EX.NO:1 **WORKING WITH PIG**

Aim:

Algorithm:

Step 1: Install Apache Pig on your system or use a pre-configured Hadoop environment.

Step 2: Create sample datasets in a structured format such as CSV or tab-delimited files.

Step 3: Use the LOAD statement to read data into a Pig relation

Step 4: Specify the delimiter and schema.

Step 5: Sort and group the data based on a specific field

Step 6: Join two datasets on a common field

Step 7: Select specific columns from the dataset

Step 8: Filter the data based on a condition

Program:

1. Sorting Data

In Pig, you can sort data using the ORDER BY clause. By default, the ORDER BY operation sorts the data in ascending order. To sort data in descending order, you can use the DESC keyword.

Example: Sorting data by a column

Load the data from a file

```
data = LOAD 'data.txt' USING PigStorage(',') AS (id:int, name:chararray, age:int);

Sort the data by the 'age' column in ascending order
sorted_data = ORDER data BY age ASC;

Store the sorted data to a file
STORE sorted_data INTO 'sorted_data.txt' USING PigStorage(',');
```

Grouping data is typically done using the GROUP BY clause. This allows you to group the data based on a certain field.

Example: Grouping data by a column

Load the data from a file

```
data = LOAD 'data.txt' USING PigStorage(',') AS (id:int, name:chararray, age:int, department:chararray);

Group the data by the 'department' column
```

```
grouped_data = GROUP data BY department;

Store the grouped data to a file
STORE grouped_data INTO 'grouped_data.txt' USING PigStorage(',');
```

3. Joining Data

Pig supports inner and outer joins. To perform a join, you use the JOIN keyword. Here's how you can join two datasets on a common field.

Example: Joining two datasets on the 'id' column

```
Load the data from two files
data1 = LOAD 'data1.txt' USING PigStorage(',') AS (id:int, name:chararray);
data2 = LOAD 'data2.txt' USING PigStorage(',') AS (id:int, age:int);

    Join the two datasets on the 'id' column
joined_data = JOIN data1 BY id, data2 BY id;

Store the joined data to a file
STORE joined_data INTO 'joined_data.txt' USING PigStorage(',');
```

4. Projecting Data (Selecting Specific Columns)

To select specific columns from a dataset, you can use the FOREACH and GENERATE statements. This is commonly referred to as "projection".

Example: Projecting specific columns

```
Load the data from a file
data = LOAD 'data.txt' USING PigStorage(',') AS (id:int, name:chararray, age:int, department:chararray);

    Project only 'name' and 'age' columns
projected_data = FOREACH data GENERATE name, age;

    Store the projected data to a file
STORE projected_data INTO 'projected_data.txt' USING PigStorage(',');
```

5. Filtering Data

Filtering data is done using the FILTER statement. You can apply any condition to filter out data you don't need.

Example: Filtering data by a condition

Load the data from a file

```
data = LOAD 'data.txt' USING PigStorage(',') AS (id:int, name:chararray, age:int, department:chararray);
```

Filter the data to include only records where age > 30

```
filtered_data = FILTER data BY age > 30;
```

Store the filtered data to a file

```
STORE filtered_data INTO 'filtered_data.txt' USING PigStorage(',');
```

Result :

EX.NO:2 **WORKING WITH HIVE**

Aim:

Description:

Apache Hive is a data warehouse infrastructure that facilitates querying and managing large data sets which resides in distributed storage system. It is built on top of Hadoop and developed by Facebook. Hive provides a way to query the data using a SQL-like query language called HiveQL(Hive query Language).

Internally, a compiler translates HiveQL statements into MapReduce jobs, which are then submitted to Hadoop framework for execution.

Hive looks very much similar like traditional database with SQL access. However, because Hive is based on Hadoop and MapReduce operations, there are several key differences:

As Hadoop is intended for long sequential scans and Hive is based on Hadoop, you would expect queries to have a very high latency. It means that Hive would not be appropriate for those applications that need very fast response times, as you can expect with a traditional RDBMS database.

User Interface:

Hive is a data warehouse infrastructure software that can create interaction between user and HDFS. The user interfaces that Hive supports are Hive Web UI, Hive command line, and Hive HD Insight (In Windows server).

Meta Store:

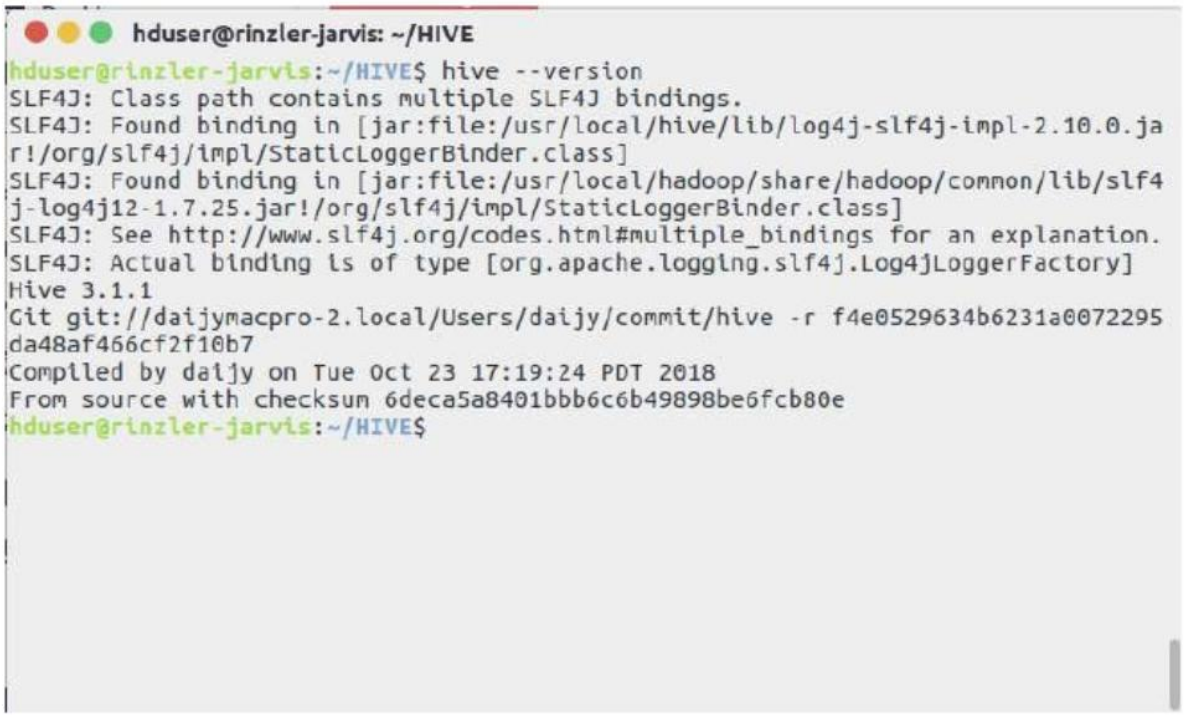
Hive chooses respective database servers to store the schema or Metadata of tables, databases, columns in a table, their data types, and HDFS mapping.

HiveQL Process Engine:

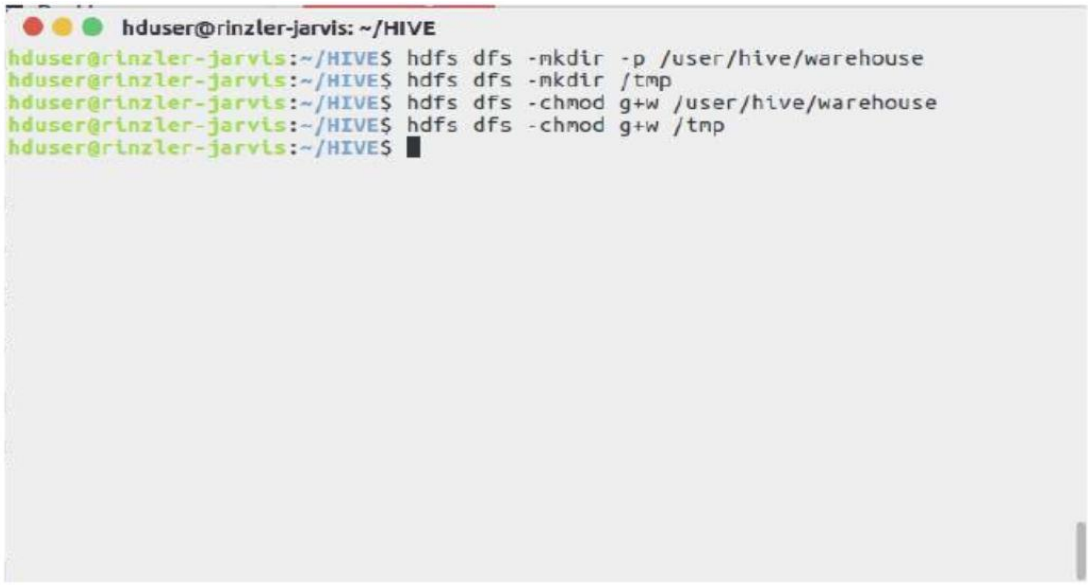
HiveQL is similar to SQL for querying on schema info on the Metastore. It is one of the replacements of traditional approach for MapReduce program. Instead of writing MapReduce program in Java, we can write a query for MapReduce job and process it.

Installing Pig

1. Download the Hive Files from Apache
2. Extract the files to a convenient location. (/usr/local).
3. Edit the system variable to include the Pig files.
4. Check Pig version to check if its working properly



5. Create Hive directories within HDFS and give them read/write permissions. The directory 'warehouse' is the location to store the table or data related to hive.



6. Set Hadoop path in hive-env.sh

Result:

EX.NO:3 BIG DATA STREAM PROCESSING

Aim:

Tools and Technologies:

- **Hadoop Distributed File System (HDFS)** or **Amazon Redshift** (or any cloud data warehouse)
- **Apache Hive** or **Amazon Athena** for querying
- **SQL** for data manipulation and analysis

Pre-requisites:

- Install and configure Hadoop or Redshift.
- Install Hive or any SQL interface.
- Sample dataset (e.g., sales data in CSV format).

Step 1: Create a Database in the Data Warehouse

1. **Open Hive/SQL client:**
   Launch Hive shell or SQL Workbench (for Redshift).
2. **Create a new database:**
   CREATE DATABASE sales_db;
   USE sales_db;
3. **Verify the database:**
   SHOW DATABASES;

Step 2: Create and Load Tables

1. **Create a table for sales data:**

```
CREATE TABLE sales_data (
    sale_id INT,
    product_name STRING,
    category STRING,
    price FLOAT,
    quantity_sold INT,
    sale_date DATE
)
```

```
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;
```

2. **Load data into the table:**

   Assume sales_data.csv is stored in HDFS or S3.

```
LOAD DATA INPATH '/user/hadoop/sales_data.csv' INTO TABLE sales_data;
```

3. **Verify data loading:**

```
SELECT * FROM sales_data LIMIT 10;
```

Step 3: Distribute Data Across Nodes

1. **Enable partitioning (optional for large datasets):**

```
CREATE TABLE sales_data_partitioned (
    sale_id INT,
    product_name STRING,
    category STRING,
    price FLOAT,
    quantity_sold INT
)
PARTITIONED BY (sale_date DATE)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;
```

2. **Insert data into partitioned table:**

```
sql
CopyEdit
INSERT OVERWRITE TABLE sales_data_partitioned
PARTITION (sale_date)
SELECT sale_id, product_name, category, price, quantity_sold, sale_date FROM sales_data;
```

Step 4: Analyze Data in the Data Warehouse

1. **Total Sales by Category:**

```
SELECT category, SUM(price * quantity_sold) AS total_sales
FROM sales_data
GROUP BY category;
```

2. **Top 5 Best-Selling Products:**

```
SELECT product_name, SUM(quantity_sold) AS total_units
```

```
FROM sales_data
GROUP BY product_name
ORDER BY total_units DESC
LIMIT 5;
```

3. **Monthly Sales Trend:**

```
SELECT MONTH(sale_date) AS month, SUM(price * quantity_sold) AS monthly_sales
FROM sales_data
GROUP BY MONTH(sale_date)
ORDER BY month;
```

Step 5: Clean Up

1. **Drop the table if needed:**

```
DROP TABLE sales_data;
DROP TABLE sales_data_partitioned;
```

2. **Drop the database:**

```
DROP DATABASE sales_db CASCADE;
```

Aim:

- To import data into a visualization tool.
- To create interactive visualizations and dashboards.
- To analyze insights from large datasets using visualization tools.

Tools Required:

- **Google BigQuery/Big Sheets** (for Google Cloud users)
- **Microsoft Power BI** (Desktop or Cloud)
- **Tableau Public/Desktop** (Free or Licensed version)
- **Sample Dataset** (e.g., Global Superstore, Sales Data CSV)

Pre-requisites:

- Install **Power BI** or **Tableau** on your system.
- Set up a **Google Cloud** account for **Big Sheets**.
- Download the sample dataset.

A. Using Google Big Sheets (via Google BigQuery BI Engine)

1. **Login to Google Cloud Platform (GCP):**
    - Go to Google Cloud Console.
    - Enable **BigQuery API**.
2. **Upload Dataset:**
    - Open **BigQuery** > Create a new dataset.
    - Click **Create Table** > Upload sales_data.csv.

- Configure schema automatically.

3. **Connect BigQuery to Big Sheets:**
   - Enable **Connected Sheets** in Google Sheets.
   - In Google Sheets, go to **Data** → **Connect to BigQuery**.
   - Select your project and dataset.
4. **Create Visualizations:**
   - Use **Explore** in Google Sheets to generate charts.
   - Create **Pivot Tables** and **Bar/Line/Geo Charts** for analysis.
5. **Analysis:**
   - Visualize **sales trends** over time.
   - Plot **sales by region** using a Geo Chart.

B. Using Microsoft Power BI

1. **Open Power BI Desktop:**
   - Download from Microsoft Power BI.
2. **Import Dataset:**
   - Click **Home** → **Get Data** → **Text/CSV**.
   - Select the sales_data.csv file.
3. **Data Transformation (Optional):**
   - Use **Power Query Editor** to clean and shape the data.
4. **Create Visualizations:**
   - **Bar Chart:** Total Sales by Category.
   - **Pie Chart:** Sales Distribution by Region.
   - **Line Chart:** Monthly Sales Trend.
   - **Map Visualization:** Sales by Country.
5. **Interactive Dashboard:**
   - Add **slicers** for filtering by **year**, **category**, or **region**.
   - Use **drill-down** features for deeper insights.
6. **Publish Report:**
   - Click **Publish** to upload to **Power BI Service**.

C. Using Tableau

1. **Open Tableau Desktop/Tableau Public:**
   - Download from Tableau.
2. **Connect to Data:**
   - Click **Connect** → **Text File**.
   - Load sales_data.csv.

3. **Data Preparation:**
    - Drag and drop fields to explore data relationships.
4. **Create Visualizations:**
    - **Bar Chart:** Product Sales by Category.
    - **Heat Map:** Sales Density by Region.
    - **Line Graph:** Sales Trend over Time.
    - **Tree Map:** Contribution of each product to total sales.
5. **Dashboard Design:**
    - Combine multiple visualizations into one **interactive dashboard**.
    - Add filters for dynamic views.
6. **Publish to Tableau Public:**
    - Click **File → Save to Tableau Public** for sharing.

Result:

EX.NO:4 **Hadoop Installation and Configuration**

AIM:

PROCEDURE:

Step 1: Click here to download the Java 8 Package. Save this file in your home directory.

Step 2: Extract the Java Tar File.

***Command*:** tar -xvf jdk-8u101-linux-i586.tar.gz

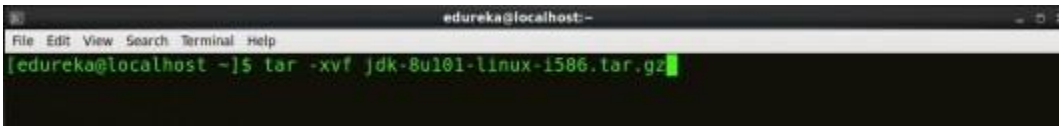


Fig: Hadoop Installation – Extracting Java Files

Step 3: Download the Hadoop 2.7.3 Package.

***Command*:** wget https://archive.apache.org/dist/hadoop/core/hadoop-2.7.3/hadoop- 2.7.3.tar.gz

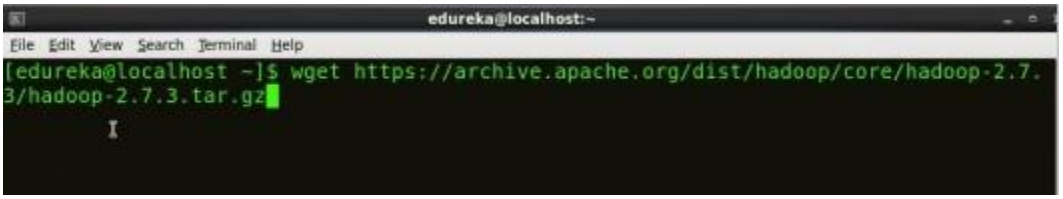


Fig: Hadoop Installation – Downloading Hadoo

Step 4: Extract the Hadoop tar File.

Command: tar -xvf hadoop-2.7.3.tar.gz

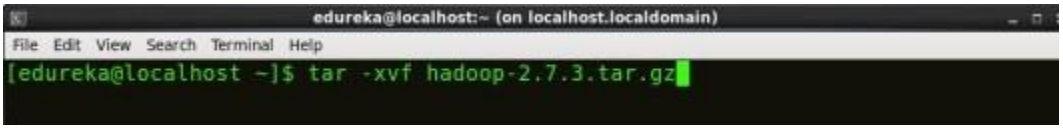
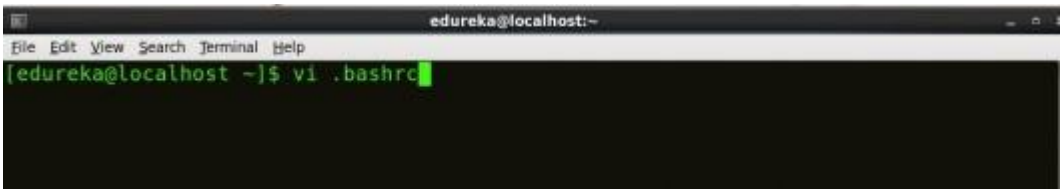


Fig: Hadoop Installation – Extracting Hadoop Files

Step 5: Add the Hadoop and Java paths in the bash file (.bashrc). Open**.** **bashrc** file. Now, add Hadoop and Java Path as shown below.

Command**:** vi .bashrc



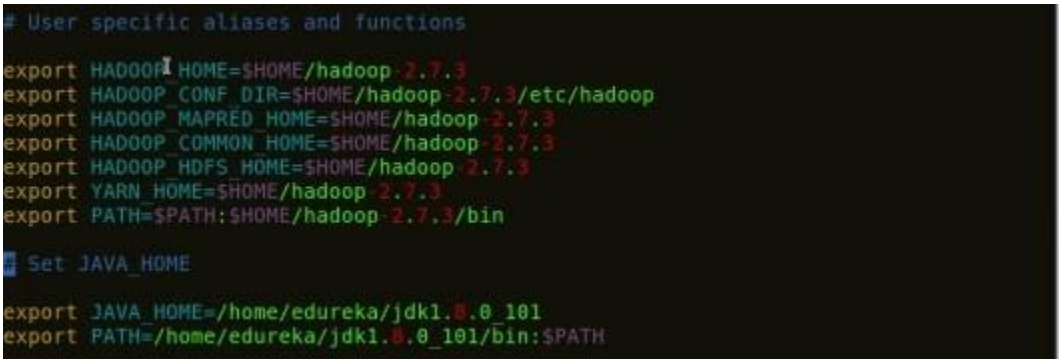


Fig: Hadoop Installation – Setting Environment Variable

Then, save the bash file and close it.

For applying all these changes to the current Terminal, execute the source command.

Command**:** source .bashrc

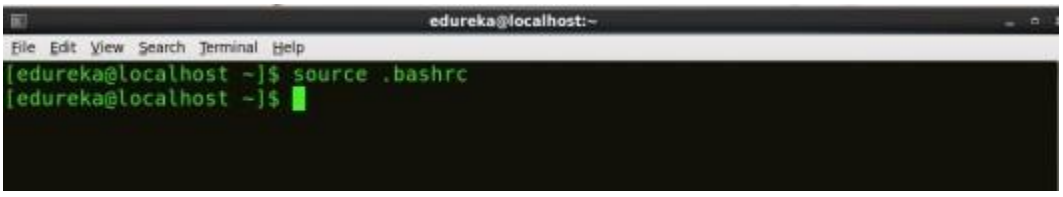


Fig: Hadoop Installation – Refreshing environment variables

To make sure that Java and Hadoop have been properly installed on your system and can be accessed through the Terminal, execute the java -version and hadoop version commands.

***Command*:** java -version

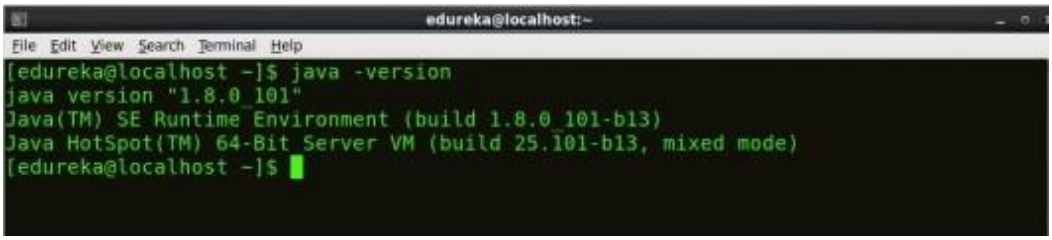


Fig: Hadoop Installation – Checking Java Version

***Command*:** hadoop version

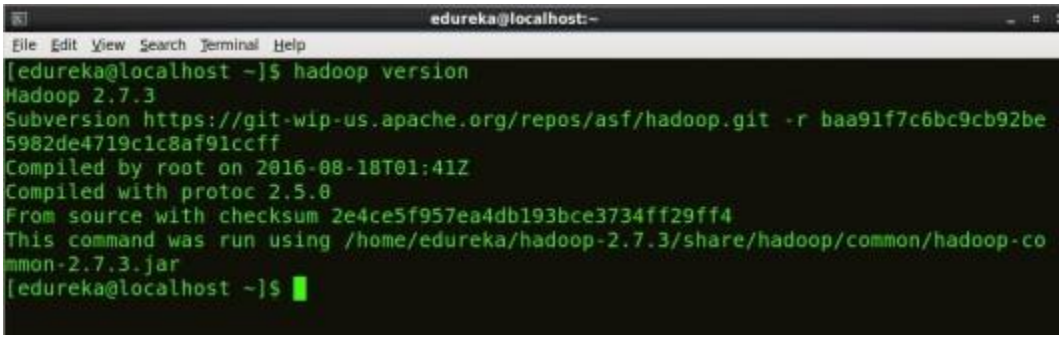


Fig: Hadoop Installation – Checking Hadoop Version

Step 6: Edit the **Hadoop Configuration files**.

Command: cd hadoop-2.7.3/etc/hadoop/

Command: ls

All the Hadoop configuration files are located in **hadoop-2.7.3/etc/hadoop** directory as you

can see in the snapshot below:

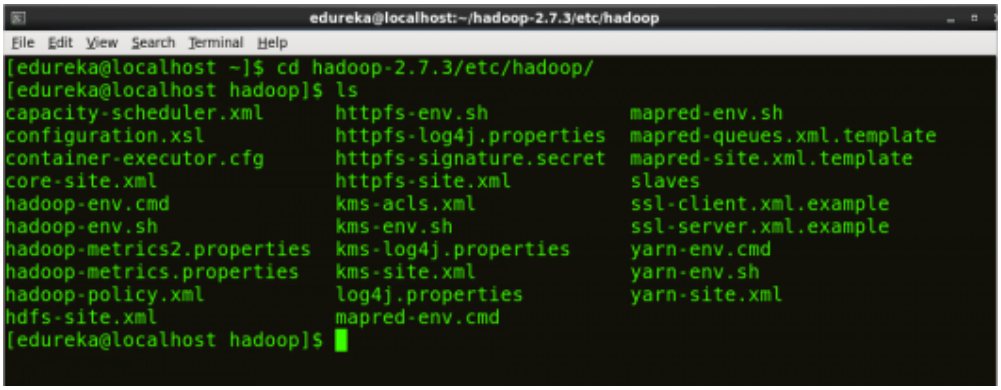


Fig: Hadoop Installation – Hadoop Configuration Files

Step 7: Open *core-site.xml* and edit the property mentioned below inside configuration tag:

core-site.xml informs Hadoop daemon where NameNode runs in the cluster. It contains configuration settings of Hadoop core such as I/O settings that are common to HDFS & MapReduce.

***Command*:** vi core-site.xml

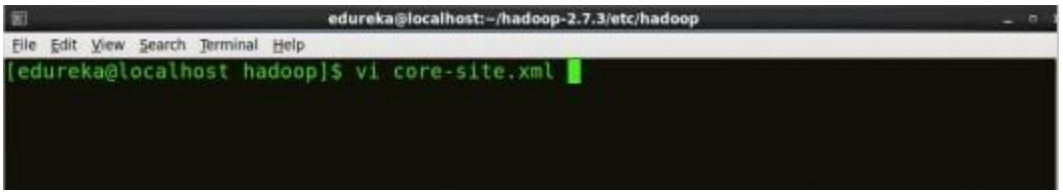


Fig: Hadoop Installation – Configuring core-site.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:9000</value>
</property>
</configuration>
```

Step 8: Edit *hdfs-site.xml* and edit the property mentioned below inside **configuration tag:**

hdfs-site.xml contains configuration settings of HDFS daemons (i.e. NameNode, DataNode, Secondary NameNode). It also includes the replication factor and block size of HDFS.

***Command*:** vi hdfs-site.xml

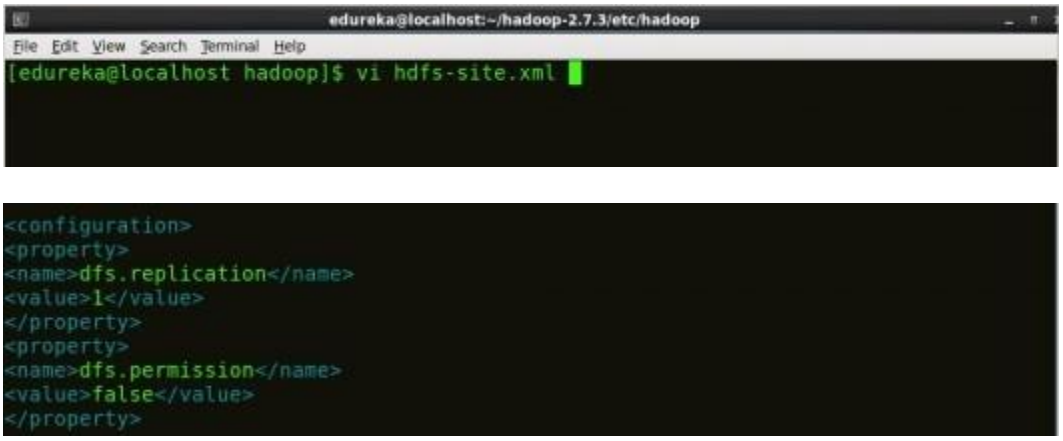


Fig: Hadoop Installation – Configuring hdfs-site.xml

```
1
2           <?xml version="1.0" encoding="UTF-8"?>
3  <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
4                      <configuration>
5                        <property>
6                 <name>dfs.replication</name>
7                       <value>1</value>
8                         </property>
9                         <property>
10                 <name>dfs.permission</name>
11                    <value>false</value>
12                       </property>
                      </configuration>
```

Step 9: Edit the *mapred-site.xml* file and edit the property mentioned below **inside configuration tag:**

mapred-site.xml contains configuration settings of MapReduce application like number of JVM that can run in parallel, the size of the mapper and the reducer process, CPU cores available for a process, etc.

In some cases, mapred-site.xml file is not available. So, we have to create the mapred- site.xml file using mapred-site.xml template.

***Command*:** cp mapred-site.xml.template mapred-site.xml

***Command*:** vi mapred-site.xml.

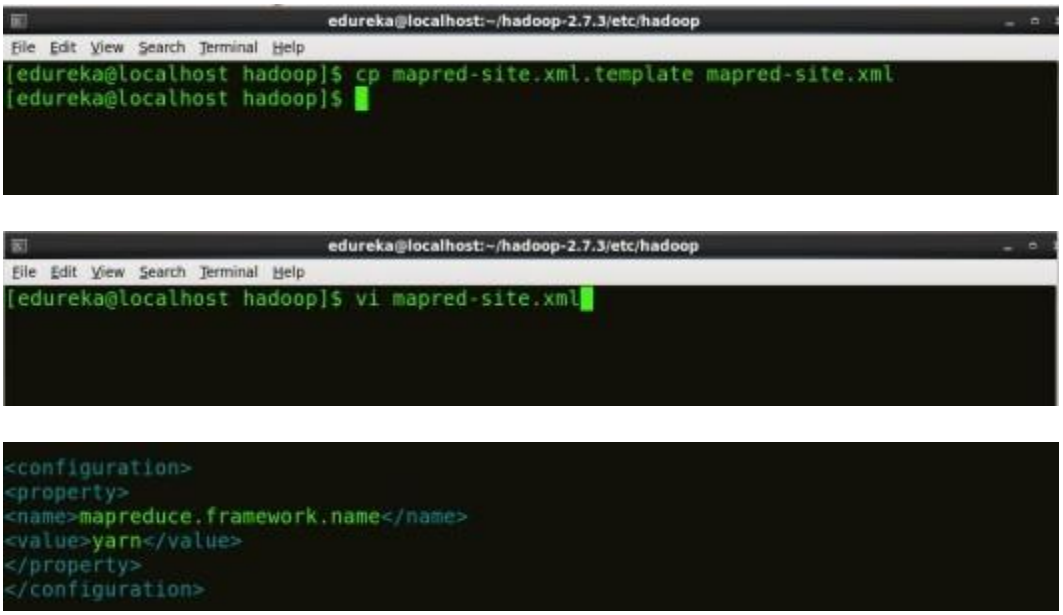


Fig: Hadoop Installation – Configuring mapred-site.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
</configuration>
```

Step 10: Edit *yarn-site.xml* and edit the property mentioned below inside configuration tag: *yarn-site.xml* contains configuration settings of ResourceManager and NodeManager like application memory management size, the operation needed on program & algorithm, etc.

***Command*:** vi yarn-site.xml

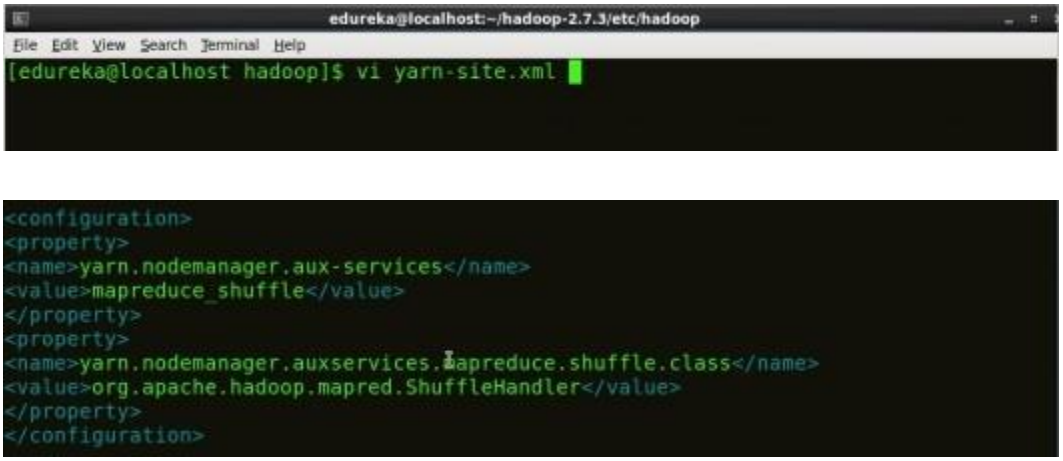


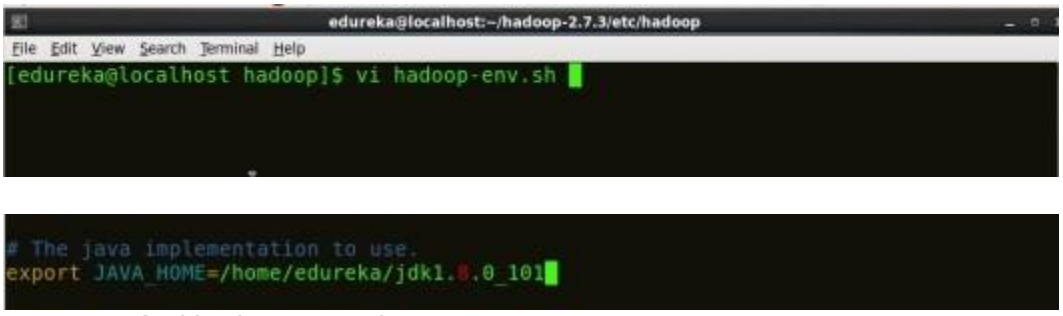
Fig: Hadoop Installation – Configuring yarn-site.xml

Step 11: Edit *hadoop-env.sh* and add the Java Path as mentioned below:

hadoop-env.sh contains the environment variables that are used in the script to run

```
<?xml version="1.0">
  <configuration>
    <property>
  <name>yarn.nodemanager.aux-
    services</name>
<namevarn.nodemanager.auxservices.mapreduce.shuff
                    name>
   <value>org.apache.hadoop.mapred.ShuffleHandl
                    er</value>
```

Hadoop like Java home path, etc.



***Command*:** vi hadoop–env.sh

Fig: Hadoop Installation – Configuring hadoop-env.sh

Step 12: Go to Hadoop home directory and format the NameNode. ***Command*:** cd
***Command*:** cd hadoop-2.7.3

***Command*:** bin/hadoop namenode -format

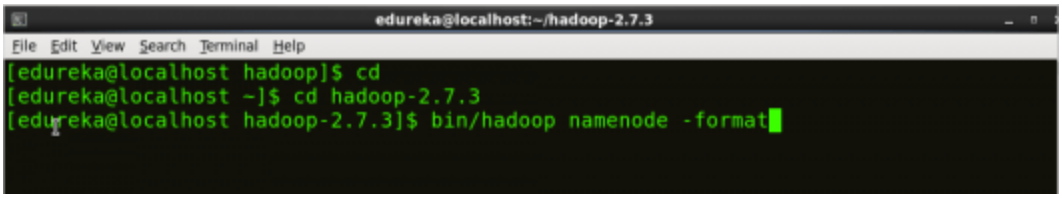


Fig: Hadoop Installation – Formatting NameNode

This formats the HDFS via NameNode. This command is only executed for the first time. Formatting the file system means initializing the directory specified by the dfs.name.dir variable.

Never format, up and running Hadoop filesystem. You will lose all your data stored in the HDFS.

Step 13: Once the NameNode is formatted, go to hadoop-2.7.3/sbin directory and start all the daemons.

Command: cd hadoop-2.7.3/sbin
Either you can start all daemons with a single command or do it individually.

Command: ./start-all.sh

The above command is a combination of ***start-dfs.sh, start-yarn.sh*** & ***mr-jobhistory-daemon.sh***

Or you can run all the services individually as below:

Start NameNode:

The NameNode is the centerpiece of an HDFS file system. It keeps the directory tree of all files

stored in the HDFS and tracks all the file stored across the cluster.

Command: ./hadoop-daemon.sh start namenode

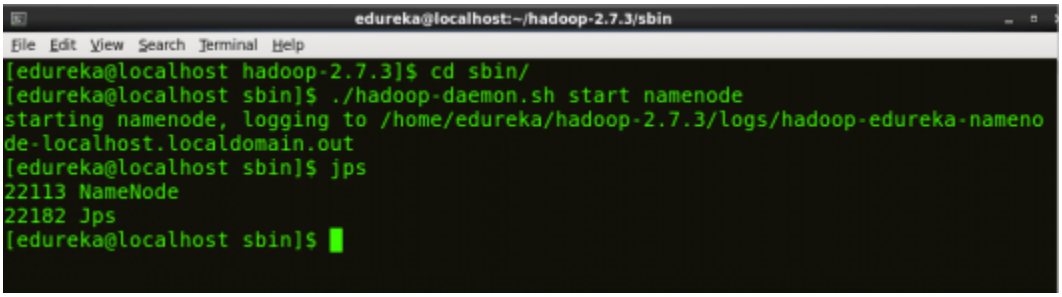


Fig: Hadoop Installation – Starting NameNode

Start DataNode:

On startup, a DataNode connects to the Namenode and it responds to the requests from the Namenode for different operations.

Command: ./hadoop-daemon.sh start datanode

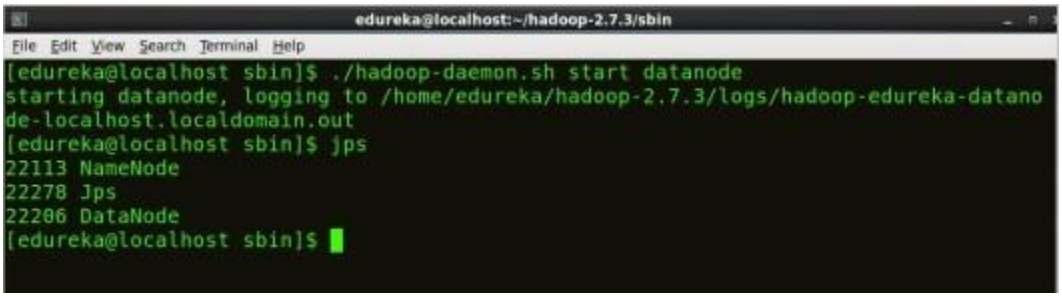


Fig: Hadoop Installation – Starting DataNode

Start ResourceManager:

ResourceManager is the master that arbitrates all the available cluster resources and thus helps in managing the distributed applications running on the YARN system. Its work is to manage each NodeManagers and the each application's ApplicationMaster.

Command: ./yarn-daemon.sh start resourcemanager

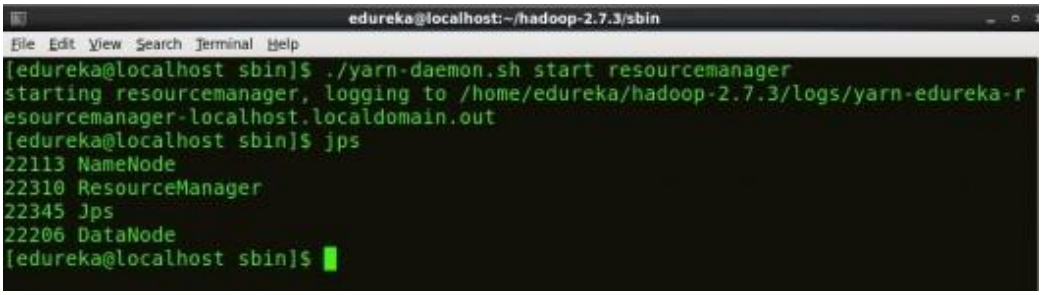


Fig: Hadoop Installation – Starting ResourceManager

Start NodeManager:

The NodeManager in each machine framework is the agent which is responsible for managing containers, monitoring their resource usage and reporting the same to the ResourceManager.

Command: ./yarn-daemon.sh start nodemanager

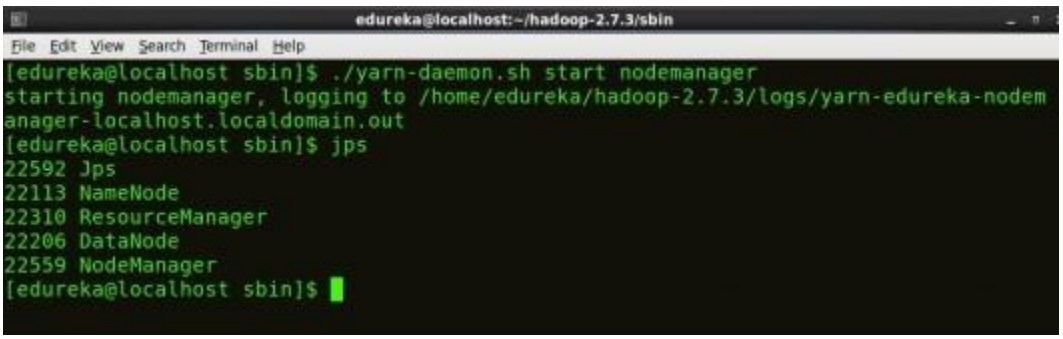


Fig: Hadoop Installation – Starting NodeManager

Start JobHistoryServer:

JobHistoryServer is responsible for servicing all job history related requests from client.

***Command*:** ./mr-jobhistory-daemon.sh start historyserver

Step 14: To check that all the Hadoop services are up and running, run the below command.

Command: jps

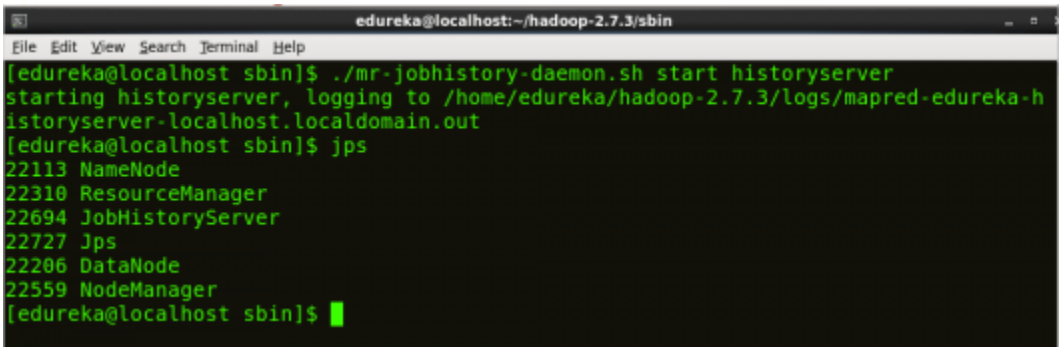


Fig: Hadoop Installation – Checking Daemons

Step 15: Now open the Mozilla browser and go to **localhost**:**50070/dfshealth.html** to check the NameNode interface.

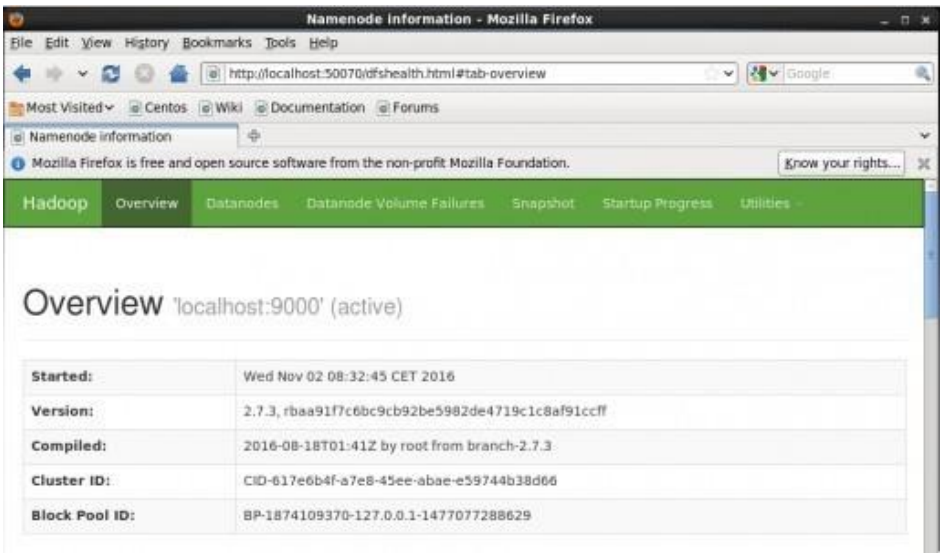


Fig: Hadoop Installation – Starting WebUI

Congratulations, you have successfully installed a single node Hadoop cluster

RESULT:

EX NO: 5 **FILE MANAGEMENT IN HADOOP**

AIM :

PROCEDURE:

Hint: A typical Hadoop workflow creates data files (such as log files) elsewhere and copies them intoHDFS using one of the above command line utilities.

RESOURCES:

VMWare stack, 4 GB RAM, Hard Disk 80 GB.

PROGRAM LOGIC:

Adding Files and Directories to HDFS

Before you can run Hadoop programs on data stored in HDFS, you„ll need to put the data into HDFS first. Let„s create a directory and put a file in it. HDFS has a default working directory of

/user/$USER, where $USER is your login user name. This directory isn„t automatically created for you, though, so let„s create it with the mkdir command. For the purpose of illustration, we use chuck. You should substitute your user name in the example commands.

hadoop fs -mkdir /user/chuck

hadoop fs -put

hadoop fs -put example.txt /user/chuck

Retrieving Files from HDFS

The Hadoop command get copies files from HDFS back to the local filesystem. To retrieve example.txt,we can run the following command:

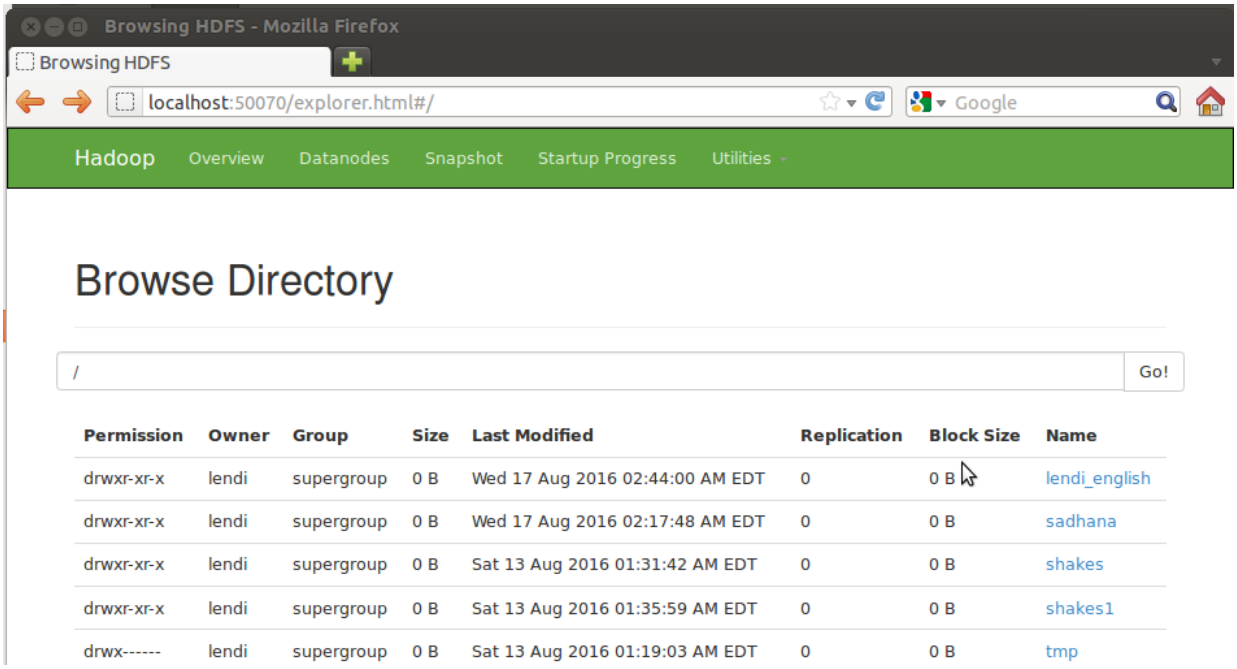
hadoop fs -cat example.txt

Deleting files from HDFS

hadoop fs -rm example.txt

- Command for creating a directory in hdfs is “hdfs dfs –mkdir /lendicse”.

  Adding directory is done through the command “hdfs dfs –put lendi_english /”.INPUT/OUTPUT:



Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	lendi	supergroup	0 B	Wed 17 Aug 2016 02:44:00 AM EDT	0	0 B	lendi_english
drwxr-xr-x	lendi	supergroup	0 B	Wed 17 Aug 2016 02:17:48 AM EDT	0	0 B	sadhana
drwxr-xr-x	lendi	supergroup	0 B	Sat 13 Aug 2016 01:31:42 AM EDT	0	0 B	shakes
drwxr-xr-x	lendi	supergroup	0 B	Sat 13 Aug 2016 01:35:59 AM EDT	0	0 B	shakes1
drwx------	lendi	supergroup	0 B	Sat 13 Aug 2016 01:19:03 AM EDT	0	0 B	tmp

RESULT:

EX.No:6 **Install Virtual box/VMware Workstation**

AIM:

PROCEDURE:

Step 1- Download Link

Link for downloading the software is https://www.vmware.com/products/workstation-pro/workstation-pro-evaluation.html. Download the software for windows. Good thing is that there is no signupprocess. Click and download begins. Software is around 541 MB.

Step 2- Download the installer file

It should probably be in the download folder by default, if you have not changed the settings in your browser. File name should be something like VMware-workstation-full-15.5.1-15018445.exe. This file name can change depending on the version of the software currently available for download. But for now, till the next version is available, they will all be VMware Workstation 15 Pro.

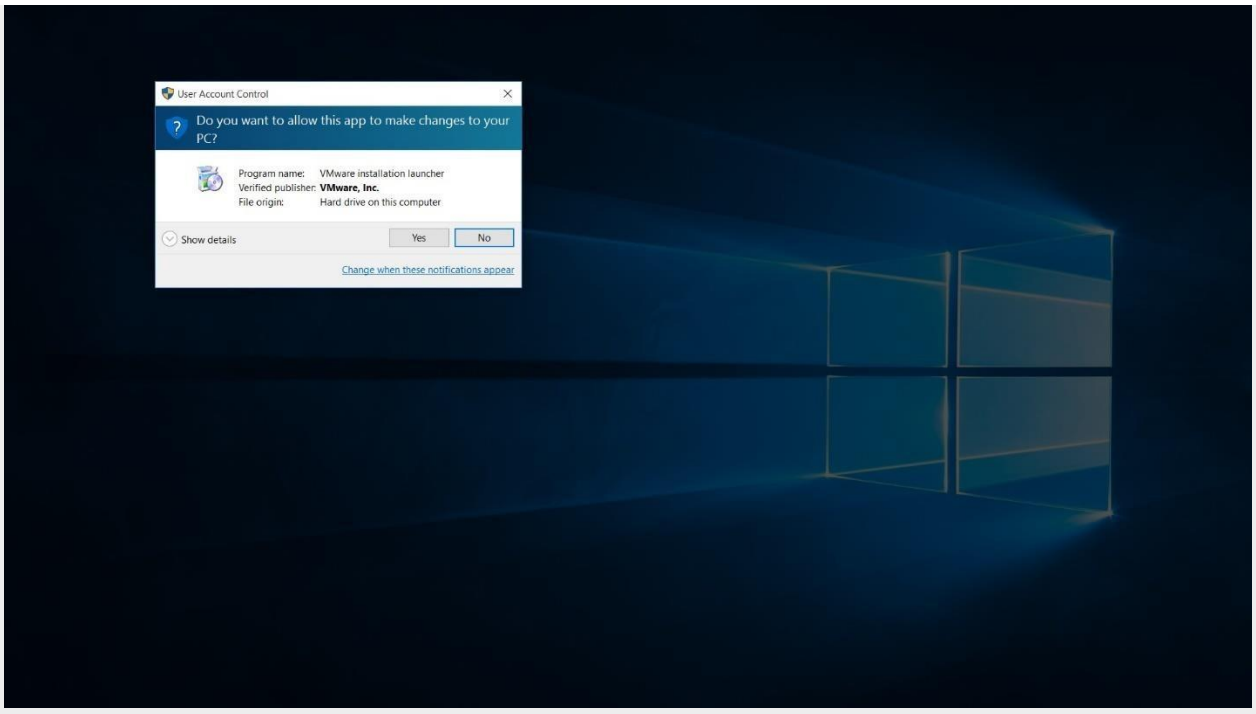
Step 3- Locate the downloaded installer file

For demonstration purpose, I have placed the downloaded installer on my desktop. Find the installer onyour system and double click to launch the application.

VMware workstation 15 pro for windows 10 installer file screenshot.

Step 4- User Access Control (UAC) Warning

Now you should see User Access Control (UAC) dialog box. Click yes to continue.



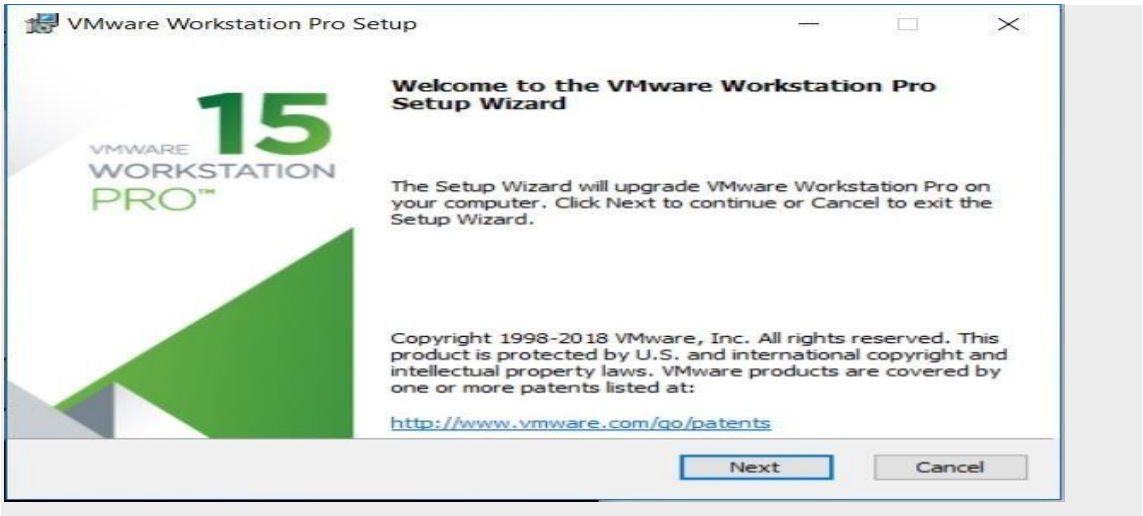
VMware Workstation 12 Pro installer windows 10 UAC screenshot

Initial Splash screen will appear. Wait for the process to complete.

VMware Workstation 15 Installation Splash Screen

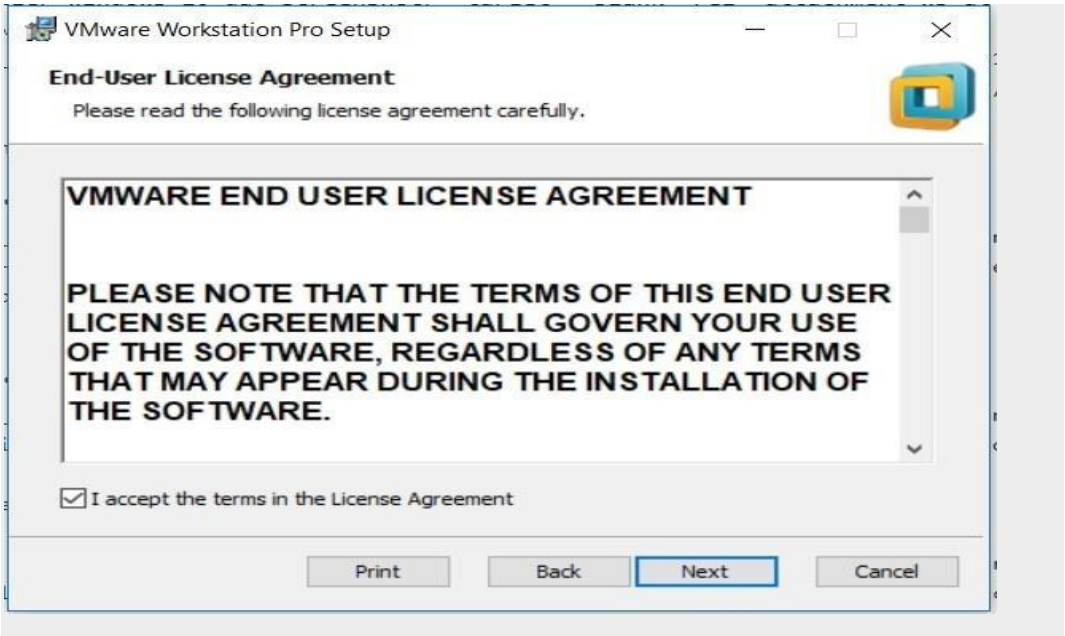
Step 5- VMware Workstation Setup wizard

Now you will see VMware Workstation setup wizard dialog box. Click next to continue.



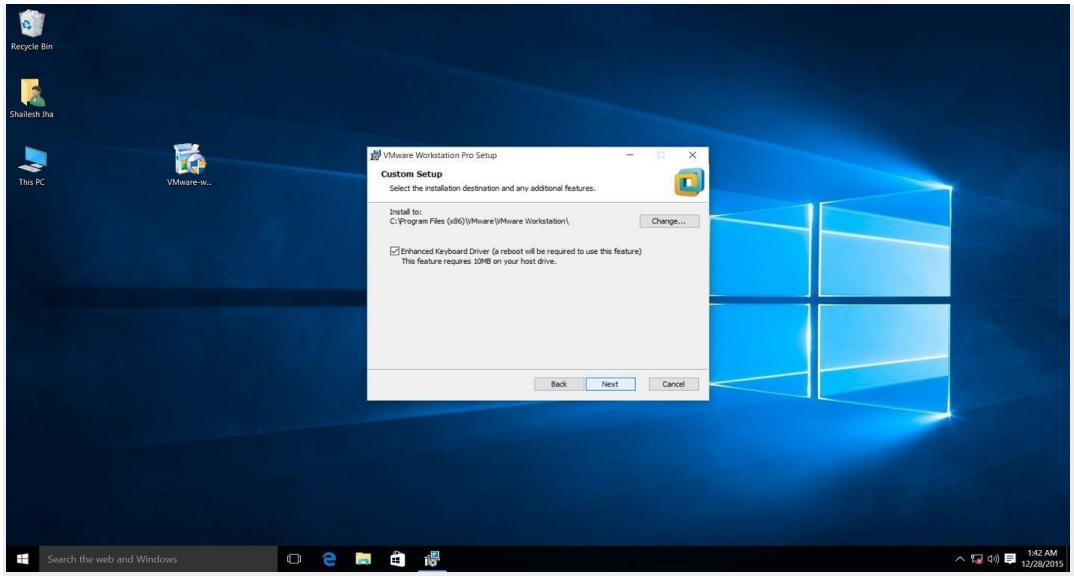
Step 6- End User Licence Agreement

This time you should see End User Licence Agreement dialog box. Check “I accept the terms in theLicence Agreement” box and press next to continue.



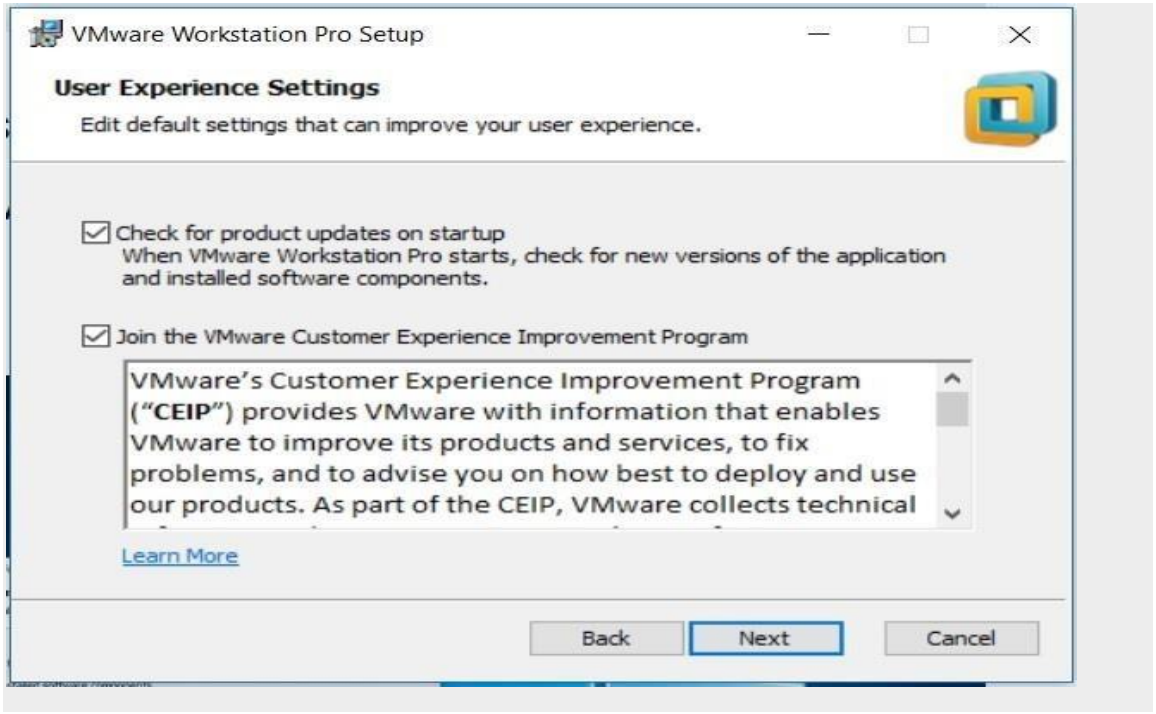
Step 7- Custom Setup options

Select the folder in which you would like to install the application. There is no harm in leaving the defaults as it is. Also select Enhanced Keyboard Driver check box.



Step 8- User Experience Settings

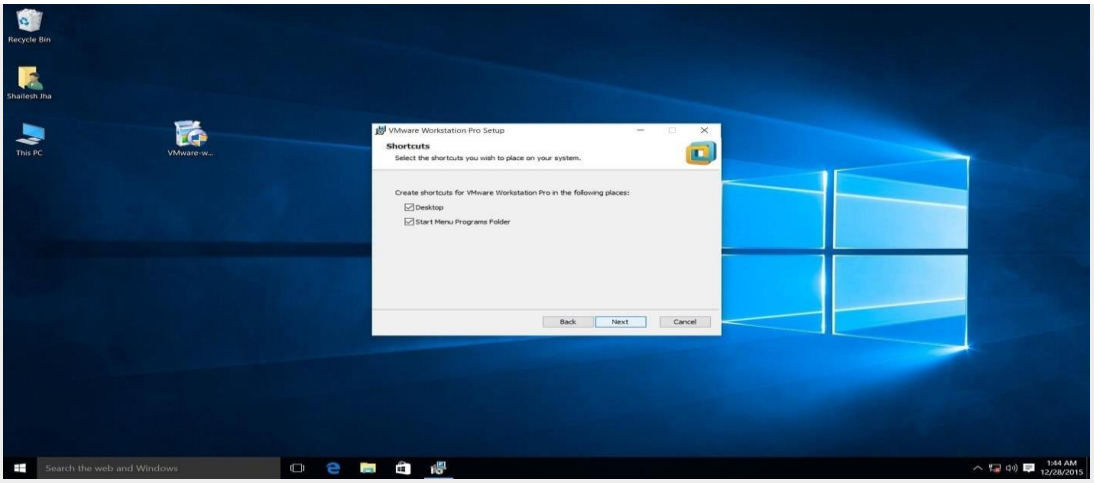
Next you are asked to select “Check for Updates” and “Help improve VMware Workstation Pro”. Do as you wish. I normally leave it to defaults that is unchecked.



Step 9- Application Shortcuts preference

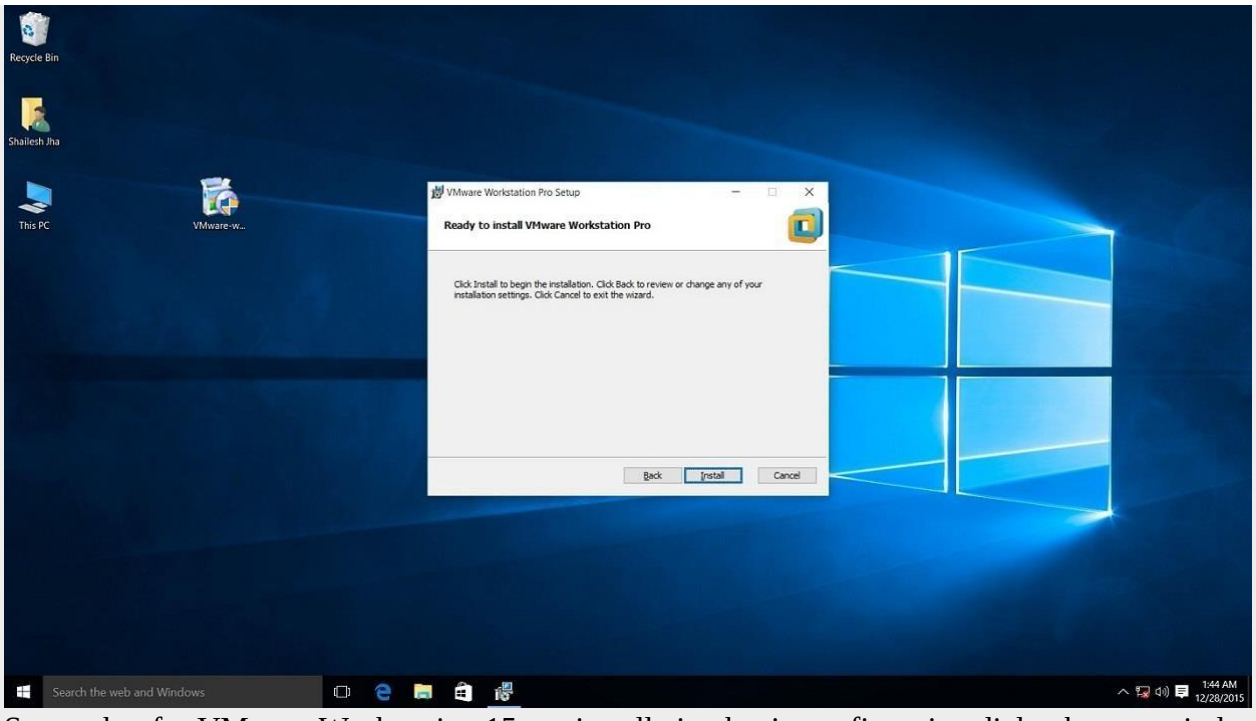
Next step is to select the place you want the shortcut icons to be placed on your system to launch

theapplication. Please select both the options, desktop and start menu and click next.



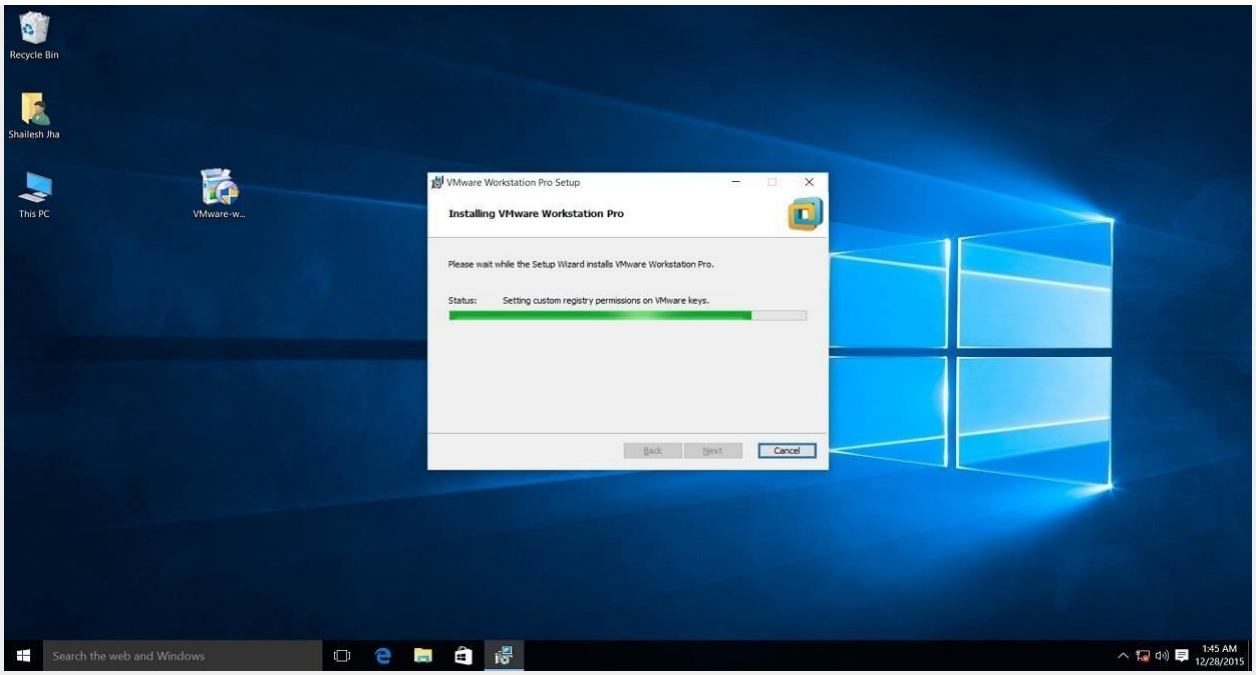
Step 10- Installation begins

Now you see the begin installation dialog box. Click install to start the installation process.



Screenshot for VMware Workstation 15 pro installation begin confirmation dialog box on windows 10.

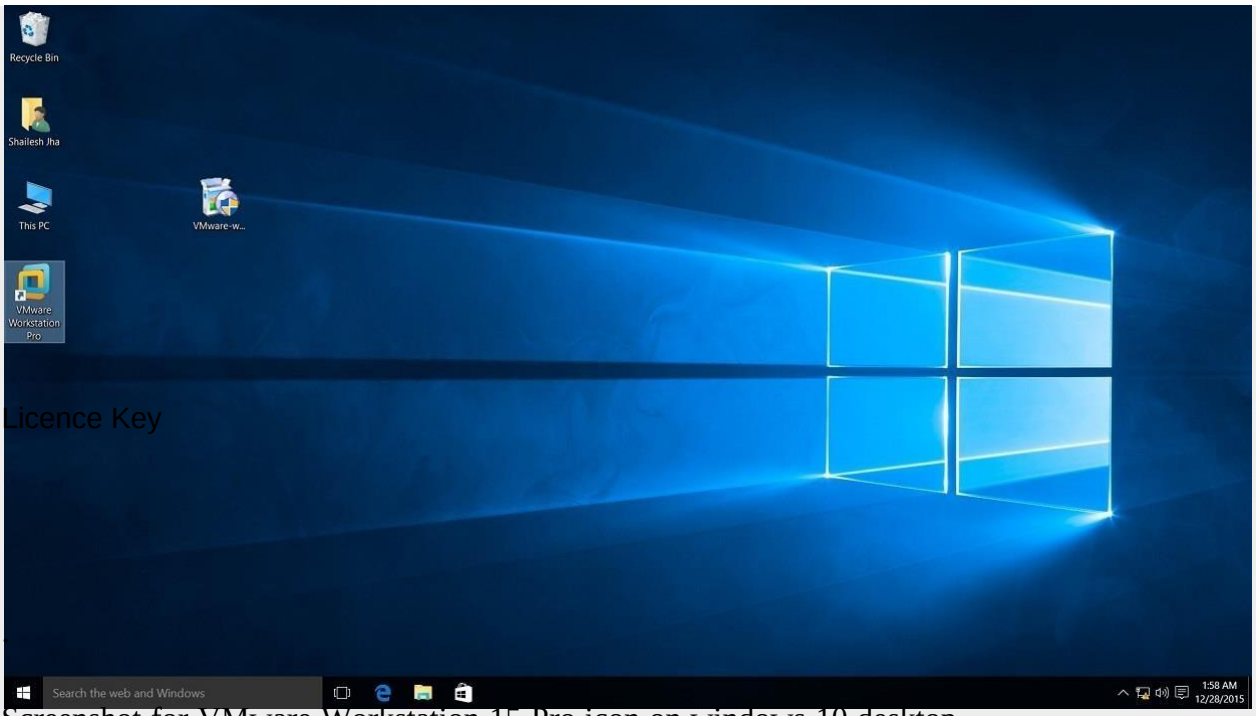
Below screenshot shows Installation in progress. Wait for this to complete.



At the end you will see installation complete dialog box. Click finish and you are done with the installation process. You may be asked to restart your computer. Click on Yes to restart.

Step 11- Launch VMware Workstation

After the installation completes, you should see VMware Workstation icon on the desktop. Double click on it to launch the application.

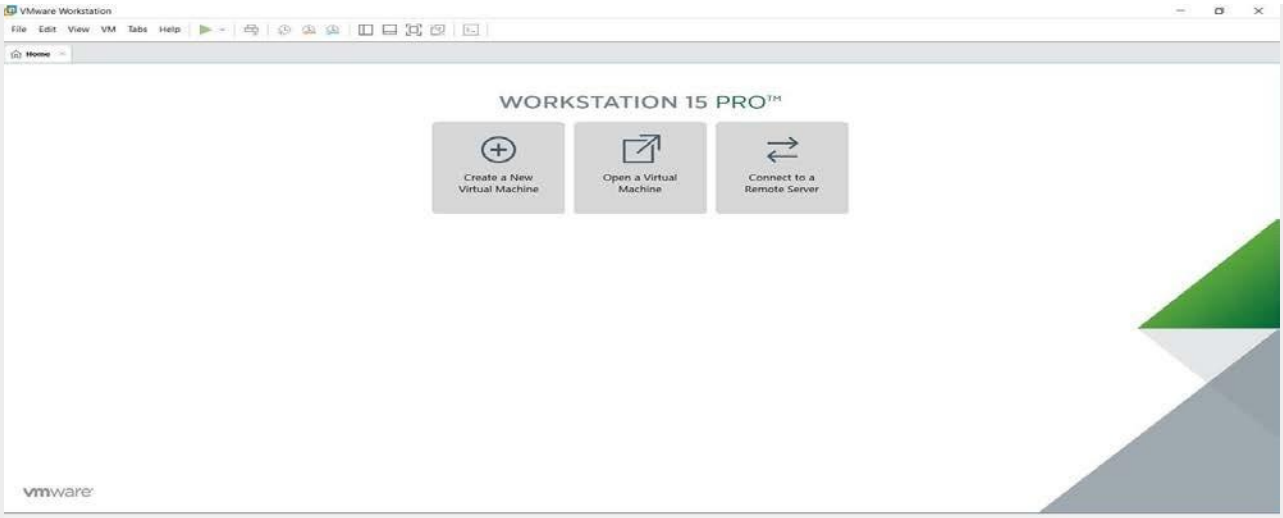


Licence Key

Screenshot for VMware Workstation 15 Pro icon on windows 10 desktop

Step 12 – License key

f you see the dialog box asking for licence key, click on trial or enter the licence key. Then what you have is the VMware Workstation 15 Pro running on your windows 10 desktop. If don't have the licence key, you will have 30days trial.



VMware Workstation 15 Pro home screen.

Step – 13

License key:

At some point of time if you decide to buy the License key, you can enter the License key by going to **Help- >Enter a License Key** You can enter the 25 character license keyin the dialog box shown below and click OK. Now you have the license version of the software.

Result:

EX.NO : 7 **Installing and Running the Google App Engine**

AIM:

PROCEDURE:

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

Pre--Requisites: Python 2.5.4

If you don't already have Python 2.5.4 installed in your computer, download and Install Python 2.5.4 from:

http://www.python.org/download/releases/2.5.4/

Download and Install

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

http://code.google.com/appengine/downloads.html

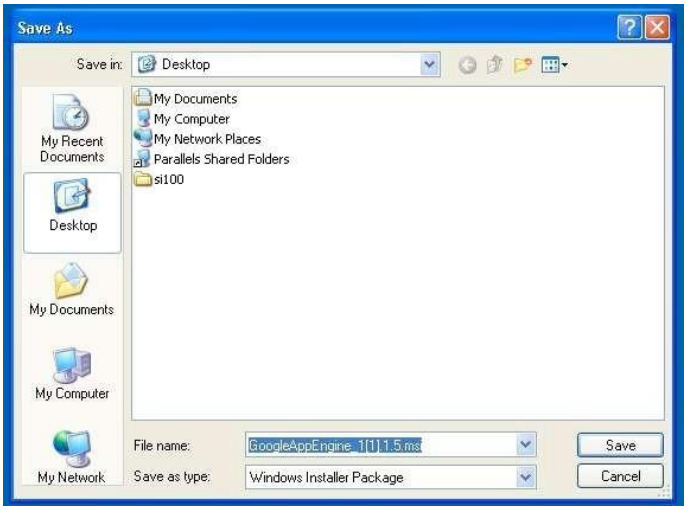
Download the Google App Engine SDK

Before downloading, please read the Terms that govern your use of the App Engine SDK.

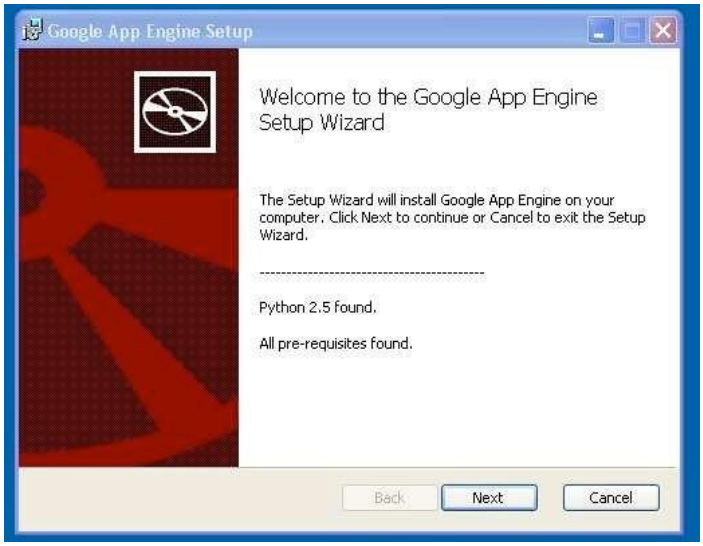
Please note: The App Engine SDK is under **active development**, please keep this in mind as you explore its capabilities. See the SDK Release Notes for the information on the most recent changes to the App Engine SDK. If you discover any issues, please feel free to notify us via our Issue Tracker.

Platform	Version	Package	Size	SHA1 Checksum
Windows	1.1.5 - 10/03/08	GoogleAppEngine_1.1.5.msi	2.5 MB	e974312b4aefc0b3873ff0d93eb4c525d5e88c30
Mac OS X	1.1.5 - 10/03/08	GoogleAppEngineLauncher-1.1.5.dmg	3.6 MB	f62208ac01c1b3e39796e58100d5f1b2f052d3e7
Linux/Other Platforms	1.1.5 - 10/03/08	google_appengine_1.1.5.zip	2.6 MB	cbb9ce817bdabf1c4f181d9544864e55ee253de1

Download the Windows installer – the simplest thing is to download it to yourDesktop or another folder that you remember.



Double Click on the **GoogleApplicationEngine** installer.



Click through the installation wizard, and it should install the App Engine. If you donot have Python 2.5, it will install Python 2.5 as well.

Once the install is complete you can discard the downloaded installer

Making your First Application

Now you need to create a simple application. We could use the "+" option to havethe launcher make us an application – but instead we will do it by hand to get a better sense of what is going on.
Make a folder for your Google App Engine applications. I am going to make theFolder on my Desktop called "**apps**" – the path to this folder is:

C:\Documents and Settings\csev\Desktop\apps

And then make a sub-folder in within **apps** called "**ae-01-trivial**" – the path to this folder would be:

C:\ Documents and Settings \csev\Desktop\apps\ae--01--trivial

Using a text editor such as JEdit (www.jedit.org), create a file called **app.yaml** in the **ae--01--trivial** folder with the following contents:

```
application: ae-01-trivialversion: 1
runtime: python
api_version: 1

handlers:
- url: /.*
script: index.py
```

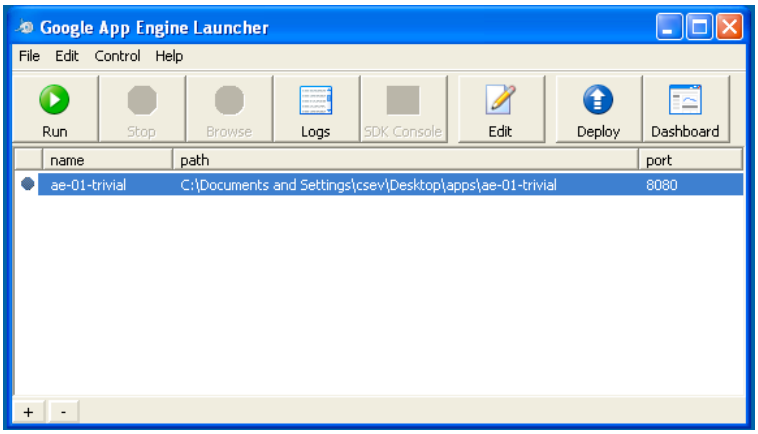
Note: Please do not copy and paste these lines into your text editor – you might endup with

strange characters – simply type them into your editor.

Then create a file in the **ae--01--trivial** folder called **index.py** with three lines in it:

```
print 'Content-Type: text/plain'print ' '
print 'Hello there Chuck'
```

Then start the **GoogleAppEngineLauncher** program that can be found under **Applications**. Use the **File --> Add Existing Application** command and navigate into the **apps** directory and select the **ae--01--trivial** folder. Once you have addedthe application, select it so that you can control the application using the launcher.



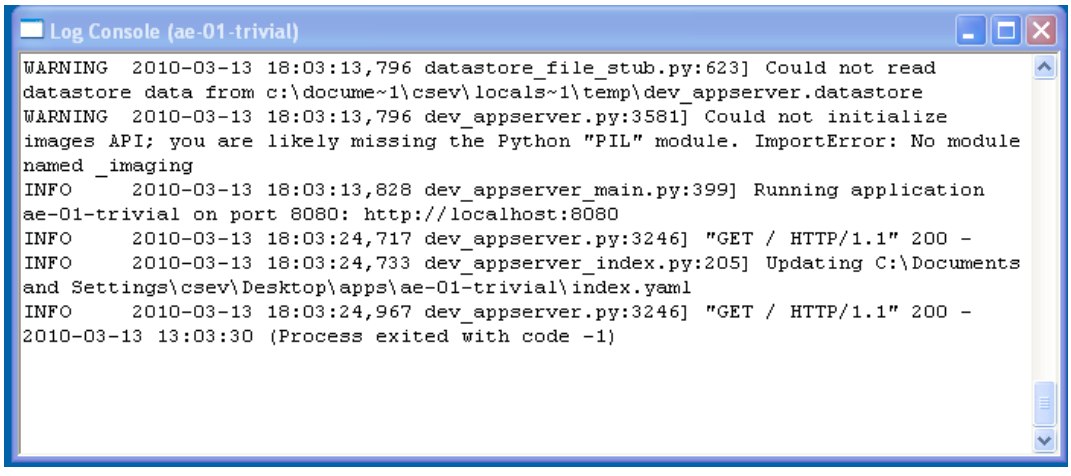
Once you have selected your application and press **Run**. After a few moments your application will start and the launcher will show a little green icon next to your application. Then press **Browse** to open a browser pointing at your application which is running at **http://localhost:8080/**

Paste **http://localhost:8080** into your browser and you should see your application as follows:

Just for fun, edit the **index.py** to change the name “Chuck” to your own name andpress Refresh in the browser to verify your updates.

Watching the Log

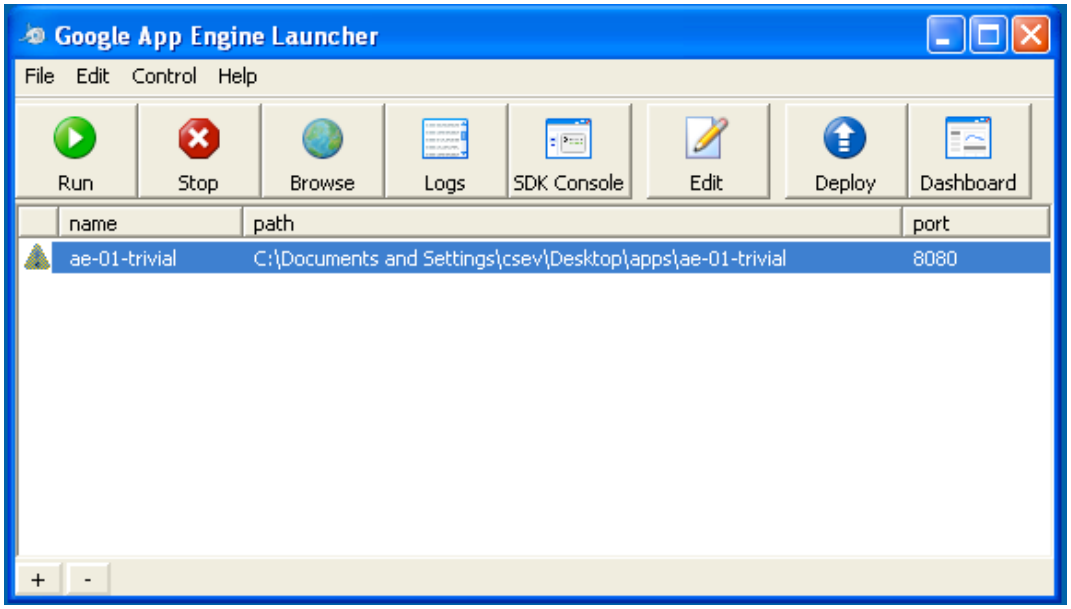
You can watch the internal log of the actions that the web server is performing whenyou are interacting with your application in the browser. Select your application in the Launcher and press the **Logs** button to bring up a log window:



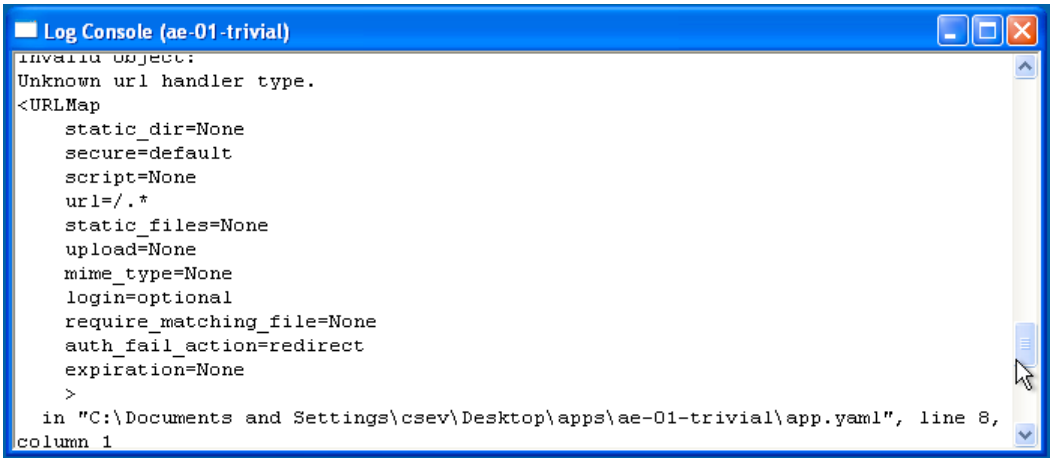
Each time you press **Refresh** in your browser – you can see it retrieving the outputwith a **GET** request.

Dealing With Errors

With two files to edit, there are two general categories of errors that you may encounter. If you make a mistake on the **app.yaml** file, the App Engine will not startand your launcher will show a yellow icon near your application:

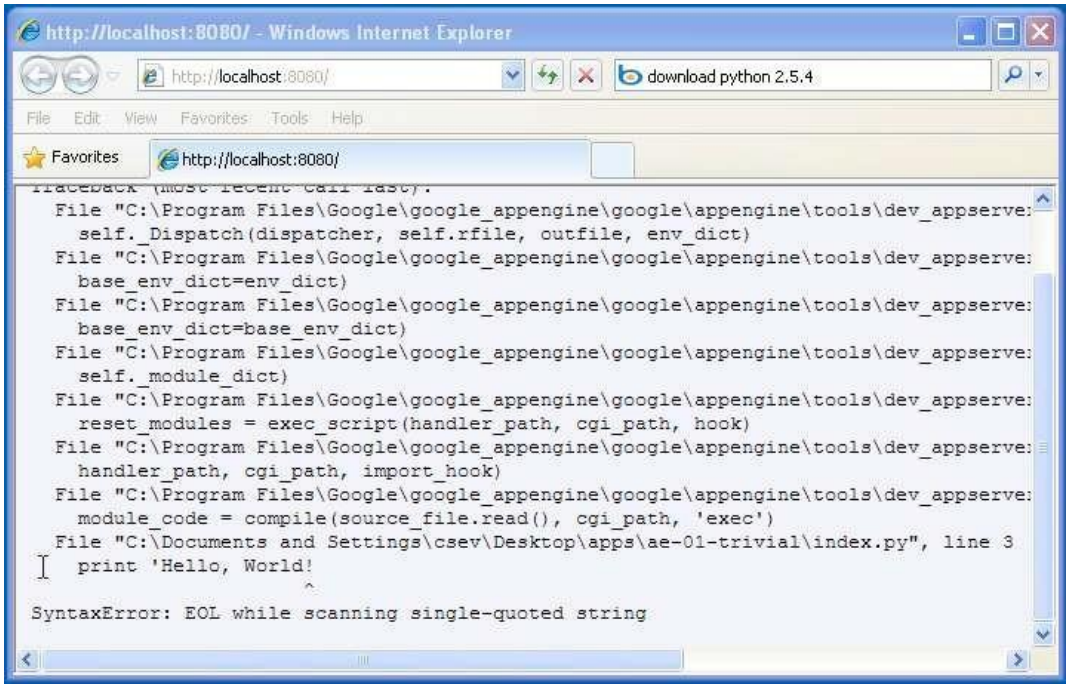


To get more detail on what is going wrong, take a look at the log for the application:



In this instance – the mistake is mis-indenting the last line in the **app.yaml** (line 8).

If you make a syntax error in the **index.py** file, a Python trace back error will appearin your browser.



The error you need to see is likely to be the last few lines of the output – in this caseI made a Python syntax error on line one of our one-line application.

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

When you make a mistake in the **app.yaml** file – you must the fix the mistake and attempt to start the application again.
If you make a mistake in a file like i**ndex.py**, you can simply fix the file and press refresh in your browser – there is no need to restart the server.

Shutting Down the Server

To shut down the server, use the Launcher, select your application and press the **Stop** button.

RESULT:

EX NO: 8 INSTALL A C COMPILER IN THE VIRTUAL MACHINE AND EXECUTE A SAMPLE PROGRAM

AIM:

PROCEDURE:

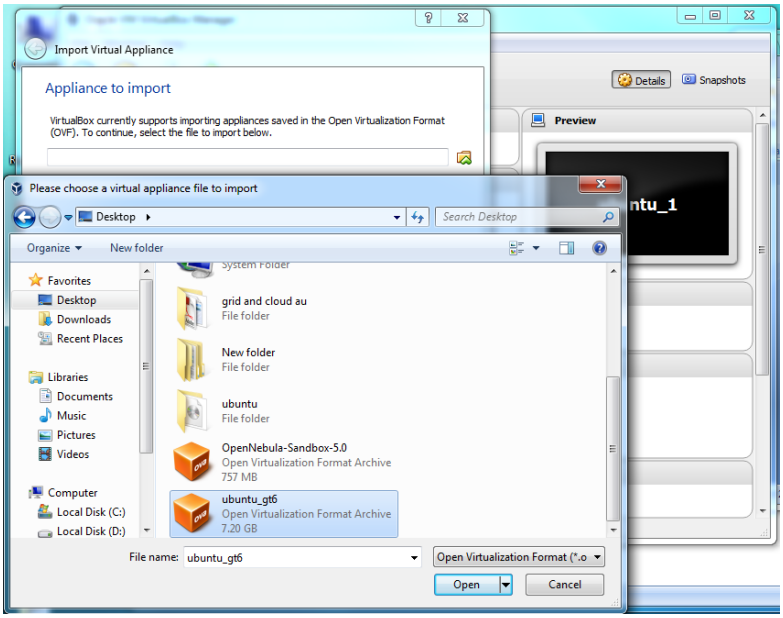
REQUIREMENTS:

1. ORACLE VIRTUAL BOX
2. OPEN NEBULA SANDBOX
3. UBUNTU Gt6.Ova

STEP 1:

ubuntu_gt6 installation:

- Open Virtual box
- File →import Appliance
- Browse ubuntu_gt6**.**ova file
- Then go to setting, select Usb and choose USB 1.1
- Then Start the ubuntu_gt6
- Login using username: dinesh, password:99425.



STEP 2:

Open the terminal

STEP 3:

//to install gcc

Sudo add-apt repository ppa:ubuntu-toolchain-r/test

sudo apt-get update

sudo apt-get install gcc-6 gcc-6-base

STEP 4:

To type a sample c program and save it

gedit hello.c

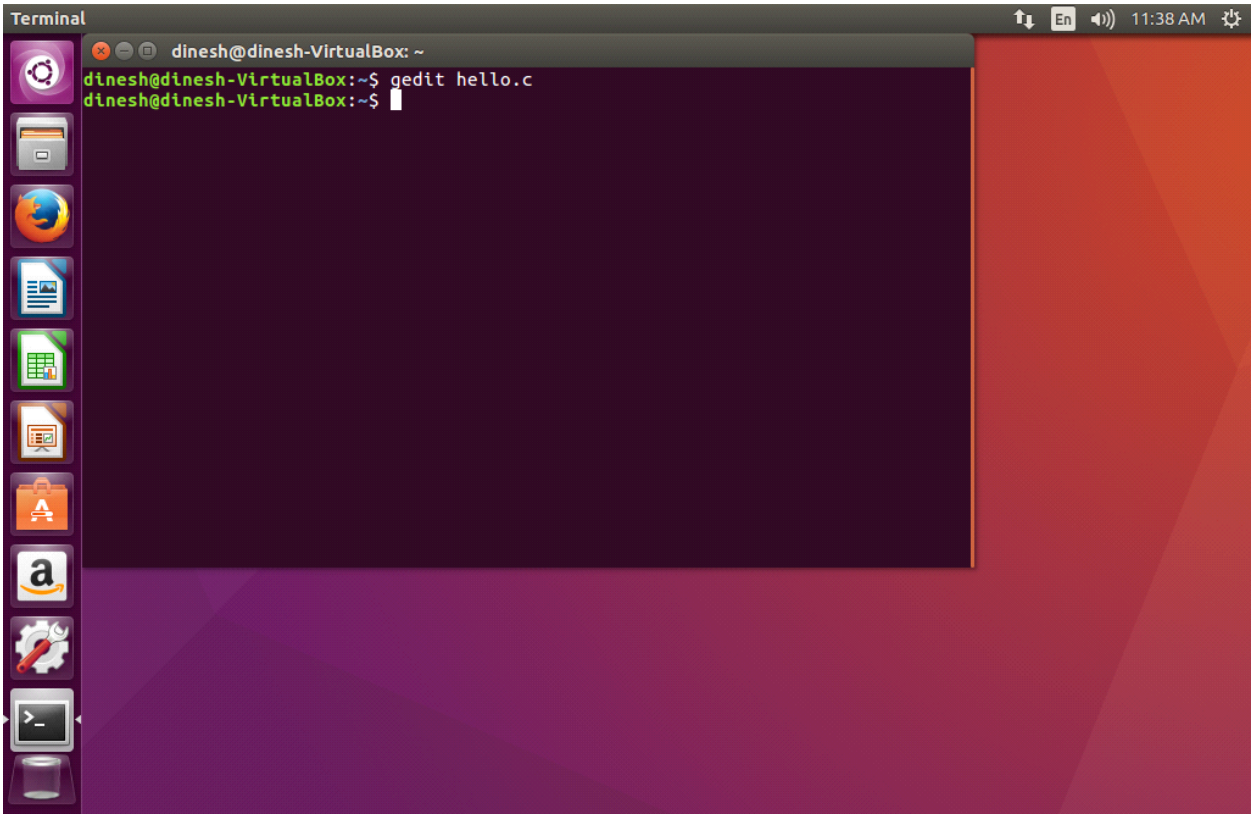
STEP 5:

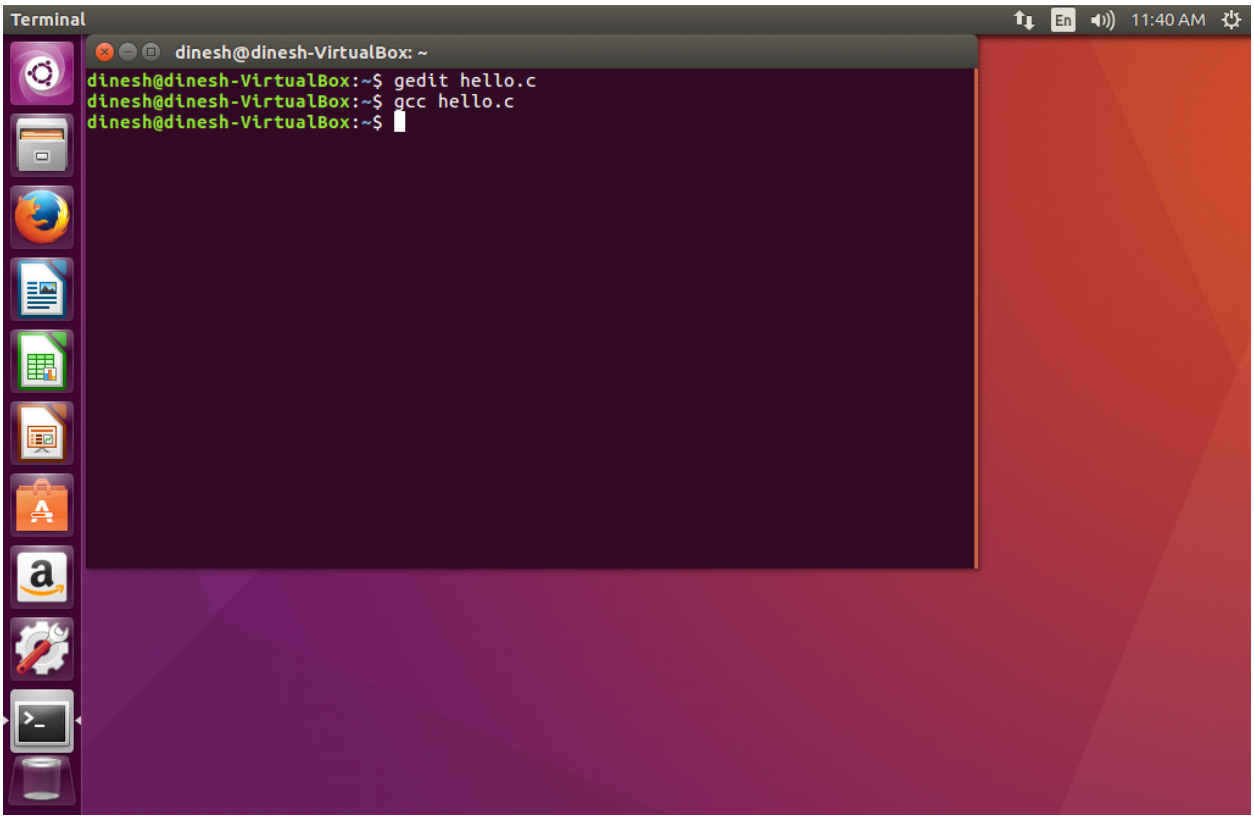
To compile and run a sample c program

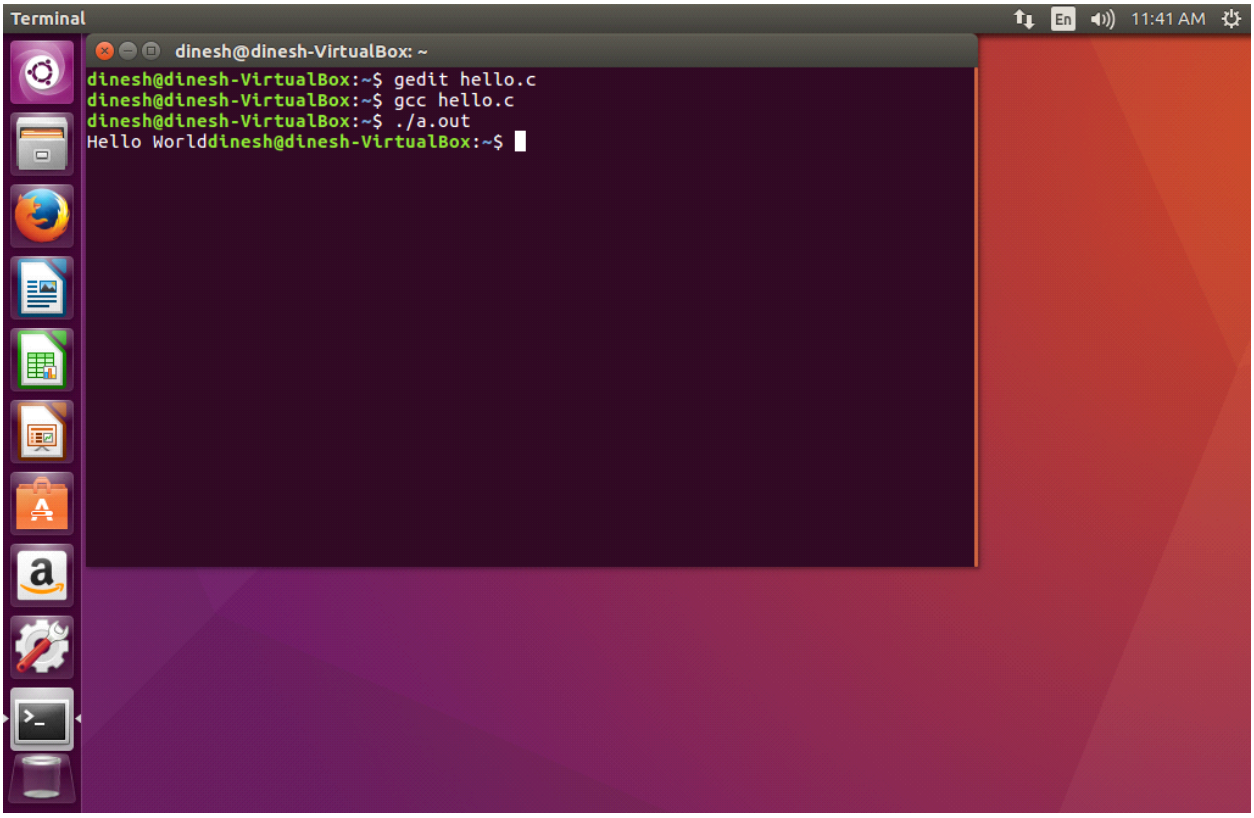
gcc hello.c

./a.out

OUTPUT:


Terminal
En
11:38 AM
dinesh@dinesh-VirtualBox: ~
dinesh@dinesh-VirtualBox:~$ gedit hello.c
dinesh@dinesh-VirtualBox:~$


Terminal
En
11:40 AM
dinesh@dinesh-VirtualBox: ~
dinesh@dinesh-VirtualBox:~$ gedit hello.c
dinesh@dinesh-VirtualBox:~$ gcc hello.c
dinesh@dinesh-VirtualBox:~$


Terminal
En
11:41 AM
dinesh@dinesh-VirtualBox: ~
dinesh@dinesh-VirtualBox:~$ gedit hello.c
dinesh@dinesh-VirtualBox:~$ gcc hello.c
dinesh@dinesh-VirtualBox:~$ ./a.out
Hello Worlddinesh@dinesh-VirtualBox:~$

RESULT: