

---

Cloud Computing Lab Program Report

BANUPRASAD B

NNM22AD017

November 14 , 2024

---

# Contents

<b>1 Experiment 1</b>	<b>4</b>
1.1 Aim . . . . .	4
1.2 Procedure . . . . .	4
1.3 Result . . . . .	13
<b>2 Experiment 2</b>	<b>14</b>
2.1 Aim . . . . .	14
2.2 Procedure . . . . .	14
2.3 Result . . . . .	16
<b>3 Experiment 3</b>	<b>17</b>
3.1 Bare-Metal Hypervisor . . . . .	17
3.2 Advantages of Bare-Metal Hypervisors . . . . .	17
3.3 Top Bare-Metal Hypervisors . . . . .	17
3.4 Overview of Popular Bare-Metal Hypervisors . . . . .	18
3.5 Hosted Hypervisors . . . . .	18
<b>4 Experiment 4</b>	<b>21</b>
4.1 Aim . . . . .	21
4.2 Procedure . . . . .	21
4.3 Result . . . . .	25
<b>5 Experiment 5</b>	<b>26</b>
5.1 Aim . . . . .	26
5.2 Procedure . . . . .	26
5.3 Result . . . . .	33

# List of Figures

1	Official website of VM VirtualBox . . . . .	4
2	VirtualBox Setup . . . . .	5
3	VirtualBox Custom Setup . . . . .	6
4	VirtualBox Setup . . . . .	6
5	VirtualBox Setup . . . . .	7
6	VirtualBox Setup . . . . .	7
7	Creation of VM . . . . .	8
8	Configuration of Hardware . . . . .	8
9	Configuration of Virtual HardDisk . . . . .	9
10	Starting the VM . . . . .	9
11	Installing Ubuntu . . . . .	10
12	Selecting the software language . . . . .	10
13	Installation Type . . . . .	11
14	Username and Password . . . . .	12
15	Setting up Ubuntu . . . . .	12
16	Ubuntu homepage . . . . .	13
17	VM (Ubuntu) terminal . . . . .	14
18	sudo apt install build-essential . . . . .	15
19	gcc –version . . . . .	15
20	Hello World program in C . . . . .	16
21	Installer . . . . .	21
22	Official website of Notepad++ . . . . .	22
23	select WINE application . . . . .	22
24	Notepad++ Setup . . . . .	23
25	Notepad++ Setup Wizard . . . . .	23
26	First view of Notepad++ . . . . .	24
27	Creating and saving the new file . . . . .	25
28	snapshot of creating a new project . . . . .	26
29	Shows the project details - project ID . . . . .	27
30	Configure the app engine . . . . .	27
31	Select the region . . . . .	28

32	Installing essential dependencies . . . . .	29
33	Follow the setup instructions to set the application . . . . .	30
34	Installing express . . . . .	30
35	Index.js hello world program . . . . .	31
36	Runtime nodejs 18 . . . . .	32
37	Snapshot of the Output . . . . .	33

# 1 Experiment 1

Install Virtual box/VMware Workstation with different flavors of Linux or windows OS on top of windows7 or 8.

## 1.1 Aim

To Install VirtualBox/VMware Workstation with different flavors of Linux or windows OS on top of windows7 or 8.

## 1.2 Procedure

### Step 1:

Open the VirtualBox website by going to <https://www.virtualbox.org/> in your Internet browser. This is where you'll download the VirtualBox setup file.

### Step 2:

Click the "Download VirtualBox" button, which is located in the middle of the page. This will take you to the downloads page.

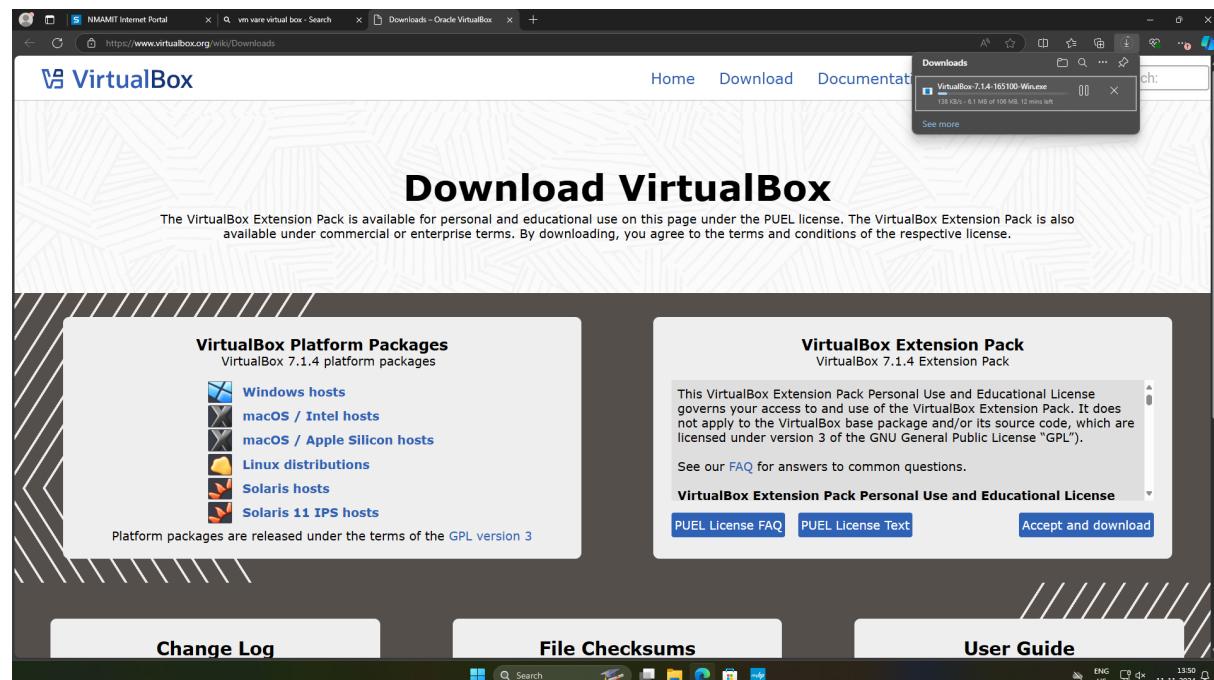


Figure 1: Official website of VM VirtualBox

### STEP 3:

Click windows host. You'll see this link below the "VirtualBox 5.2.8 platform packages" heading. The VirtualBox EXE file will begin downloading onto your computer.

### STEP 4:

Open the VirtualBox EXE file. Go to the location where the EXE file was downloaded and double-click the file. This will open the VirtualBox installation window. Navigate through the installation prompts by doing the following:

- Click **Next** on the first three pages.
- Click **Yes** when prompted.
- Click **Install**.
- Click **Yes** when prompted.
- Click **Install** when prompted. This will start the installation of VirtualBox on your computer.
- Click **Finish** when prompted. This button is in the lower-right side of the window and will close the installation window, opening VirtualBox.



Figure 2: VirtualBox Setup

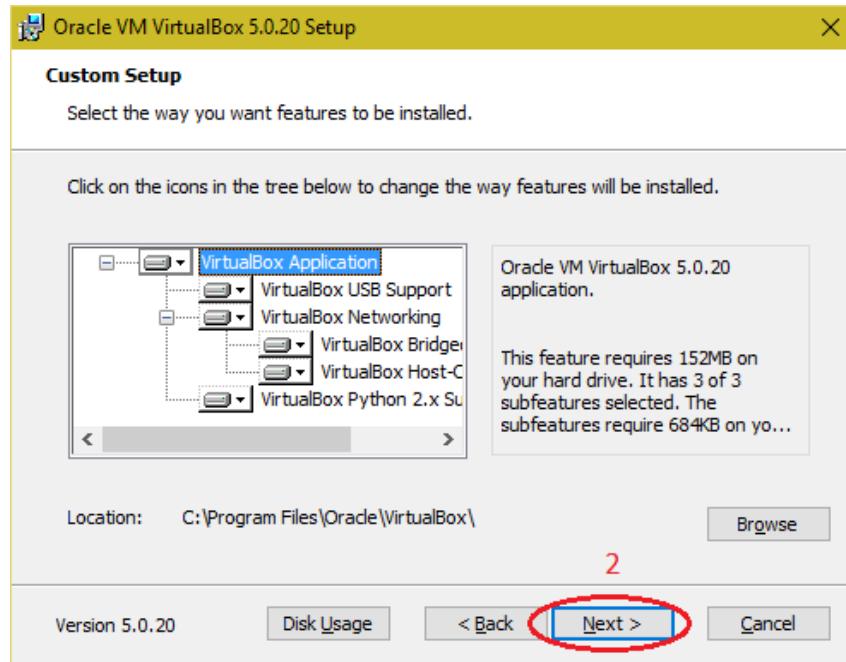


Figure 3: VirtualBox Custom Setup

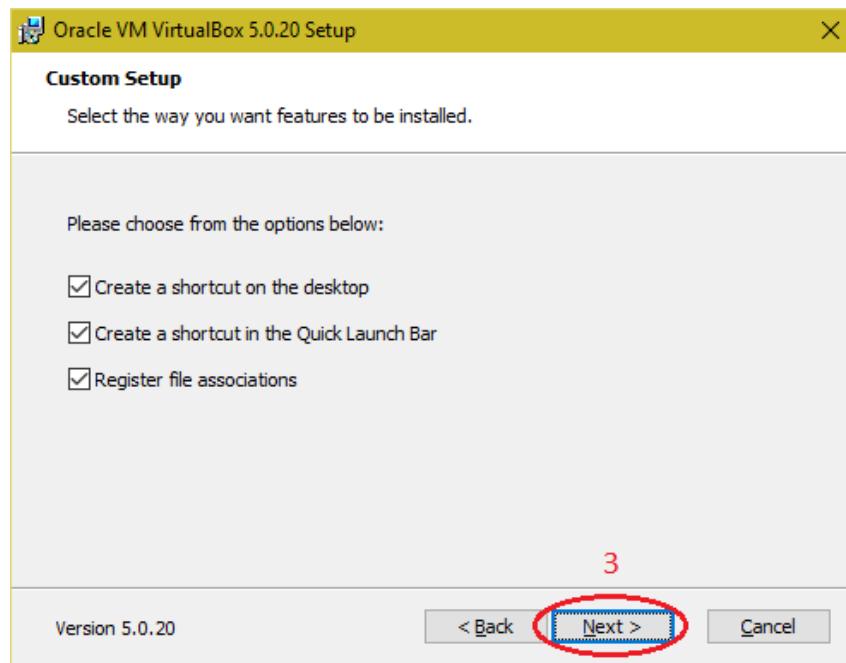


Figure 4: VirtualBox Setup

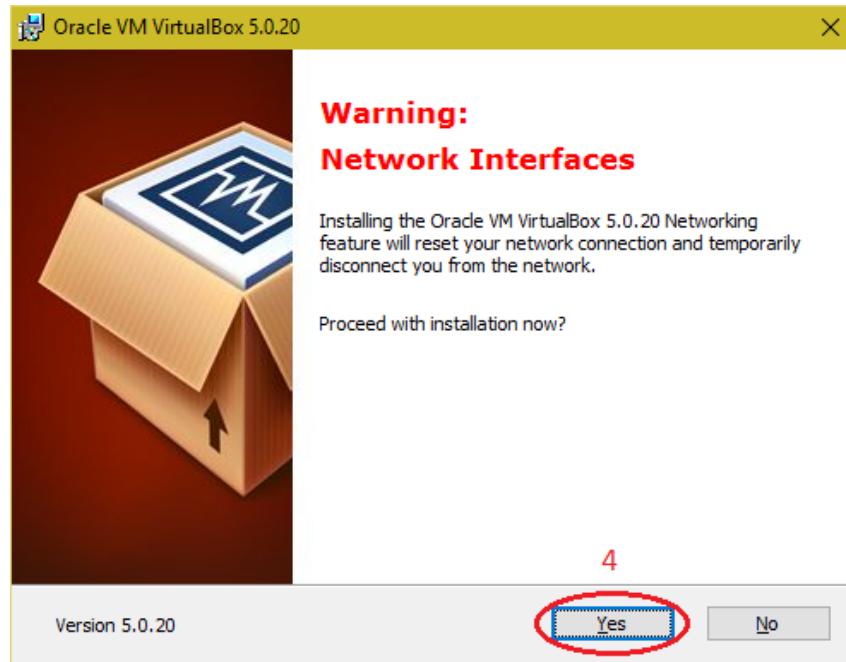


Figure 5: VirtualBox Setup

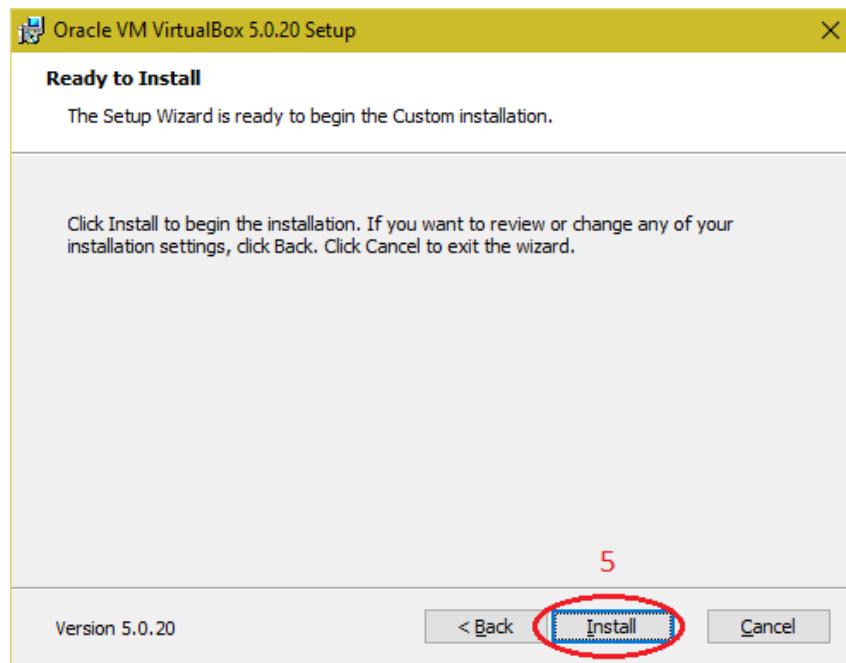


Figure 6: VirtualBox Setup

## STEP 5:

Open Oracle VM VirtualBox Manager, Type a name for the new virtual machine. (Ubuntu), I'll enter 'ubuntu6969'. Note that VirtualBox automatically changes 'Type' to Linux and 'Version' to 'Ubuntu (64 bit)'. These two options are exactly what we need

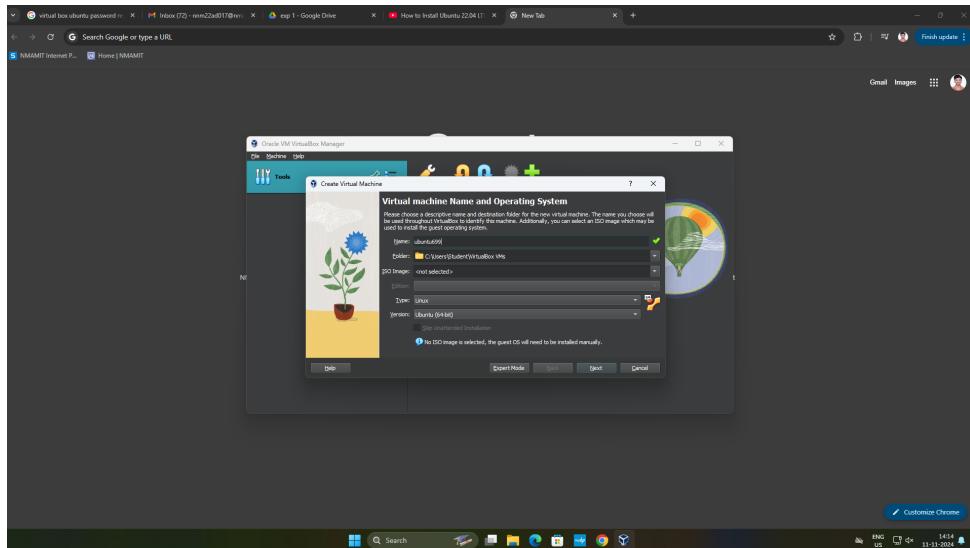


Figure 7: Creation of VM

## STEP 6:

The memory size depends on your host machine memory size and select the size of the virtual hard disk and click ‘next’ button.

Note that VirtualBox will create a swap partition with the same amount space as base memory you have entered here. So later when you are selecting the size of the virtual hard drive, make sure it is large enough since the hard drive will be splitted into root (/) and swap partitions. The root partition contains by default all your system files, program settings and documents.

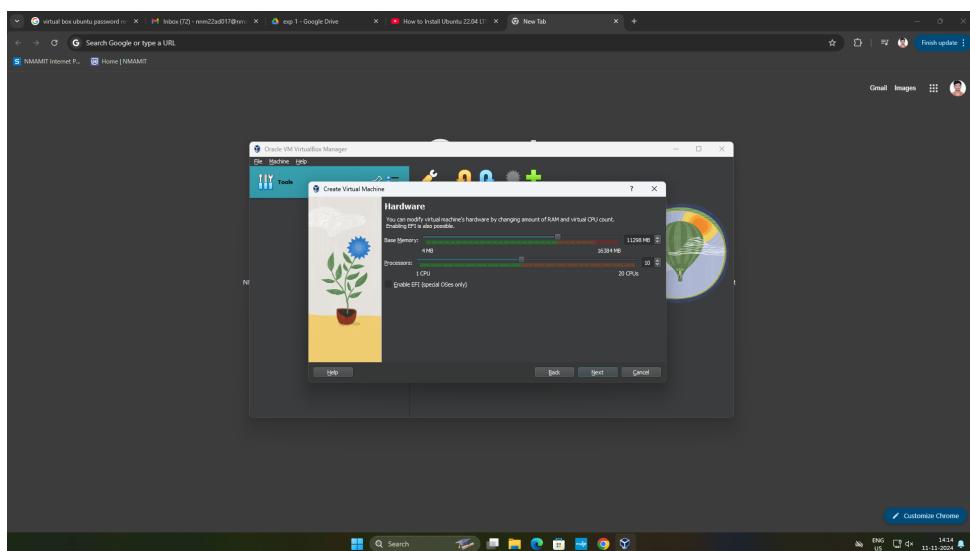


Figure 8: Configuration of Hardware

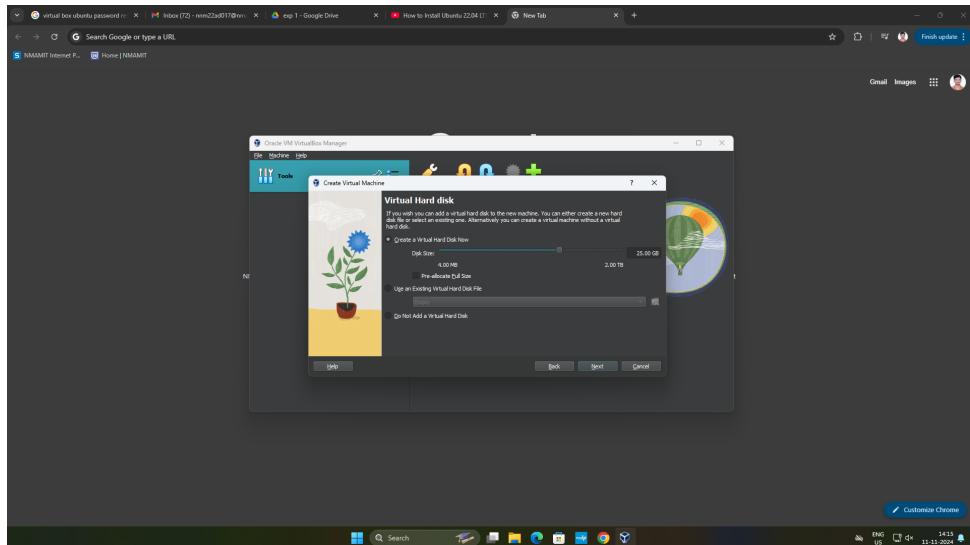


Figure 9: Configuration of Virtual HardDisk

## STEP 7:

Select the created Virtual machine and click 'start' button.

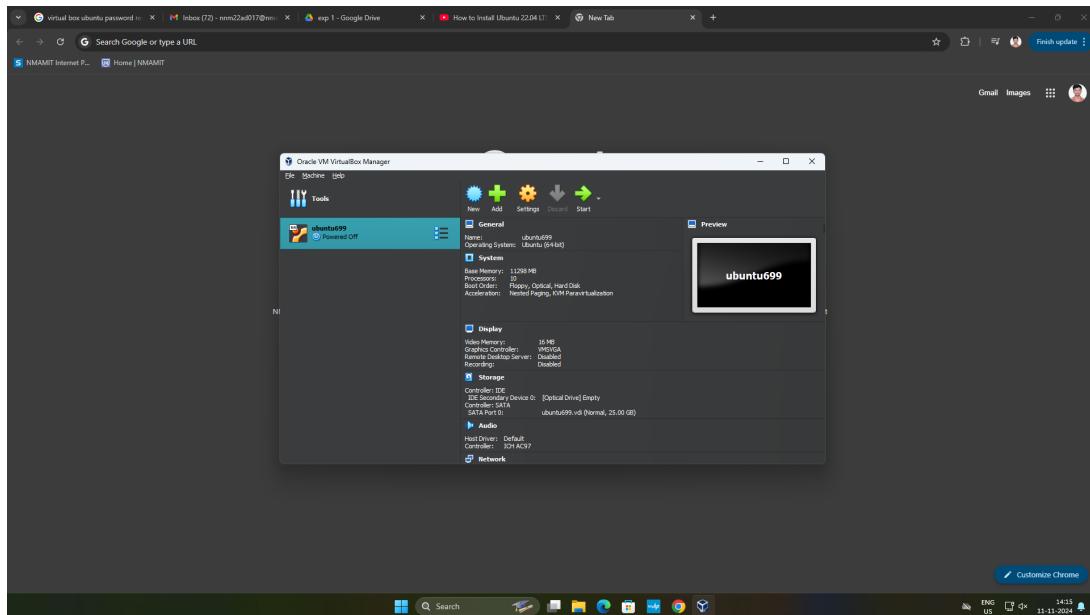


Figure 10: Starting the VM

## Install Ubuntu

Now you shall see a 'Welcome' screen. Click 'Install Ubuntu' button. Note that the installation process may differ a little bit from version to version

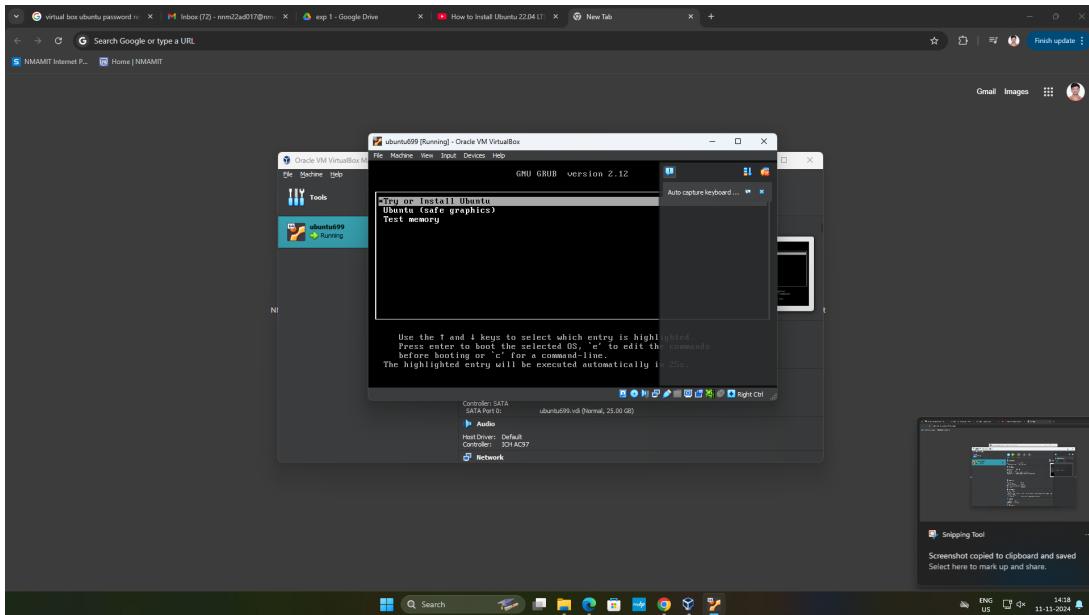


Figure 11: Installing Ubuntu

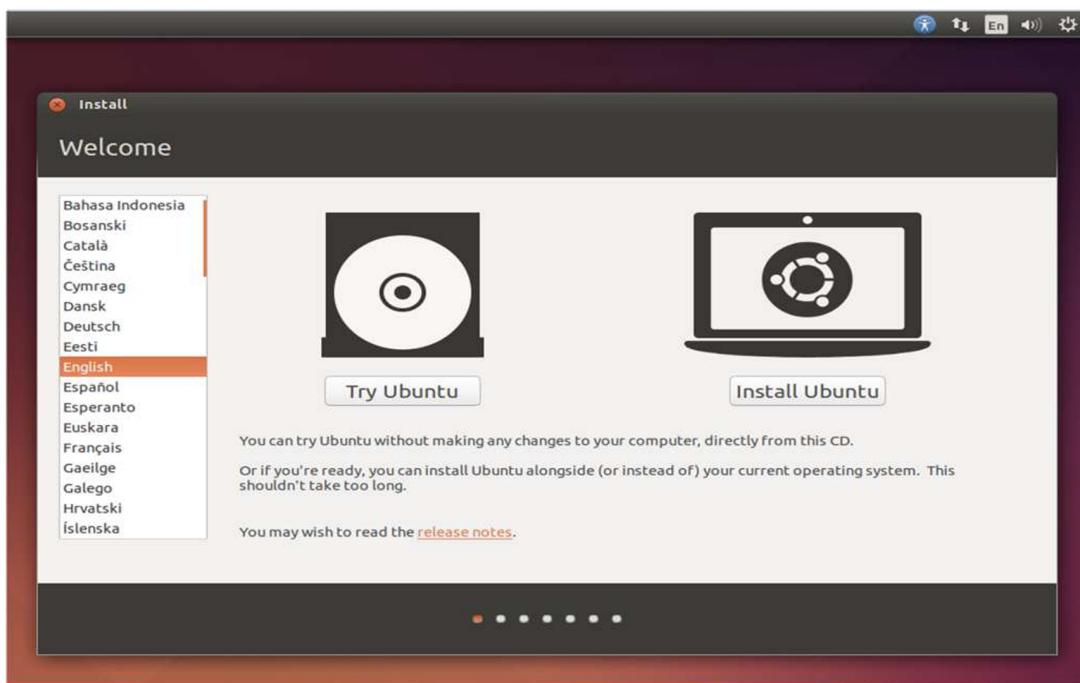


Figure 12: Selecting the software language

### STEP 8:

Click continue and make sure 'Erase disk and install Ubuntu' option is selected and click 'Install Now'

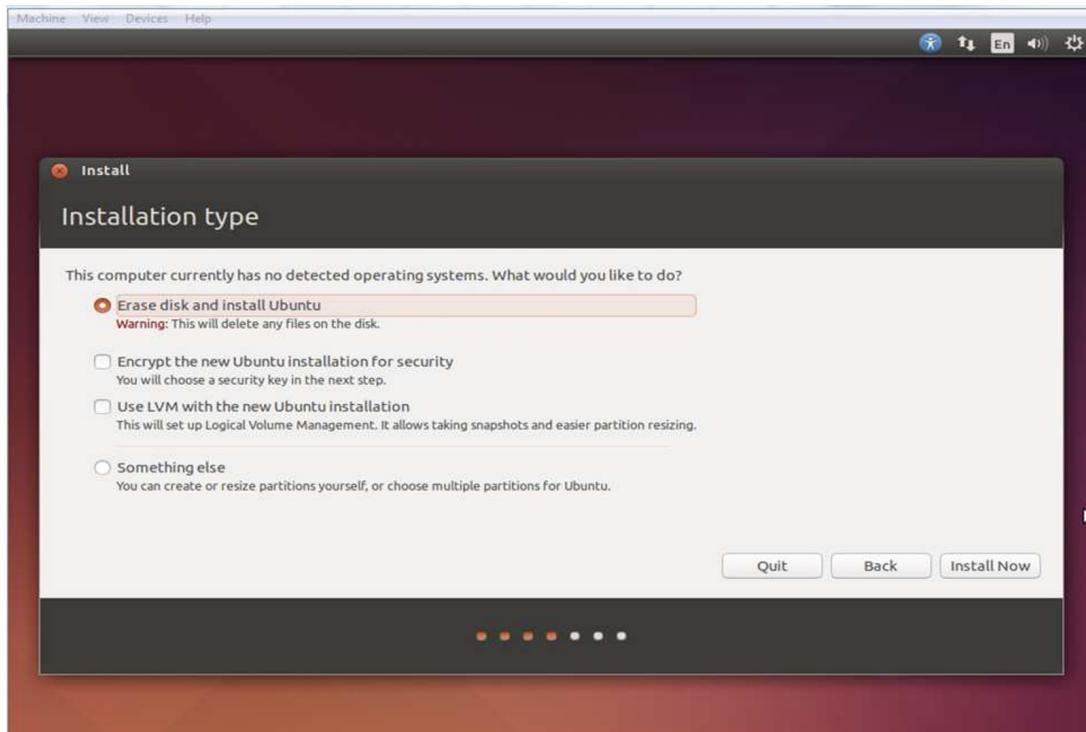


Figure 13: Installation Type

### STEP 9:

Ubuntu will ask you a few questions. If the default is good, click 'Continue', then we need to fill the username and password.

Note that this user will have root/sudo privilege. Click 'Continue' button

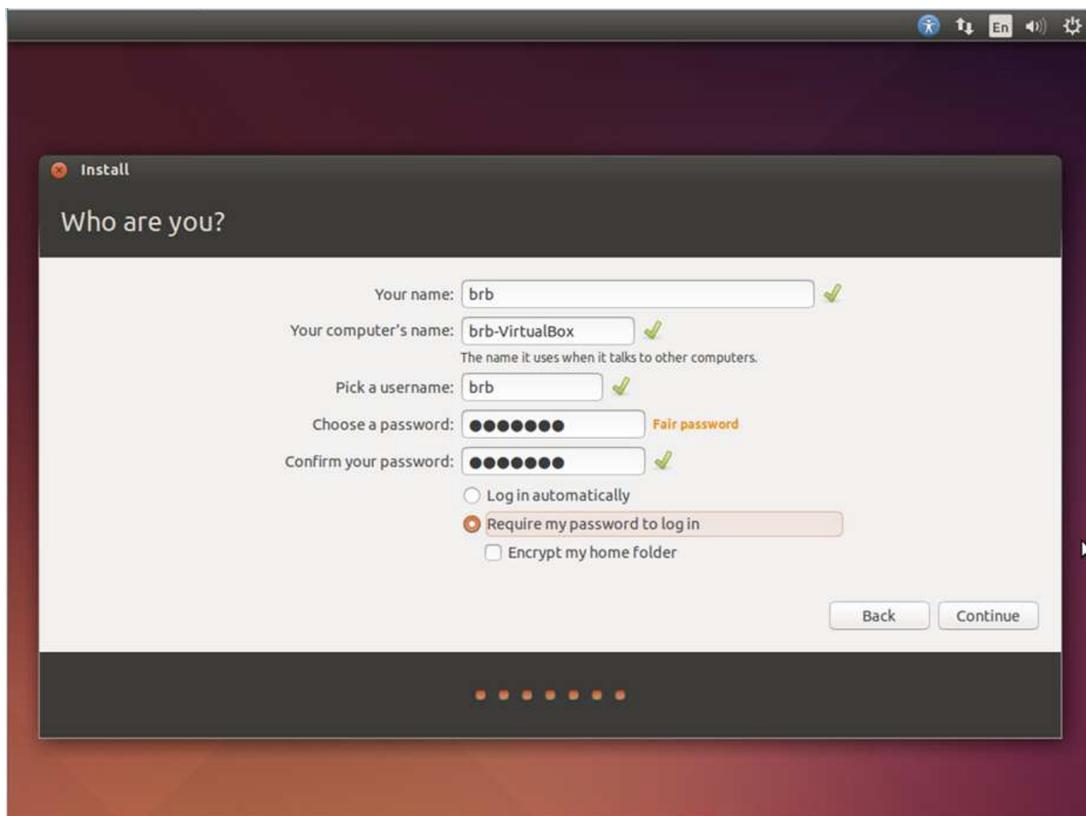


Figure 14: Username and Password

#### STEP 10:

After installation is complete, click 'Restart Now' button. When you see a screen with a black background saying 'Please remove installation media and close the tray (if any) then press ENTER

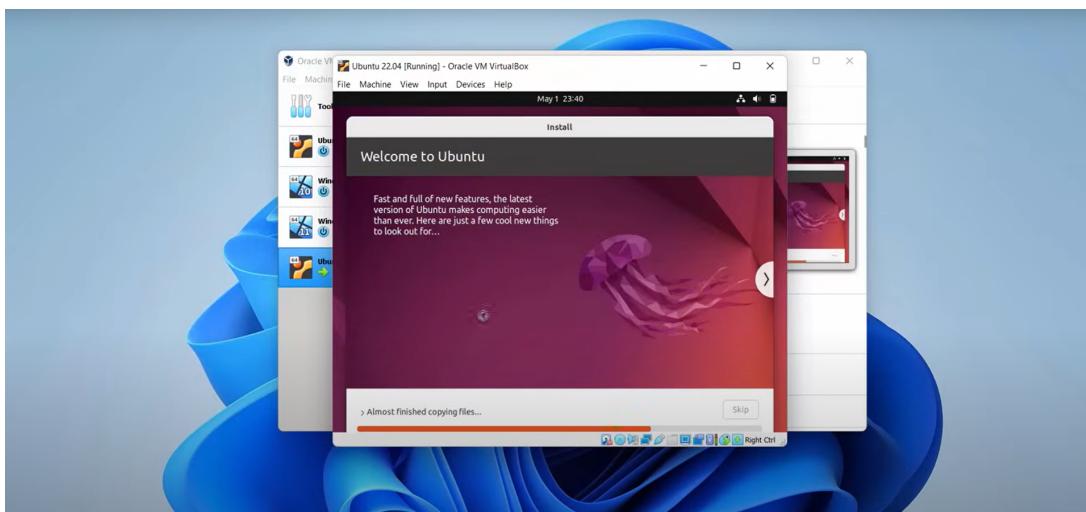


Figure 15: Setting up Ubuntu

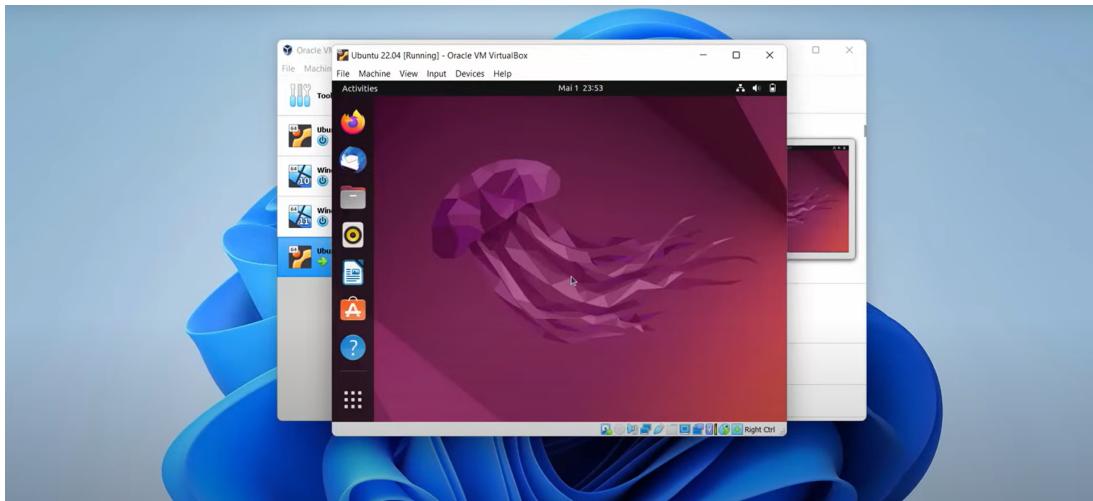


Figure 16: Ubuntu homepage

### 1.3 Result

Thus the installation of a virtual machine using Virtualbox/VMware Workstation on top of windows7 or 8 is done successfully.

## 2 Experiment 2

Install a C compiler in the virtual machine created using VirtualBox and execute simple programs.

### 2.1 Aim

To find the procedure to install a C Compiler in the Virtual Machine and execute a C program.

### 2.2 Procedure

#### STEP 1:

Open the Virtual Machine, Start your Ubuntu virtual machine from Virtual- Box.

#### STEP 2:

Update System Packages: Once Ubuntu loads, open the Terminal

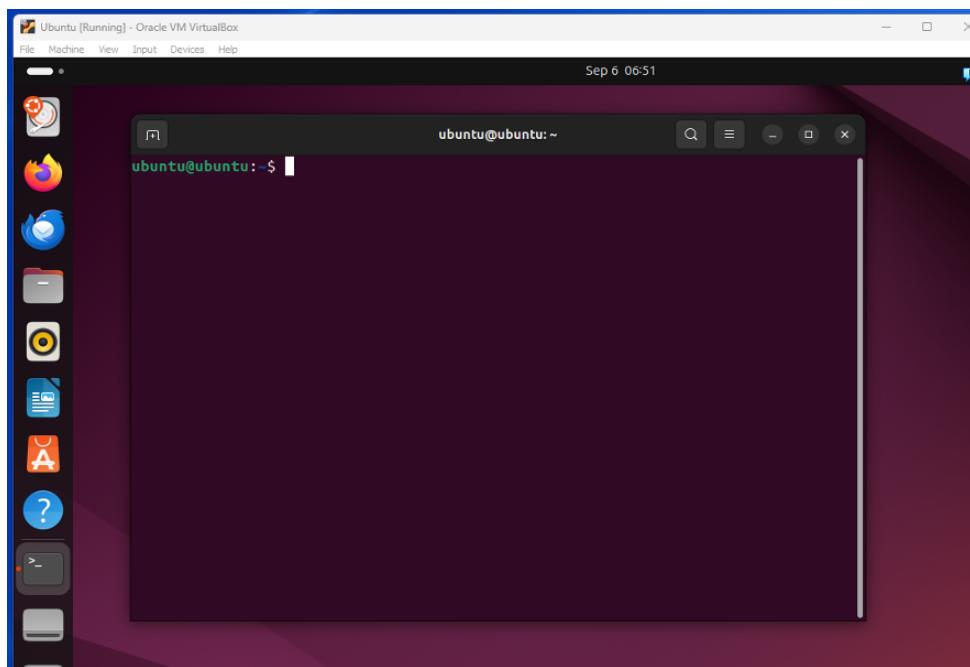


Figure 17: VM (Ubuntu) terminal

Enter the following command to update the system packages:

```
sudo apt update
```

```
sudo apt upgrade
```

**STEP 3:**

**INSTALLATION OF C COMPILER:** In the terminal, install the required build essentials, including the GCC (GNU Compiler Collection) C compiler

```
Sudo apt install gcc
```

```
Sudo apt install build-essential
```

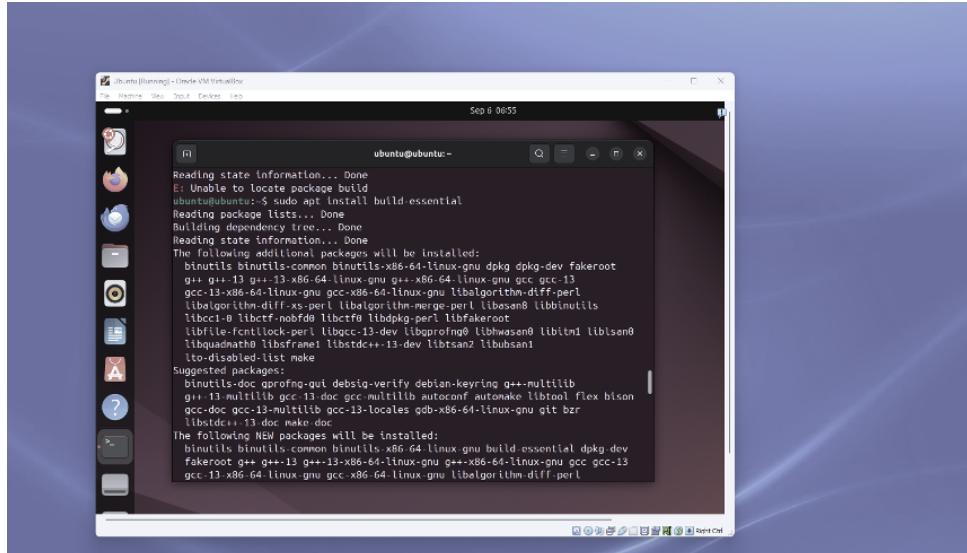


Figure 18: sudo apt install build-essential

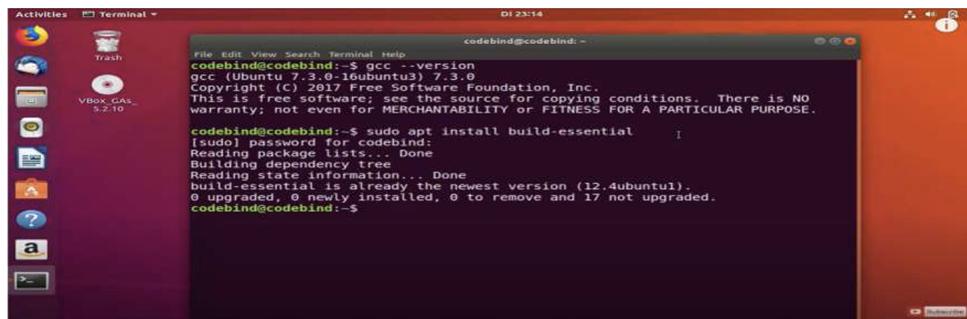


Figure 19: gcc --version

**STEP 4:**

After the compiler is installed, create a new C program file. In the terminal, run:

```
touch hello.c
```

This will open the editor. Write the following C code for the "Hello, World!" program:

```
#include <stdio.h>
```

```
int main() {  
    printf("Hello, World!\n");  
    return 0;  
}
```

Save the file as hello.c.

#### STEP 5:

Compile and Execute the Program C Program:

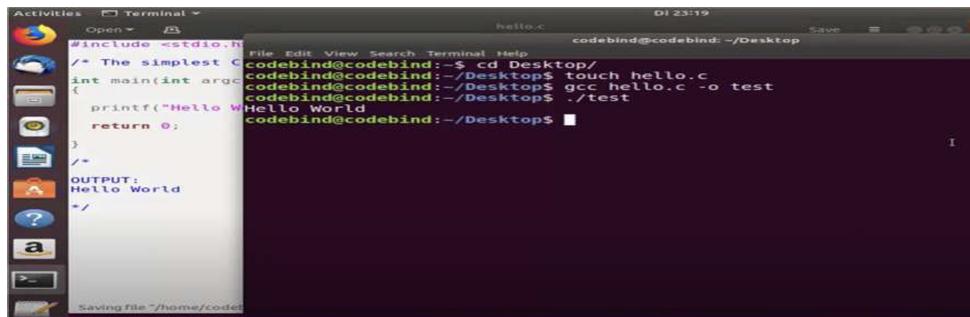
Now, compile the program using the gcc compiler. In the terminal, run:

```
gcc hello.c -o hello
```

After the compilation is successful, execute the program:

```
./hello
```

The output Hello, World! will be printed on the terminal.



```
#include <stdio.h>  
/* The simplest C program */  
int main(int argc,  
        char *argv[]){  
    printf("Hello World\n");  
    return 0;  
}  
/*  
 * OUTPUT:  
 * Hello World  
 */
```

File Edit View Search Terminal Help  
hello.c 01 23:19 codebind@codebind: ~/Desktop  
codebind@codebind:~\$ cd Desktop/  
codebind@codebind:~/Desktop\$ touch hello.c  
codebind@codebind:~/Desktop\$ gcc hello.c -o test  
codebind@codebind:~/Desktop\$ ./test  
Hello World  
codebind@codebind:~/Desktop\$

Figure 20: Hello World program in C

### 2.3 Result

Thus the C Compiler is installed successfully and executed a sample C program.

### 3 Experiment 3

Discuss bare metal hypervisor and Hosted hypervisor and list a few of them with detailed explanation.

#### 3.1 Bare-Metal Hypervisor

A Type 1 hypervisor, commonly referred to as a **bare-metal hypervisor**, represents a sophisticated virtualization framework that interfaces directly with the system's physical hardware infrastructure. As a primary control layer, it orchestrates both the underlying hardware resources and the multiple guest operating systems operating above it. By eliminating intermediary layers, this direct hardware interaction enables superior performance metrics compared to alternative virtualization approaches.

In contrast, Type 2 hypervisors, known as **hosted hypervisors**, operate within the ecosystem of a conventional operating system. This architectural design introduces an additional abstraction layer, as all hardware communications must traverse through the host operating system. The consequence of this layered structure manifests in heightened latency within the virtual machine environments, as each instruction must navigate through the host OS before reaching the physical hardware.

#### 3.2 Advantages of Bare-Metal Hypervisors

- Resource management efficiency
- System performance optimization
- Hardware utilization patterns
- Overall virtualization overhead

#### 3.3 Top Bare-Metal Hypervisors

There are several popular bare-metal hypervisors, both commercial and open-source. Some of the most well-known options include:

---

1. Kernel-based Virtual Machine (KVM)
2. Microsoft Hyper-V
3. Oracle VM Server for x86
4. VMware ESXi
5. Xen Project
6. RedHat Virtualization

### **3.4 Overview of Popular Bare-Metal Hypervisors**

#### **KVM (Kernel-based Virtual Machine)**

KVM (Kernel-based Virtual Machine) represents a unique hybrid hypervisor architecture that combines aspects of both Type 1 and Type 2 virtualization. While it leverages direct hardware virtualization capabilities like a bare-metal hypervisor.

#### **Microsoft Hyper-V**

Hyper-V is a Type 1 hypervisor that runs directly on hardware. It offers excellent resource allocation for VMs, such as dynamic memory adjustments.

#### **VMware ESXi**

VMware ESXi is a lightweight, Type 1 hypervisor with a small footprint of only 150MB.

### **3.5 Hosted Hypervisors**

Hosted hypervisors (Type 2) run on top of an existing operating system. Some popular hosted hypervisors include:

1. VirtualBox
  2. QEMU
  3. Parallels Desktop
  4. VMware Workstation Pro
-

## **VirtualBox**

Oracle VM VirtualBox is a versatile, free, and open-source hosted hypervisor that functions across multiple platforms including Windows, Linux, macOS, and Solaris. It offers comprehensive virtualization features including support for diverse guest operating systems, snapshot management for system state preservation, live VM migration capabilities, advanced networking configurations (NAT, bridged, host-only), and seamless file sharing between host and guest systems. The platform enhances user experience through features like seamless mode integration, multi-monitor support, and cross-platform compatibility, while the VirtualBox Extension Pack provides additional functionality such as USB device support and remote desktop capabilities. For advanced users and automation needs, VirtualBox includes a robust command-line interface (VBoxManage) alongside its traditional GUI, making it an ideal choice for both personal and professional virtualization requirements, particularly among developers and system administrators requiring local virtual environments.

## **QEMU**

QEMU (Quick Emulator) operates as a Type 2 hypervisor that employs dynamic binary translation for processor emulation, particularly effective when integrated with KVM for enhanced performance. It excels in supporting multiple CPU architectures (x86, ARM, PowerPC, MIPS, SPARC), making it invaluable for cross-platform virtualization needs. The platform offers comprehensive features including snapshot management, network emulation capabilities, and extensive disk and device emulation support. QEMU's versatility makes it particularly useful in testing and development environments, especially when actual hardware access is limited. When combined with KVM, it achieves notable performance improvements through hardware acceleration, while maintaining its robust capabilities for complex network configurations and storage device emulation, making it a comprehensive solution for both development and production environments.

## **Parallels Desktop**

Parallels Desktop is a sophisticated hosted hypervisor, officially sanctioned by Microsoft for running Windows on Apple Silicon processors (M1, M2), that offers seamless integration between macOS and Windows environments. It combines robust security features including VM encryption and multi-factor authentication with advanced capabilities such

---

as cloud integration (Azure, AWS), GPU acceleration, and DirectX/OpenGL support. The platform's standout features include Coherence Mode for native-like Windows application execution, comprehensive VM configuration options, and optimized performance for resource-intensive applications. With seamless macOS integration supporting features like Time Machine backup and Retina display compatibility, along with efficient cross-platform functionality, Parallels Desktop delivers a high-performance virtualization solution particularly suited for both individual users and enterprises requiring sophisticated graphics capabilities and development environments on Apple hardware.

### **VMware Workstation Pro**

VMware Workstation Pro is a powerful hosted hypervisor designed for creating and managing virtual machines (VMs) on Windows and Linux systems, catering primarily to developers, IT professionals, and system administrators. It features cloning capabilities, allowing users to quickly create linked or full clones of VMs for testing or deployment purposes. The remote management functionality enables control over local and remote VMs via vCenter and ESXi hosts, enhancing operational flexibility. Security is prioritized through encryption, safeguarding sensitive data within VMs, which is crucial for compliance with security policies. Users can utilize snapshot functionality to save and restore VM states, facilitating error recovery and configuration management. Advanced network simulation capabilities allow for the creation of isolated networks for testing without impacting the host system, while virtual networking options enable custom network setups with NAT and bridged connections. VMware Workstation Pro supports a wide array of operating systems, optimizing resource allocation for CPU, memory, and disk space according to user needs. Additionally, it integrates seamlessly with VMware vSphere, allowing for the migration of VMs between environments, thus providing a robust hybrid cloud experience.

## 4 Experiment 4

To demonstrates application-level virtualization

### 4.1 Aim

To demonstrates application-level virtualization by installing and using Notepad++ on a Linux system via Wine, which allows running Windows applications on Linux.

### 4.2 Procedure

#### STEP 1:

Ensure Wine is Installed in the System, To install Wine on a Linux system, open a terminal and run the following command:

```
sudo apt update
```

```
sudo apt install wine64
```



Figure 21: Installer

#### STEP 2:

Download the Notepad++ Installer: Visit the official Notepad++ website to download the installer for Linux systems using Wine.

---

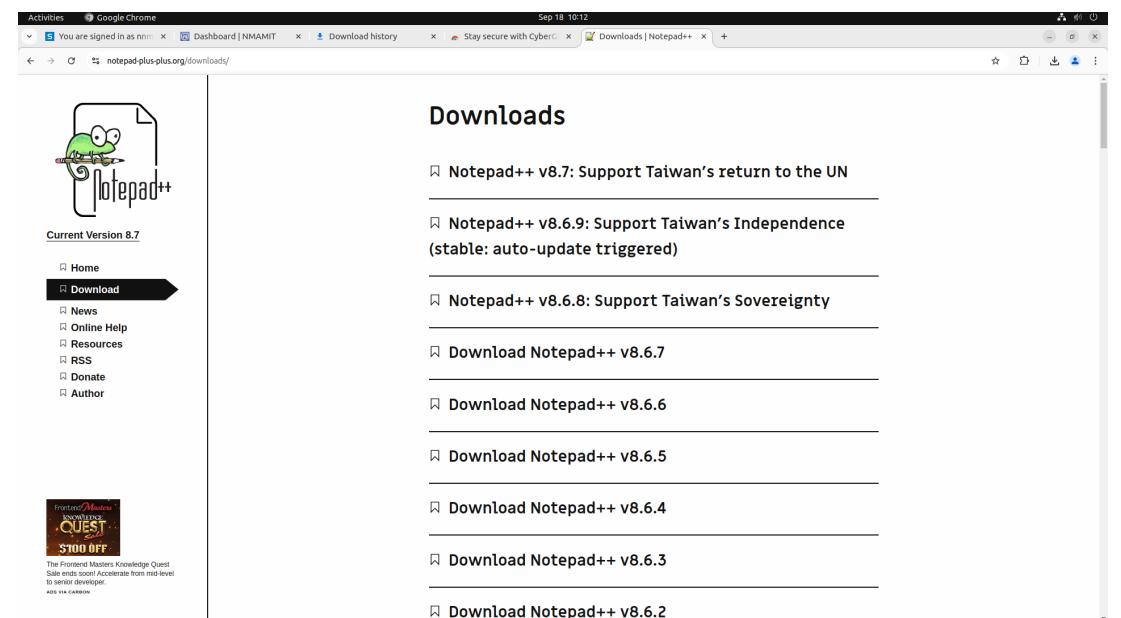


Figure 22: Official website of Notepad++

### STEP 3:

Select Wine as the Application to Run the Installer: In the context menu, select Open With Wine or Run with Wine

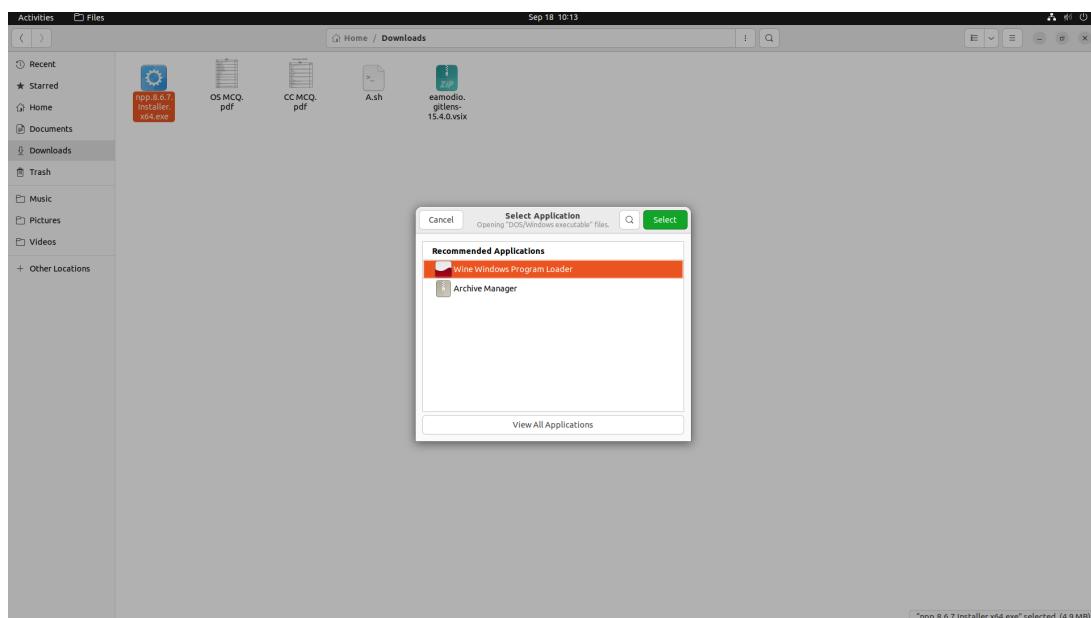


Figure 23: select WINE application

### STEP 4:

Follow the Instructions in the Notepad++ Installation Wizard: During the Wine application launch, the Notepad++ installation wizard will initiate automatically. To proceed

with the installation, click "I Agree" to accept the terms and conditions, then choose your preferred installation directory - either opt for the default Wine directory or specify a custom location of your choice.

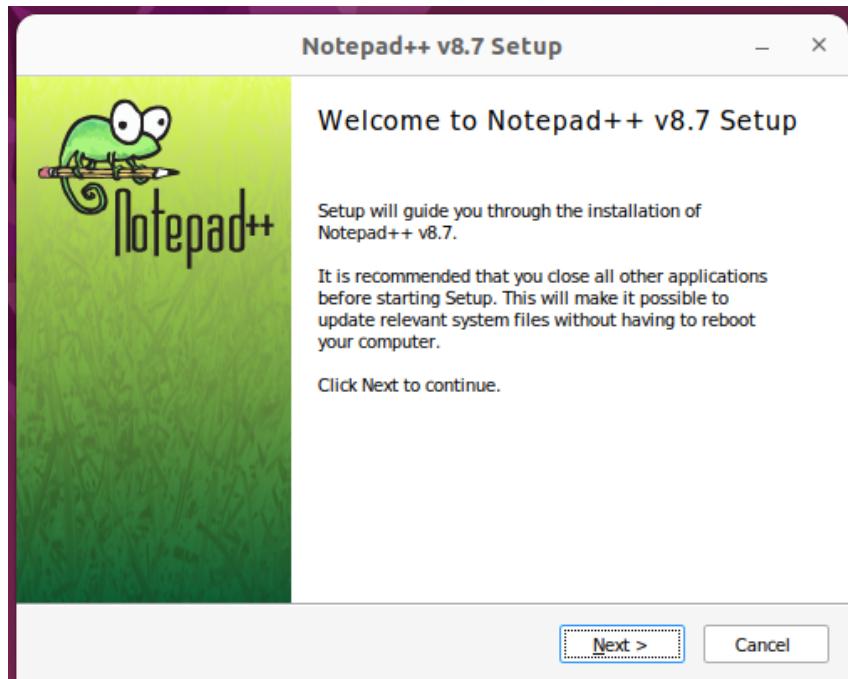


Figure 24: Notepad++ Setup

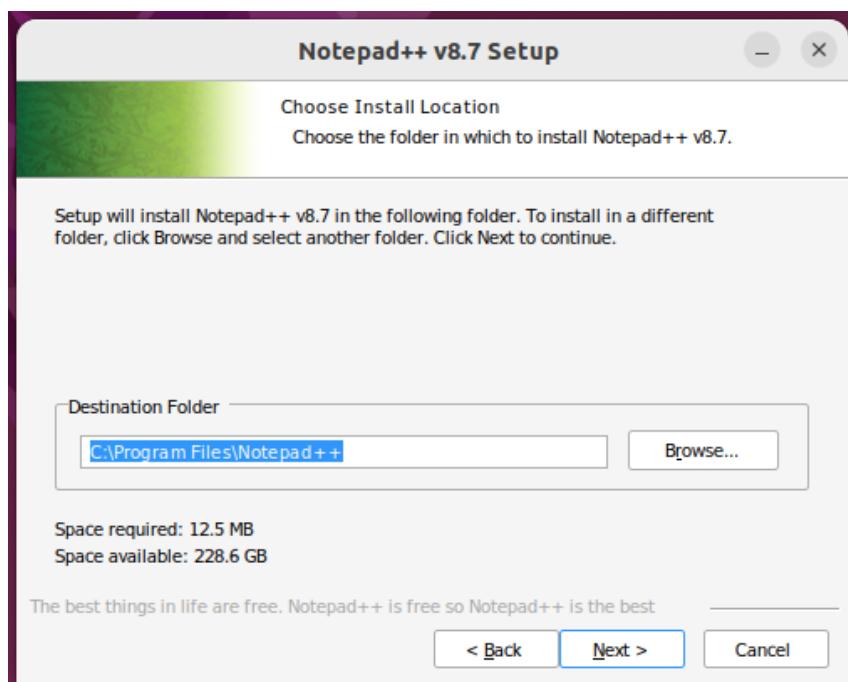


Figure 25: Notepad++ Setup Wizard

---

## STEP 5:

---

Click Install and Finish the Setup: After confirming all installation settings, click Install to begin the process. Once the installation is complete, click Finish to exit the installer.

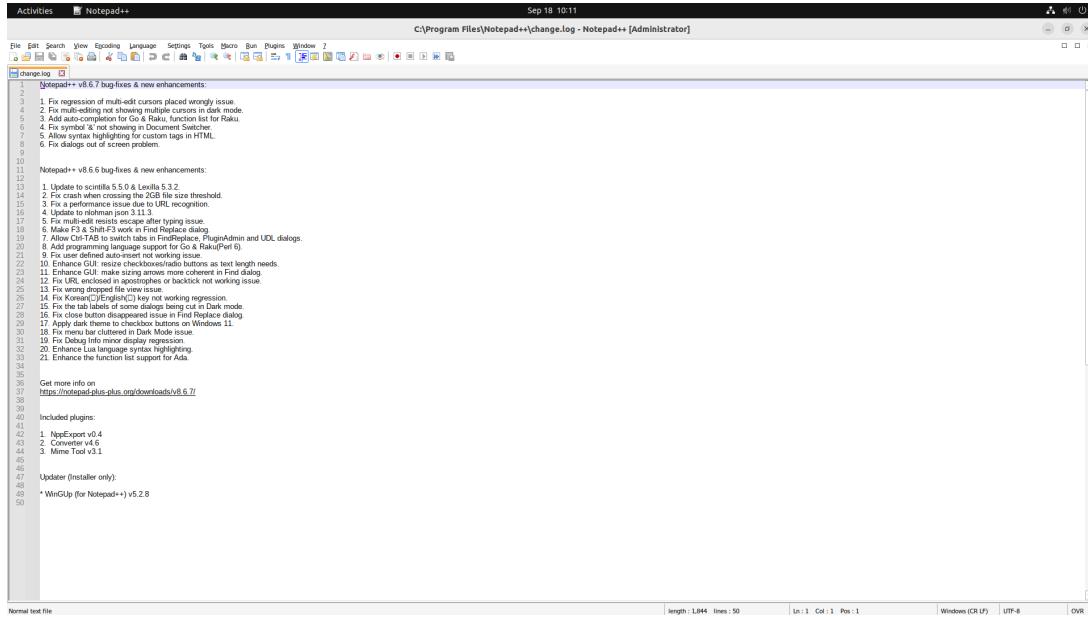


Figure 26: First view of Notepad++

### STEP 6:

Create and Save a New Text File in Notepad++: When Notepad++ launches, create a new text file by selecting File→New from the menu. Enter your desired text content. Once you've added the content, save your work by clicking File→Save As and choose your preferred storage location for the file.

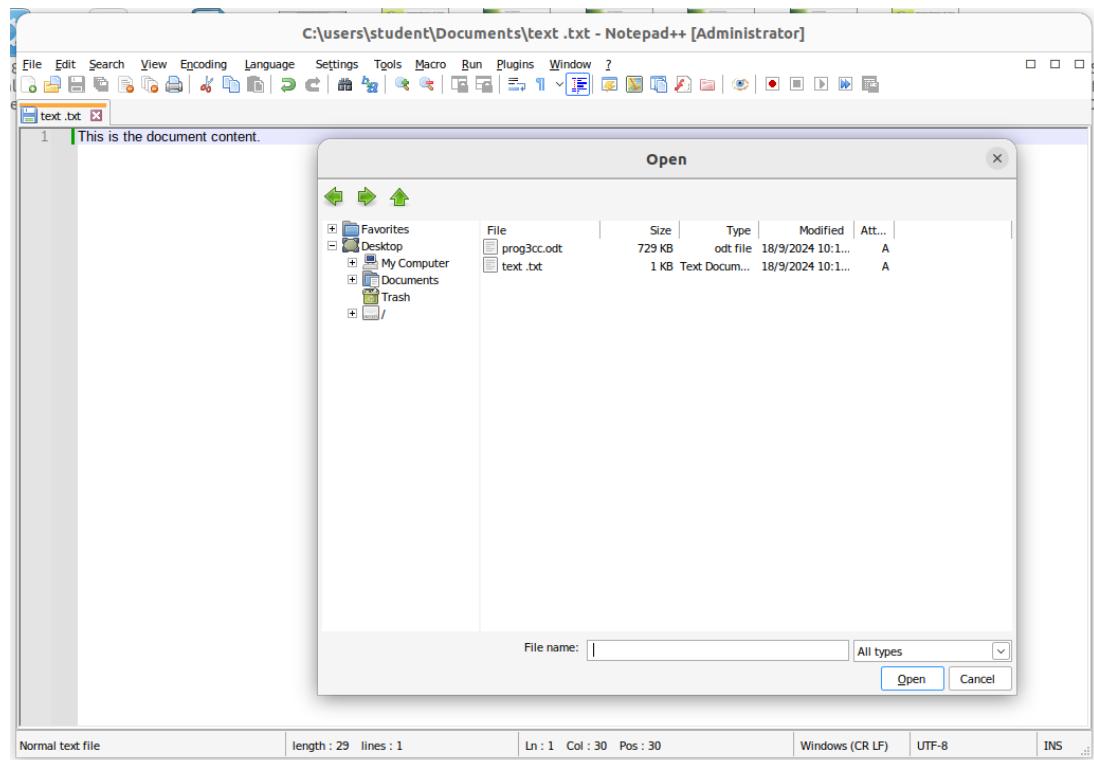


Figure 27: Creating and saving the new file

### 4.3 Result

In this experiment we successfully demonstrated application-level virtualization by running Notepad++ (a Windows application) efficiently on a Linux system through Wine, proving cross-platform compatibility without requiring native Windows OS installation.

## 5 Experiment 5

Demonstrating and Implementing PaaS to Deploy a Simple Web Application

### 5.1 Aim

To Demonstrate and Implementing PaaS to Deploy a Simple Web Application

### 5.2 Procedure

#### Prerequisites

- **Google Cloud Account:** Ensure that you have a Google Cloud Account.
- **Google Cloud SDK:** Install the Google SDK on your machine to manage GCP resources from the command line.

To deploy a simple "Hello World" application using Platform as a Service (PaaS) on Google Cloud Platform (GCP), we will use Google App Engine with a Node.js application. The steps are as follows:

#### STEP 1:

To begin with Google Cloud deployment, access the Google Cloud Console, create a new project by selecting "New Project" from the project dropdown menu, assign a project name while noting down the Project ID for future reference, and ensure billing is activated for the project to enable deployment capabilities.

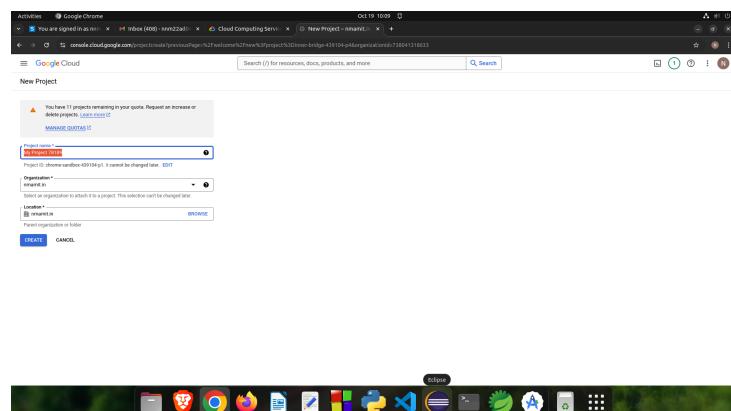


Figure 28: snapshot of creating a new project

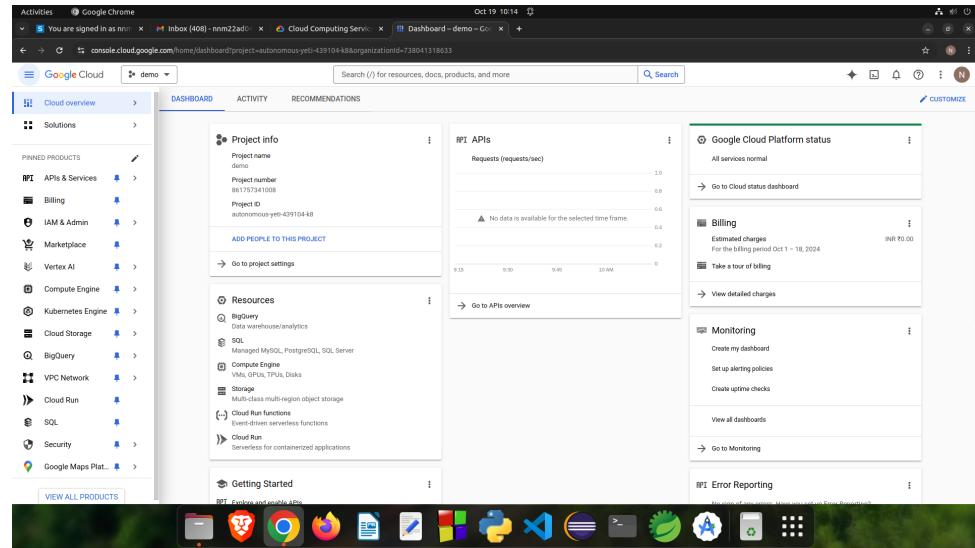


Figure 29: Shows the project details - project ID

## STEP 2:

Navigate to the App Engine page to enable your application, select your project and click "Create Application," then choose an appropriate region for deployment. Next, proceed with Node.js installation if not already present on your system using the installation command.

```
sudo apt install npm
```

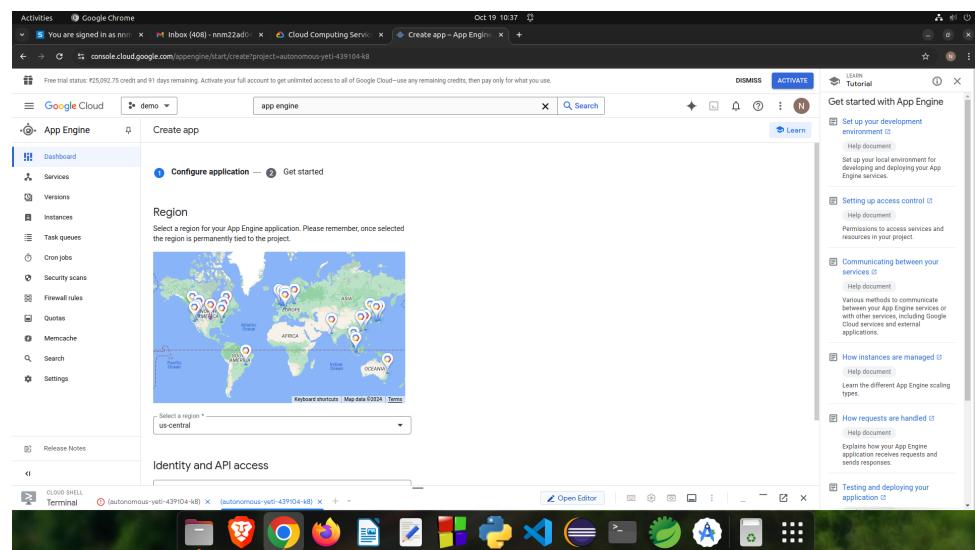


Figure 30: Configure the app engine

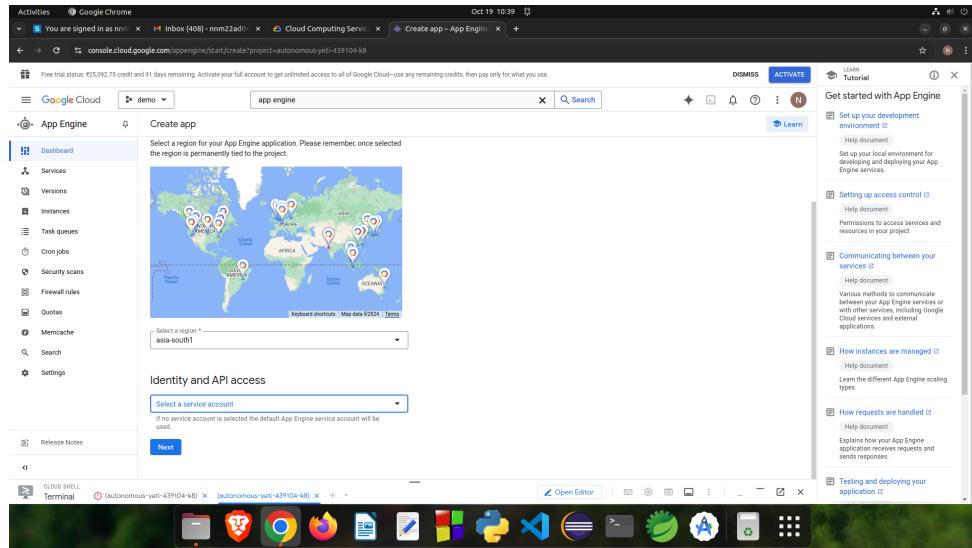


Figure 31: Select the region

### STEP 3:

Initialize Google Cloud SDK, App Engine and Open the Terminal, Authenticate with Google Cloud:

- Run the following command to log in:

```
gcloud auth login
```

### Set the Project:

- Set the project by running:

```
gcloud config set project YOUR_PROJECT_ID
```

### Initialize App Engine:

- Initialize App Engine by running:

```
gcloud app create project=YOUR_PROJECT_ID
```

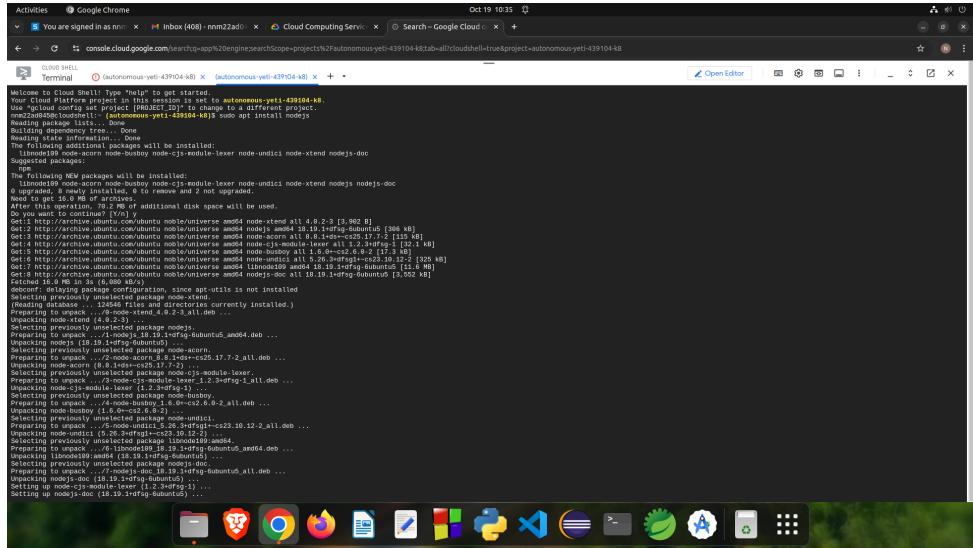


Figure 32: Installing essential dependencies

## **Step 4: Create a Sample Node.js Application**

## **1. Create a New Directory for Your Project:**

- Create a directory called ‘hello-world-app‘ and navigate into it:

```
mkdir hello-world-app
```

```
cd hello-world-app
```

## **2. Initialize the Node.js Project:**

- Initialize the Node.js project by running:

```
npm init -y
```

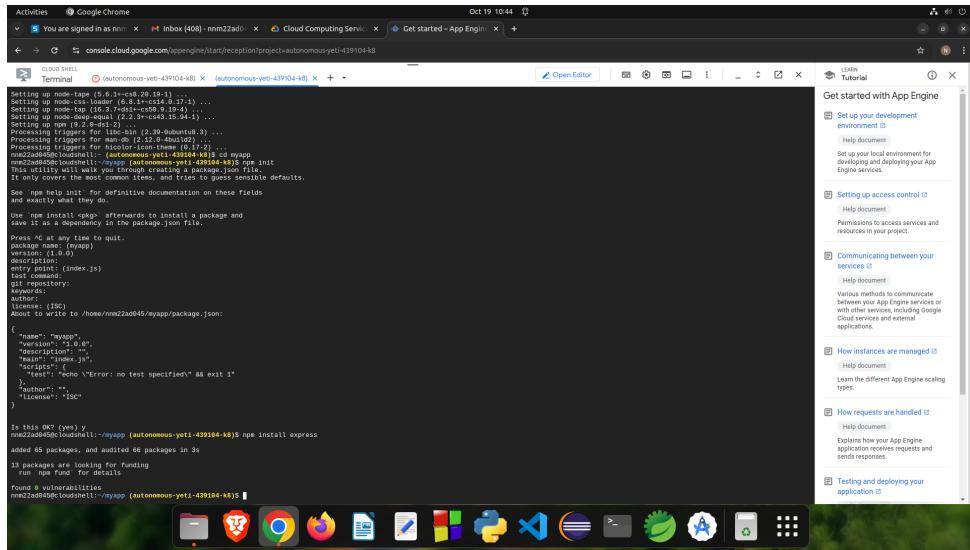


Figure 33: Follow the setup instructions to set the application

### **3. Install Express:**

- Install Express by running:

```
npm install express
```

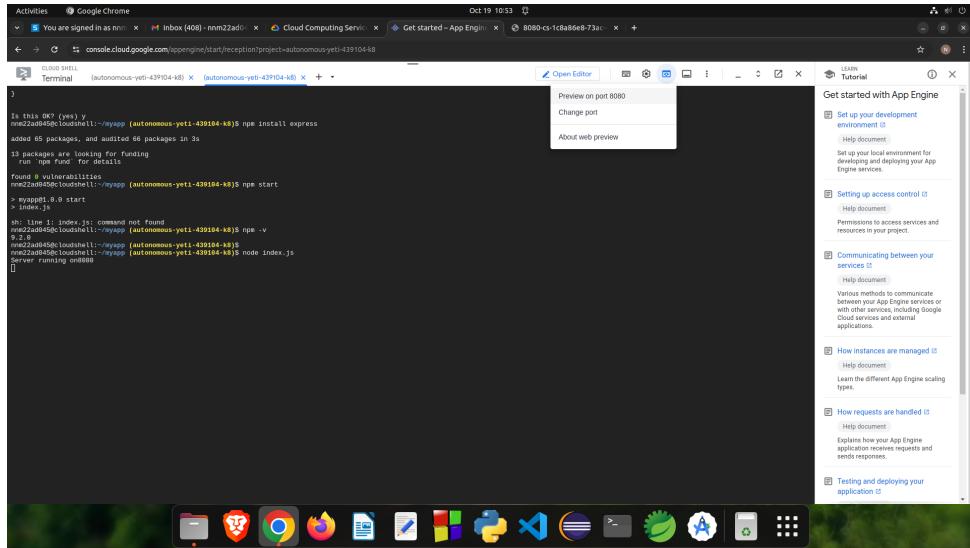


Figure 34: Installing express

#### **4. Create the Application:**

- Create a file ‘index.js‘ and add the following code:

```
const express = require('express');
```

```

const app = express();
const PORT = process.env.PORT || 8080;

app.get('/', (req, res) => {
  res.send("Hello World!");
});

app.listen(PORT, () => {
  console.log('Server is running on port ${PORT}');
});

```

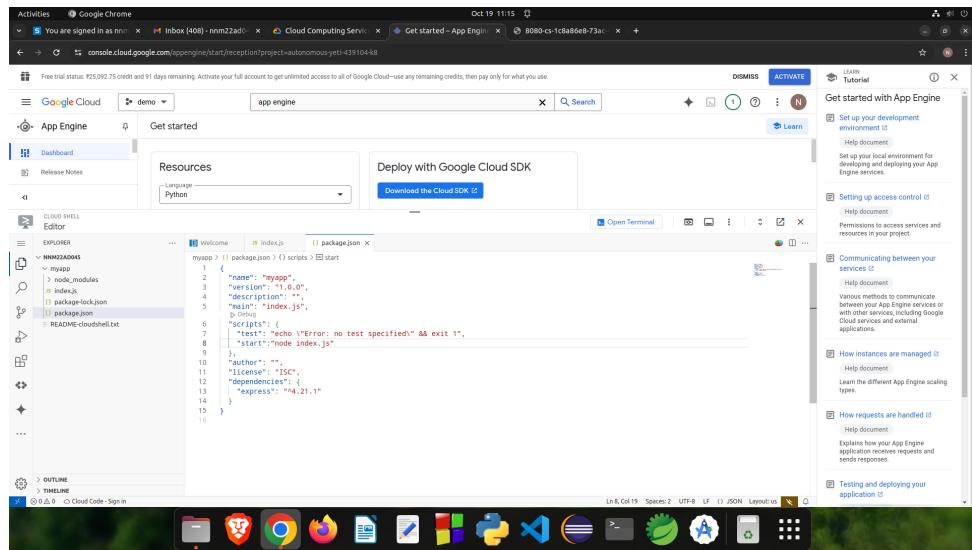


Figure 35: Index.js hello world program

## 5. Create an app.yaml File:

- In the 'hello-world-app' directory, create an 'app.yaml' file with the following content:

```
runtime: nodejs18
```

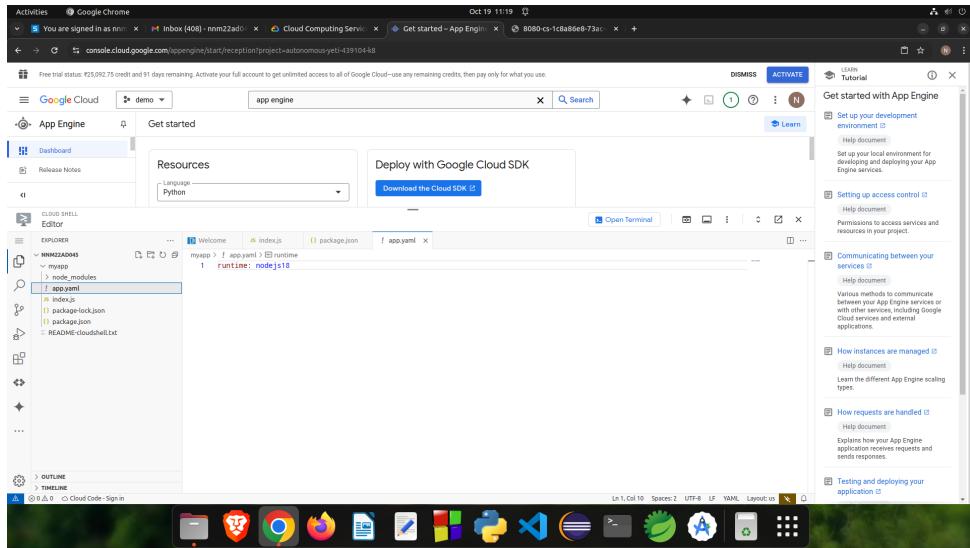


Figure 36: Runtime nodejs 18

## Step 5: Deploy to Google App Engine

### 1. Deploy the Application:

- Run the following command to deploy the application:

```
gcloud app deploy
```

## Step 6: Launch and Test the Application

### 1. Open the Application:

- Run the following command to open the application in the browser:

```
gcloud app browse
```

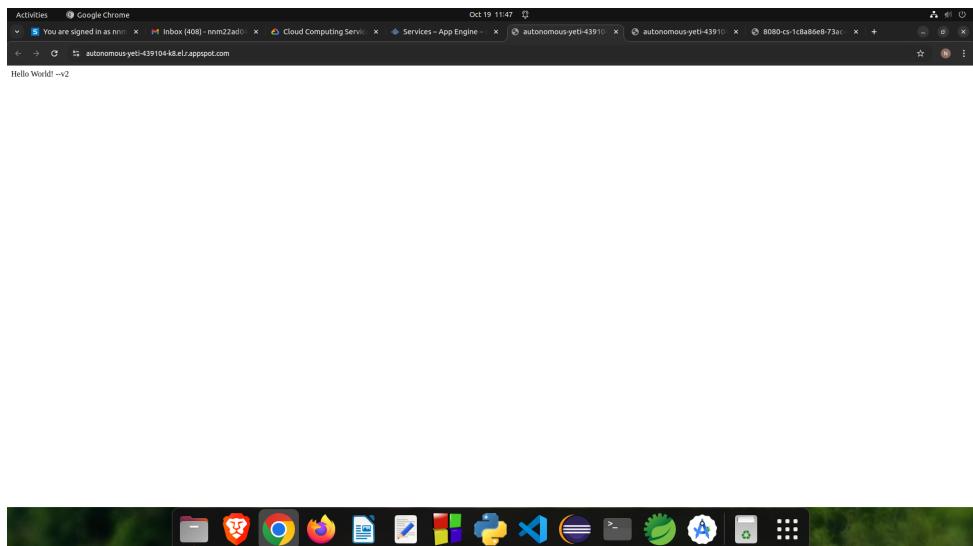


Figure 37: Snapshot of the Output

### 5.3 Result

With these steps, the "Hello World" application has been successfully deployed using Google Cloud Platform's PaaS (Google App Engine) with Node.js.