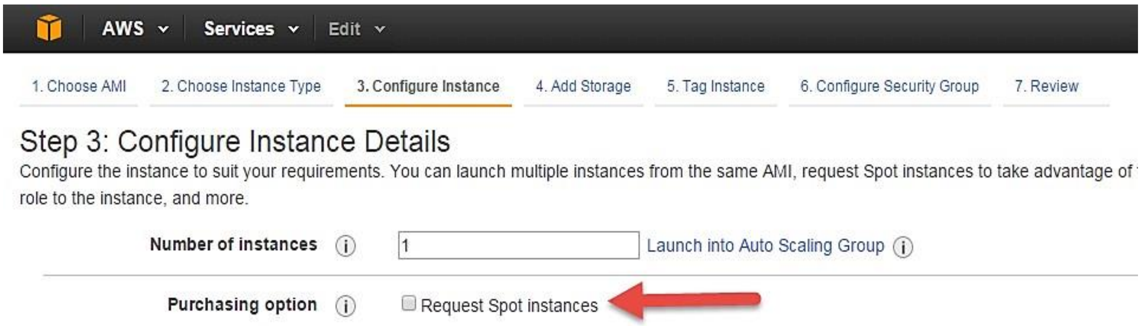
Step 3: Configure Instance Details

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

Number of instances: 1 · Launch into Auto Scaling Group

Purchasing option: □ Request Spot instances

Step 2) Under Purchasing Options, keep the option of 'Request Spot Instances' unchecked as ofnow. (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: 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 role to the instance, and more.

Number of instances: 1 · Launch into Auto Scaling Group

Purchasing option: □ Request Spot instances

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

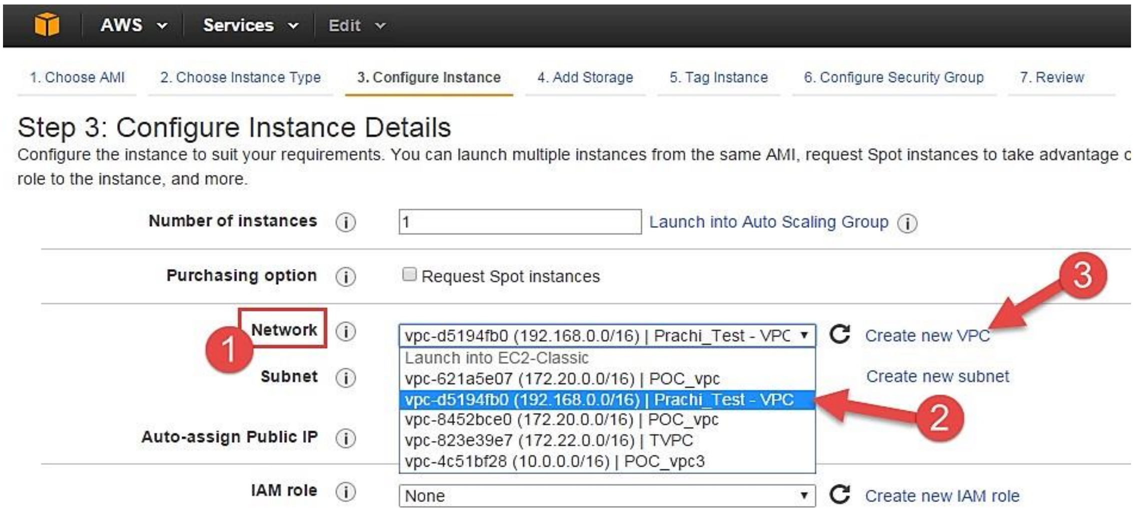
- You have to decide here, in which VPC (Virtual Private Cloud) you want to launch your instance and under which subnets inside your VPC. It is better to determine and plan this prior to launching the instance. Your AWS architecture set-up should include IP ranges for your subnets etc. pre-planned for 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 yourVPC.

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 c 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

Launch into EC2-Classic
vpc-621a5e07 (172.20.0.0/16) | POC_vpc
vpc-d5194fb0 (192.168.0.0/16) | Prachi_Test - VPC
vpc-8452bce0 (172.20.0.0/16) | POC_vpc
vpc-823e39e7 (172.22.0.0/16) | TVPC
vpc-4c51bf28 (10.0.0.0/16) | POC_vpc3

Subnet — Create new subnet

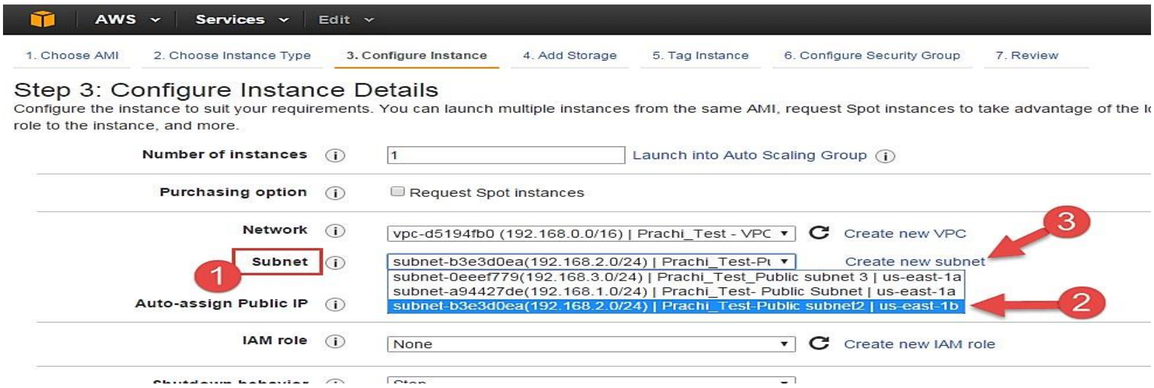
Auto-assign Public IP

IAM role: None — 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 l 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-Pu — Create new subnet

subnet-0eeef779(192.168.3.0/24) | Prachi_Test_Public subnet 3 | us-east-1a
subnet-a94427de(192.168.1.0/24) | Prachi_Test- Public Subnet | us-east-1a
subnet-b3e3d0ea(192.168.2.0/24) | Prachi_Test-Public subnet2 | us-east-1b

Auto-assign Public IP

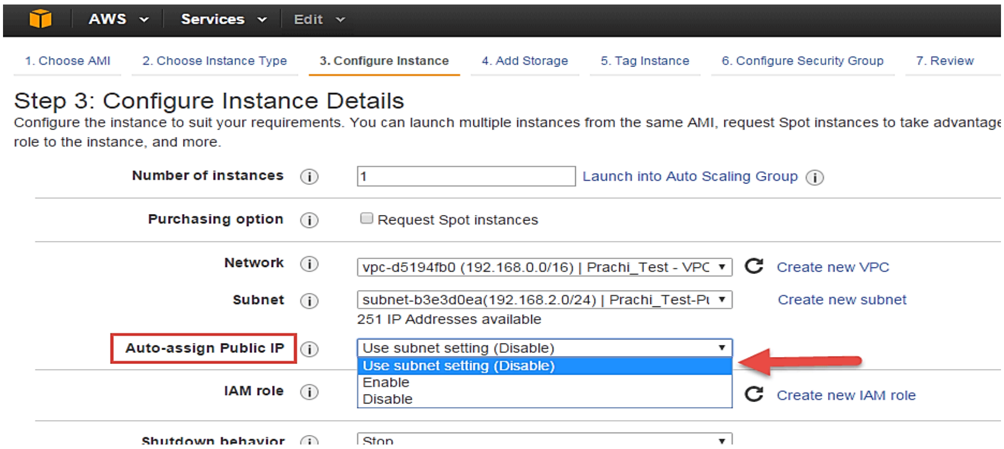
IAM role: None — Create new IAM role

- Once your instance is launched in a public subnet, AWS will assign a dynamic public IPto 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 itmanually 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.



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 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-Pu — Create new subnet

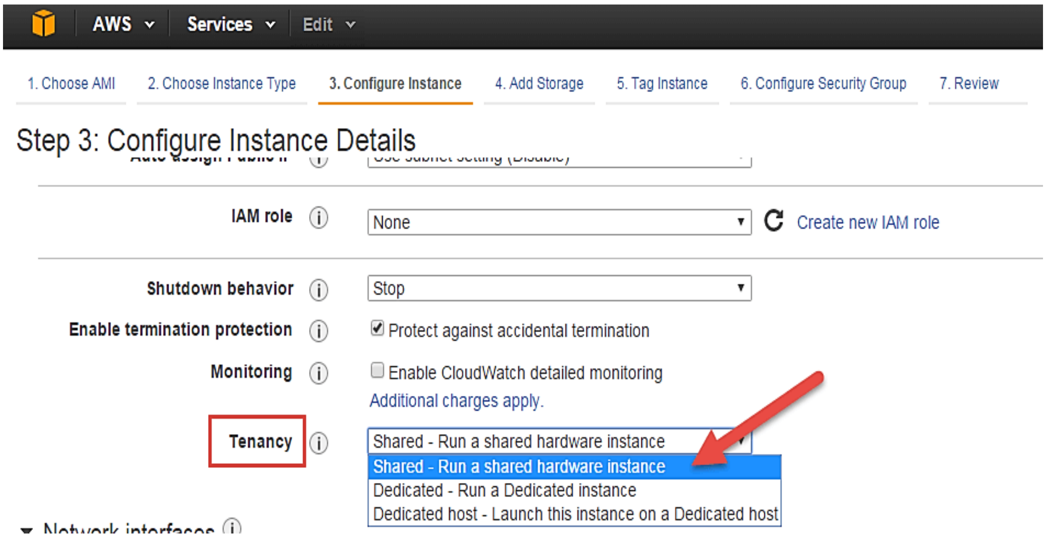
251 IP Addresses available

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

- Use subnet setting (Disable)
- Enable
- Disable

IAM role — Create new IAM role

Shutdown behavior: Stop



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

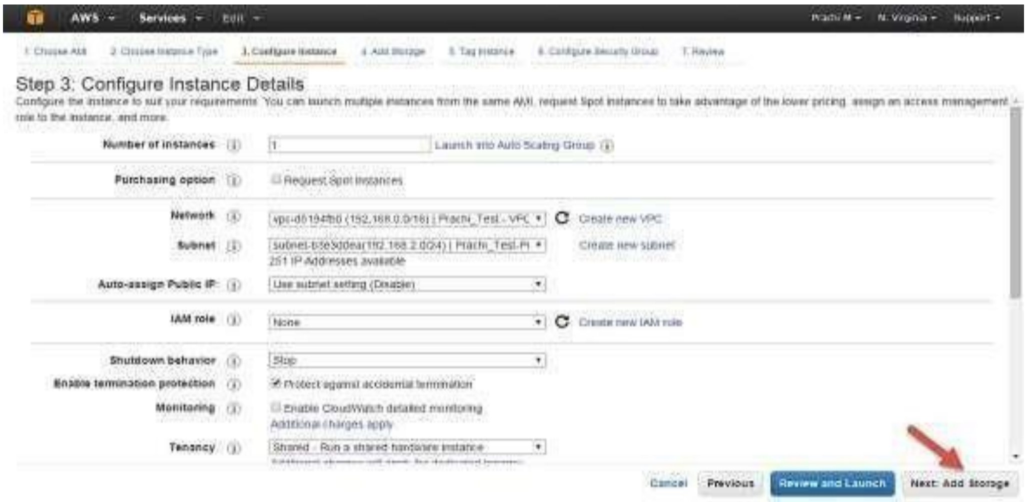
Monitoring: □ Enable CloudWatch detailed monitoring

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



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

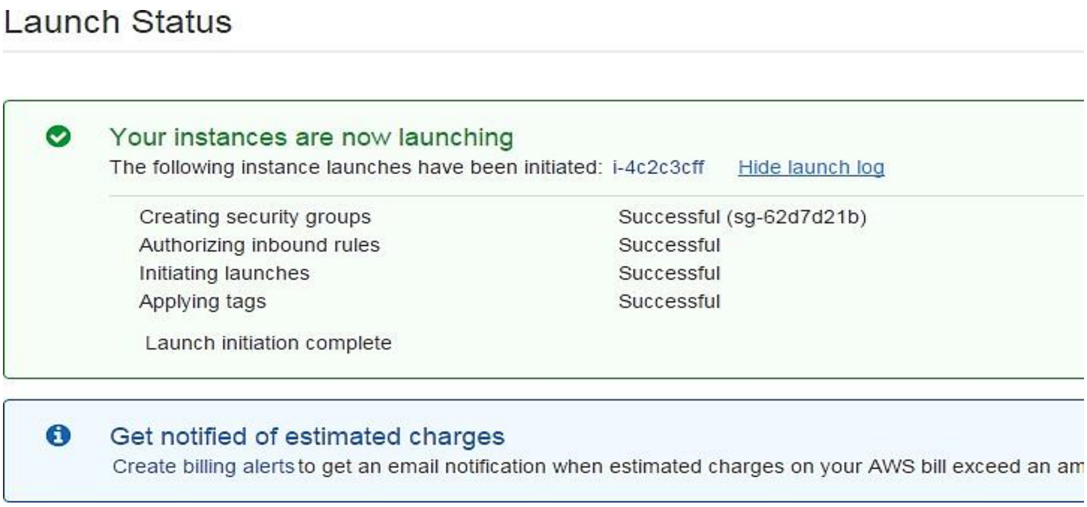
Shutdown behavior: Stop

Enable termination protection: ☑ Protect against accidental termination

Monitoring: □ Enable CloudWatch detailed monitoring

Additional charges apply

Tenancy: Shared - Run a shared hardware instance



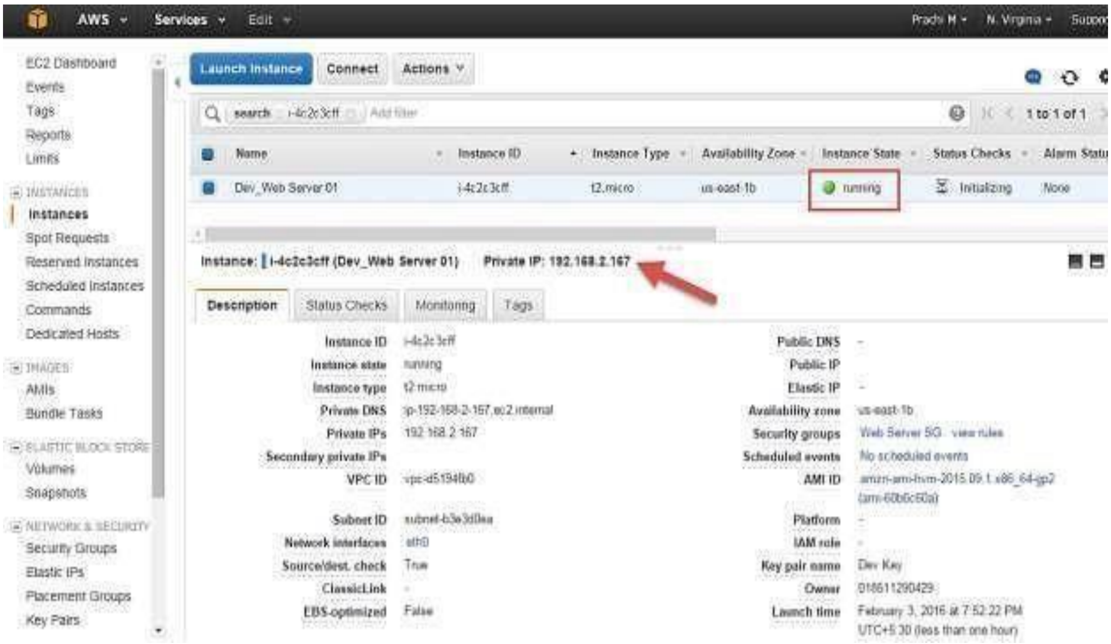
Launch Status

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



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Statu
Dev_Web Server 01	i-4c2c3cff	t2.micro	us-east-1b	running	Initializing	None

Instance: i-4c2c3cff (Dev_Web Server 01) Private IP: 192.168.2.167

Instance ID	i-4c2c3cff	Public DNS	-
Instance state	running	Public IP	
Instance type	t2.micro	Elastic IP	-
Private DNS	ip-192-168-2-167.ec2.internal	Availability zone	us-east-1b
Private IPs	192.168.2.167	Security groups	Web Server SG . view rules
Secondary private IPs		Scheduled events	No scheduled events
VPC ID	vpc-d5194fb0	AMI ID	amzn-ami-hvm-2015.09.1.x86_64-gp2 (ami-60b6c60a)
Subnet ID	subnet-b3e3d0ea	Platform	-
Network interfaces	eth0	IAM role	-
Source/dest. check	True	Key pair name	Dev Key
ClassicLink	-	Owner	018611290429
EBS-optimized	False	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 :

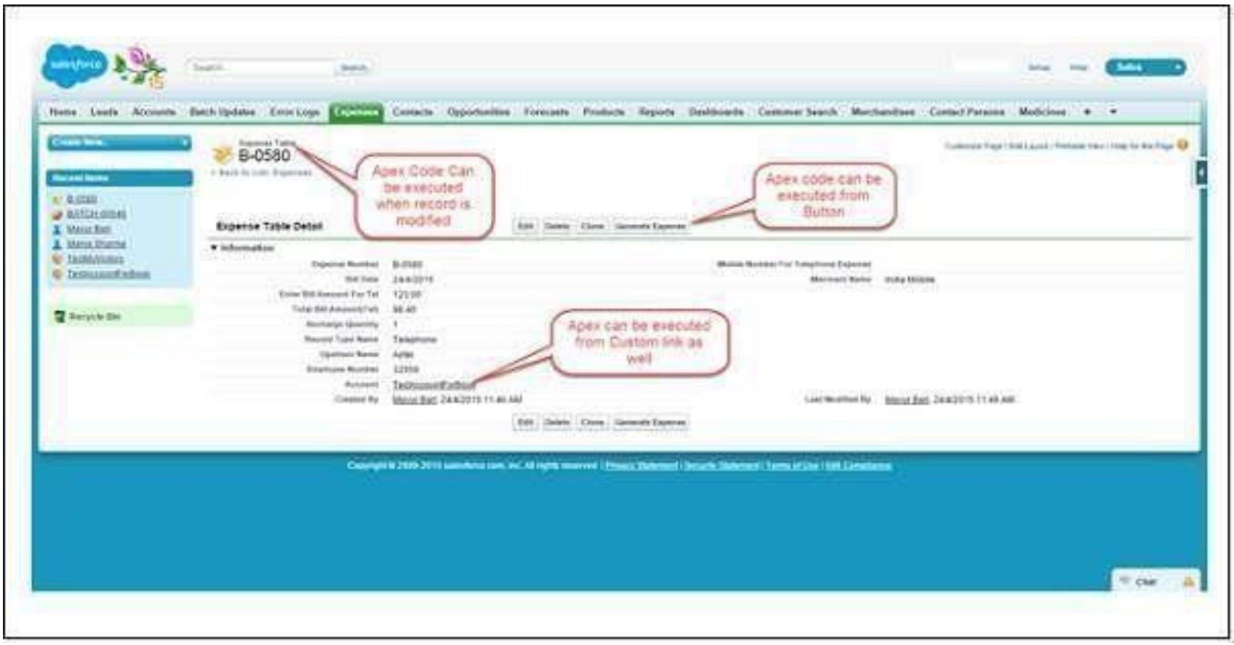
- Ubuntu 20.04, Web application i.e. salesforce.com

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.



Apex Code Can be executed when record is modified

Apex code can be executed from Button

Apex can be executed from Custom link as well

.

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.

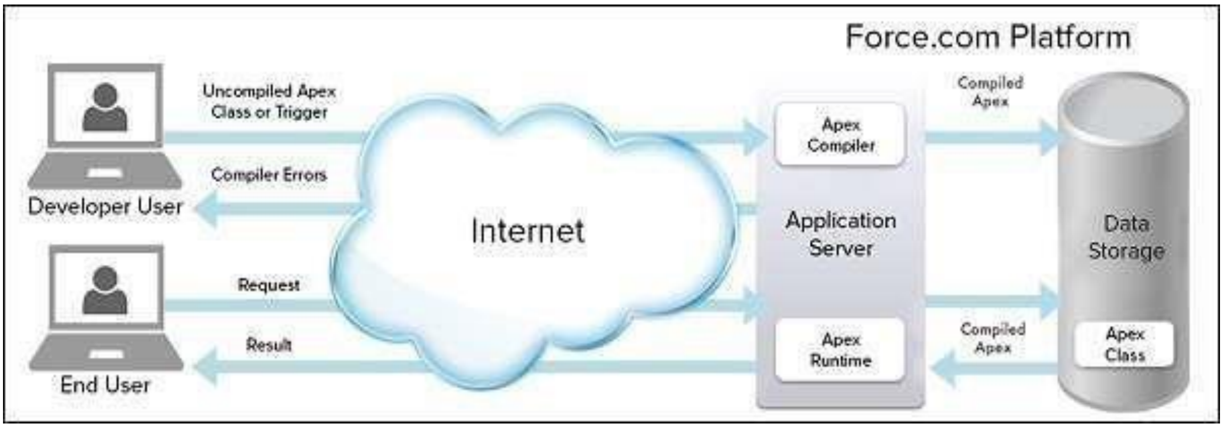
 Create complex business processes that are not supported by existing workflow functionality or flows.

 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 I 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 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, VMwareESXi cloud

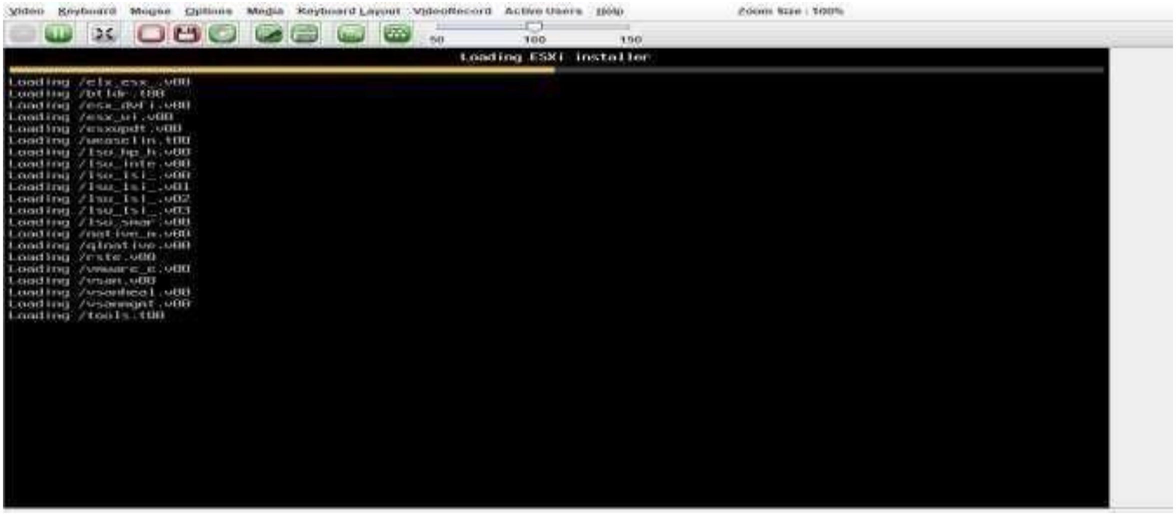
Theory:

Here we are installing VMwareESXi cloud

- Host/NodeESXi installation:-
- **ESXiHardwareRequirements:-**

- ESXi6.7requiresahostmachinewithatleasttwoCPUcores.
- ESXi6.7supports64-bitx86processors
- ESXi6.7requirestheNX/XDbit to be enabled for the CPU in the BIOS.
- ESXi6.7requiresaminimumof4GBofphysicalRAM.Itisrecommended to provide atleast 8 GB of RAM to run virtual machines in typical productionenvironments.
- Tosupport64-bitvirtualmachines,support for hardware virtualization (IntelVT-xor AMDRVI) mustbeenabledonx64CPUs.
- One or more Gigabit or faster Ethernet controllers. For a list of supportednetwork adapter models.
- SCSI disk oralocal,non-network,RAIDLUN with unpartitioned space for the virtualmachines.

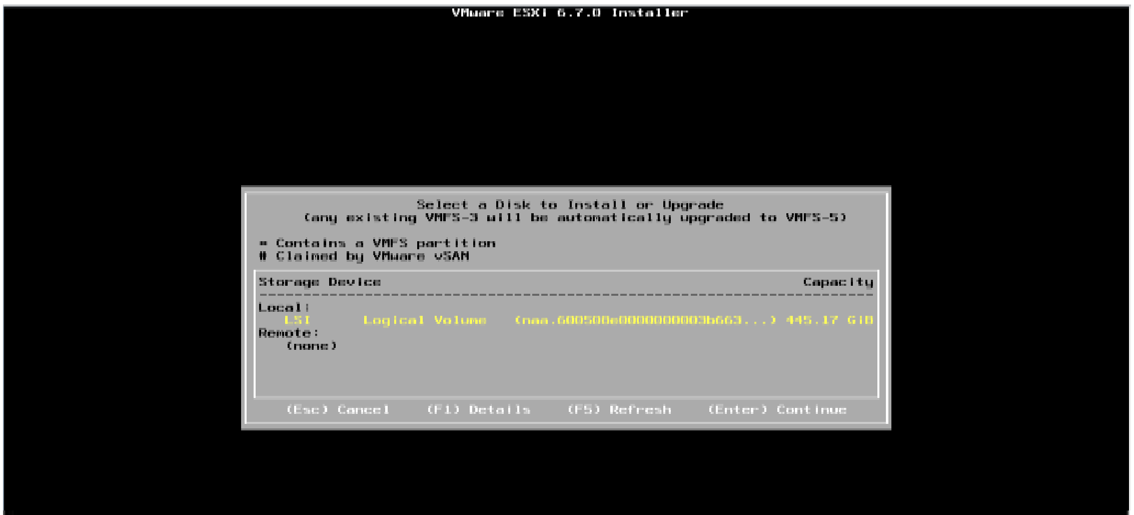
ForSerialATA(SATA), a disk connected through supported SAS controller or supported on board SATA controllers. SATA disks are considered remote not local. These disks are not used as a scratch partition by default be cause they are seen as remote.



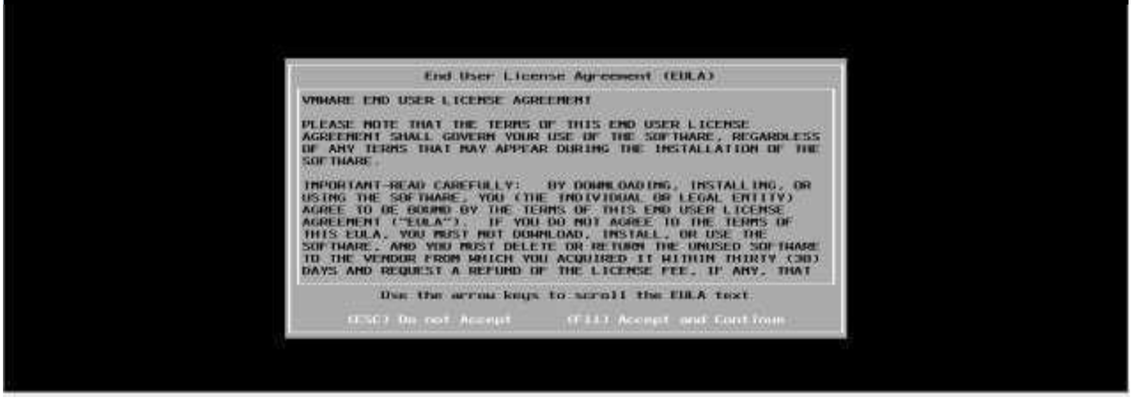
```
                         Loading ESXi installer
Loading /elx_esx_.v00
Loading /btldr.t00
Loading /esx_dvfi.v00
Loading /esx_ui.v00
Loading /esxupdt.v00
Loading /weaselin.t00
Loading /lsu_hp_h.v00
Loading /lsu_inte.v00
Loading /lsu_lsi_.v00
Loading /lsu_lsi_.v01
Loading /lsu_lsi_.v02
Loading /lsu_lsi_.v03
Loading /lsu_smar.v00
Loading /native_m.v00
Loading /qlnative.v00
Loading /rste.v00
Loading /vmware_e.v00
Loading /vsan.v00
Loading /vsanheal.v00
Loading /vsanmgmt.v00
Loading /tools.t00
```

ESXiInstaller:

Accept Agreement:

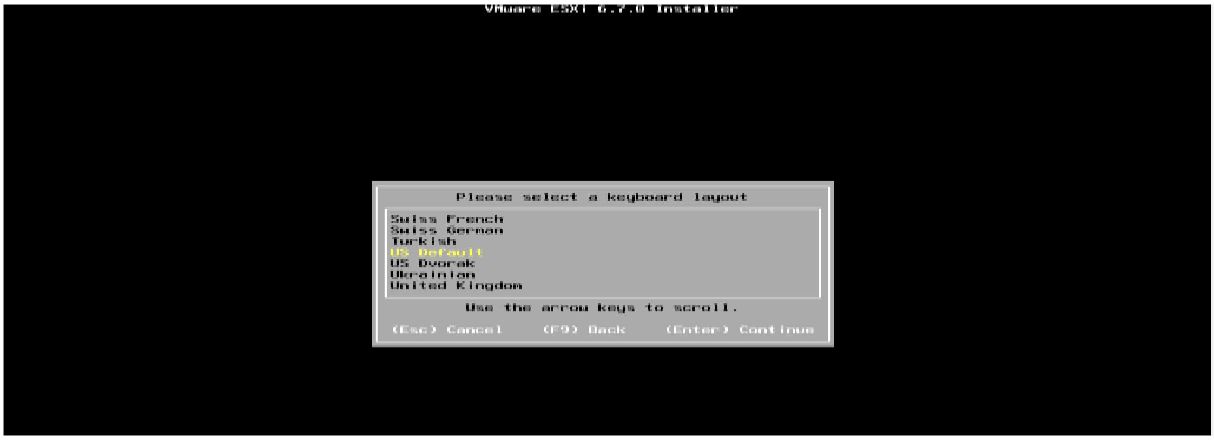


```
                       VMware ESXi 6.7.0 Installer

                    Select a Disk to Install or Upgrade
        (any existing VMFS-3 will be automatically upgraded to VMFS-5)

  * Contains a VMFS partition
  # Claimed by VMware vSAN

  Storage Device                                                    Capacity
  --------------------------------------------------------------------------
  Local:
     LSI      Logical Volume   (naa.600508e0000000003b663...) 445.17 GiB
  Remote:
     (none)

     (Esc) Cancel     (F1) Details     (F5) Refresh     (Enter) Continue
```



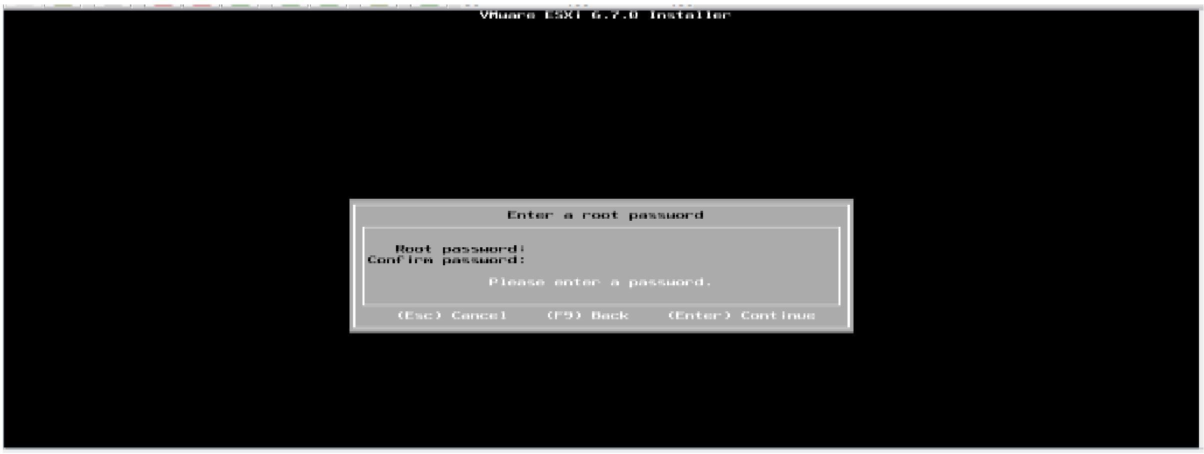
```
                 End User License Agreement (EULA)

VMWARE END USER LICENSE AGREEMENT

PLEASE NOTE THAT THE TERMS OF THIS END USER LICENSE
AGREEMENT SHALL GOVERN YOUR USE OF THE SOFTWARE, REGARDLESS
OF ANY TERMS THAT MAY APPEAR DURING THE INSTALLATION OF THE
SOFTWARE.

IMPORTANT-READ CAREFULLY:   BY DOWNLOADING, INSTALLING, OR
USING THE SOFTWARE, YOU (THE INDIVIDUAL OR LEGAL ENTITY)
AGREE TO BE BOUND BY THE TERMS OF THIS END USER LICENSE
AGREEMENT ("EULA").  IF YOU DO NOT AGREE TO THE TERMS OF
THIS EULA, YOU MUST NOT DOWNLOAD, INSTALL, OR USE THE
SOFTWARE, AND YOU MUST DELETE OR RETURN THE UNUSED SOFTWARE
TO THE VENDOR FROM WHICH YOU ACQUIRED IT WITHIN THIRTY (30)
DAYS AND REQUEST A REFUND OF THE LICENSE FEE, IF ANY, THAT

            Use the arrow keys to scroll the EULA text

       (ESC) Do not Accept        (F11) Accept and Continue
```

Select storage :



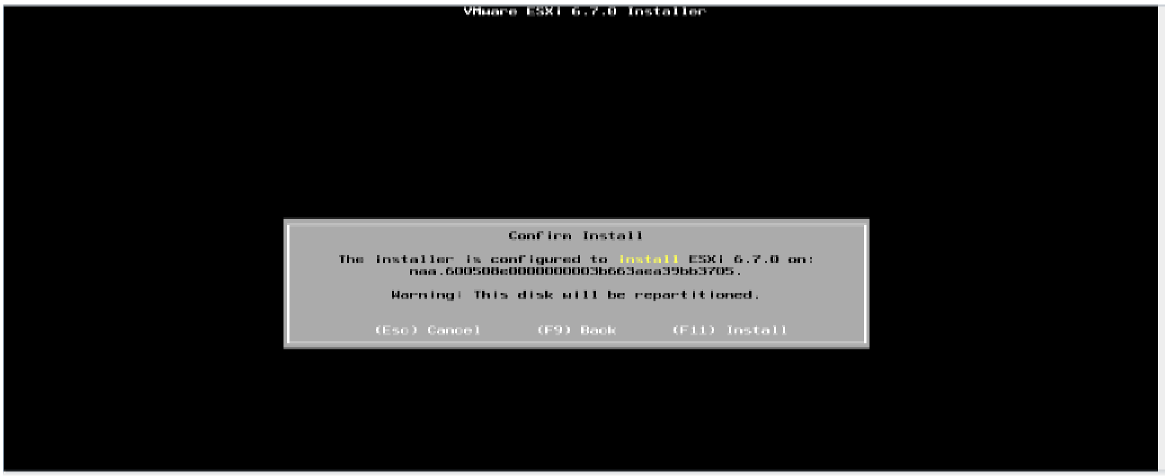
```
VMware ESXi 6.7.0 Installer

Please select a keyboard layout

Swiss French
Swiss German
Turkish
US Default
US Dvorak
Ukrainian
United Kingdom

Use the arrow keys to scroll.

(Esc) Cancel      (F9) Back      (Enter) Continue
```

Select Keyboard Layout :



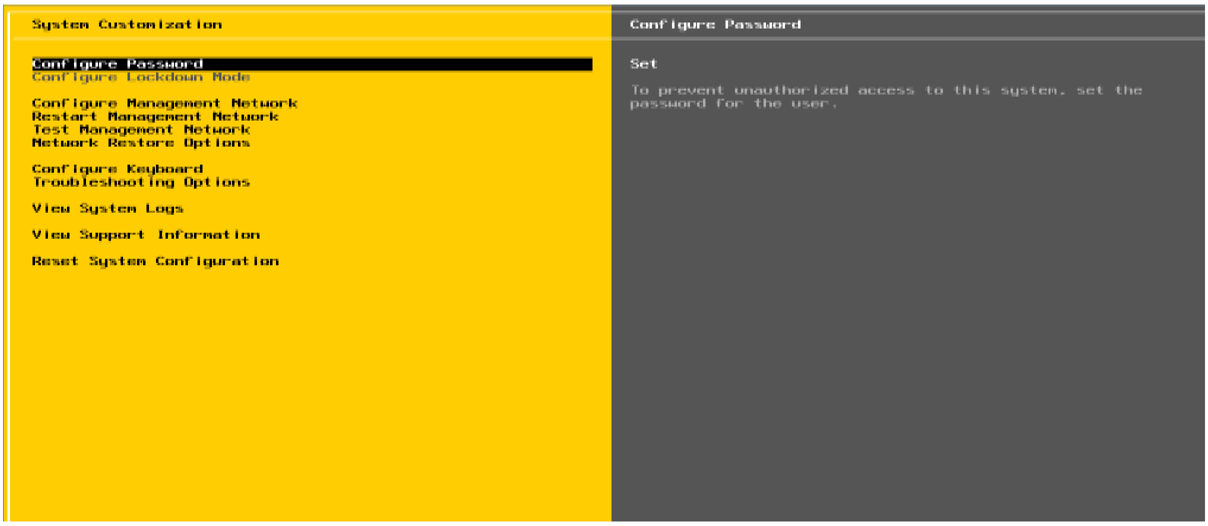
```
VMware ESXi 6.7.0 Installer

Enter a root password

    Root password:
 Confirm password:

          Please enter a password.

(Esc) Cancel      (F9) Back      (Enter) Continue
```

Set NodeESXi Root Password :



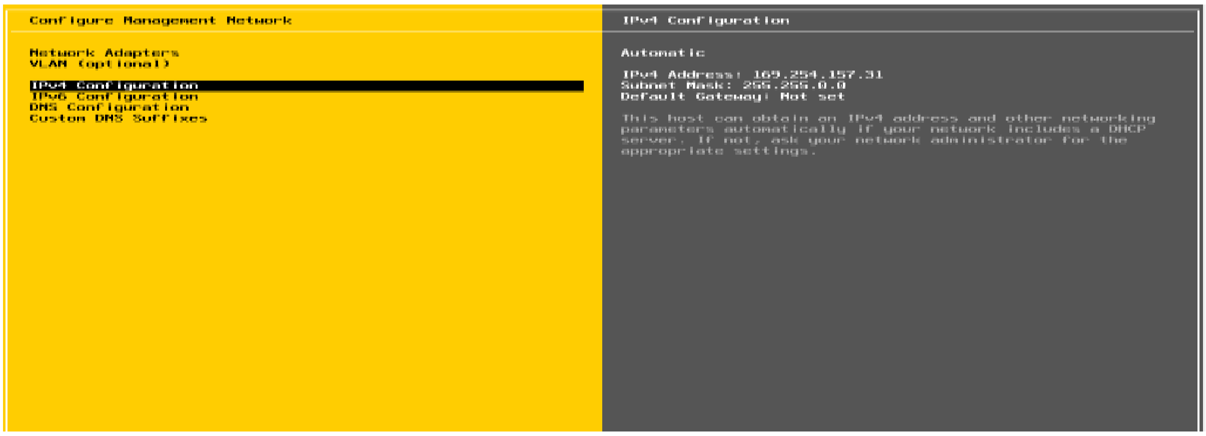
```
VMware ESXi 6.7.0 Installer

Confirm Install

The installer is configured to install ESXi 6.7.0 on:
naa.600508e00000000003b663aea39bb3705.

Warning! This disk will be repartitioned.

(Esc) Cancel      (F9) Back      (F11) Install
```

Installation complete (Reboot)CLII interface to configuration



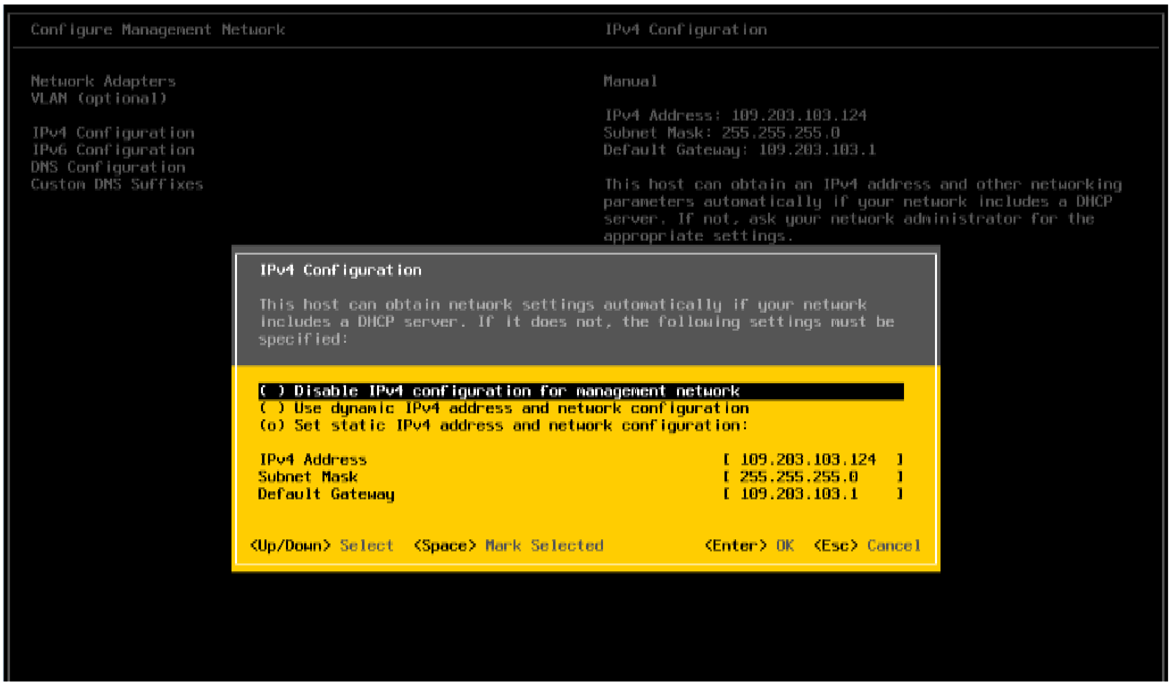
```
System Customization                              Configure Password

Configure Password                                Set
Configure Lockdown Mode
                                                  To prevent unauthorized access to this system, set the
Configure Management Network                      password for the user.
Restart Management Network
Test Management Network
Network Restore Options

Configure Keyboard
Troubleshooting Options

View System Logs

View Support Information

Reset System Configuration
```

CLI Interface to Configuration:



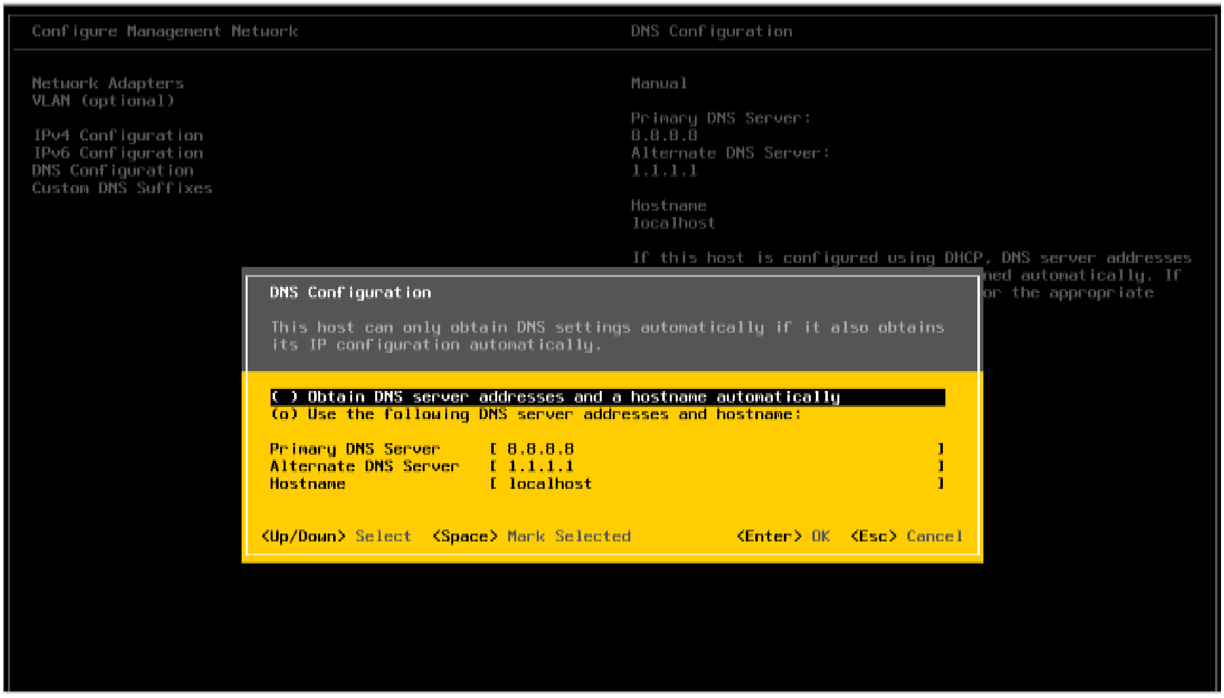
```
Configure Management Network                      IPv4 Configuration

Network Adapters                                  Automatic
VLAN (optional)
                                                  IPv4 Address: 169.254.157.31
IPv4 Configuration                                Subnet Mask: 255.255.0.0
IPv6 Configuration                                Default Gateway: Not set
DNS Configuration
Custom DNS Suffixes                               This host can obtain an IPv4 address and other networking
                                                  parameters automatically if your network includes a DHCP
                                                  server. If not, ask your network administrator for the
                                                  appropriate settings.
```

Configure Management Network



```
Configure Management Network                      IPv4 Configuration

Network Adapters                                  Manual
VLAN (optional)
                                                  IPv4 Address: 109.203.103.124
IPv4 Configuration                                Subnet Mask: 255.255.255.0
IPv6 Configuration                                Default Gateway: 109.203.103.1
DNS Configuration
Custom DNS Suffixes                               This host can obtain an IPv4 address and other networking
                                                  parameters automatically if your network includes a DHCP
                                                  server. If not, ask your network administrator for the
                                                  appropriate settings.

IPv4 Configuration

This host can obtain network settings automatically if your network
includes a DHCP server. If it does not, the following settings must be
specified:

( ) Disable IPv4 configuration for management network
( ) Use dynamic IPv4 address and network configuration
(o) Set static IPv4 address and network configuration:

IPv4 Address                                   [ 109.203.103.124  ]
Subnet Mask                                    [ 255.255.255.0    ]
Default Gateway                                [ 109.203.103.1    ]

<Up/Down> Select   <Space> Mark Selected          <Enter> OK   <Esc> Cancel
```

Set IPV4

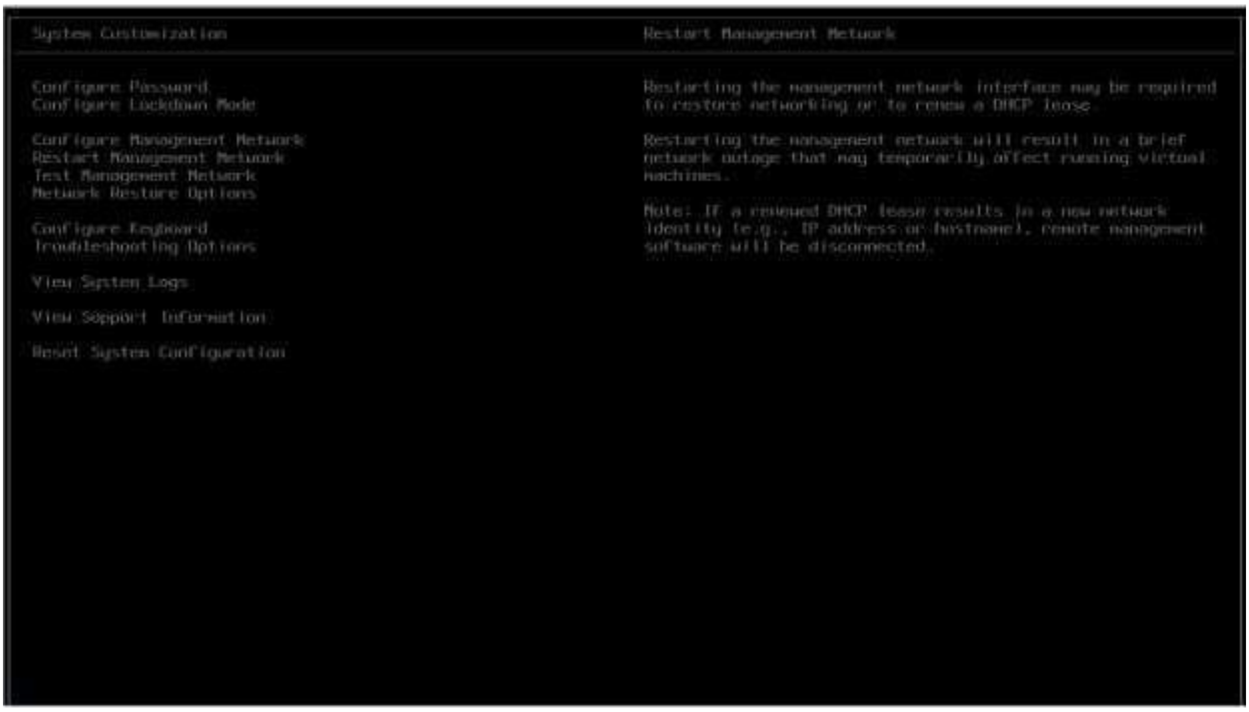


```
Configure Management Network                          DNS Configuration

Network Adapters                                      Manual
VLAN (optional)
                                                      Primary DNS Server:
IPv4 Configuration                                    8.8.8.8
IPv6 Configuration                                    Alternate DNS Server:
DNS Configuration                                     1.1.1.1
Custom DNS Suffixes
                                                      Hostname
                                                      localhost

                                                      If this host is configured using DHCP, DNS server addresses
                                                                                ned automatically. If
                                                                                or the appropriate
```

```
DNS Configuration

This host can only obtain DNS settings automatically if it also obtains
its IP configuration automatically.

( ) Obtain DNS server addresses and a hostname automatically
(o) Use the following DNS server addresses and hostname:

Primary DNS Server      [ 8.8.8.8                                   ]
Alternate DNS Server    [ 1.1.1.1                                   ]
Hostname                [ localhost                                 ]

<Up/Down> Select   <Space> Mark Selected             <Enter> OK   <Esc> Cancel
```

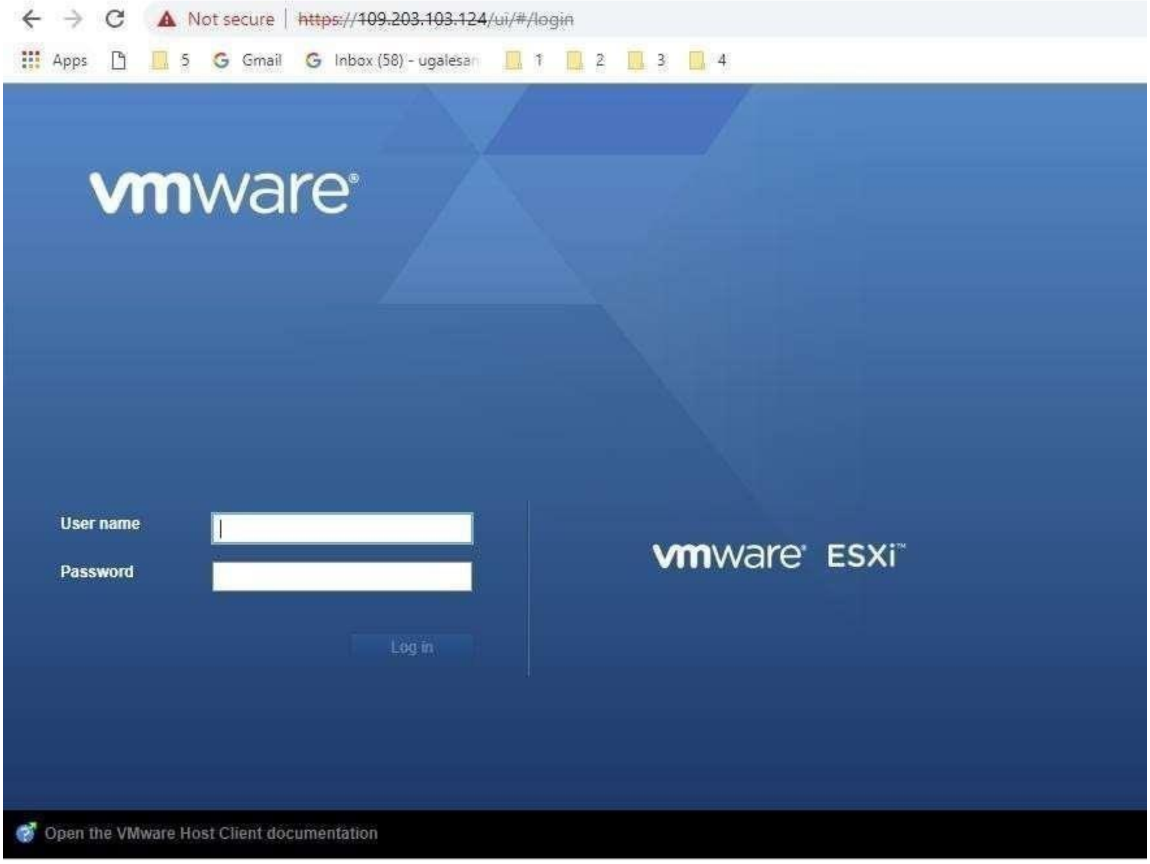
Set DNSeriver :

Restart Management Network



```
System Customization                                  Restart Management Network

Configure Password                                    Restarting the management network interface may be required
Configure Lockdown Mode                               to restore networking or to renew a DHCP lease.

Configure Management Network                          Restarting the management network will result in a brief
Restart Management Network                            network outage that may temporarily affect running virtual
Test Management Network                               machines.
Network Restore Options
                                                      Note: If a renewed DHCP lease results in a new network
Configure Keyboard                                    identity (e.g., IP address or hostname), remote management
Troubleshooting Options                               software will be disconnected.

View System Logs

View Support Information

Reset System Configuration
```

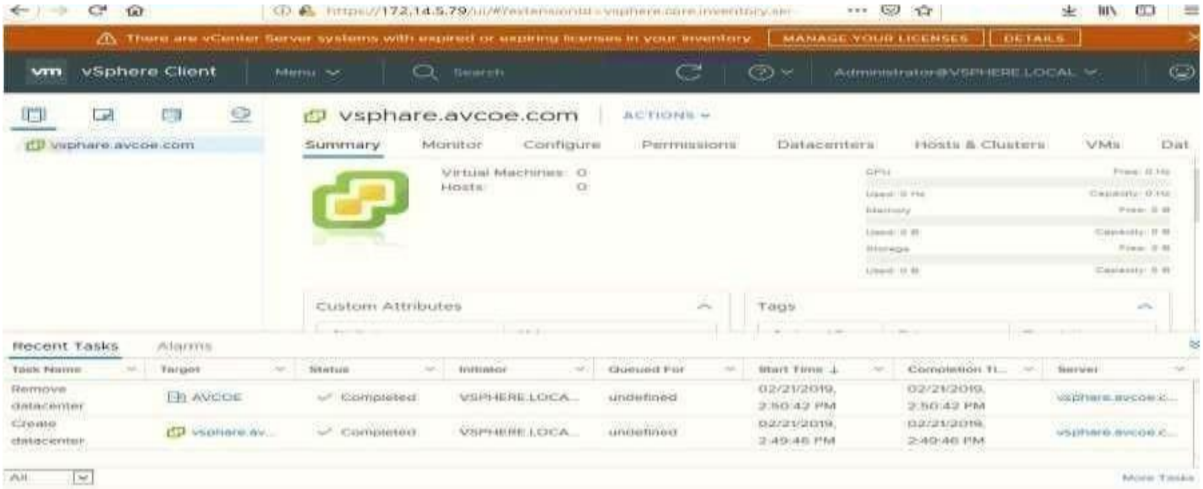
GUIAccess :



ClusterSetup

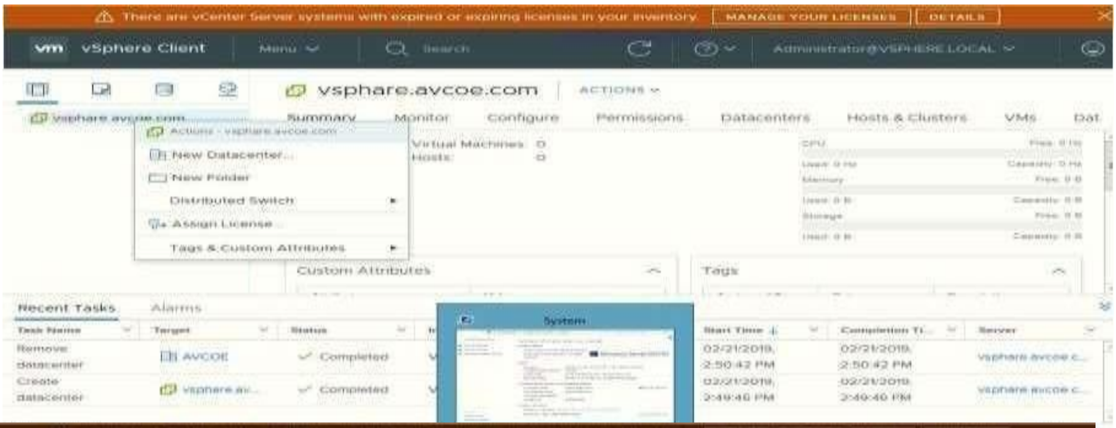
- **CreatingDatacenter**
- **CreatingCluster**
- **Adding Hosts incluster**
- **Resourcesafteraddingcluster.**
- **DRS**
- **Failover**

VCenter Access:

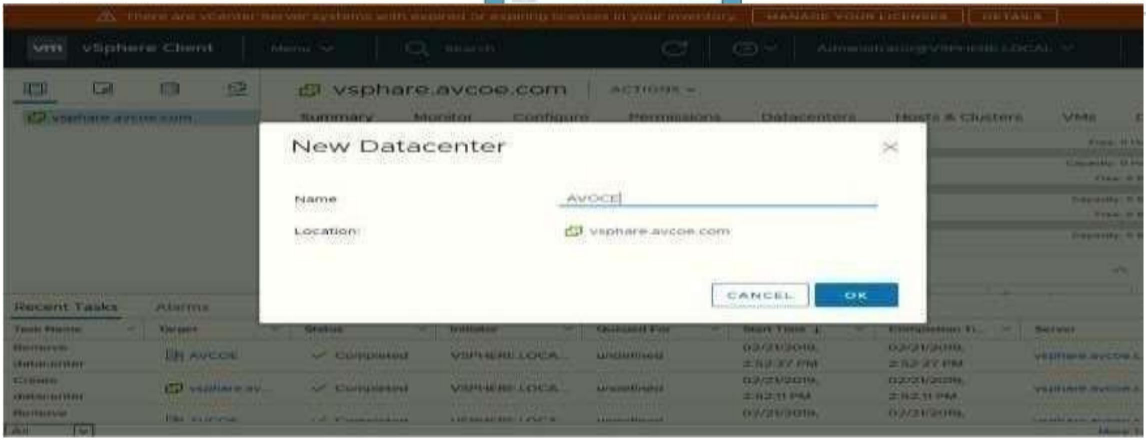


Task Name	Target	Status	Initiator	Queued For	Start Time ↓	Completion Ti...	Server
Remove datacenter	AVCOE	Completed	VSPHERE.LOCA...	undefined	02/21/2019, 2:50:42 PM	02/21/2019, 2:50:42 PM	vsphare.avcoe.c...
Create datacenter	vsphare.av...	Completed	VSPHERE.LOCA...	undefined	02/21/2019, 2:49:46 PM	02/21/2019, 2:49:46 PM	vsphare.avcoe.c...

Create Datacenter:



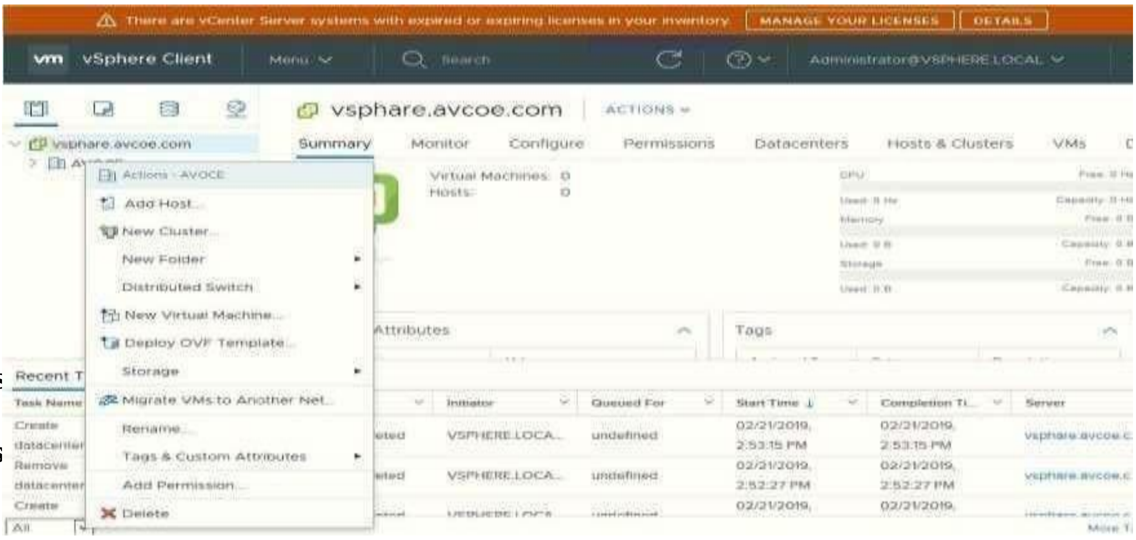
Task Name	Target	Status	Start Time	Completion Time	Server
Remove datacenter	AVCOE	✓ Completed	02/21/2019, 2:50:42 PM	02/21/2019, 2:50:42 PM	vsphare.avcoe.c...
Create datacenter	vsphare.av...	✓ Completed	02/21/2019, 2:49:46 PM	02/21/2019, 2:49:46 PM	vsphare.avcoe.c...



Name: AVOCE

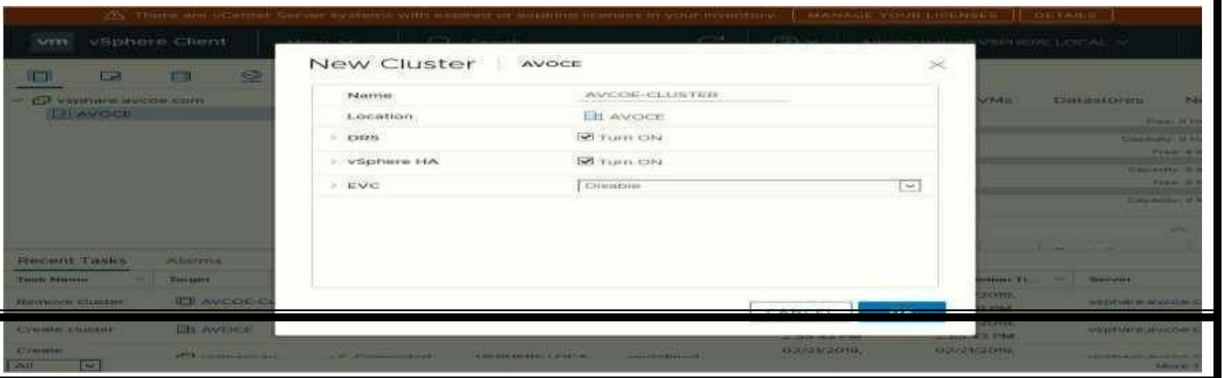
Location: vsphare.avcoe.com

Task Name	Target	Status	Initiator	Queued For	Start Time	Completion Time	Server
Remove datacenter	AVCOE	✓ Completed	VSPHERE.LOCA...	undefined	02/21/2019, 2:52:27 PM	02/21/2019, 2:52:27 PM	vsphare.avcoe.c...
Create datacenter	vsphare.av...	✓ Completed	VSPHERE.LOCA...	undefined	02/21/2019, 2:52:11 PM	02/21/2019, 2:52:11 PM	vsphare.avcoe.c...



Create clus

Assign clus



Name: AVCOE-CLUSTER

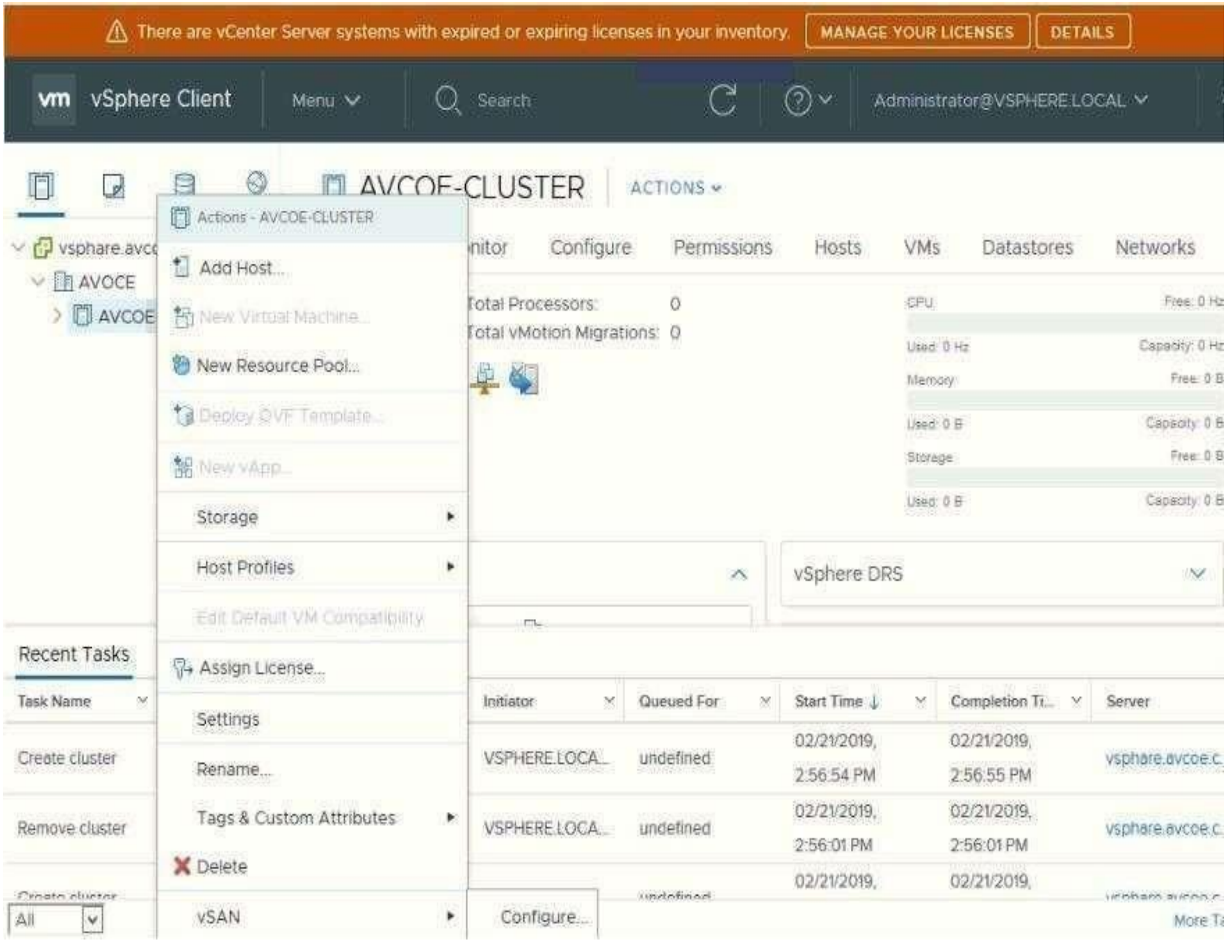
Location: AVOCE

DRS ☑ Turn ON

vSphere HA ☑ Turn ON

EVC Disable

Add host .:

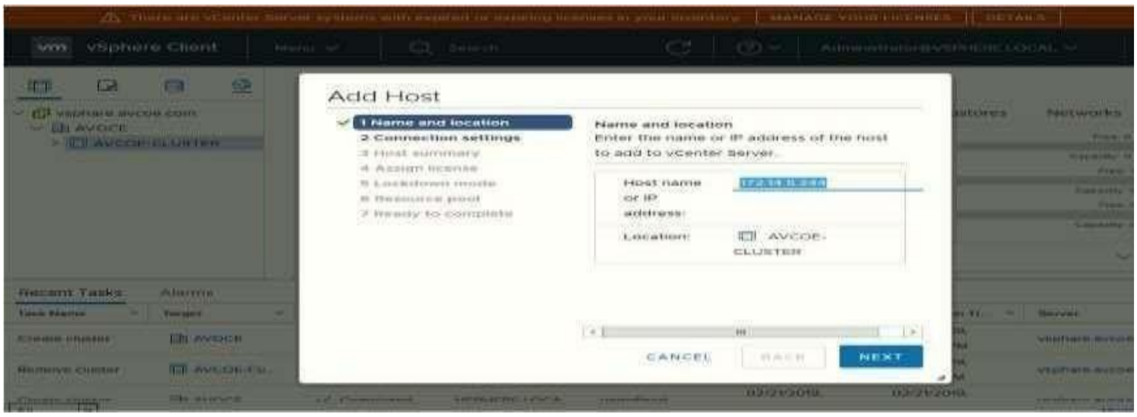


Total Processors: 0
Total vMotion Migrations: 0

CPU Free: 0 Hz
Used: 0 Hz Capacity: 0 Hz
Memory Free: 0 B
Used: 0 B Capacity: 0 B
Storage Free: 0 B
Used: 0 B Capacity: 0 B

Task Name	Initiator	Queued For	Start Time	Completion Time	Server
Create cluster	VSPHERE.LOCA...	undefined	02/21/2019, 2:56:54 PM	02/21/2019, 2:56:55 PM	vsphare.avcoe.c...
Remove cluster	VSPHERE.LOCA...	undefined	02/21/2019, 2:56:01 PM	02/21/2019, 2:56:01 PM	vsphare.avcoe.c...

Add host IP :

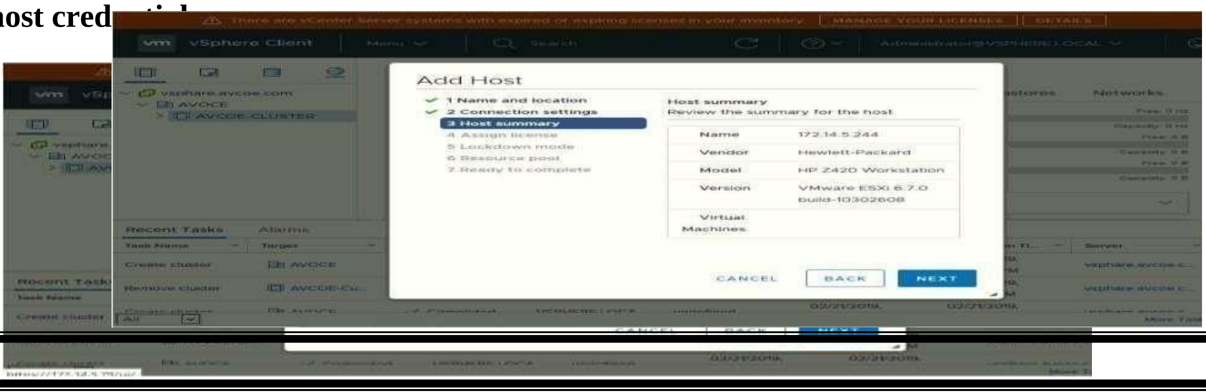


1 Name and location
2 Connection settings
3 Host summary
4 Assign license
5 Lockdown mode
6 Resource pool
7 Ready to complete

Name and location
Enter the name or IP address of the host to add to vCenter Server.

Host name or IP address: 172.14.5.244
Location: AVCOE-CLUSTER

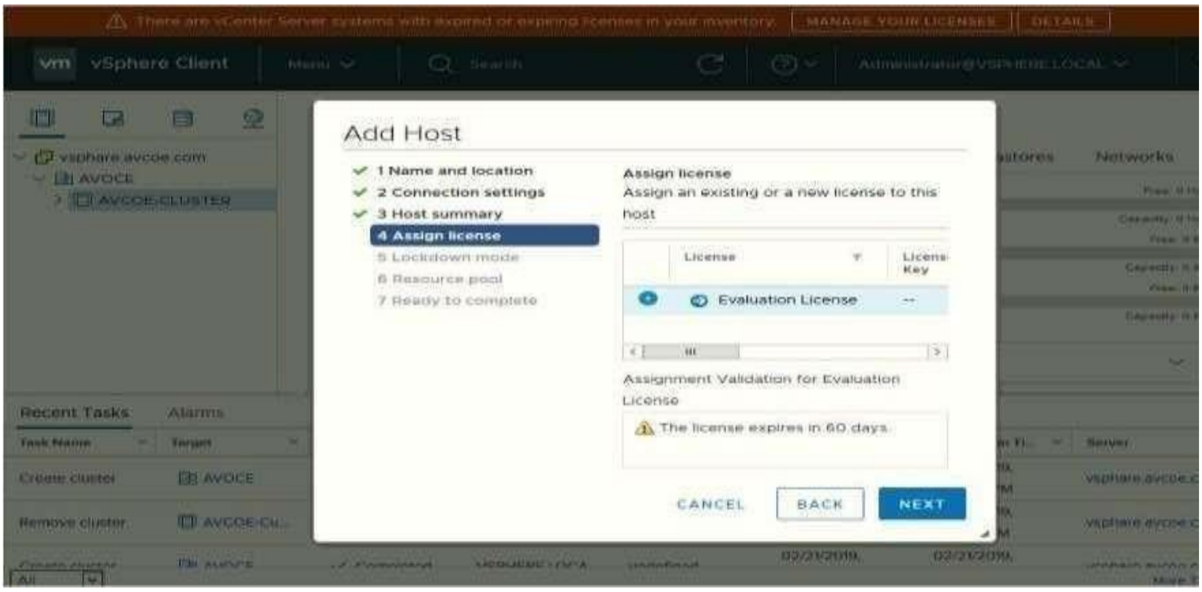
Enter host cred



1 Name and location
2 Connection settings
3 Host summary
4 Assign license
5 Lockdown mode
6 Resource pool
7 Ready to complete

Host summary
Review the summary for the host

Name	172.14.5.244
Vendor	Hewlett-Packard
Model	HP Z420 Workstation
Version	VMware ESXi 6.7.0 build-10302608
Virtual Machines	



There are vCenter Server systems with expired or expiring licenses in your inventory. MANAGE YOUR LICENSES DETAILS

Add Host

- ✓ 1 Name and location
- ✓ 2 Connection settings
- ✓ 3 Host summary
- 4 Assign license
- 5 Lockdown mode
- 6 Resource pool
- 7 Ready to complete

Assign license
Assign an existing or a new license to this host

	License	License Key
●	Evaluation License	--

Assignment Validation for Evaluation License

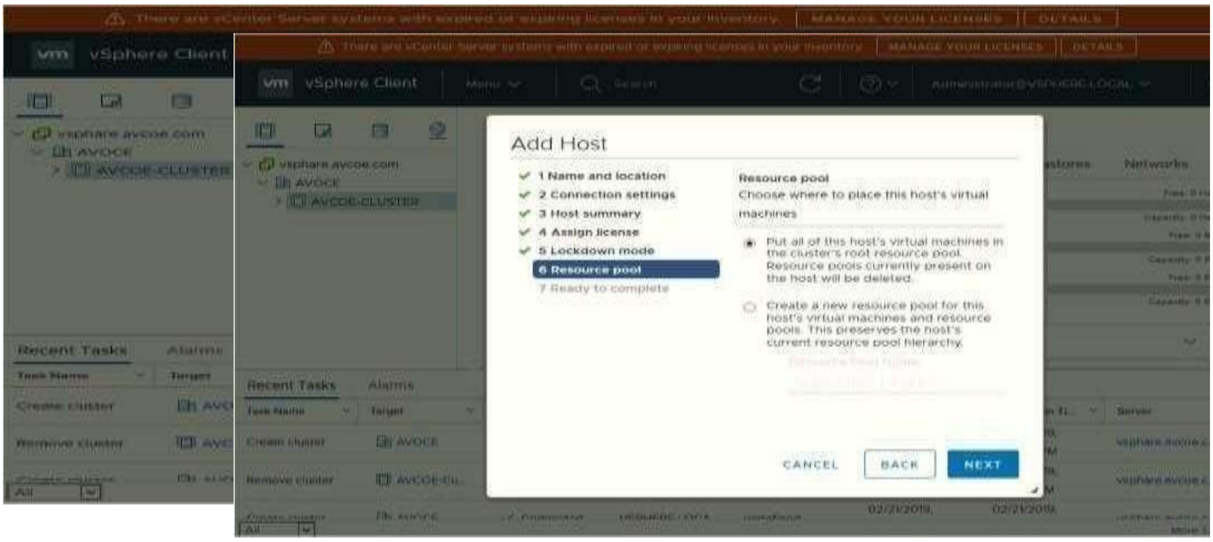
⚠ The license expires in 60 days.

CANCEL　BACK　NEXT

Hot summary :

Lock Down mode:

Add Host In Pool:



There are vCenter Server systems with expired or expiring licenses in your inventory. MANAGE YOUR LICENSES DETAILS

Add Host

- ✓ 1 Name and location
- ✓ 2 Connection settings
- ✓ 3 Host summary
- ✓ 4 Assign license
- ✓ 5 Lockdown mode
- 6 Resource pool
- 7 Ready to complete

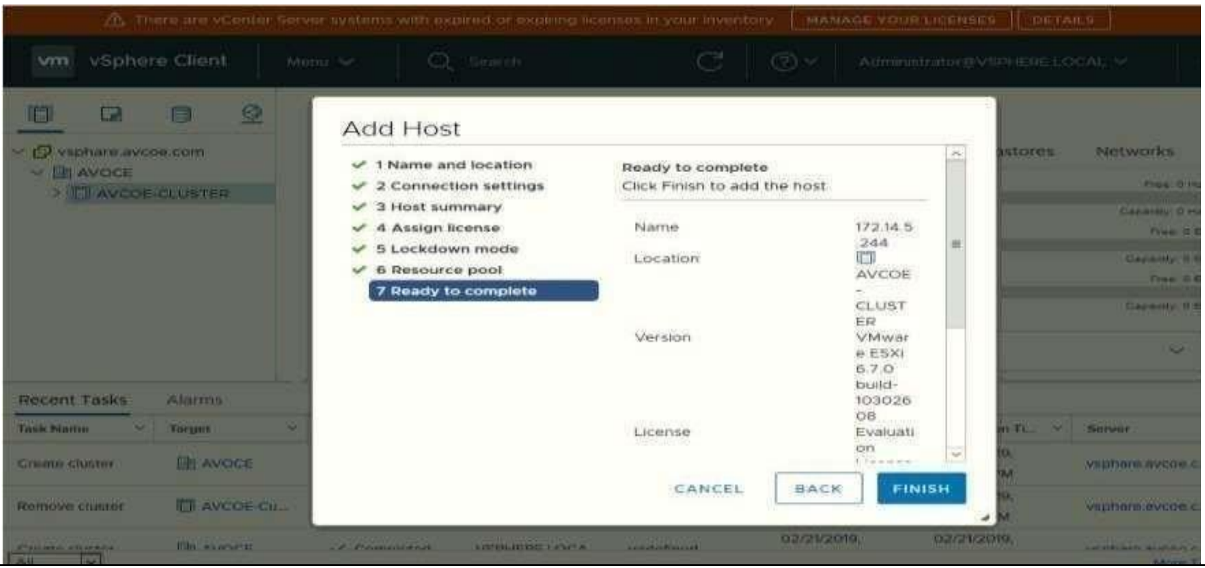
Resource pool
Choose where to place this host's virtual machines

● Put all of this host's virtual machines in the cluster's root resource pool. Resource pools currently present on the host will be deleted.

○ Create a new resource pool for this host's virtual machines and resource pools. This preserves the host's current resource pool hierarchy.

CANCEL　BACK　NEXT

Finish:



There are vCenter Server systems with expired or expiring licenses in your inventory. MANAGE YOUR LICENSES DETAILS

Add Host

- ✓ 1 Name and location
- ✓ 2 Connection settings
- ✓ 3 Host summary
- ✓ 4 Assign license
- ✓ 5 Lockdown mode
- ✓ 6 Resource pool
- 7 Ready to complete

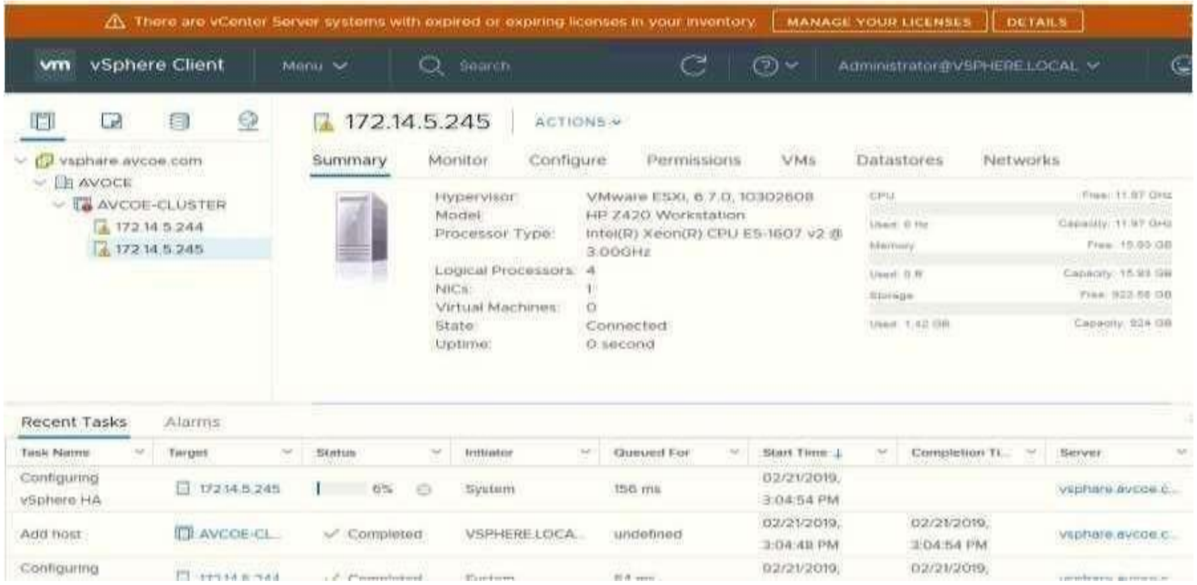
Ready to complete
Click Finish to add the host

Name	172.14.5.244
Location	AVCOE-CLUSTER
Version	VMware ESXi 6.7.0 build-10302608
License	Evaluation

CANCEL　BACK　FINISH

Host View and View Config:

Cluster View and Configuration:



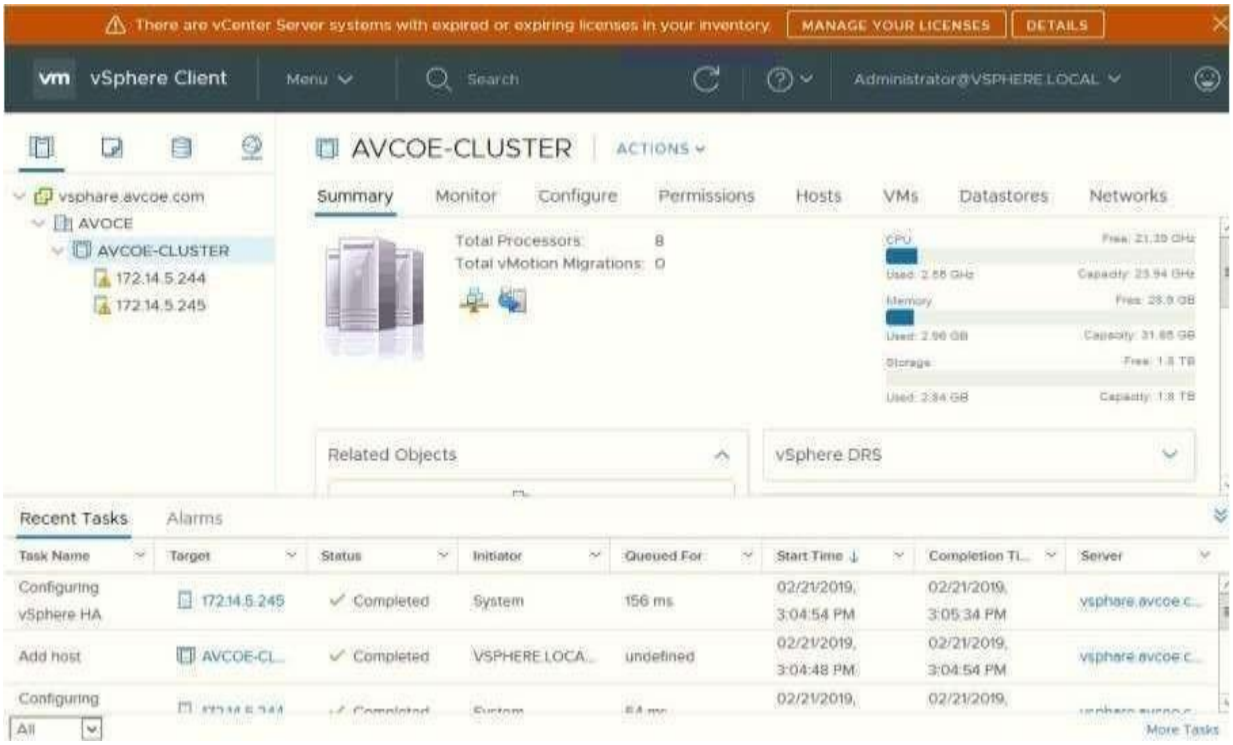
⚠ There are vCenter Server systems with expired or expiring licenses in your inventory. MANAGE YOUR LICENSES DETAILS

172.14.5.245

Hypervisor:	VMware ESXi, 6.7.0, 10302608
Model:	HP Z420 Workstation
Processor Type:	Intel(R) Xeon(R) CPU E5-1607 v2 @ 3.00GHz
Logical Processors:	4
NICs:	1
Virtual Machines:	0
State:	Connected
Uptime:	0 second

CPU Free: 11.97 GHz
Used: 0 Hz Capacity: 11.97 GHz
Memory Free: 15.93 GB
Used: 0 B Capacity: 15.93 GB
Storage Free: 922.58 GB
Used: 1.42 GB Capacity: 924 GB

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



⚠ There are vCenter Server systems with expired or expiring licenses in your inventory. MANAGE YOUR LICENSES DETAILS

AVCOE-CLUSTER

Total Processors:	8
Total vMotion Migrations:	0

CPU Free: 21.35 GHz
Used: 2.58 GHz Capacity: 23.94 GHz
Memory Free: 28.9 GB
Used: 2.96 GB Capacity: 31.85 GB
Storage Free: 1.8 TB
Used: 2.34 GB Capacity: 1.8 TB

Task Name	Target	Status	Initiator	Queued For	Start Time ↓	Completion Ti...	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	vsphare.avcoe.c...
Add host	AVCOE-CL...	✓ Completed	VSPHERE.LOCA...	undefined	02/21/2019, 3:04:48 PM	02/21/2019, 3:04:54 PM	vsphare.avcoe.c...
Configuring					02/21/2019,	02/21/2019,	

Conclusion: Like this we have configure V Sphere Private Cloud

Artificial Intelligence

Part I : Artificial Intelligence	
Suggested List of Laboratory Experiments/Assignments	
Sr. No.	**Group A** **All assignments are compulsory**
1.	Implement depth first search algorithm and Breadth First Search algorithm, Use an undirected graph and develop a recursive algorithm for searching all the vertices of a graph or tree data structure.
2.	Implement A star Algorithm for any game search problem.
3.	Implement Greedy search algorithm for any of the following application: I. Selection Sort II. Minimum Spanning Tree III. Single-Source Shortest Path Problem IV. Job Scheduling Problem V. Prim's Minimal Spanning Tree Algorithm VI. Kruskal's Minimal Spanning Tree Algorithm VII. Dijkstra's Minimal Spanning Tree Algorithm
	Group B
4.	Implement a solution for a Constraint Satisfaction Problem using Branch and Bound and Backtracking for n-queens problem or a graph coloring problem.
5.	Develop an elementary catboat for any suitable customer interaction application.
	Group C
6.	Implement any one of the following Expert System I. Information management II. Hospitals and medical facilities III. Help desks management IV. Employee performance evaluation V. Stock market trading VI. Airline scheduling and cargo schedules

Practical No-1

TITLE: Implement DFS and BFS Algorithm. Use and Undirected Graph and develop a Recursive Algorithm for searching all the vertices of the graph or tree data structure.

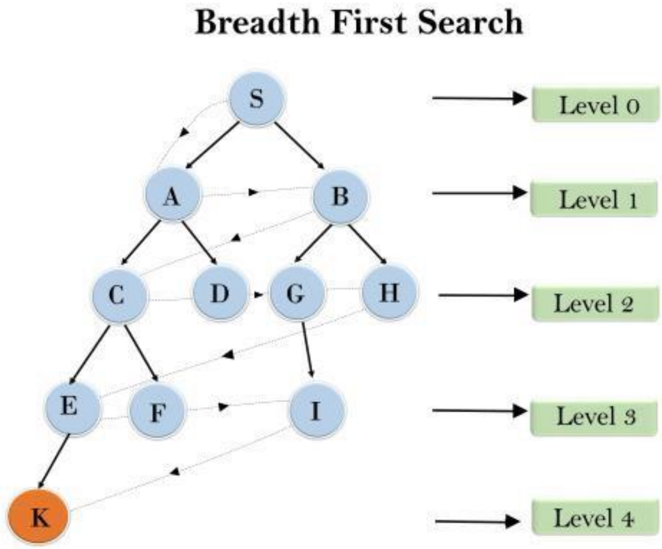
AIM: Aim of this practical is to develop DFS and BFS algorithm programs in programming language.

OBJECTIVES: Based on above main aim following are the objectives

1. To study BFS
2. To study DFS
3. To apply algorithmic logic in implementation of program.

1. Breadth-first Search:

- Breadth-first search is the most common search strategy for traversing a tree or graph. This algorithm searches breadthwise in a tree or graph, so it is called breadth-first search.
- BFS algorithm starts searching from the root node of the tree and expands all successor node at the current level before moving to nodes of next level.
- The breadth-first search algorithm is an example of a general-graph search algorithm.
- Breadth-first search implemented using FIFO queue data structure.



In the above tree structure, we have shown the traversing of the tree using BFS algorithm from the root node S to goal node K. BFS search algorithm traverse in layers, so it will follow the path which is shown by the dotted arrow, and the traversed path will be:

S---> A--->B---->C--->D---->G--->H--->E---->F---->I >K

- **Time Complexity:** Time Complexity of BFS algorithm can be obtained by the number of nodes traversed in BFS until the shallowest Node. Where the d= depth of shallowest solution and b is a node at every state.
- **$T (b) = 1+b^2+b^3+. \quad + b^d= O (b^d)$**
- **Space Complexity:** Space complexity of BFS algorithm is given by the Memory size of frontier which is $O(b^d)$.
- **Completeness:** BFS is complete, which means if the shallowest goal node is at some finite depth, then BFS will find a solution.
- **Optimality:** BFS is optimal if path cost is a non-decreasing function of the depth of the node.

Algorithm:

1. Pick any node, visit the adjacent unvisited vertex, mark it as visited, display it, and insert it in a queue.
2. If there are no remaining adjacent vertices left, remove the first vertex from the queue.
3. Repeat step 1 and step 2 until the queue is empty or the desired node is found

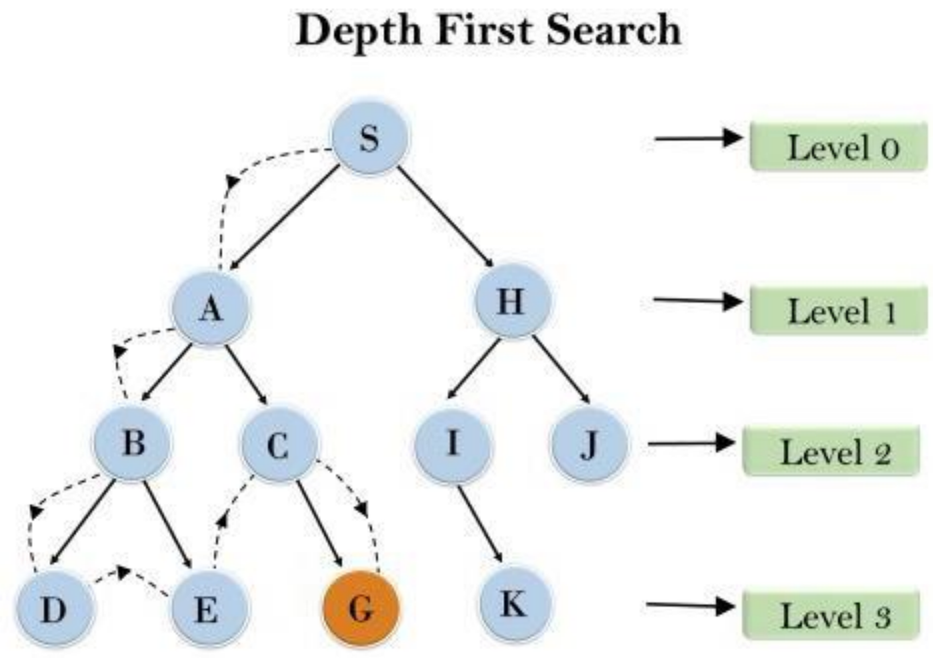
2. Depth-first Search:

- Depth-first search isa recursive algorithm for traversing a tree or graph data structure.
- It is called the depth-first search because it starts from the root node and follows each path to its greatest depth node before moving to the next path.
- DFS uses a stack data structure for its implementation.
- The process of the DFS algorithm is similar to the BFS algorithm.

Example: In the below search tree, we have shown the flow of depth-first search, and it will follow the order as:

Root node--->Left node > right node.

It will start searching from root node S, and traverse A, then B, then D and E, after traversing E, it will backtrack the tree as E has no other successor and still goal node is not found. After backtracking it will traverse node C and then G, and here it will terminate as it found goal node.



- **Completeness:** DFS search algorithm is complete within finite state space as it will expand every node within a limited search tree.
- **Time Complexity:** Time complexity of DFS will be equivalent to the node traversed by the algorithm. It is given by:

$$\mathbf{T(n)= 1+ n^{2}+ n^{3} +.......+ n^{m}=O(n^{m})}$$

- **Where, m= maximum depth of any node and this can be much larger than d (Shallowest solution depth)**
- **Space Complexity:** DFS algorithm needs to store only single path from the root node, hence space complexity of DFS is equivalent to the size of the fringe set, which is **O(bm)**.
- **Optimal:** DFS search algorithm is non-optimal, as it may generate a large number of steps or high cost to reach to the goal node.

Algorithm:

1. We will start by putting any one of the graph's vertex on top of the stack.
2. After that take the top item of the stack and add it to the visited list of the vertex.
3. Next, create a list of that adjacent node of the vertex. Add the ones which aren't in the visited list of vertexes to the top of the stack.
4. Lastly, keep repeating steps 2 and 3 until the stack is empty.

Conclusion: Thus we have studied BFS & DFS algorithm in detail and implemented using recursive function.

Practical No-2

TITLE: Implement A* Algorithm for any game search problem

AIM: Aim of this practical is to apply algorithmic logic for any game

OBJECTIVES: Based on above main aim following are the objectives

1. To study A* algorithm
2. To study different game search problems
3. To determine performance A* algorithm in various games

MOTIVATION: To approximate the shortest path in real-life situations, like- in maps, games where there can be many hindrances.

INTRODUCTION

A* Search Algorithm:

A* Search algorithm is one of the best and popular technique used in path-finding and graph traversals. A* Search algorithms, unlike other traversal techniques, it has “brains”. What it means is that it is really a smart algorithm which separates it from the other conventional algorithms. This fact is cleared in detail in below sections. And it is also worth mentioning that many games and web-based maps use this algorithm to find the shortest path very efficiently (approximation). Consider a square grid having many obstacles and we are given a starting cell and a target cell. We want to reach the target cell (if possible) from the starting cell as quickly as possible. Here A* Search Algorithm comes to the rescue. What A* Search Algorithm does is that at each step it picks the node according to a value-‘f’ which is a parameter equal to the sum of two other parameters – ‘g’ and ‘h’. At each step it picks the node/cell having the lowest ‘f’, and process that node/cell. We define ‘g’ and ‘h’ as simply as possible below

g = the movement cost to move from the starting point to a given square on the grid, following the path generated to get there.

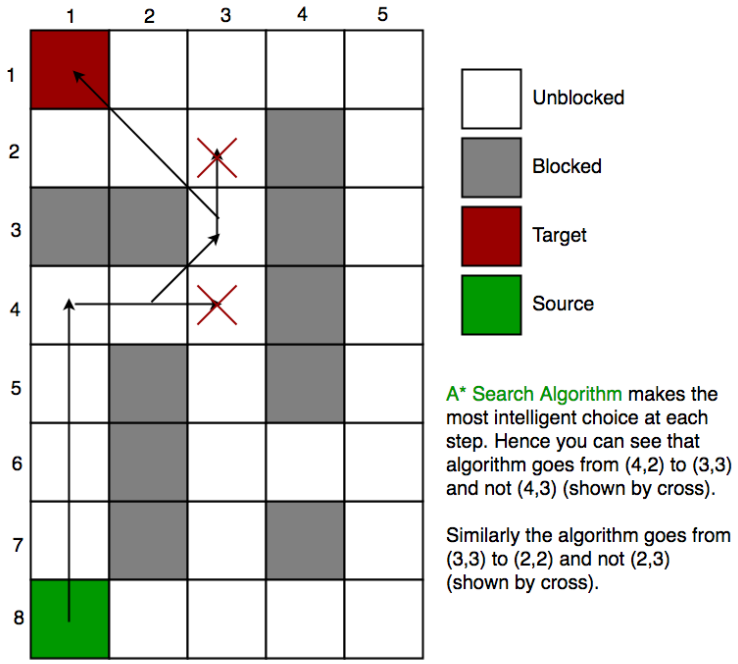
h = the estimated movement cost to move from that given square on the grid to the final destination. This is often referred to as the heuristic, which is nothing but a kind of smart guess. We really don’t know the actual distance until we find the path, because all sorts of things can be in the way (walls, water, etc.). There can be many ways to calculate this ‘h’ which are discussed in the later sections.

Algorithm

We create two lists – Open List and Closed List (just like Dijkstra Algorithm)

```
// A* Search Algorithm
1.  Initialize the open list
2.  Initialize the closed list
    put the starting node on the open
    list (you can leave its f at zero)

3.  while the open list is not empty
    a) find the node with the least f on
       the open list, call it "q"

    b) pop q off the open list

    c) generate q's 8 successors and set their
       parents to q

    d) for each successor
        i) if successor is the goal, stop search
          successor.g = q.g + distance between
                              successor and q
          successor.h = distance from goal to
          successor (This can be done using many
          ways, we will discuss three heuristics-
          Manhattan, Diagonal and Euclidean
          Heuristics)

          successor.f = successor.g + successor.h

        ii) if a node with the same position as
          successor is in the OPEN list which has a
          lower f than successor, skip this successor

        iii) if a node with the same position as
          successor  is in the CLOSED list which has
          a lower f than successor, skip this successor
          otherwise, add  the node to the open list
     end (for loop)

    e) push q on the closed list
    end (while loop)
```

So suppose as in the below figure if we want to reach the target cell from the source cell, then the A* Search algorithm would follow path as shown below. Note that the below figure is made by considering Euclidean Distance as a heuristics.



A* Search Algorithm makes the most intelligent choice at each step. Hence you can see that algorithm goes from (4,2) to (3,3) and not (4,3) (shown by cross).

Similarly the algorithm goes from (3,3) to (2,2) and not (2,3) (shown by cross).

Heuristics:

We can calculate g but how to calculate h ?

A) Either calculate the exact value of h (which is certainly time consuming).

OR

B) Approximate the value of h using some heuristics (less time consuming).

A) Exact Heuristics –

We can find exact values of h, but that is generally very time consuming.

Below are some of the methods to calculate the exact value of h.

1) Pre-compute the distance between each pair of cells before running the A* Search Algorithm.

2) If there are no blocked cells/obstacles then we can just find the exact value of h without any pre-computation using the distance formula/Euclidean Distance

B) Approximation Heuristics –

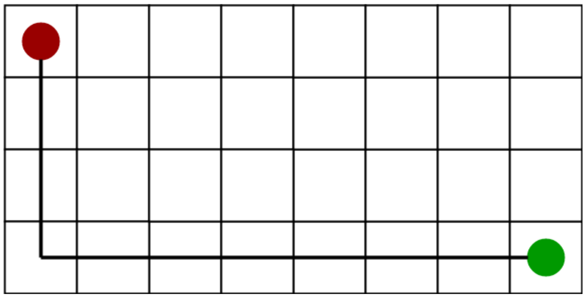
There are generally three approximation heuristics to calculate h –

1) **Manhattan Distance** – It is nothing but the sum of absolute values of differences in the goal's x and y coordinates and the current cell's x and y coordinates respectively, i.e.,

```
h = abs (current_cell.x – goal.x) +
abs (current_cell.y – goal.y)
```

When to use this heuristic? – When we are allowed to move only in four directions only (right, left, top, bottom)

The Manhattan Distance Heuristics is shown by the below figure (assume red spot as source cell and green spot as target cell).



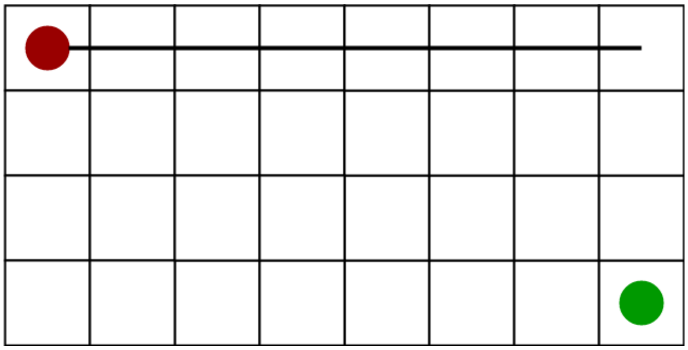
2) Diagonal Distance-

It is nothing but the maximum of absolute values of differences in the goal's x and y coordinates and the current cell's x and y coordinates respectively, i.e.,

```
dx = abs(current_cell.x – goal.x)

dy = abs(current_cell.y – goal.y)


h = D * (dx + dy) + (D2 - 2 * D) * min(dx, dy)

where D is length of each node(usually = 1) and D2 is diagonal
distance between each node (usually = sqrt(2) ).
```

When to use this heuristic? – When we are allowed to move in eight directions only (similar to a move of a King in Chess)

The Diagonal Distance Heuristics is shown by the below figure (assume red spot as source cell and green spot as target cell).



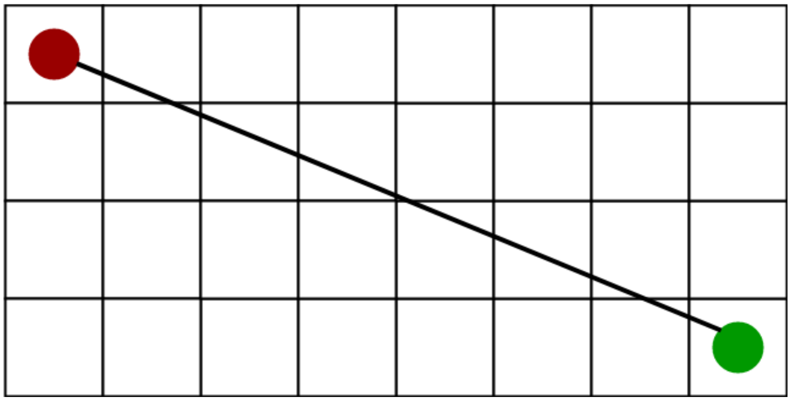
3) Euclidean Distance-

As it is clear from its name, it is nothing but the distance between the current cell and the goal cell using the distance formula

```
h = sqrt ( (current_cell.x – goal.x)2 +
            (current_cell.y – goal.y)2 )
```

When to use this heuristic? – When we are allowed to move in any directions.

The Euclidean Distance Heuristics is shown by the below figure (assume red spot as source cell and green spot as target cell).



Implementation:

We can use any data structure to implement open list and closed list but for best performance, we use a set data structure of C++ STL(implemented as Red-Black Tree) and a boolean hash table for a closed list.

The implementations are similar to Dijsktra's algorithm. If we use a Fibonacci heap to implement the open list instead of a binary heap/self-balancing tree, then the performance will become better (as Fibonacci heap takes O(1) average time to insert into open list and to decrease key)

CONCLUSIONS: Thus we have studied A* algorithm in detail and implemented it in game search problem

Practical No-3

TITLE: Implement Greedy search algorithm for Kruskal's Minimal Spanning Tree Algorithm

AIM: To apply Greedy search algorithm mechanism in Kruskal’s MST.

OBJECTIVES: Based on above main aim following are the objectives

1. To study Greedy Search algorithm
2. To study Kruskal’s minimum spanning tree
3. To determine complexity of algorithms

INTRODUCTION

Greedy Algorithms:

Greedy is an algorithmic paradigm that builds up a solution piece by piece, always choosing the next piece that offers the most obvious and immediate benefit. So the problems where choosing locally optimal also leads to global solution are best fit for Greedy.

Structure of a Greedy Algorithm:

Greedy algorithms take all of the data in a particular problem, and then set a rule for which elements to add to the solution at each step of the algorithm. In the animation above, the set of data is all of the numbers in the graph, and the rule was to select the largest number available at each level of the graph. The solution that the algorithm builds is the sum of all of those choices.

Kruskal’s Minimum Spanning Tree Algorithm:

What is Minimum Spanning Tree?

Given a connected and undirected graph, a spanning tree of that graph is a subgraph that is a tree and connects all the vertices together. A single graph can have many different spanning trees. A minimum spanning tree (MST) or minimum weight spanning tree for a weighted, connected, undirected graph is a spanning tree with a weight less than or equal to the weight of every other spanning tree. The weight of a spanning tree is the sum of weights given to each edge of the spanning tree.

How many edges does a minimum spanning tree has? –
A minimum spanning tree has (V – 1) edges where V is the number of vertices in the given graph.

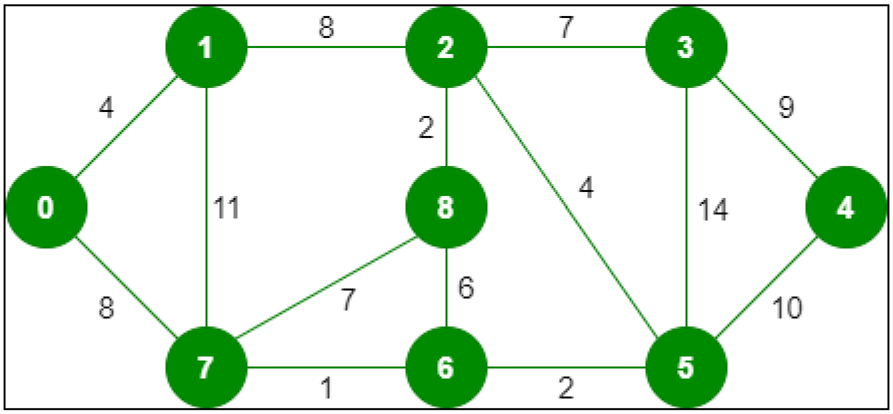
Below are the steps for finding MST using Kruskal's algorithm

1. Sort all the edges in non-decreasing order of their weight.

2. Pick the smallest edge. Check if it forms a cycle with the spanning tree formed so far. If cycle is not formed, include this edge. Else, discard it.

3. Repeat step 2 until there are (V-1) edges in the spanning tree.

The algorithm is a Greedy Algorithm. The Greedy Choice is to pick the smallest weight edge that does not cause a cycle in the MST constructed so far. Let us understand it with an example: Consider the below input graph.



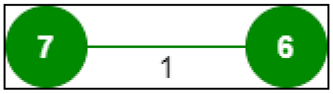
The graph contains 9 vertices and 14 edges. So, the minimum spanning tree formed will be having (9 – 1) = 8 edges.

After sorting:

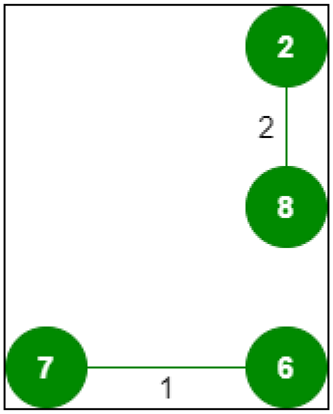
Weight	Src	Dest
1	7	6
2	8	2
2	6	5
4	0	1
4	2	5
6	8	6
7	2	3
7	7	8
8	0	7
8	1	2
9	3	4
10	5	4
11	1	7
14	3	5

Now pick all edges one by one from the sorted list of edges

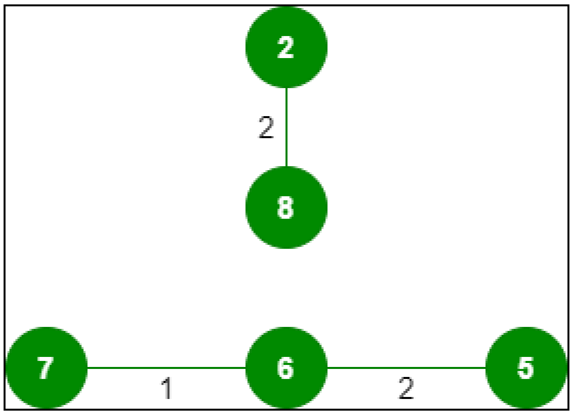
1. Pick edge 7-6: No cycle is formed, include it.



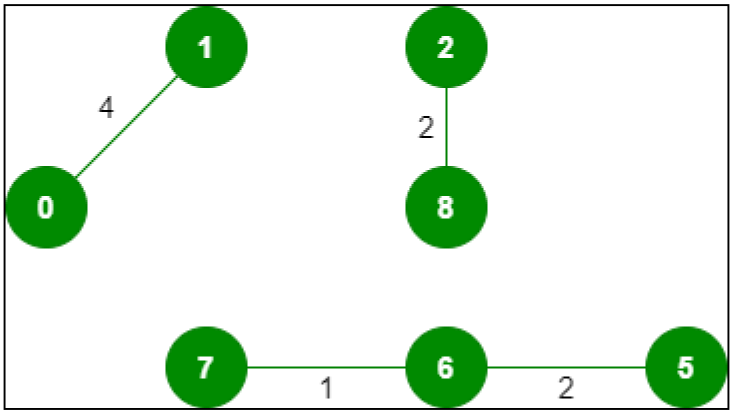
2. Pick edge 8-2: No cycle is formed, include it.



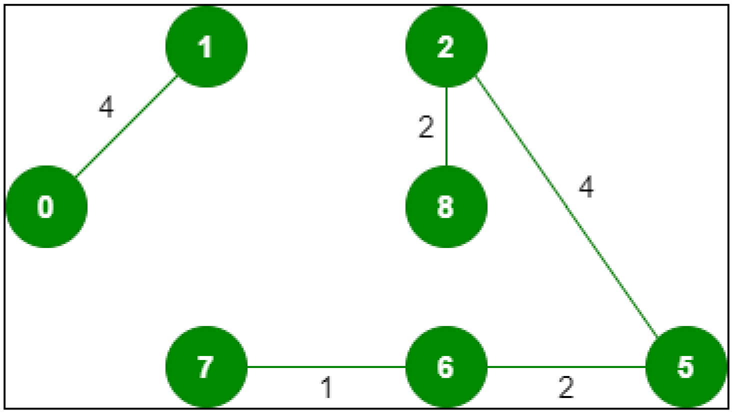
3. Pick edge 6-5: No cycle is formed, include it.



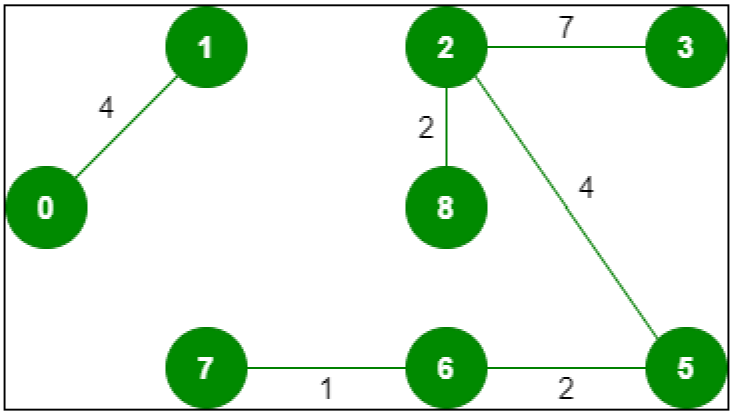
4. Pick edge 0-1: No cycle is formed, include it.



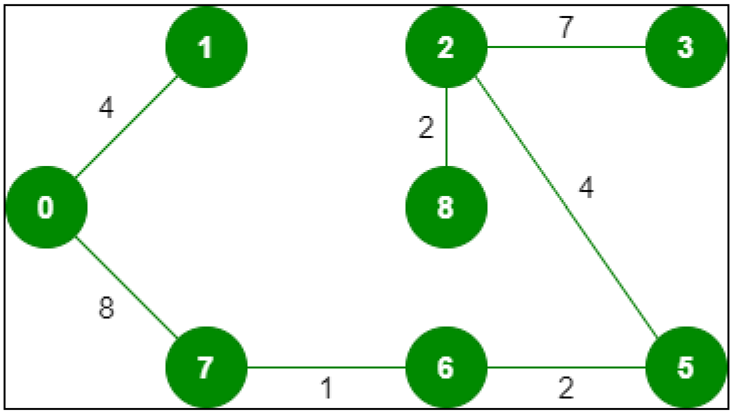
5. Pick edge 2-5: No cycle is formed, include it.



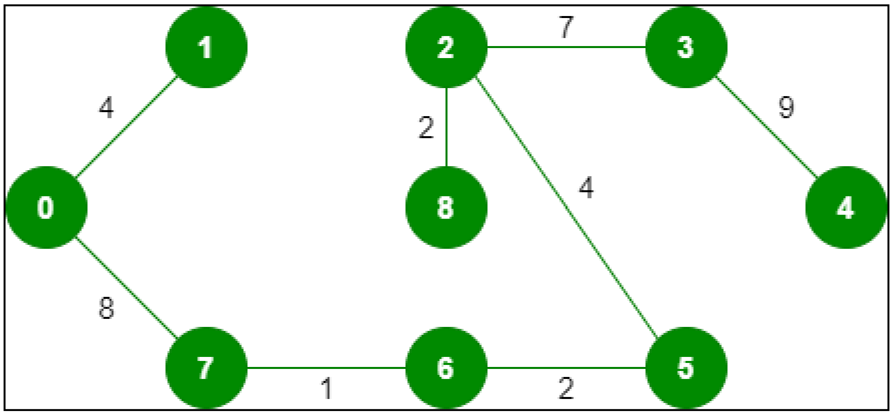
6. Pick edge 8-6: Since including this edge results in the cycle, discard it.
7. Pick edge 2-3: No cycle is formed, include it.



8. Pick edge 7-8: Since including this edge results in the cycle, discard it.
9. Pick edge 0-7: No cycle is formed, include it.



10. Pick edge 1-2: Since including this edge results in the cycle, discard it.
11. Pick edge 3-4: No cycle is formed, include it.



Since the number of edges included equals (V – 1), the algorithm stops here.

Time Complexity:

O(ElogE) or O(ElogV). Sorting of edges takes O(ELogE) time. After sorting, we iterate through all edges and apply the find-union algorithm. The find and union operations can take at most O(LogV) time. So overall complexity is O(ELogE + ELogV) time. The value of E can be at most O(V2), so O(LogV) is O(LogE) the same. Therefore, the overall time complexity is O(ElogE) or O(ElogV)

Conclusion: Thus we have studied how to implement Greedy Search Algorithm in Kruskal’s Minimum Spanning Tree.

Practical No-4

TITLE: Implement a solution for a Constraint Satisfaction Problem using Branch and Bound and Backtracking for n-queens problem or a graph coloring problem

AIM: To use Branch & Bound and Backtracking to solve CSP & n-queens or graph coloring problem.

OBJECTIVES: Based on above main aim following are the objectives

1. To study Branch & Bound and Backtracking methods.
2. To study Constraint Satisfaction Problem
3. To study n-queen or graph coloring problem

Theory:

Constraint Satisfaction Problems:

The objective of every problem-solving technique is one, i.e., to find a solution to reach the goal. Although, in adversarial search and local search, there were no constraints on the agents while solving the problems and reaching to its solutions. By the name, it is understood that constraint satisfaction means solving a problem under certain constraints or rules.

Constraint satisfaction is a technique where a problem is solved when its values satisfy certain constraints or rules of the problem. Such type of technique leads to a deeper understanding of the problem structure as well as its complexity.

Constraint satisfaction depends on three components, namely:

X: It is a set of variables.

D: It is a set of domains where the variables reside. There is a specific domain for each variable.

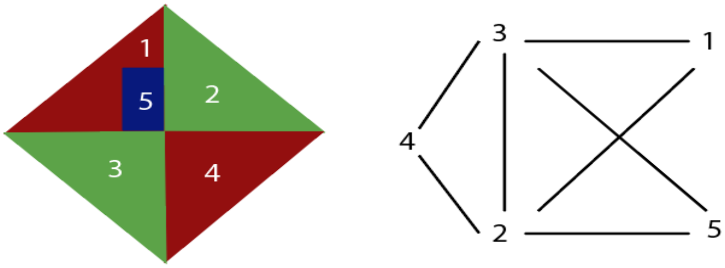
C: It is a set of constraints which are followed by the set of variables.

In constraint satisfaction, domains are the spaces where the variables reside, following the problem specific constraints. These are the three main elements of a constraint satisfaction technique. The constraint value consists of a pair of {scope, rel}. The scope is a tuple of variables which participate in the constraint and rel is a relation which includes a list of values which the variables can take to satisfy the constraints of the problem.

CSP Problems:

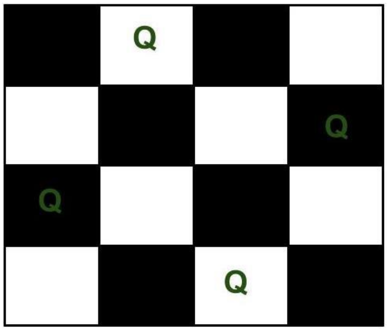
Constraint satisfaction includes those problems which contains some constraints while solving the problem. CSP includes the following problems:

Graph Coloring: The problem where the constraint is that no adjacent sides can have the same color.



Graph Coloring

n-queen problem: In n-queen problem, the constraint is that no queen should be placed either diagonally, in the same row or column. The N Queen is the problem of placing N chess queens on an N×N chessboard so that no two queens attack each other.



Backtracking Algorithm to solve n-queen problem:

The idea is to place queens one by one in different columns, starting from the leftmost column. When we place a queen in a column, we check for clashes with already placed queens. In the current column, if we find a row for which there is no clash, we mark this row and column as part of the solution. If we do not find such a row due to clashes then we backtrack and return false.

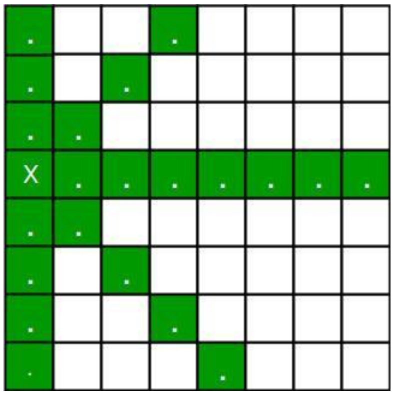
```
1) Start in the leftmost column
```

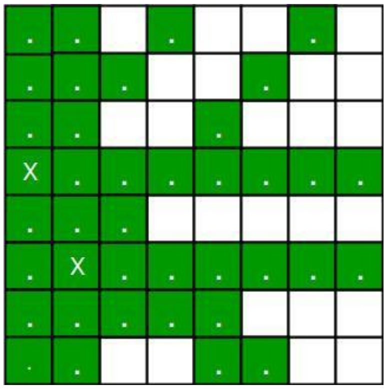
```
2) If all queens are placed
    return true
3) Try all rows in the current column.
   Do following for every tried row.
    a) If the queen can be placed safely in this row
       then mark this [row, column] as part of the
       solution and recursively check if placing
       queen here leads to a solution.
    b) If placing the queen in [row, column] leads to
       a solution then return true.
    c) If placing queen doesn't lead to a solution then
       unmark this [row, column] (Backtrack) and go to
       step (a) to try other rows.
3) If all rows have been tried and nothing worked,
   return false to trigger backtracking.
```

N Queen Problem using Branch and Bound:

In backtracking solution we backtrack when we hit a dead end. In Branch and Bound solution, after building a partial solution, we figure out that there is no point going any deeper as we are going to hit a dead end.

“The idea is to place queens one by one in different columns, starting from the leftmost column. When we place a queen in a column, we check for clashes with already placed queens. In the current column, if we find a row for which there is no clash, we mark this row and column as part of the solution. If we do not find such a row due to clashes, then we backtrack and return false.”





1. For the 1st Queen, there are total 8 possibilities as we can place 1st Queen in any row of first column. Let’s place Queen 1 on row 3.
2. After placing 1st Queen, there are 7 possibilities left for the 2nd Queen. But wait, we don’t really have 7 possibilities. We cannot place Queen 2 on rows 2, 3 or 4 as those cells are under attack from Queen 1. So, Queen 2 has only 8 – 3 = 5 valid positions left.
3. After picking a position for Queen 2, Queen 3 has even fewer options as most of the cells in its column are under attack from the first 2 Queens.

We need to figure out an efficient way of keeping track of which cells are under attack. In previous solution we kept an 8--by--8 Boolean matrix and update it each time we placed a queen, but that required linear time to update as we need to check for safe cells.

Basically, we have to ensure 4 things:

1. No two queens share a column.

2. No two queens share a row.

3. No two queens share a top-right to left-bottom diagonal.

4. No two queens share a top-left to bottom-right diagonal.

Number 1 is automatic because of the way we store the solution. For number 2, 3 and 4, we can perform updates in O(1) time. The idea is to keep three Boolean arrays that tell us which rows and which diagonals are occupied.

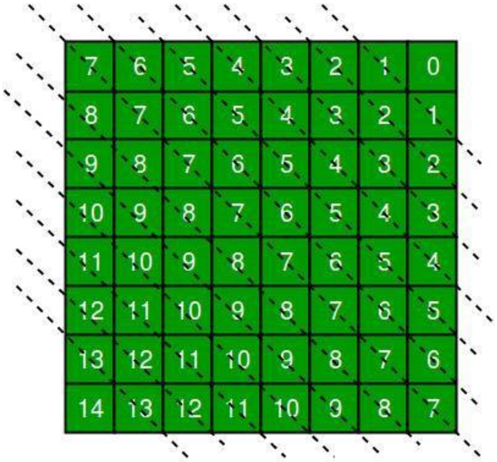
Let’s do some pre-processing first. Let’s create two N x N matrix one for / diagonal and other one for \ diagonal. Let’s call them slashCode and backslashCode respectively. The trick is to fill them in such a way that two queens sharing a same /-diagonal will have the same value in matrix slashCode, and if they share same \-diagonal, they will have the same value in backslashCode matrix.

For an N x N matrix, fill slashCode and backslashCode matrix using below formula –

slashCode[row][col] = row + col

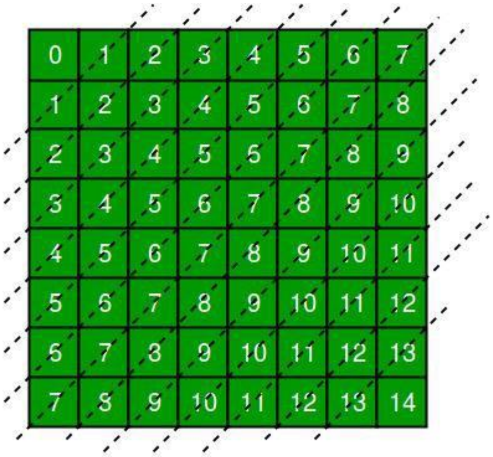
backslashCode[row][col] = row – col + (N-1)

Using above formula will result in below matrices



7	6	5	4	3	2	1	0
8	7	6	5	4	3	2	1
9	8	7	6	5	4	3	2
10	9	8	7	6	5	4	3
11	10	9	8	7	6	5	4
12	11	10	9	8	7	6	5
13	12	11	10	9	8	7	6
14	13	12	11	10	9	8	7

r -c + 7



0	1	2	3	4	5	6	7
1	2	3	4	5	6	7	8
2	3	4	5	6	7	8	9
3	4	5	6	7	8	9	10
4	5	6	7	8	9	10	11
5	6	7	8	9	10	11	12
6	7	8	9	10	11	12	13
7	8	9	10	11	12	13	14

r + c

The ‘N – 1’ in the backslash code is there to ensure that the codes are never negative because we will be using the codes as indices in an array.

Now before we place queen i on row j, we first check whether row j is used (use an array to store row info). Then we check whether slash code (j + i) or backslash code (j – i + 7) are used (keep two arrays that will tell us which diagonals are occupied). If yes, then we have to try a different location for queen i. If not, then we mark the row and the two diagonals as used and recurse on queen i + 1. After the recursive call returns and before we try another position for queen i, we need to reset the row, slash code and backslash code as unused again

Conclusion: Thus we have studied how to solve n-queen (Constraint Satisfaction Problem) using Backtracking and Branch & Bound.

Practical No-5

TITLE: Develop an elementary chatbot for any suitable customer interaction application

AIM: To create Chatbot for customer interaction

OBJECTIVES: Based on above main aim following are the objectives

1. To study theory of Chatbot
2. To create Chatbot

Theory:

Chatbots:

A chatbot can be defined as a computer program that converses with users over the internet as any human would with natural language and sentiments. It is a program or piece of software that automatically responds to messages sent over a website chatbox, email, social media messaging app, or text.

For a deeper understanding of Chatbot, we can define it as a computer program that impersonates human conversations in its natural format, which may include text (since the advent of bots) or spoken language using artificial intelligence (AI) techniques such as Natural Language Processing (NLP) and audio analysis. One of the primary aspects of an AI-based bot is that it is dynamic.

A chatbot is a smart application that reduces human work and helps an organization to solve basic queries of the customer. Today most of the companies, business from different sector makes use of chatbot in a different way to reply their customer as fast as possible. chatbots also help in increasing traffic of site which is top reason of business to use chatbots.

Chatbot asks for basic information of customers like name, email address, and the query. If a query is simple like product fault, booking mistake, need some information then without any human connection it can solve it automatically and If some problem is high then It passes the details to the human head and helps customer to connect with organization manager easily. And most of the customers like to deal and talk with a chatbot.

How do the Chatbots function?

The main technology that lies behind chatbots is NLP and Machine Learning.

When a question is presented to a chatbot, a series or complex algorithms process the received input, understand what the user is asking, and based on that, determines the answer suitable to the question.

Chatbots have to rely on the ability of the algorithms to detect the complexity of both text and spoken words. Some chatbots perform very well to the point it becomes difficult to differentiate whether the user is a machine or a human.

However, handling complex conversations is a huge challenge; where there is a usage of various figures of speech, it may be difficult for machines to understand.

Why we need Chatbots?

Cost and Time Effective: Humans cannot be active on-site 24/7 but chatbots can and the replying power of chatbots is much fast than humans.

Cheap Development cost: with the advancement in technology many tools are developed that help easy development and integration of chatbots with little investment.

Human Resource: Today Chatbots can also talk with text o speech technology so it gives the feel as a human is talking on another side.

Business Branding: Businesses are changing with technology and chatbot is one out of them. Chatbot also helps in advertising, branding of organization product and services and give daily updates to users.

Types of Chatbots:

1. **Rule-based chatbots:** Chatbots follow a set of established rules or flows to respond to questions posted by a user. All your simple applications contain rule-based chatbots, which respond to queries based on the rules they are trained on. For instance, a weather application, where you ask for weather forecast and it fetches the data from different sources and responds with the information.

   Rule-based chatbots may not be able to hold complex conversations. It can only accomplish the tasks it is programmed to perform unless more improvements are made by the developer.

2. **Self-learning chatbots: S**elf-learning bots are highly efficient because they are capable to grab and identify the user’s intent on their own. They are build using advanced tools and techniques of Machine Learning, Deep Learning, and NLP. Self-learning bots are further divided into 2 subcategories.

   a. **Retrieval-based chatbots**:- Retrieval-based it is somewhat the same as Rule-based where predefined input patterns and responses are embedded.

b. **Generative-Based chatbots**:- It is based on the same phenomenon as Machine Translation build using sequence 2 sequences neural network.

Most of the organization uses self-learning chatbot along with embedding some rules like Hybrid version of both methods which makes chatbot powerful to handle each situation during a conversation with a customer.

Chatbot Development Platforms:

IBM Watson: Watson is one of the most preferred platforms when it comes to building AI chatbots. The advantage of Watson is its capability to serve different verticals and manage complex interactions with ease.

Microsoft Azure Bot Service: The Azure bot service provides the developer with SDK and portal, along with a bot connector service that will allow the developer to connect to any social media platform. The SDK also helps with debugging your bot and provides a large selection of sample bots that can be used as building blocks for your bot. This Cloud-based service is accessible from almost anywhere and provides multiple language support.

QnA Maker: This is another bot from Microsoft, which is exactly as the name suggests. It can be of great help to any business that is asked frequent questions from their customers regarding their products. QnA Maker allows you to develop and train your bots for answering simple questions, based on your FAQ URLs, any structured documents, and manuals for the product within a matter of minutes.

Chatbot Deployment Platforms:

Once chatbots are developed, they need to be deployed to a deployment platform. You will have to choose a deployment platform based on your customer base. However, the use of chatbots revolves mostly around social media platforms or virtual assistant features in various devices. Let us look at some of the emerging bot platform ecosystems.

Facebook Messenger: With over 1 billion users, there is no denying that Facebook has a wide reach around the world. For developers who are developing bots, this is a great platform to reach out to a bigger audience. Facebook has been investing in bot development and has provided tools for users to create bots for their specific needs without writing a single line of code. Fast food joints like Burger King has leveraged the use of bots to serve their customers by taking their order via Facebook. Many businesses have used Facebook to their advantage and improved ways of serving their customer base.

Skype for Business: This is another popular instant messaging platform utilized by many businesses around the world for their internal or external communication. Bots like Skyscanner allow you to make travel arrangements right in your Skype window. In addition, it helps you to find the most affordable travel options. Bots like Bing Image Preview and Getty Images allow you to search for images right from your Skype search bar.

Kik: It is an instant messaging platform, used for internal communication in businesses. One of the most popular bots on this platform is The Weather Channel. It forecasts the weather for you and lets you know if there is going to be any change in the weather. This is great for traveling professionals as they can plan their schedules accordingly.

There are other messaging platforms with interesting bots that have been used to make business operations smoother and easier.

How to Make a Chatbot in Python?

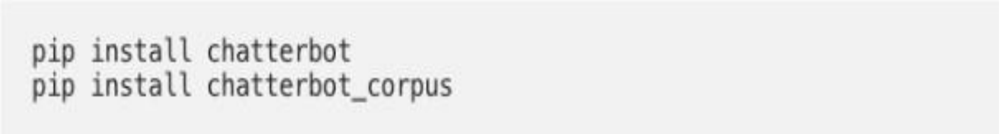
To build a chatbot in Python, you have to import all the necessary packages and initialize the variables you want to use in your chatbot project. Also, remember that when working with text data, you need to perform data preprocessing on your dataset before designing an ML model.

This is where tokenizing helps with text data – it helps fragment the large text dataset into smaller, readable chunks (like words). Once that is done, you can also go for lemmatization that transforms a word into its lemma form. Then it creates a pickle file to store the python objects that are used for predicting the responses of the bot.

Another vital part of the chatbot development process is creating the training and testing datasets.

1. **Prepare the Dependencies**

   The first step in creating a chatbot in Python with the ChatterBot library is to install the library in your system. It is best if you create and use a new Python virtual environment for the installation. To do so, you have to write and execute this command in your Python terminal:

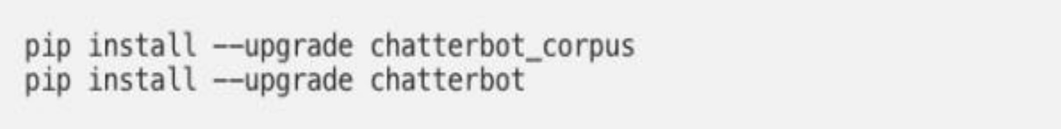


```
pip install chatterbot
pip install chatterbot_corpus
```

You can also install ChatterBot's latest development version directly from GitHub. For this, you will have to write and execute the following command:

pip install git+git://github.com/gunthercox/ChatterBot.git@master

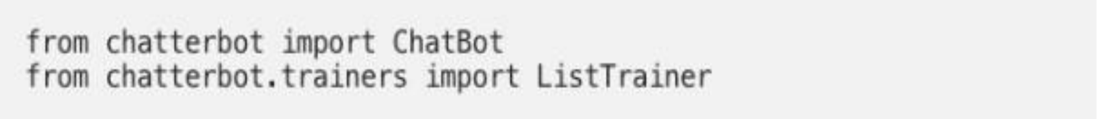
If you wish to upgrade the command, you can do so as well:



```
pip install --upgrade chatterbot_corpus
pip install --upgrade chatterbot
```

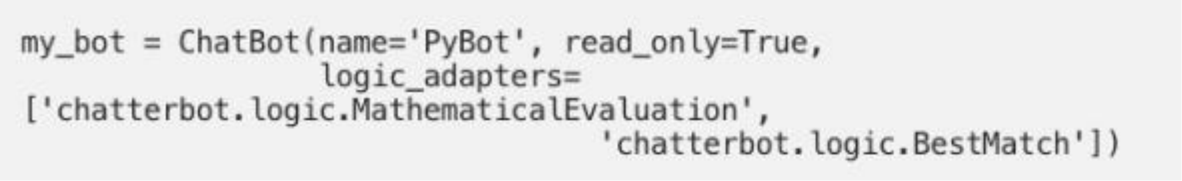
Now that your setup is ready, we can move on to the next step to create chatbot using python.

2. **Import Classes:** Importing classes is the second step in the Python chatbot creation process. All you need to do is import two classes – ChatBot from chatterbot and ListTrainer from chatterbot.trainers. To do this, you can execute the following command:



```
from chatterbot import ChatBot
from chatterbot.trainers import ListTrainer
```

3. **Create and Train the Chatbot:** This is the third step on creating chatbot in python. The chatbot you are creating will be an instance of the class "ChatBot." After creating a new ChatterBot instance, you can train the bot to improve its performance. Training ensures that the bot has enough knowledge to get started with specific responses to specific inputs. You have to execute the following command now:

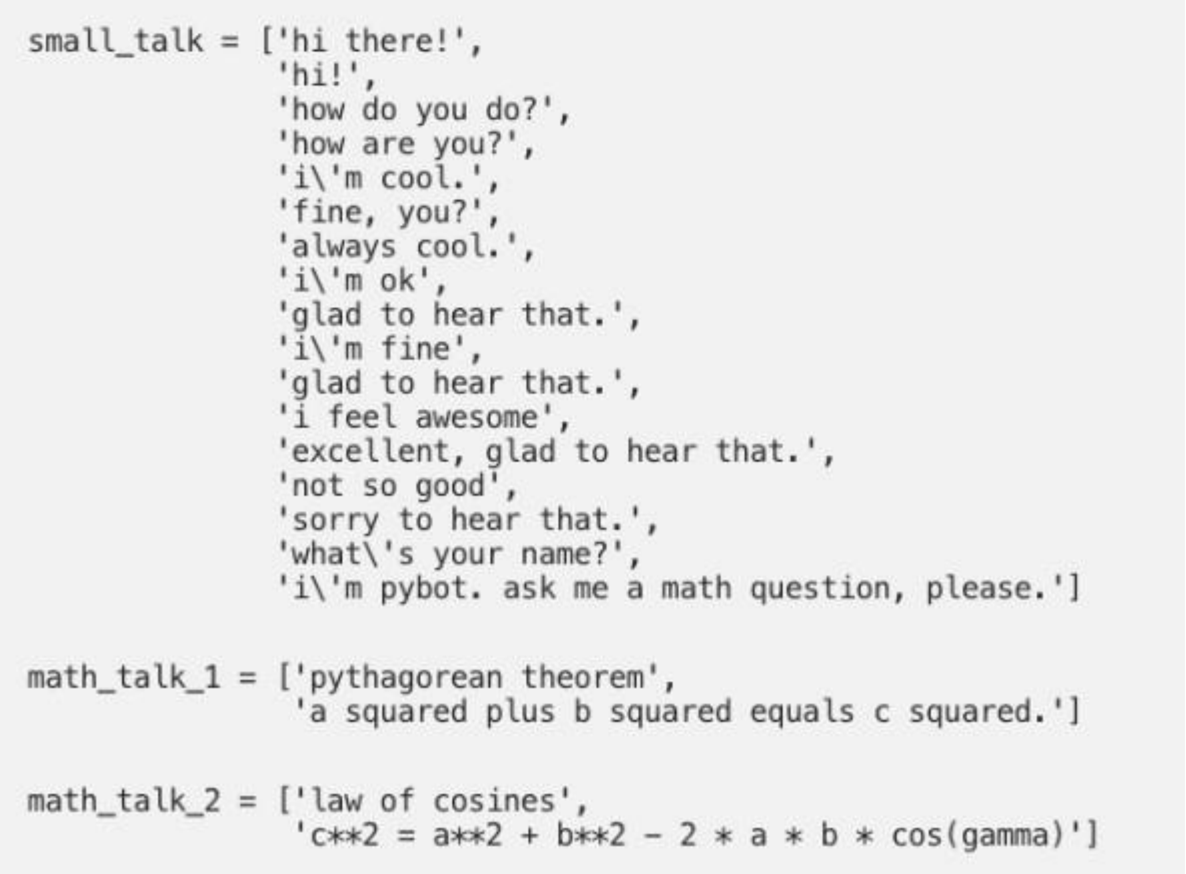


```
my_bot = ChatBot(name='PyBot', read_only=True,
                 logic_adapters=
['chatterbot.logic.MathematicalEvaluation',
                                 'chatterbot.logic.BestMatch'])
```

Here, the argument (that corresponds to the parameter name) represents the name of your Python chatbot. If you wish to disable the bot's ability to learn after the training, you can include the "read_only=True" command. The command "logic_adapters" denotes the list of adapters used to train the chatbot.

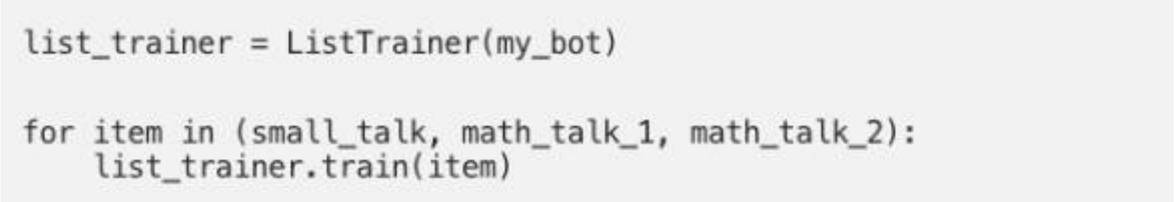
While the "chatterbot.logic.MathematicalEvaluation" helps the bot to solve math problems, the "chatterbot.logic.BestMatch" helps it to choose the best match from the list of responses already provided.

Since you have to provide a list of responses, you can do it by specifying the lists of strings that can be later used to train your Python chatbot, and find the best match for each query. Here’s an example of responses you can train your chatbot using python to learn:



```
small_talk = ['hi there!',
              'hi!',
              'how do you do?',
              'how are you?',
              'i\'m cool.',
              'fine, you?',
              'always cool.',
              'i\'m ok',
              'glad to hear that.',
              'i\'m fine',
              'glad to hear that.',
              'i feel awesome',
              'excellent, glad to hear that.',
              'not so good',
              'sorry to hear that.',
              'what\'s your name?',
              'i\'m pybot. ask me a math question, please.']

math_talk_1 = ['pythagorean theorem',
               'a squared plus b squared equals c squared.']

math_talk_2 = ['law of cosines',
               'c**2 = a**2 + b**2 - 2 * a * b * cos(gamma)']
```

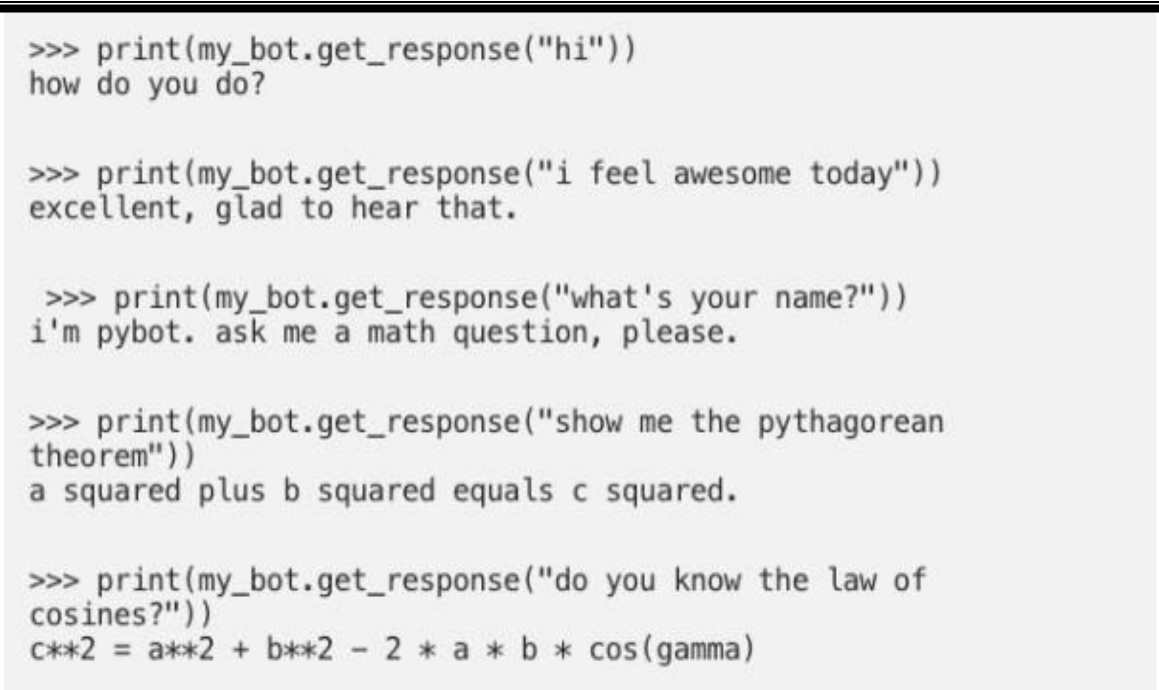
You can also create and train the bot by writing an instance of “ListTrainer” and supplying it with a list of strings like so:



```
list_trainer = ListTrainer(my_bot)

for item in (small_talk, math_talk_1, math_talk_2):
    list_trainer.train(item)
```

Now, your Python chatbot is ready to communicate.

4. **Communicate with the Python Chatbot:** To interact with your Python chatbot, you can use the .get_response() function. This is how it should look while communicating:

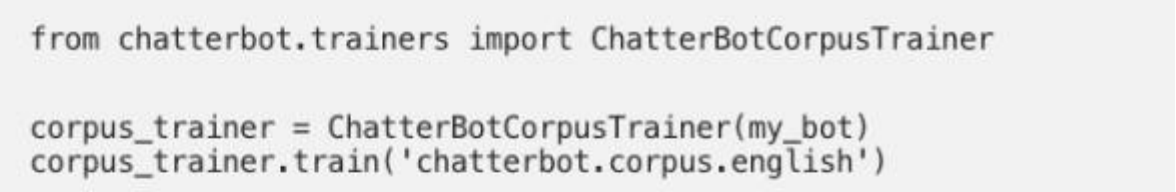


```
>>> print(my_bot.get_response("hi"))
how do you do?

>>> print(my_bot.get_response("i feel awesome today"))
excellent, glad to hear that.

 >>> print(my_bot.get_response("what's your name?"))
i'm pybot. ask me a math question, please.

>>> print(my_bot.get_response("show me the pythagorean
theorem"))
a squared plus b squared equals c squared.

>>> print(my_bot.get_response("do you know the law of
cosines?"))
c**2 = a**2 + b**2 - 2 * a * b * cos(gamma)
```

However, it is essential to understand that the chatbot using python might not know how to answer all your questions. Since its knowledge and training is still very limited, you have to give it time and provide more training data to train it further.

5 Train your Python Chatbot with a Corpus of Data:

In this last step of how to make a chatbot in Python, for training your python chatbot even further, you can use an existing corpus of data. Here’s an example of how to train your Python chatbot with a corpus of data provided by the bot itself:



```
from chatterbot.trainers import ChatterBotCorpusTrainer

corpus_trainer = ChatterBotCorpusTrainer(my_bot)
corpus_trainer.train('chatterbot.corpus.english')
```

The good thing is that ChatterBot offers this functionality in many different languages. So, you can also specify a subset of a corpus in a language you would prefer.

Conclusion: Thus we have successfully implemented elementary Chatbot for Customer interaction application.

Practical No-6

TITLE: Implement any one of the following Expert System

I. Information management

II. Hospitals and medical facilities

III. Help desks management

IV. Employee performance evaluation

V. Stock market trading

VI. Airline scheduling and cargo schedules

AIM: To create Expert System for any of above chosen topic.

OBJECTIVES: Based on above main aim following are the objectives

1. To study theory of Expert System
2. To create Expert System

Theory:

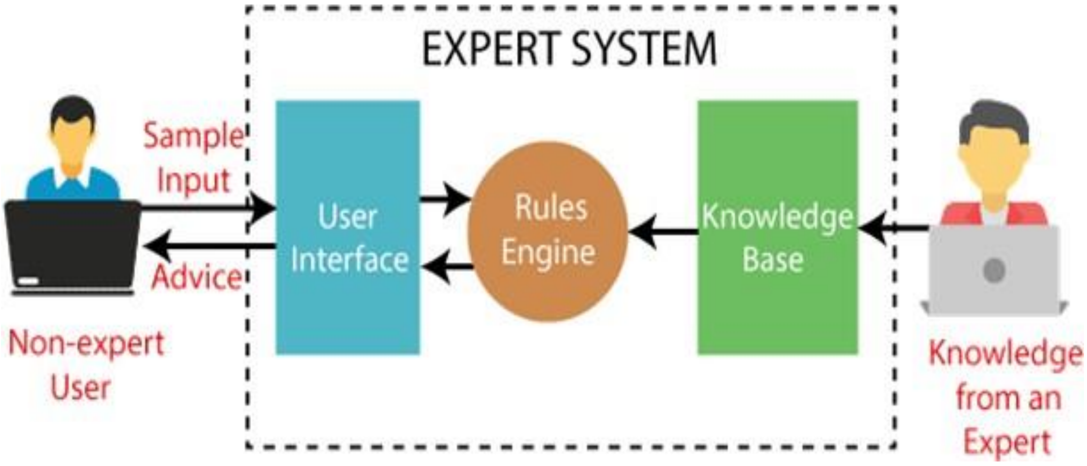
Expert System:

An expert system is a computer program that is designed to solve complex problems and to provide decision-making ability like a human expert. It performs this by extracting knowledge from its knowledge base using the reasoning and inference rules according to the user queries.

The expert system is a part of AI, and the first ES was developed in the year 1970, which was the first successful approach of artificial intelligence. It solves the most complex issue as an expert by extracting the knowledge stored in its knowledge base. The system helps in decision making for complex problems using both facts and heuristics like a human expert. It is called so because it contains the expert knowledge of a specific domain and can solve any complex problem of that particular domain. These systems are designed for a specific domain, such as medicine, science, etc.

The performance of an expert system is based on the expert's knowledge stored in its knowledge base. The more knowledge stored in the KB, the more that system improves its performance. One of the common examples of an ES is a suggestion of spelling errors while typing in the Google search box.

Below is the block diagram that represents the working of an expert system:



Below are some popular examples of the Expert System:

DENDRAL: It was an artificial intelligence project that was made as a chemical analysis expert system. It was used in organic chemistry to detect unknown organic molecules with the help of their mass spectra and knowledge base of chemistry.

MYCIN: It was one of the earliest backward chaining expert systems that was designed to find the bacteria causing infections like bacteraemia and meningitis. It was also used for the recommendation of antibiotics and the diagnosis of blood clotting diseases.

PXDES: It is an expert system that is used to determine the type and level of lung cancer. To determine the disease, it takes a picture from the upper body, which looks like the shadow. This shadow identifies the type and degree of harm.

CaDeT: The CaDet expert system is a diagnostic support system that can detect cancer at early stages.

Characteristics of Expert System:

High Performance: The expert system provides high performance for solving any type of complex problem of a specific domain with high efficiency and accuracy.

Understandable: It responds in a way that can be easily understandable by the user. It can take input in human language and provides the output in the same way.

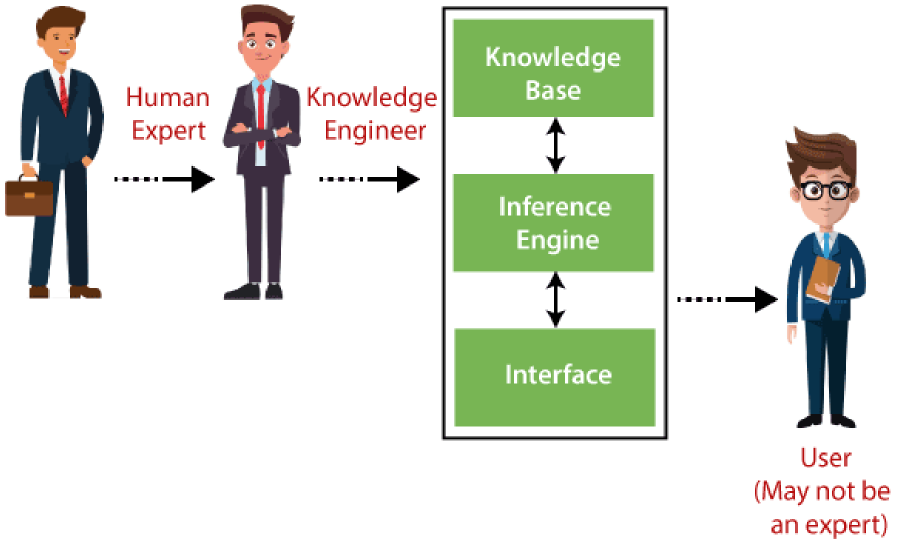
Reliable: It is much reliable for generating an efficient and accurate output.

Highly responsive: ES provides the result for any complex query within a very short period of time.

Components of Expert System

An expert system mainly consists of three components:

- User Interface
- Inference Engine
- Knowledge Base



1. User Interface

With the help of a user interface, the expert system interacts with the user, takes queries as an input in a readable format, and passes it to the inference engine. After getting the response from the inference engine, it displays the output to the user. In other words, it is an interface that helps a non-expert user to communicate with the expert system to find a solution.

2. Inference Engine (Rules of Engine)

- The inference engine is known as the brain of the expert system as it is the main processing unit of the system. It applies inference rules to the knowledge base to derive a conclusion or deduce new information. It helps in deriving an error-free solution of queries asked by the user.
- With the help of an inference engine, the system extracts the knowledge from the knowledge base.
- There are two types of inference engine:

- **Deterministic Inference engine:** The conclusions drawn from this type of inference engine are assumed to be true. It is based on facts and rules.
- **Probabilistic Inference engine:** This type of inference engine contains uncertainty in conclusions, and based on the probability.

Inference engine uses the below modes to derive the solutions:

- **Forward Chaining:** It starts from the known facts and rules, and applies the inference rules to add their conclusion to the known facts.
- **Backward Chaining:** It is a backward reasoning method that starts from the goal and works backward to prove the known facts.

3. Knowledge Base

The knowledgebase is a type of storage that stores knowledge acquired from the different experts of the particular domain. It is considered as big storage of knowledge. The more the knowledge base, the more precise will be the Expert System.

It is similar to a database that contains information and rules of a particular domain or subject. One can also view the knowledge base as collections of objects and their attributes. Such as a Lion is an object and its attributes are it is a mammal, it is not a domestic animal, etc.

Explain your implemented Expert System: