K.R. MANGALAM UNIVERSITY

THE COMPLETE WORLD OF EDUCATION

Cloud Computing

Lab File (ETCA362A)

Submitted To:

Mrs. Neetu GUPTA
(Assistant Professor)

SOET Department

Submitted By:

Harsh
Shokeen
1912730002

BTechTIIIE

Cloud Computing Lab Index

Q.1 Case Study on Cloud Computing and different Cloud vendors like AWS, AZURE, IBM, GCP

Abstract:

This case study aims to provide an in-depth analysis and comparison of cloud computing services offered by major cloud vendors, namely Amazon Web Services (AWS), Microsoft Azure, IBM Cloud, and Google Cloud Platform (GCP). Cloud computing has revolutionized the IT landscape, enabling organizations to leverage scalable and cost-effective solutions. By examining the offerings of these prominent vendors, this case study provides insights into the features, capabilities, pricing models, and customer experiences associated with each cloud platform.

Introduction

Cloud computing refers to the delivery of computing services, including storage, processing power, software applications, and databases, over the internet. It eliminates the need for organizations to invest in and manage their own physical infrastructure, such as servers and data centers, by providing access to shared and scalable resources hosted on remote servers.

The significance of cloud computing in modern IT infrastructure can be understood through the following key points:

- Scalability and Flexibility: Cloud computing offers unprecedented scalability, allowing organizations to quickly scale their resources up or down based on demand. This flexibility enables businesses to adapt to changing needs without incurring significant costs or experiencing downtime.

- Cost Efficiency: Cloud computing follows a pay-as-you-go model, where organizations pay only for the resources they consume. This eliminates the need for large upfront investments in hardware and infrastructure. Additionally, cloud providers handle maintenance and updates, reducing operational costs.

- Global Accessibility: Cloud computing enables access to data, applications, and services from anywhere with an internet connection. This fosters collaboration and remote work capabilities, facilitating seamless communication and productivity across geographically dispersed teams.

- Disaster Recovery and Data Protection: Cloud providers often offer robust disaster recovery and data backup solutions. By storing data in multiple locations, organizations can minimize the risk of data loss due to hardware failures, natural disasters, or other unforeseen events.

- Innovation and Time-to-Market: Cloud computing allows businesses to experiment and innovate rapidly. It provides easy access to a wide range of tools, frameworks, and development platforms, enabling faster development and deployment of applications and services.

- Security and Compliance: Cloud vendors invest heavily in security measures and compliance certifications to protect customer data. They often have dedicated teams of security experts and offer features such as encryption, access controls, and identity management to ensure data confidentiality and integrity.

- Big Data and Analytics: Cloud computing provides the infrastructure and services required for processing and analyzing large volumes of data. It offers scalable storage and computing capabilities, enabling organizations to leverage data-driven insights for decision-making and gaining a competitive edge.

Cloud computing has revolutionized the IT industry, empowering organizations of all sizes to access advanced technologies and capabilities that were previously limited to large enterprises. It has become an essential component of modern IT infrastructure, enabling agility, cost savings, scalability, and innovation for businesses across various sectors.

Amazon Web Services (AWS)

Amazon Web Services (AWS) is a comprehensive cloud computing platform provided by Amazon.com. It offers a vast array of services that enable organizations to build, deploy, and manage applications and infrastructure in the cloud. Here is an overview of AWS cloud services and their key features:

Compute Services:

Amazon Elastic Compute Cloud (EC2): Provides scalable virtual servers on-demand, allowing users to choose the desired computer capacity.

- AWS Lambda: Enables serverless computing, allowing developers to run code without provisioning or managing servers' storage and Content Delivery Services:
- Amazon Simple Storage Service (S3): Offers highly scalable and durable object storage for storing and retrieving data.
- Amazon Elastic Block Store (EBS): Provides persistent block-level storage volumes for EC2 instances.
- Amazon Glacier: Secure and durable storage for data archiving and long-term backup.
- Amazon Relational Database Service (RDS): Managed database service for popular relational databases like MySQL, PostgreSQL, Oracle, and SQL Server.
- Amazon DynamoDB: Fully managed NoSQL database for fast and flexible document and key-value data storage.

- Amazon Redshift: Fully managed data warehousing service for analyzing large datasets.
- Amazon Athena: Interactive query service for analyzing data stored in S3 using standard SQL.
- Amazon Kinesis: Real-time streaming data platform for collecting, processing, and analyzing data.

Case studies highlighting successful implementations of AWS in various industries:

1. **Netflix**: Netflix migrated its entire video streaming platform to AWS, taking advantage of AWS's scalability and global infrastructure to deliver seamless streaming to millions of users worldwide.
2. **Airbnb**: Airbnb relies on AWS for its infrastructure needs, enabling the company to handle peak traffic and scale rapidly during periods of high demand, ensuring a smooth booking experience for users.
3. **Capital One**: Capital One, a financial services company, migrated its infrastructure to AWS, leveraging AWS's security features and compliance certifications to ensure the protection of sensitive customer data.
4. **NASA Jet Propulsion Laboratory (JPL**): JPL uses AWS for its mission-critical workloads, including the Mars Rover project. AWS's scalability and high-performance computing capabilities enable JPL to process and analyze vast amounts of data from space missions.

Microsoft Azure

Azure, provided by Microsoft, is a comprehensive cloud computing platform that offers a wide range of services for building, deploying, and managing applications and services. Azure provides a robust and scalable infrastructure to support organizations' digital transformation initiatives. Here is an introduction to Azure cloud services and its distinguishing features:

Compute Services:

- Azure Virtual Machines: Offers scalable virtual machines that run Windows or Linux, allowing users to deploy a wide range of applications.
- Azure Functions: Provides serverless computing, allowing developers to run code in response to events without managing infrastructure.
- Azure Blob Storage: Scalable object storage for storing and retrieving large amounts of unstructured data.
- Azure Files: Fully managed file shares accessible via the Server Message Block (SMB) protocol.
- Azure Content Delivery Network (CDN): Accelerates the delivery of web content, videos, and applications globally.

- Azure Virtual Network (VNet): Provides isolated and customizable virtual networks to securely connect resources.
- Azure Load Balancer: Distributes inbound and outbound network traffic to improve availability and scalability.
- Azure Traffic Manager: Routes incoming traffic across multiple endpoints based on defined policies.
- Azure Monitor: Provides comprehensive monitoring and diagnostics for applications, infrastructure, and network resources.
- Azure Automation: Automates manual, time-consuming tasks with workflow and process automation.
- Azure Resource Manager: Simplifies resource management and allows for consistent deployment and management across Azure.
- Azure Machine Learning: Offers a cloud-based environment for building, training, and deploying machine learning models.
- Azure Cognitive Services: Provides pre-built AI models for vision, speech, language, and decision-making capabilities.
- Azure Bot Service: Enables the creation of intelligent, conversational bots using natural language understanding.

Case studies illustrating the successful adoption of Azure in different organizations:

1. **Maersk:** Maersk, a global logistics company, adopted Azure to transform its operations and improve efficiency. By leveraging Azure's scalability and data analytics capabilities, Maersk gained real-time visibility into its supply chain, optimized route planning, and enhanced customer experience.
2. **Adobe:** Adobe, a leading software company, migrated its Creative Cloud services to Azure. This migration enabled Adobe to provide a seamless experience to its users, scale resources based on demand, and achieve global availability and performance.
3. **Rolls-Royce:** Rolls-Royce, an aerospace and defense company, chose Azure as its cloud platform to power its digital transformation. Azure's AI and IoT capabilities helped Rolls-Royce collect and analyze vast amounts of data from its aircraft engines, leading to improved maintenance processes and fuel efficiency.
4. **NBC Sports:** NBC Sports, a major sports broadcasting network, relies on Azure for streaming live sports events to millions of viewers. Azure's scalability and media services enable NBC Sports to deliver high-quality video streams, handle peak demand, and provide a seamless viewing experience.

Azure offers various pricing structures and cost optimization mechanisms to help customers manage their cloud spending effectively:

Pay-as-you-go: Customers pay for Azure services based on actual usage, with no upfront costs or long-term commitments. This model provides flexibility and allows for cost control.

IBM Cloud

IBM Cloud is a comprehensive cloud computing platform offered by IBM. It provides a wide range of infrastructure, platform, and software services to support organizations in building, deploying, and managing applications and services. Here is an overview of IBM Cloud services:

- IBM Virtual Servers: Offers scalable virtual server instances with a variety of configurations, allowing users to deploy and manage their applications.
- IBM Bare Metal Servers: Provides dedicated, single-tenant servers for workloads that require high performance, security, and control.
- IBM Cloud Foundry: An open-source platform that simplifies the deployment and management of applications across multiple cloud environments.
- IBM Kubernetes Service: Managed Kubernetes service for deploying and managing containerized applications.
- IBM Watson Assistant: Enables the creation of AI-powered virtual assistants to interact with users and provide personalized experiences.
- IBM Watson Studio: A platform for data scientists and developers to build and deploy AI models using a variety of tools and frameworks.
- IBM Watson IoT Platform: Helps organizations securely connect and manage IoT devices, collect and analyze data, and gain insights.

Case studies showcasing successful utilization of IBM Cloud in diverse industries:

1. **Medtronic**: Medtronic, a global leader in medical technology, used IBM Cloud to build and deploy a secure and scalable platform for analyzing and storing medical device data. By leveraging IBM Cloud's capabilities, Medtronic improved data management, accelerated insights, and enhanced patient care.
2. **The Weather Company**: The Weather Company, an IBM subsidiary, relies on IBM Cloud to process and analyze vast amounts of weather data in real-time. By leveraging IBM Cloud's scalability and analytics capabilities, The Weather Company delivers accurate weather forecasts and insights to millions of users worldwide.
3. **Bausch + Lomb: Bausch + Lomb,** a leading eye health company, adopted IBM Cloud to enable global collaboration, data analysis, and research. IBM Cloud provided Bausch + Lomb with a secure and scalable platform for storing and processing critical healthcare data.
4. **Bank of Tokyo-Mitsubishi UFJ**: Bank of Tokyo-Mitsubishi UFJ (BTMU) utilized IBM Cloud to develop a blockchain-based payment system. By leveraging IBM Cloud's blockchain platform, BTMU enhanced transaction security, reduced costs, and improved operational efficiency in cross-border payments.

Google Cloud Platform (GCP)

Google Cloud Platform (GCP) is a suite of cloud computing services provided by Google. It offers a wide range of services that cater to various aspects of cloud computing, including infrastructure, data analytics, and machine learning. Here is an introduction to GCP services:

Infrastructure Services:

Compute Engine: Provides virtual machines (VMs) that allow users to run applications on Google's infrastructure.

App Engine: Offers a platform for building and hosting web applications, providing automatic scaling and managed infrastructure.

Kubernetes Engine: Managed Kubernetes service that simplifies the deployment, management, and scaling of containerized applications.

Cloud Storage: Provides scalable and durable object storage for storing and accessing data from anywhere in the world.

Cloud SQL: Fully managed relational database service that supports MySQL and PostgreSQL.

Cloud Bigtable: A NoSQL wide-column database suitable for large-scale, low-latency workloads.

TensorFlow: An open-source machine learning framework that is widely used for developing and deploying ML models.

Virtual Private Cloud (VPC): Provides networking functionality for creating and managing private virtual networks.

Case studies highlight successful implementation of GCP in different use cases.

1. **Spotify:** Spotify, the popular music streaming platform, migrated its infrastructure to GCP to improve scalability and reliability. By leveraging GCP's robust data storage and processing capabilities, Spotify was able to handle massive amounts of data, enhance its recommendation algorithms, and provide a seamless user experience.
2. **Home Depot**: Home Depot, a leading home improvement retailer, utilized GCP's data analytics and machine learning services to improve its inventory management and supply chain processes. By analyzing customer data and using machine learning algorithms, Home Depot optimized inventory levels, improved product availability, and enhanced customer satisfaction.
3. **Snap Inc.**: Snap Inc., the parent company of Snapchat, turned to GCP to handle its massive data requirements and support its real-time image and video processing needs. GCP's storage, analytics, and machine learning services enabled Snap Inc. to efficiently store, process, and analyze user-generated content, delivering an engaging and personalized experience to millions of users.
4. **Airbus**: Airbus, a leading aerospace manufacturer, partnered with Google Cloud to leverage its machine learning and data analytics capabilities. By utilizing GCP's tools

and technologies, Airbus enhanced aircraft maintenance and optimized fuel consumption. The collaboration also involved analyzing vast amounts of sensor data to improve flight operations and reduce costs.

5. **Niantic**: Niantic, the company behind popular augmented reality games like Pokémon GO, chose GCP to handle the massive scale and real-time demands of its gaming platforms. GCP's infrastructure and global reach provided Niantic with the necessary resources to deliver a seamless and immersive gaming experience to millions of players worldwide.

Q2. Case study of Public, Private, Hybrid Cloud.

Volkswagen Group, one of the world's largest automobile manufacturers, embarked on a digital transformation journey that involved implementing a hybrid cloud infrastructure to support its diverse IT needs. This case study showcases the successful adoption of a hybrid cloud model by Volkswagen Group.

Challenge:

Volkswagen Group faced the challenge of managing a complex IT landscape consisting of various applications, systems, and data across multiple brands and regions. They needed a flexible and scalable cloud solution that could accommodate their diverse requirements, including data privacy and compliance regulations, while ensuring cost efficiency and operational agility.

Solution:

Volkswagen Group opted for a hybrid cloud approach, combining both public and private cloud environments to address their specific needs. They partnered with several cloud providers, including AWS (public cloud) and OpenStack (private cloud), to establish a hybrid infrastructure.

Public Cloud: Volkswagen Group utilized the public cloud for certain non-sensitive workloads and applications that required scalability and cost optimization. They leveraged AWS's services to deploy applications, run simulations, and process large datasets. The public cloud provided on-demand scalability and access to a wide range of tools and services.

Private Cloud: Volkswagen Group maintained a private cloud environment built on the OpenStack platform. This private cloud facilitated the management of critical and sensitive applications, ensuring data privacy, compliance, and enhanced security controls. The private cloud offered greater control and customization options required for specific business requirements.

Hybrid Cloud Integration: Volkswagen Group implemented a robust hybrid cloud integration strategy to seamlessly connect and manage workloads across their public and private cloud environments. They utilized various tools and technologies, such as hybrid cloud management platforms and application programming interfaces (APIs), to enable data exchange, workload migration, and workload orchestration between the two cloud environments.

Benefits:

By adopting a hybrid cloud model, Volkswagen Group achieved several significant benefits:

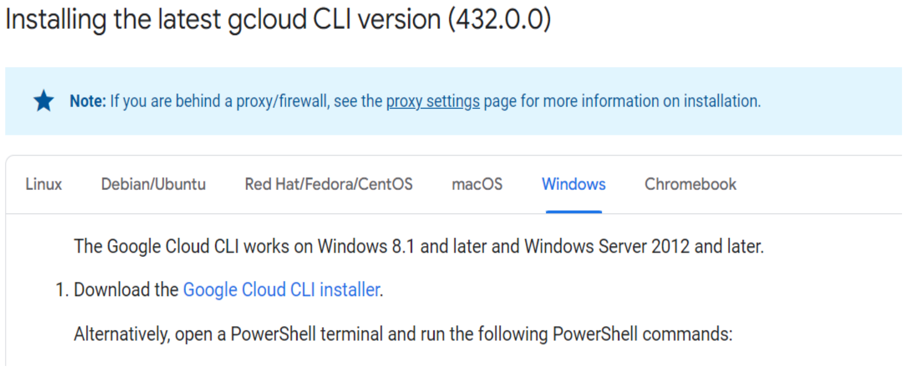
- **Flexibility and Scalability**: The hybrid cloud infrastructure allowed Volkswagen Group to scale resources up or down based on demand, ensuring optimal performance and cost efficiency for their applications and workloads.
- **Data Security and Compliance**: The private cloud component ensured that critical data and applications adhered to stringent security and compliance regulations, providing Volkswagen Group with the necessary control and governance over sensitive information.
- **Cost Optimization**: By utilizing the public cloud for non-sensitive workloads, Volkswagen Group optimized costs by paying only for the resources consumed. They could leverage the elasticity and cost-effective pricing models of the public cloud, resulting in potential cost savings.
- **Operational Efficiency**: The hybrid cloud approach enhanced operational efficiency by providing a unified management and deployment framework. Volkswagen Group could deploy workloads seamlessly across public and private clouds, enabling agility and reducing time-to-market for new applications and services.

This case study exemplifies how Volkswagen Group successfully implemented a hybrid cloud strategy, leveraging the benefits of both public and private cloud environments. The hybrid cloud infrastructure provided the necessary agility, scalability, security, and compliance capabilities required by the automotive giant, supporting their digital transformation initiatives, and ensuring efficient management of their IT ecosystem.

Q3. Install Google App Engine. Create a Hello world app and other simple web applications using python/java.

Procedure:

Step 1: Download Google Cloud SDK from the official website of Google.



Installing the latest gcloud CLI version (432.0.0)

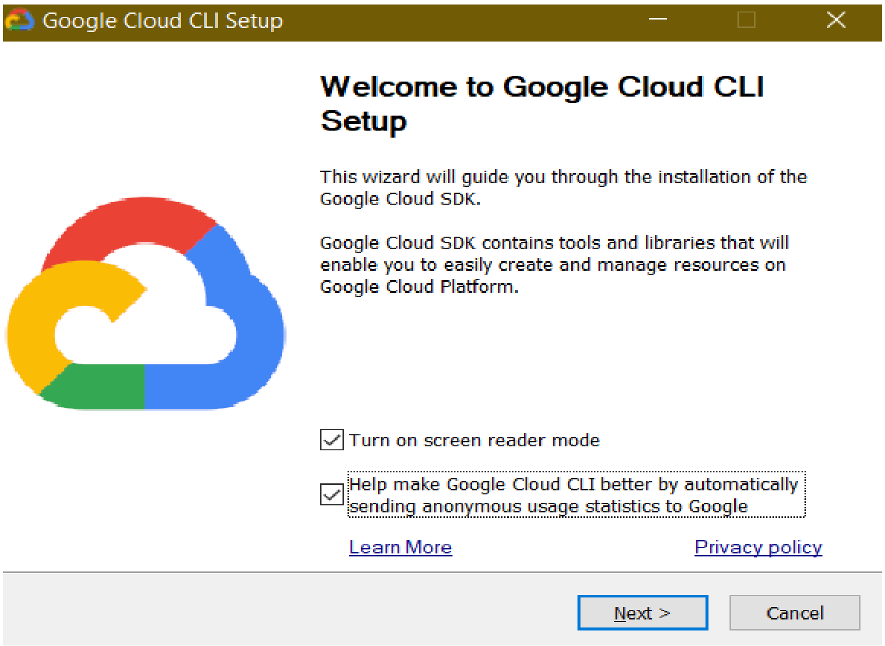
★ **Note:** If you are behind a proxy/firewall, see the proxy settings page for more information on installation.

The Google Cloud CLI works on Windows 8.1 and later and Windows Server 2012 and later.

1. Download the Google Cloud CLI installer.

Alternatively, open a PowerShell terminal and run the following PowerShell commands:

Step 2: Open the install wizard of the Google SDK/CLI(Command Line Interface)



Welcome to Google Cloud CLI Setup

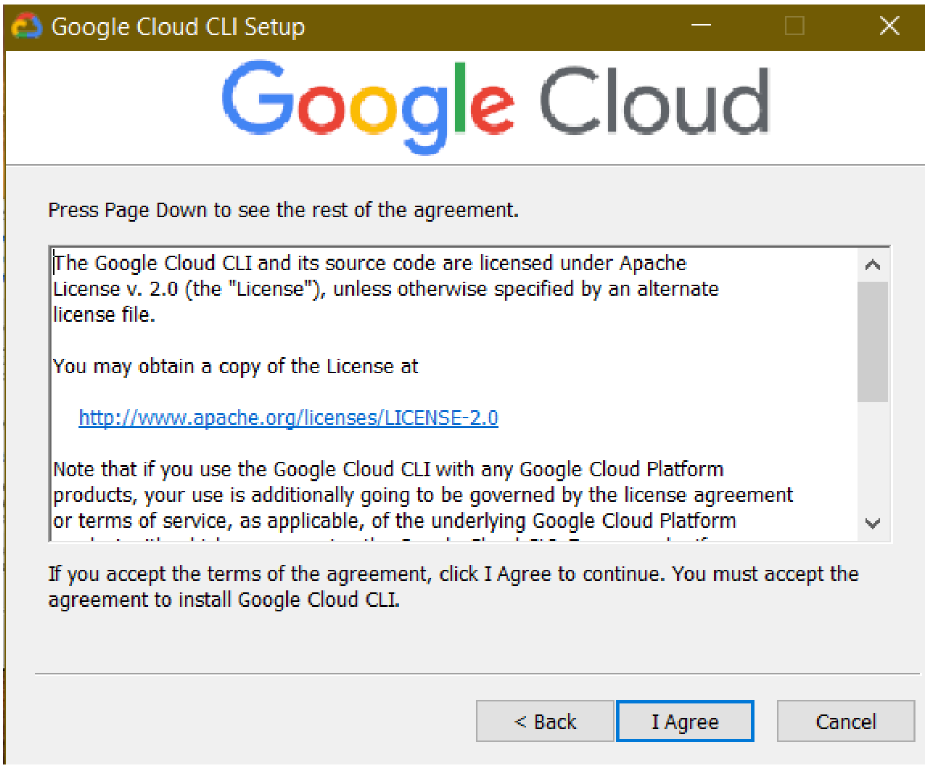
This wizard will guide you through the installation of the Google Cloud SDK.

Google Cloud SDK contains tools and libraries that will enable you to easily create and manage resources on Google Cloud Platform.

☑ Turn on screen reader mode

☑ Help make Google Cloud CLI better by automatically sending anonymous usage statistics to Google

Step 3: Agree to the steps.



Press Page Down to see the rest of the agreement.

The Google Cloud CLI and its source code are licensed under Apache License v. 2.0 (the "License"), unless otherwise specified by an alternate license file.

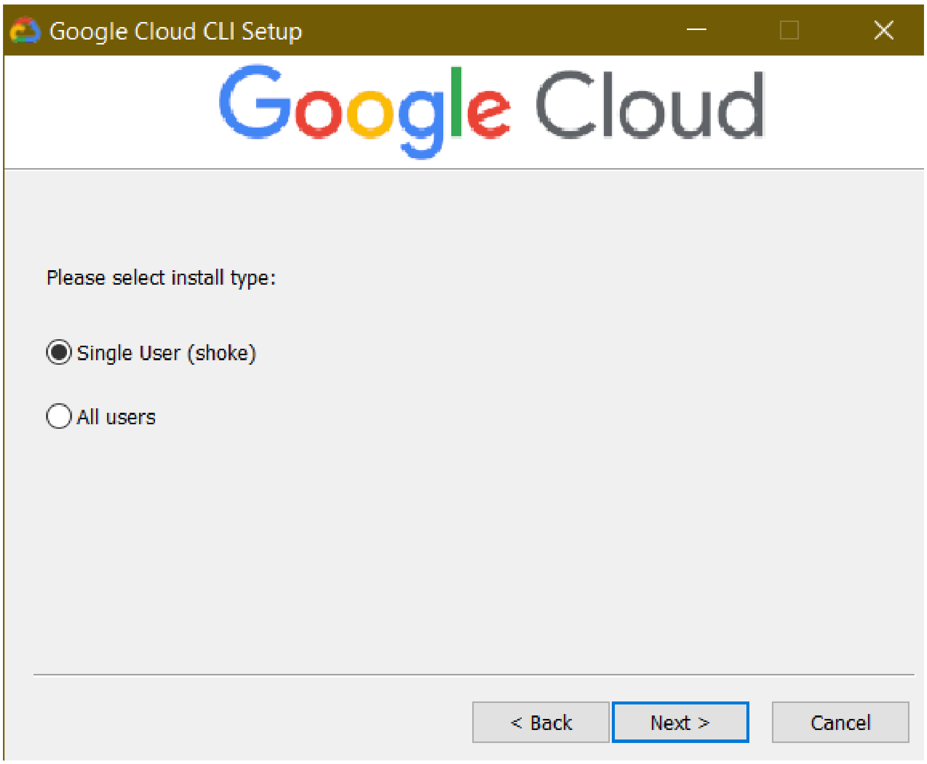
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Note that if you use the Google Cloud CLI with any Google Cloud Platform products, your use is additionally going to be governed by the license agreement or terms of service, as applicable, of the underlying Google Cloud Platform

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Google Cloud CLI.

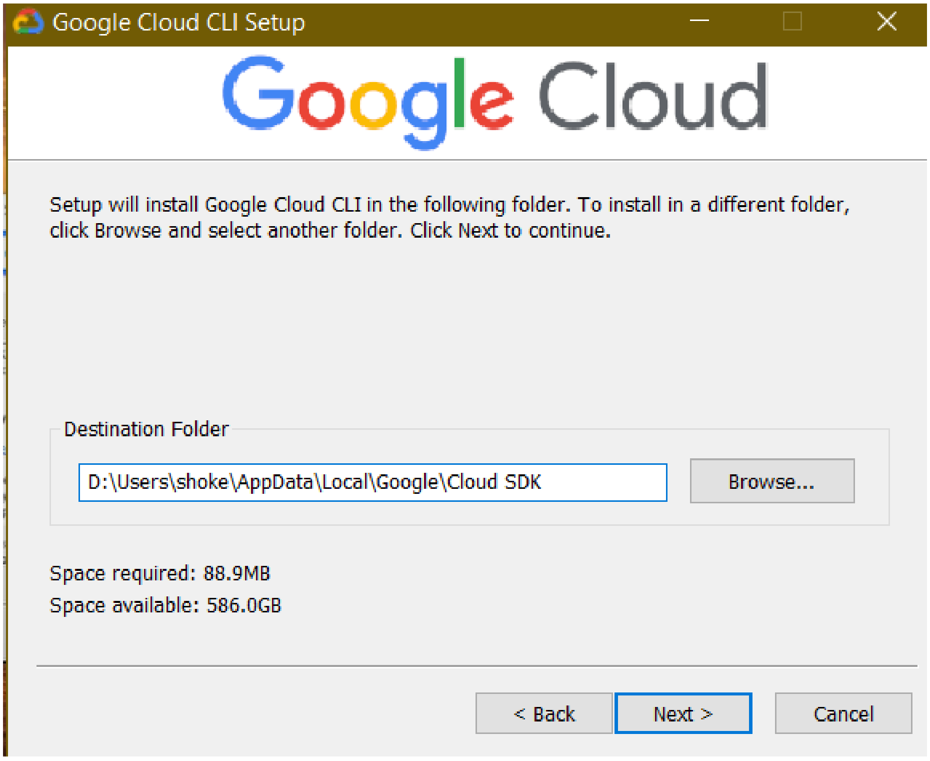
Step 4:



Please select install type:

● Single User (shoke)

○ All users



Setup will install Google Cloud CLI in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

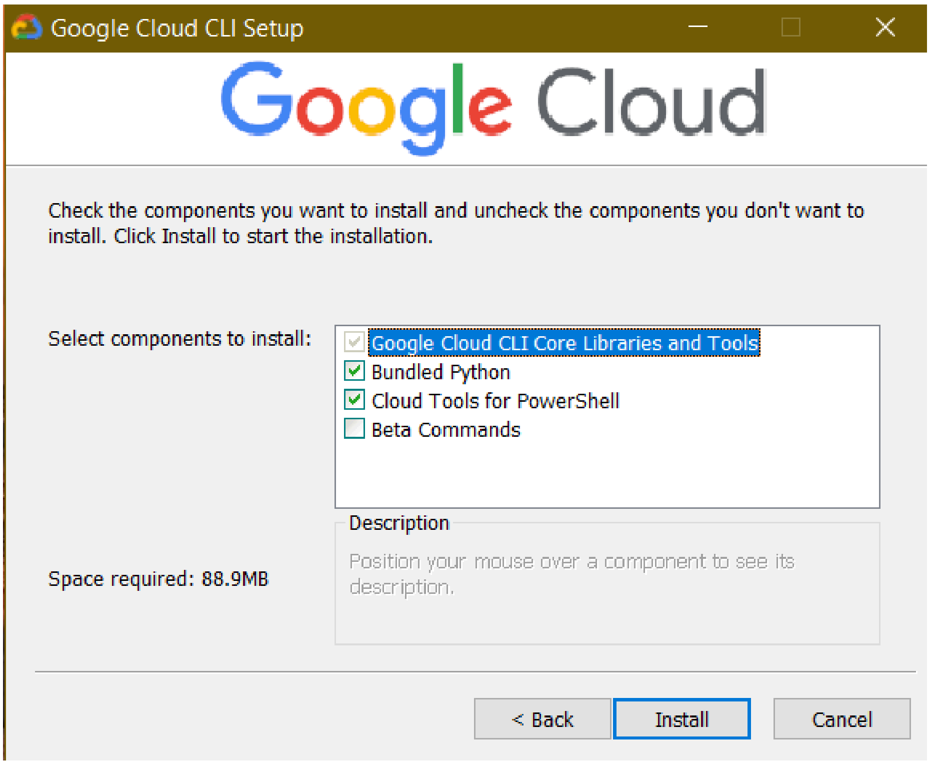
Destination Folder

D:\Users\shoke\AppData\Local\Google\Cloud SDK

Space required: 88.9MB

Space available: 586.0GB

Step 5:



Google Cloud CLI Setup

Check the components you want to install and uncheck the components you don't want to install. Click Install to start the installation.

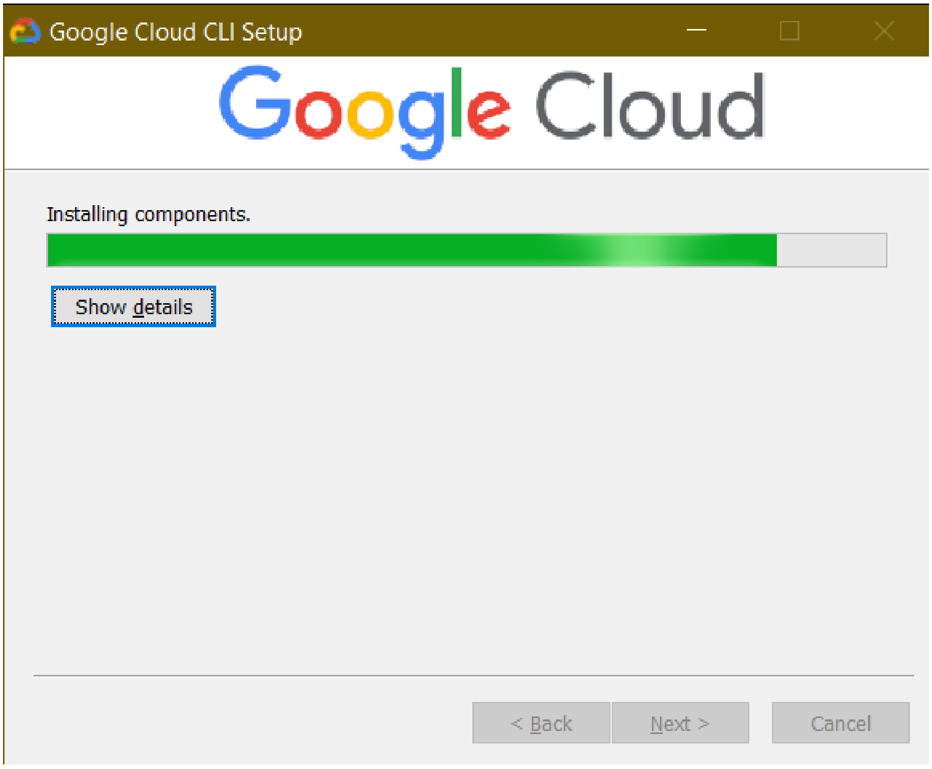
Select components to install:

☑ Google Cloud CLI Core Libraries and Tools
☑ Bundled Python
☑ Cloud Tools for PowerShell
☐ Beta Commands

Space required: 88.9MB

Description

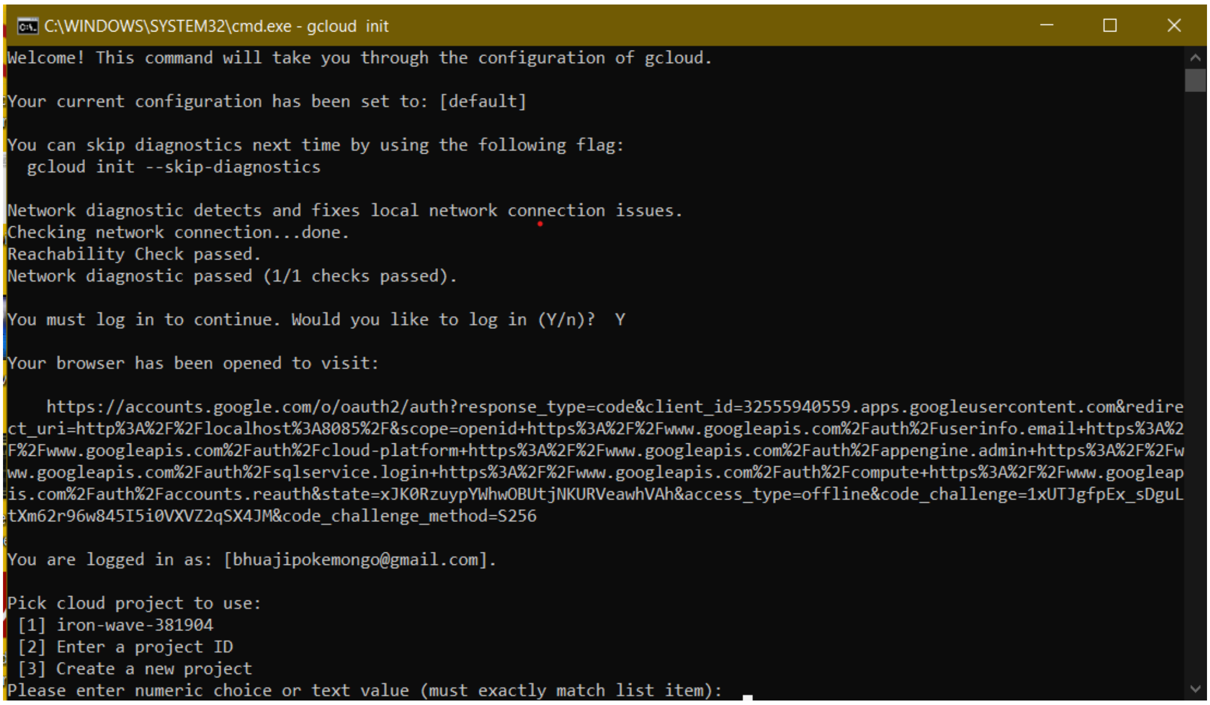
Position your mouse over a component to see its description.



Google Cloud CLI Setup

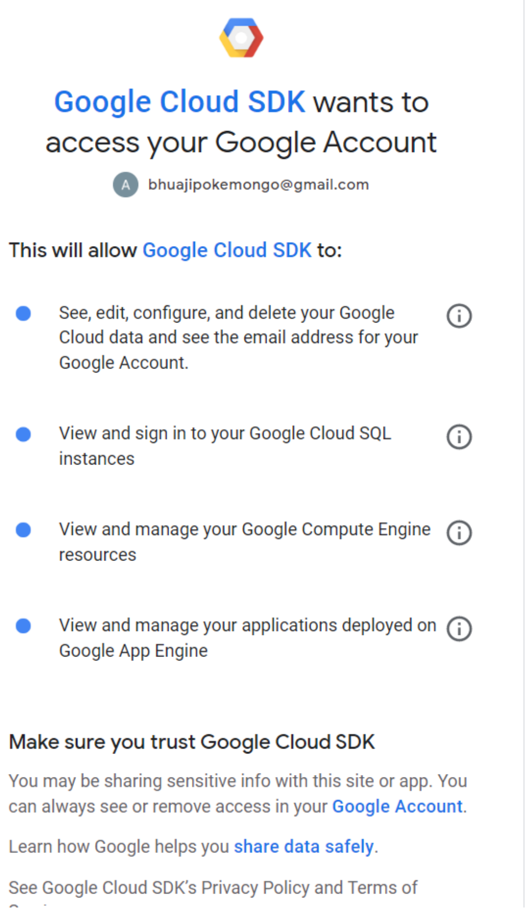
Installing components.

Step 6: After the installation of the Google Cloud SDK; Open the GCSDK and Accept the Login from your default account.



```
Welcome! This command will take you through the configuration of gcloud.

Your current configuration has been set to: [default]

You can skip diagnostics next time by using the following flag:
  gcloud init --skip-diagnostics

Network diagnostic detects and fixes local network connection issues.
Checking network connection...done.
Reachability Check passed.
Network diagnostic passed (1/1 checks passed).

You must log in to continue. Would you like to log in (Y/n)?  Y

Your browser has been opened to visit:

    https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=32555940559.apps.googleusercontent.com&redire
ct_uri=http%3A%2F%2Flocalhost%3A8085%2F&scope=openid+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2
F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fw
ww.googleapis.com%2Fauth%2Fsqlservice.login+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleap
is.com%2Fauth%2Faccounts.reauth&state=xJK0RzuypYWhwOBUtjNKURVeawhVAh&access_type=offline&code_challenge=1xUTJgfpEx_sDguL
tXm62r96w845I5i0VXVZ2qSX4JM&code_challenge_method=S256

You are logged in as: [bhuajipokemongo@gmail.com].

Pick cloud project to use:
 [1] iron-wave-381904
 [2] Enter a project ID
 [3] Create a new project
Please enter numeric choice or text value (must exactly match list item):
```

Step 7: Accept the login from your default account.



Google Cloud SDK wants to access your Google Account

bhuajipokemongo@gmail.com

This will allow Google Cloud SDK to:

- See, edit, configure, and delete your Google Cloud data and see the email address for your Google Account.
- View and sign in to your Google Cloud SQL instances
- View and manage your Google Compute Engine resources
- View and manage your applications deployed on Google App Engine

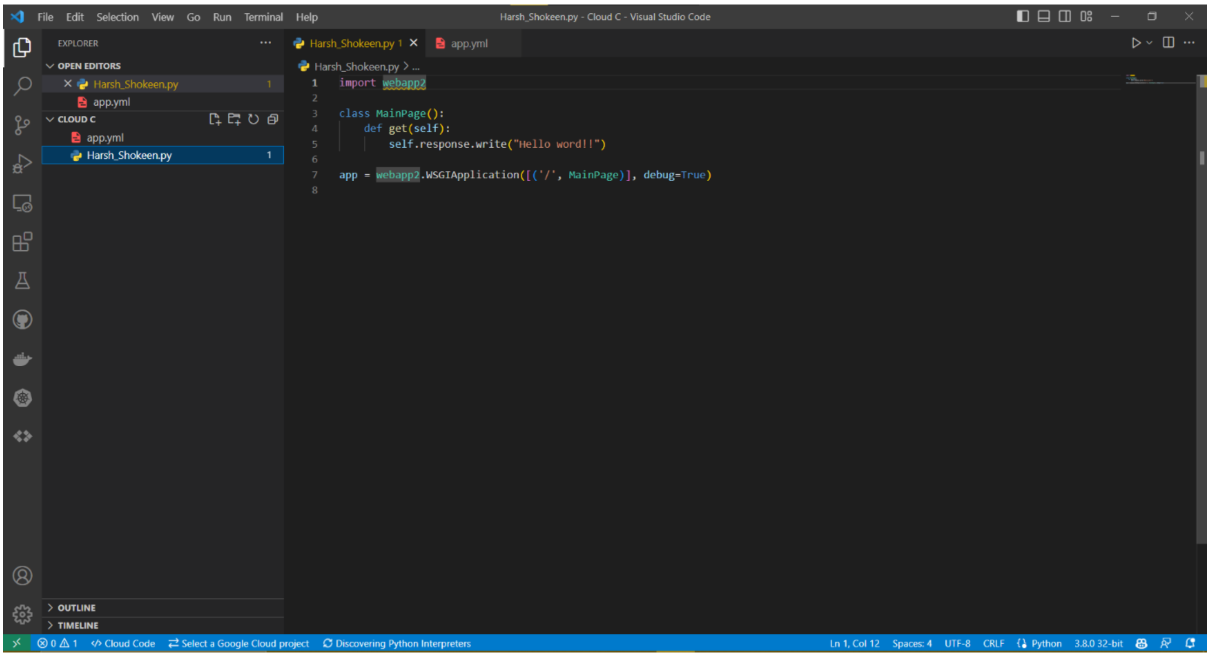
Make sure you trust Google Cloud SDK

You may be sharing sensitive info with this site or app. You can always see or remove access in your Google Account.

Learn how Google helps you share data safely.

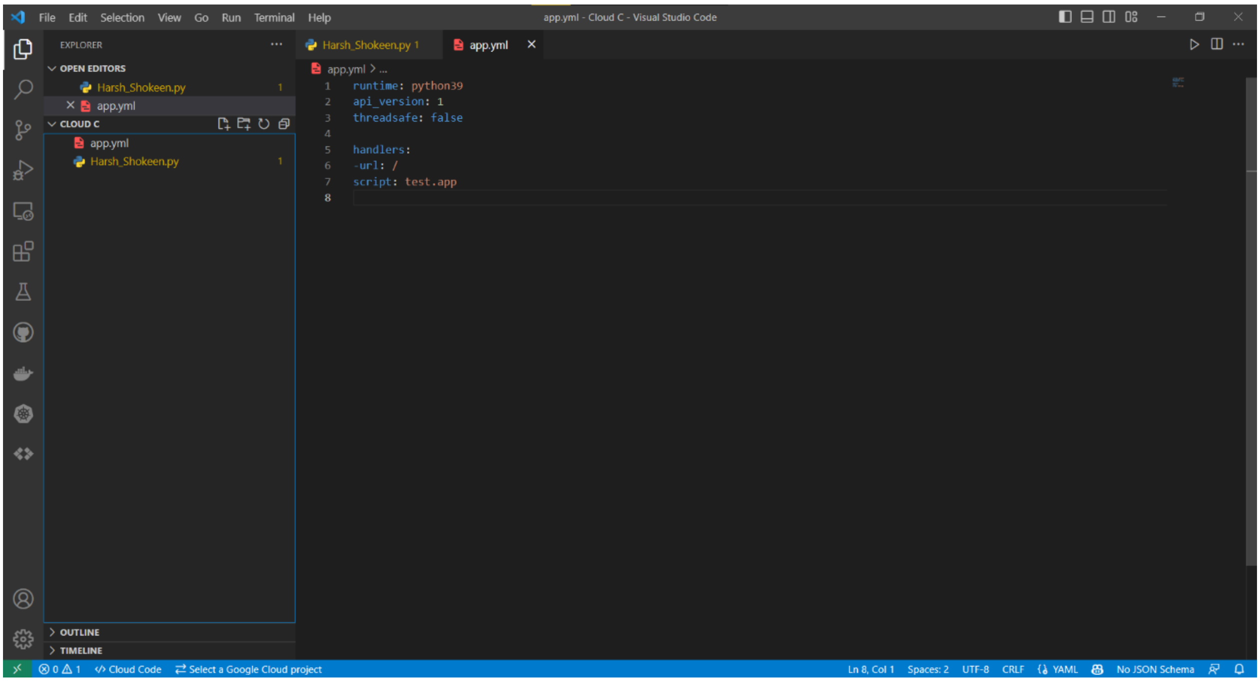
See Google Cloud SDK's Privacy Policy and Terms of

Step 8: Write the code for the Webapp in VScode



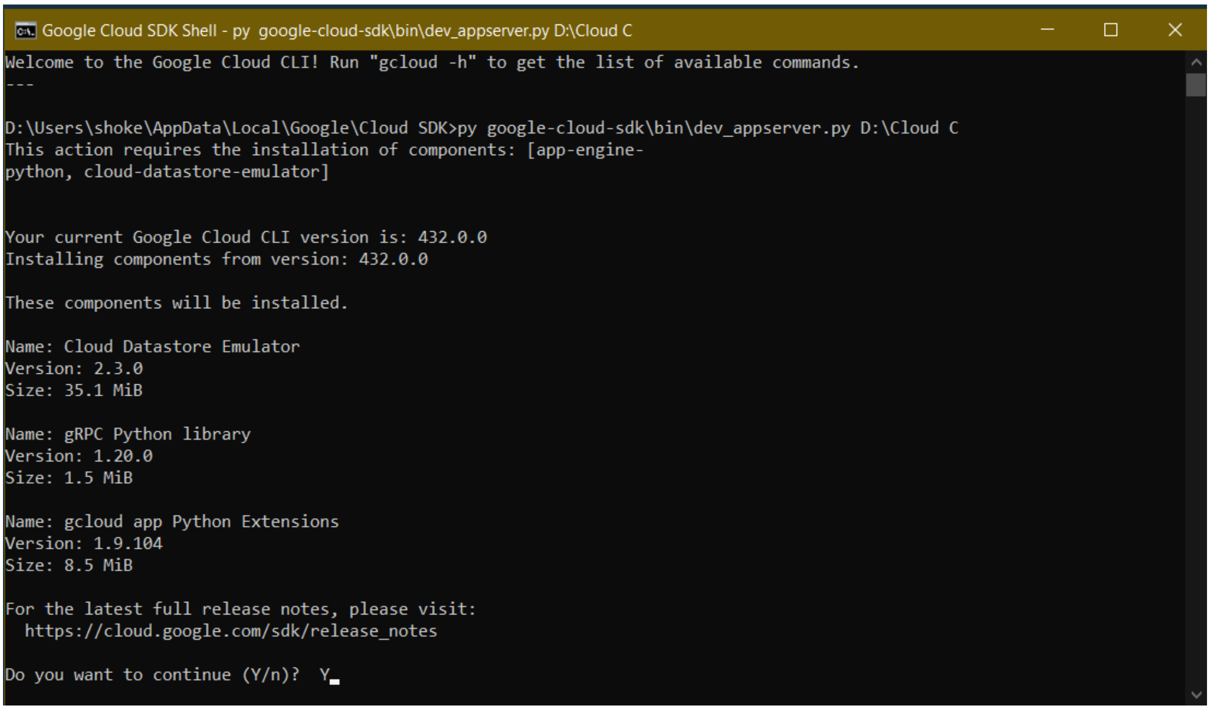
```python
import webapp2

class MainPage():
    def get(self):
        self.response.write("Hello word!!")

app = webapp2.WSGIApplication([('/', MainPage)], debug=True)
```

Step 9: Write the code for the YAML file.



```yaml
runtime: python39
api_version: 1
threadsafe: false

handlers:
-url: /
script: test.app
```

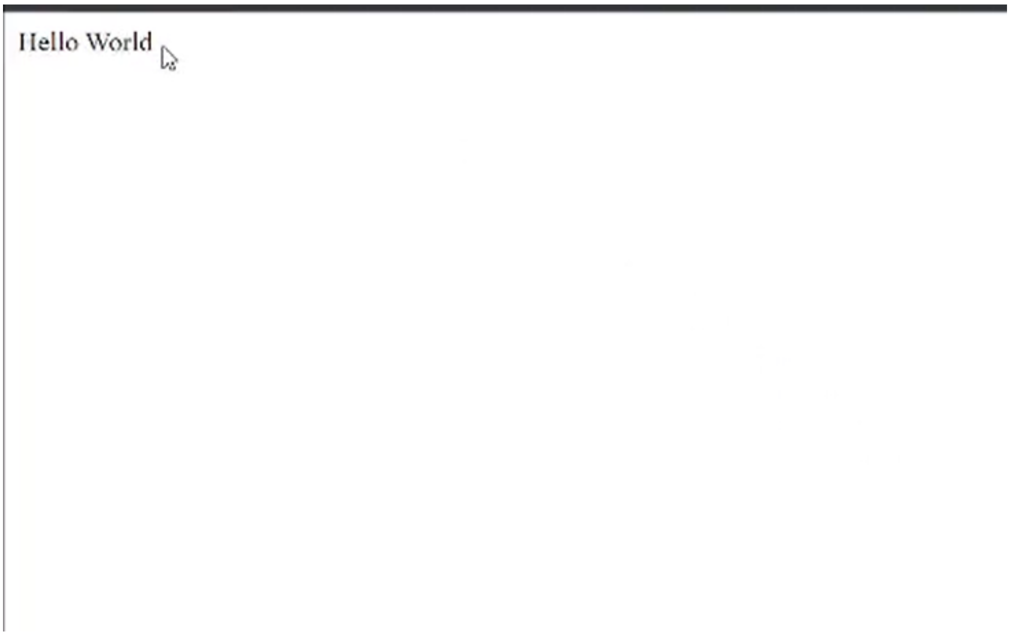
Step 10: Open the code/folder in GCSDK and run on Google App Engine



Google Cloud SDK Shell - py google-cloud-sdk\bin\dev_appserver.py D:\Cloud C

```
Welcome to the Google Cloud CLI! Run "gcloud -h" to get the list of available commands.
---

D:\Users\shoke\AppData\Local\Google\Cloud SDK>py google-cloud-sdk\bin\dev_appserver.py D:\Cloud C
This action requires the installation of components: [app-engine-
python, cloud-datastore-emulator]


Your current Google Cloud CLI version is: 432.0.0
Installing components from version: 432.0.0

These components will be installed.

Name: Cloud Datastore Emulator
Version: 2.3.0
Size: 35.1 MiB

Name: gRPC Python library
Version: 1.20.0
Size: 1.5 MiB

Name: gcloud app Python Extensions
Version: 1.9.104
Size: 8.5 MiB

For the latest full release notes, please visit:
  https://cloud.google.com/sdk/release_notes

Do you want to continue (Y/n)?  Y
```

OUTPUT:

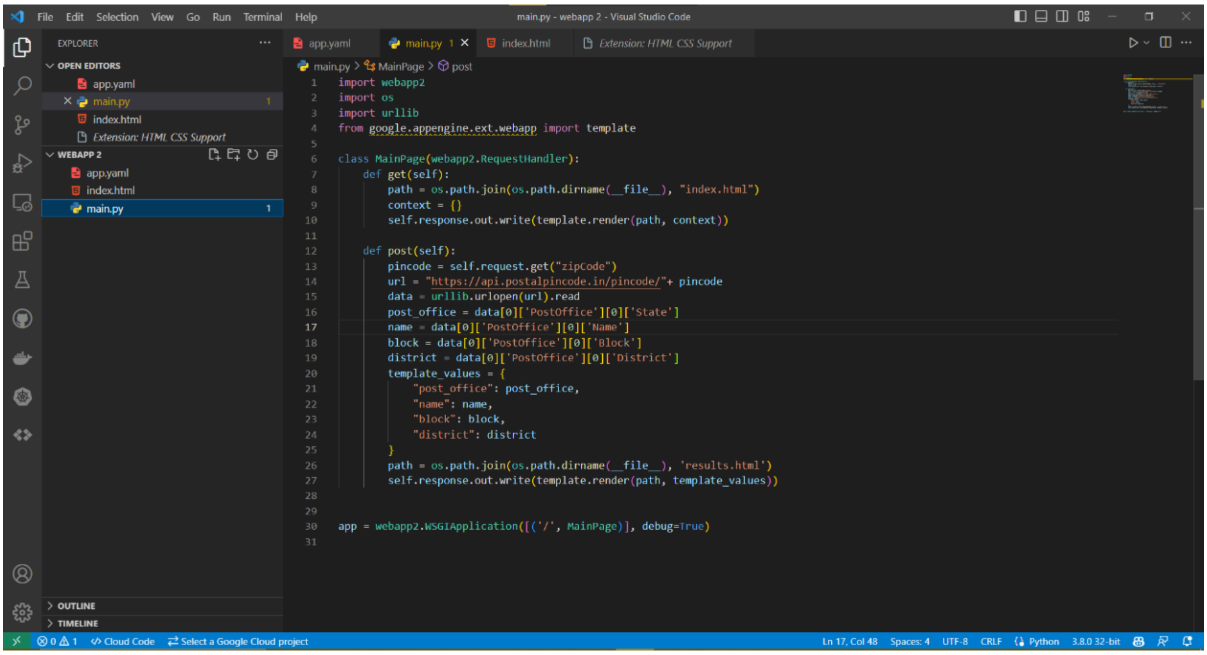


Hello World

Q4. Use GAE launcher to launch the web applications.

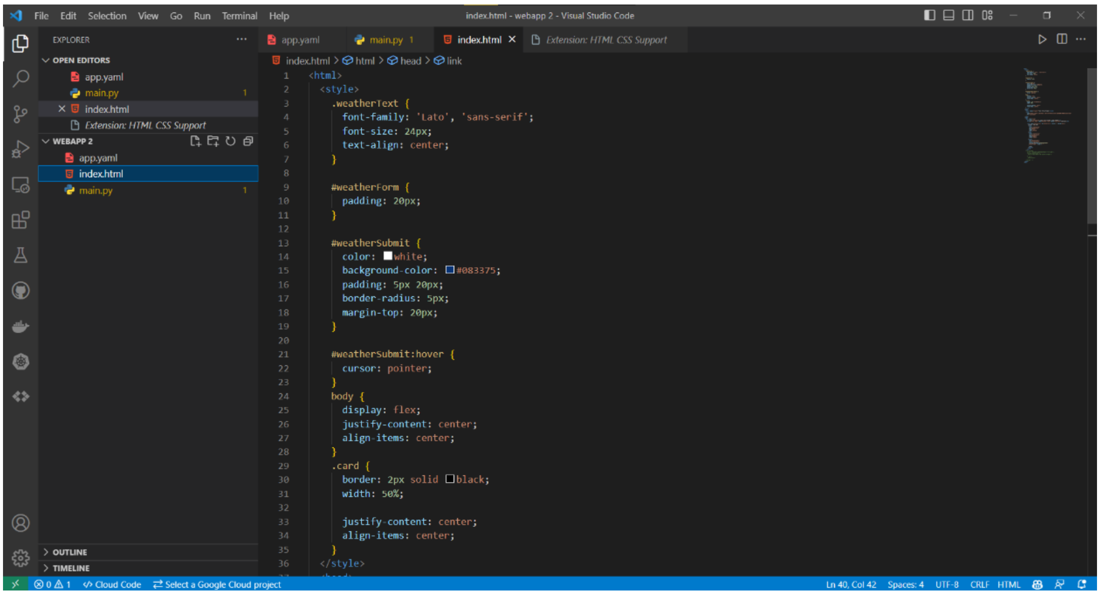
PROCEDURE:

Step 1: Write the Python code for your web application; for me it is about the fetching data of the corresponding PINCODE.



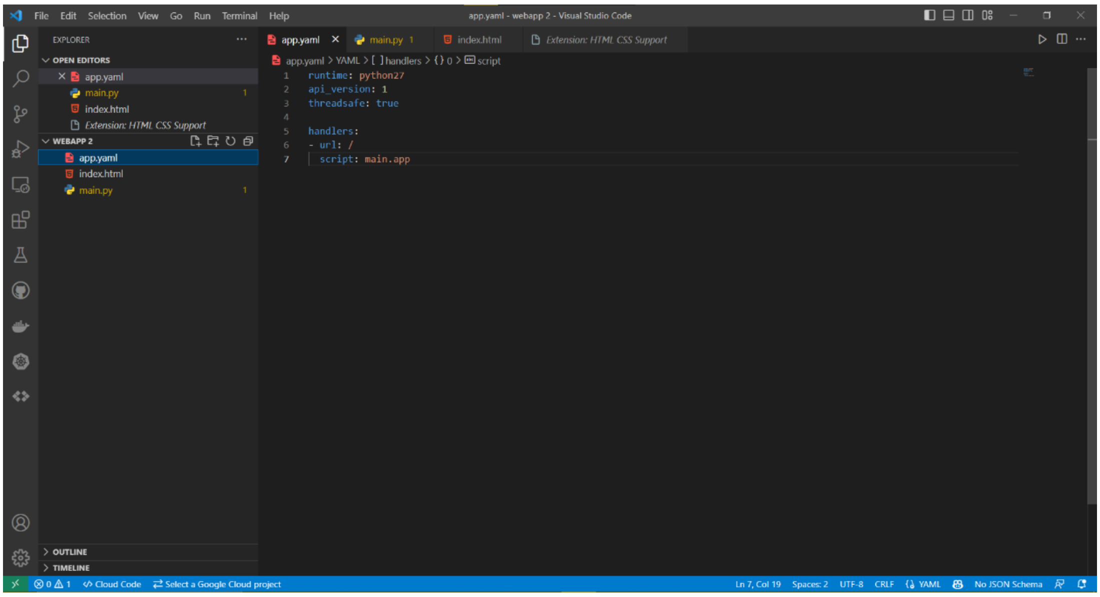
```python
import webapp2
import os
import urllib
from google.appengine.ext.webapp import template

class MainPage(webapp2.RequestHandler):
    def get(self):
        path = os.path.join(os.path.dirname(__file__), "index.html")
        context = {}
        self.response.out.write(template.render(path, context))

    def post(self):
        pincode = self.request.get("zipCode")
        url = "https://api.postalpincode.in/pincode/"+ pincode
        data = urllib.urlopen(url).read
        post_office = data[0]['PostOffice'][0]['State']
        name = data[0]['PostOffice'][0]['Name']
        block = data[0]['PostOffice'][0]['Block']
        district = data[0]['PostOffice'][0]['District']
        template_values = {
            "post_office": post_office,
            "name": name,
            "block": block,
            "district": district
        }
        path = os.path.join(os.path.dirname(__file__), 'results.html')
        self.response.out.write(template.render(path, template_values))


app = webapp2.WSGIApplication([('/', MainPage)], debug=True)
```

Step 2: Write the subsequent HHTML CODE for it specifying all the details you want to add.



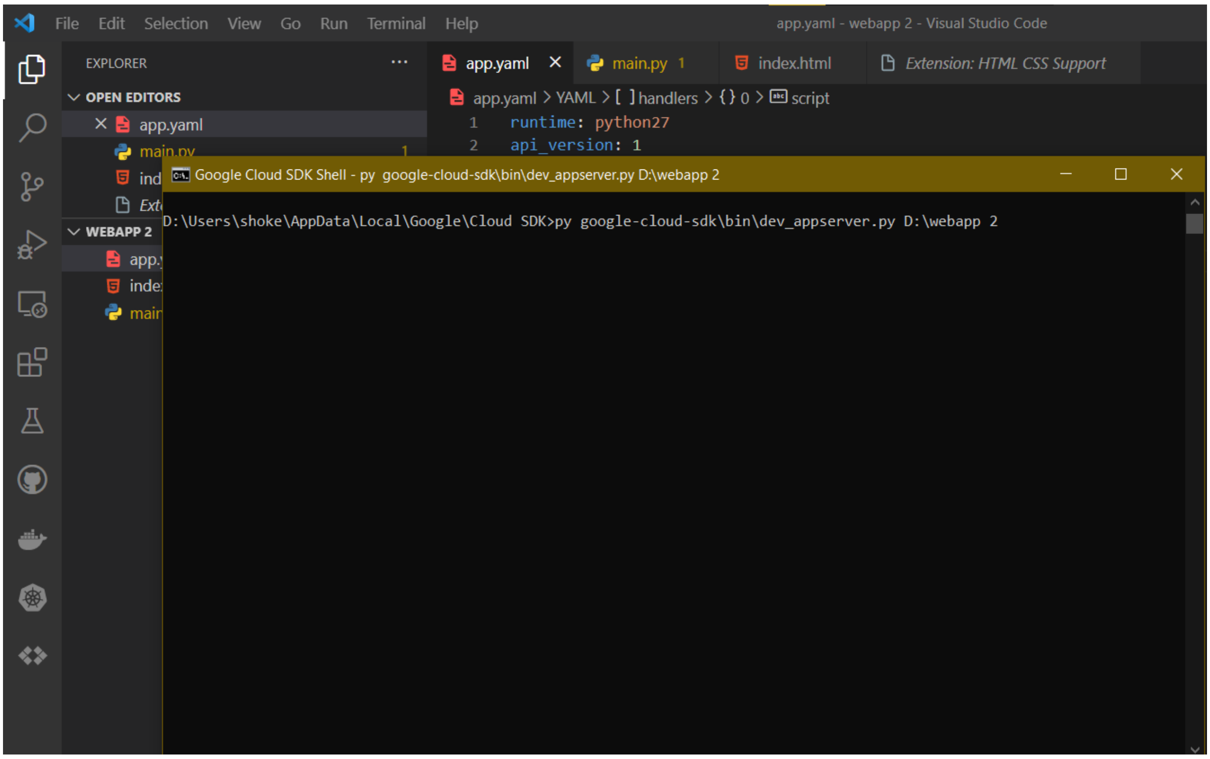
```html
<html>
  <style>
    .weatherText {
      font-family: 'Lato', 'sans-serif';
      font-size: 24px;
      text-align: center;
    }

    #weatherForm {
      padding: 20px;
    }

    #weatherSubmit {
      color: white;
      background-color: #083375;
      padding: 5px 20px;
      border-radius: 5px;
      margin-top: 20px;
    }

    #weatherSubmit:hover {
      cursor: pointer;
    }
    body {
      display: flex;
      justify-content: center;
      align-items: center;
    }
    .card {
      border: 2px solid black;
      width: 50%;

      justify-content: center;
      align-items: center;
    }
  </style>
```

Step 3: Write the corresponding YAML for it specifying threads, APIs, and script.



```yaml
runtime: python27
api_version: 1
threadsafe: true

handlers:
- url: /
  script: main.app
```

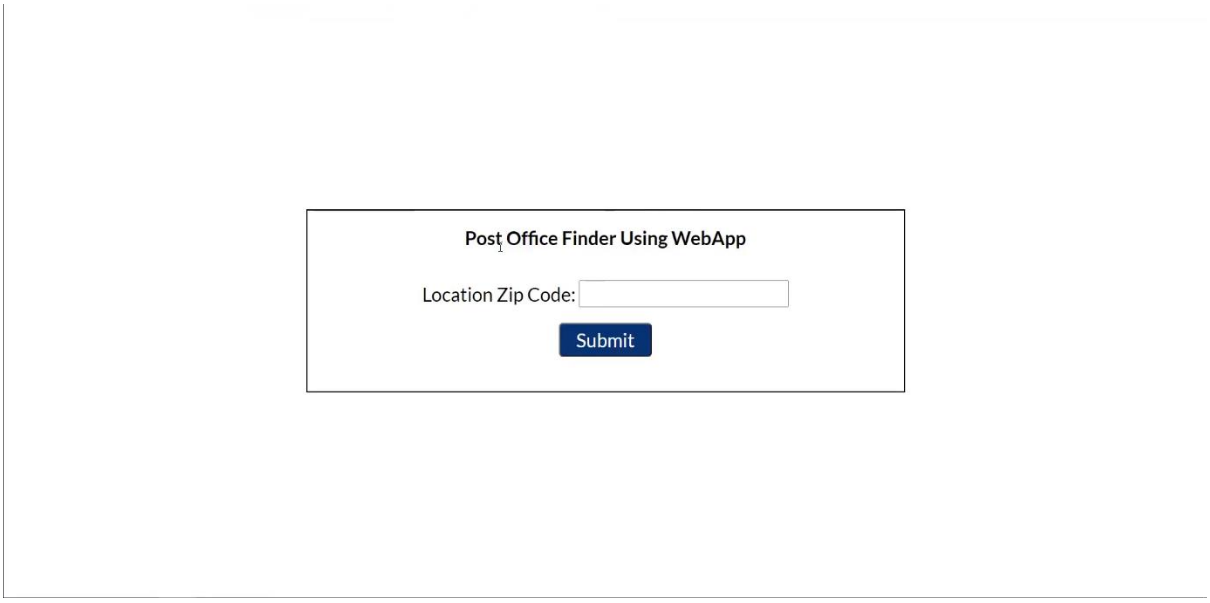
Step 4: Locate the file where your appserver.py is and run with the location where your project is in Google SDK Shell and run the command.



app.yaml - webapp 2 - Visual Studio Code
Google Cloud SDK Shell - py google-cloud-sdk\bin\dev_appserver.py D:\webapp 2

```
D:\Users\shoke\AppData\Local\Google\Cloud SDK>py google-cloud-sdk\bin\dev_appserver.py D:\webapp 2
```

OUTPUT:



Post Office Finder Using WebApp

Location Zip Code:

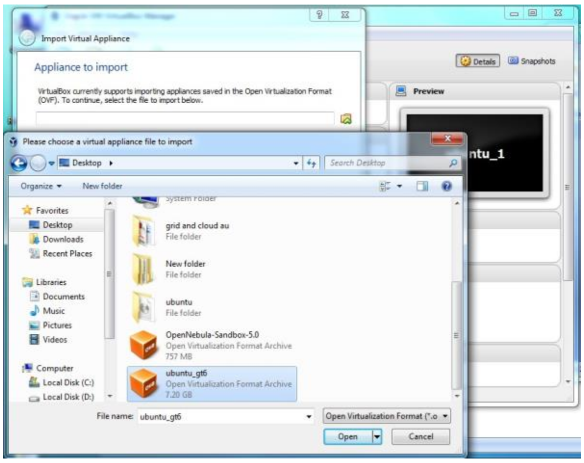
Submit

Q5. Install a C compiler in the virtual machine created using virtual box and execute Simple Programs.

PROCEDURE:

Steps to import .ova file:

1. Open Virtual box
2. File import Appliance
3. Browse ubuntu_gt6.ova file
4. Then go to setting, select Usb and choose USB 1.1
5. Then start the ubuntu_gt6
6. Login using username: Harsh, password:1912730002.



Import Virtual Appliance

Appliance to import

VirtualBox currently supports importing appliances saved in the Open Virtualization Format (OVF). To continue, select the file to import below.

Please choose a virtual appliance file to import

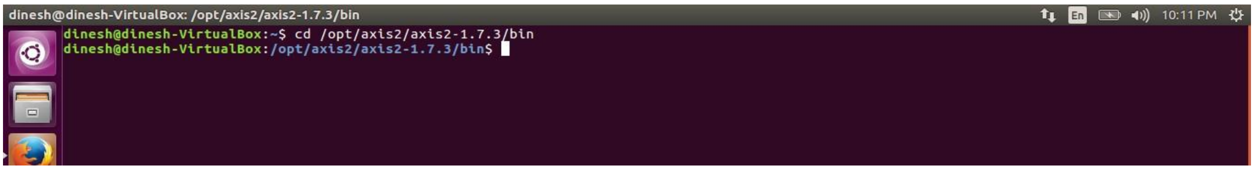
File name: ubuntu_gt6

Open Virtualization Format (*.o

Steps to run c program:

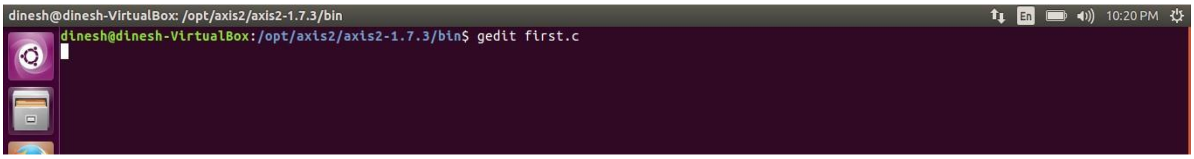
1. Open the terminal.
2. Type cd /opt/axis2/axis2-1.7.3/bin then press enter.
3. gedit hello.c
4. gcc hello.c

Step 1.Type cd /opt/axis2/axis2-1.7.3/bin then press enter.



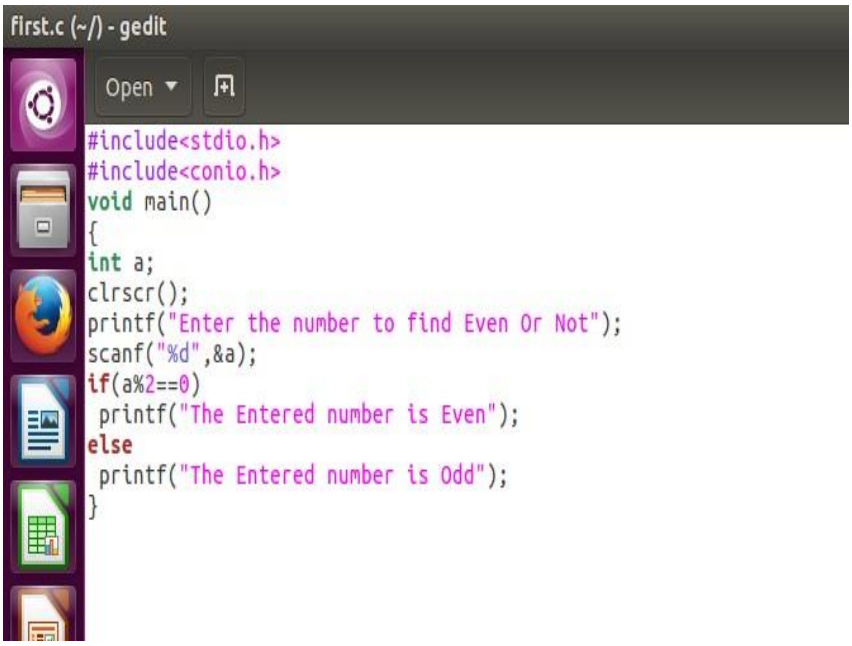
```
dinesh@dinesh-VirtualBox:~$ cd /opt/axis2/axis2-1.7.3/bin
dinesh@dinesh-VirtualBox:/opt/axis2/axis2-1.7.3/bin$ 
```

Step 2: Type gedit first.c



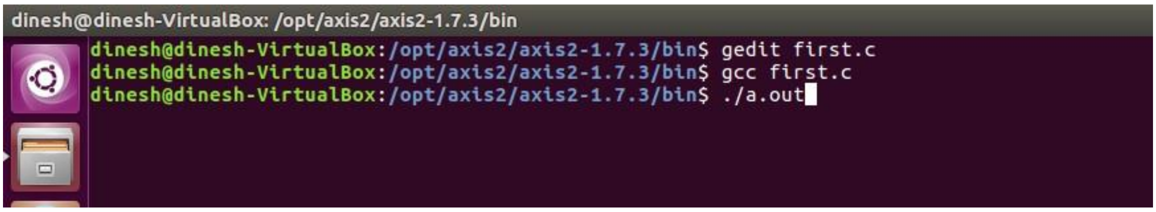
```
dinesh@dinesh-VirtualBox:/opt/axis2/axis2-1.7.3/bin$ gedit first.c
```

Step 3: Type the C Program



```
#include<stdio.h>
#include<conio.h>
void main()
{
int a;
clrscr();
printf("Enter the number to find Even Or Not");
scanf("%d",&a);
if(a%2==0)
 printf("The Entered number is Even");
else
 printf("The Entered number is Odd");
}
```

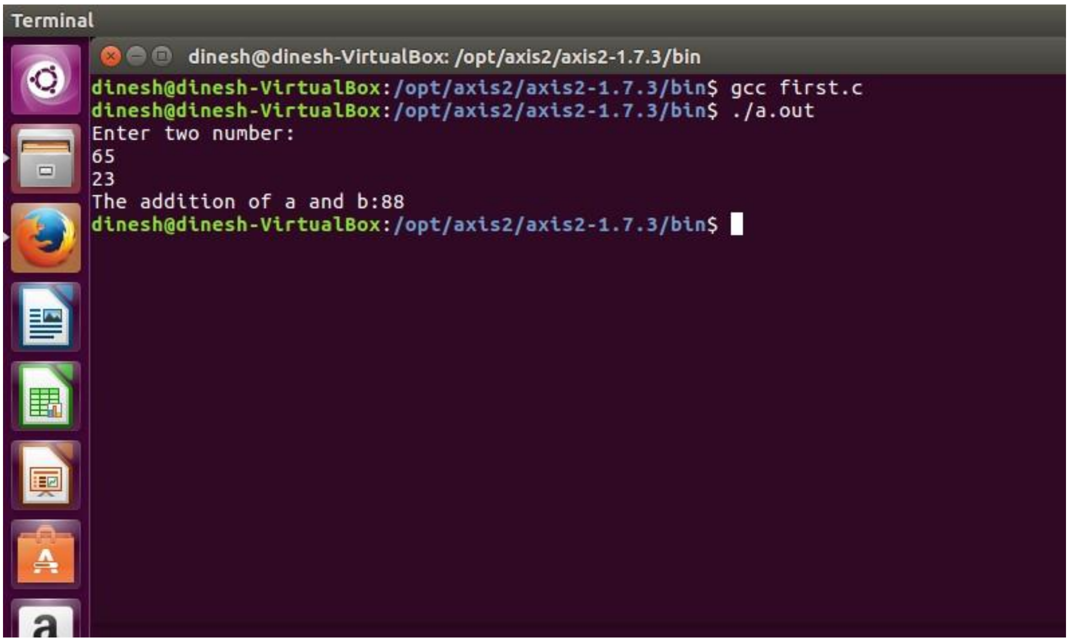
Step 4: Running the C Program



```
dinesh@dinesh-VirtualBox: /opt/axis2/axis2-1.7.3/bin
dinesh@dinesh-VirtualBox:/opt/axis2/axis2-1.7.3/bin$ gedit first.c
dinesh@dinesh-VirtualBox:/opt/axis2/axis2-1.7.3/bin$ gcc first.c
dinesh@dinesh-VirtualBox:/opt/axis2/axis2-1.7.3/bin$ ./a.out
```

Step 5: Display The output.

RESULT:



```
dinesh@dinesh-VirtualBox: /opt/axis2/axis2-1.7.3/bin
dinesh@dinesh-VirtualBox:/opt/axis2/axis2-1.7.3/bin$ gcc first.c
dinesh@dinesh-VirtualBox:/opt/axis2/axis2-1.7.3/bin$ ./a.out
Enter two number:
65
23
The addition of a and b:88
dinesh@dinesh-VirtualBox:/opt/axis2/axis2-1.7.3/bin$ 
```

Q6. Find a procedure to transfer the files from one virtual machine to another virtual machine.

Steps:

1. For this you need to share clipboard between host OS and guest OS, installing Guest Addition on both the virtual machines (probably setting bidirectional and restarting them). You copy from guest OS in the clipboard that is shared with the host OS.
2. Then you paste from the host OS to the second guest OS.
3. You can enable drag and drop too with the same method (Click on the machine, settings, general, advanced, drag and drop set to bidirectional)
4. You can have common Shared Folders on both virtual machines and use one of the directories shared as buffer to copy.
5. Installing Guest Additions, you have the possibility to set Shared Folders too. As you put a file in a shared folder from host OS or from guest OS, is immediately visible to the other. (Keep in mind that can arise some problems for date/time of the files when there are different clock settings on the different virtual machines).
6. If you use the same folder shared on more machines, you can exchange files directly copying them in this folder.

5. You can use the usual method to copy files between 2 different computers with a client-server application. (e.g., scp with sshd active for linux, winscp... you can get some info about SSH servers e.g., here)

7. You need an active server (sshd) on the receiving machine and a client on the sending machine. Of course you need to have the authorization setted (via password or, better, via an automatic authentication method).

8. Note: many Linux/Ubuntu distribution install sshd by default: you can see if it is running with pgrep sshd from a shell. You can install with sudo apt-get install openssh-server.

6. You can mount part of the file system of a virtual machine via NFS or SSHFS on the other, or you can share file and directory with Samba. You may find interesting the article Sharing files between guest and host without VirtualBox shared folders with detailed step by step instructions.

You should remember that you are dealing with a little network of machines with different operative systems, and in particular:

- Each virtual machine has its own operating system running on and acts as a physical machine.
- Each virtual machine is an instance of a program owned by a user in the hosting operating system and should undergo the restrictions of the user in the hosting OS.

E.g., Let we say that Hastur and Meow are users of the hosting machine, but they did not allow each other to see their directories (no read/write/execute authorization). When each of them runs a virtual machine, for the hosting OS those virtual machines are two normal programs owned by Hastur and Meow and cannot see the private directory of the other user. This is a restriction due to the hosting OS. It's easy to overcome it: it's enough to give authorization to read/write/execute to a directory or to choose a different directory in which both users can read/write/execute.

Windows likes mouse and Linux fingers. :-)

I mean I suggest you enable Drag & drop to be cozy with the Windows machines and the Shared folders or to be cozy with Linux.

When you will need to be fast with Linux you will feel the need of ssh-keygen and

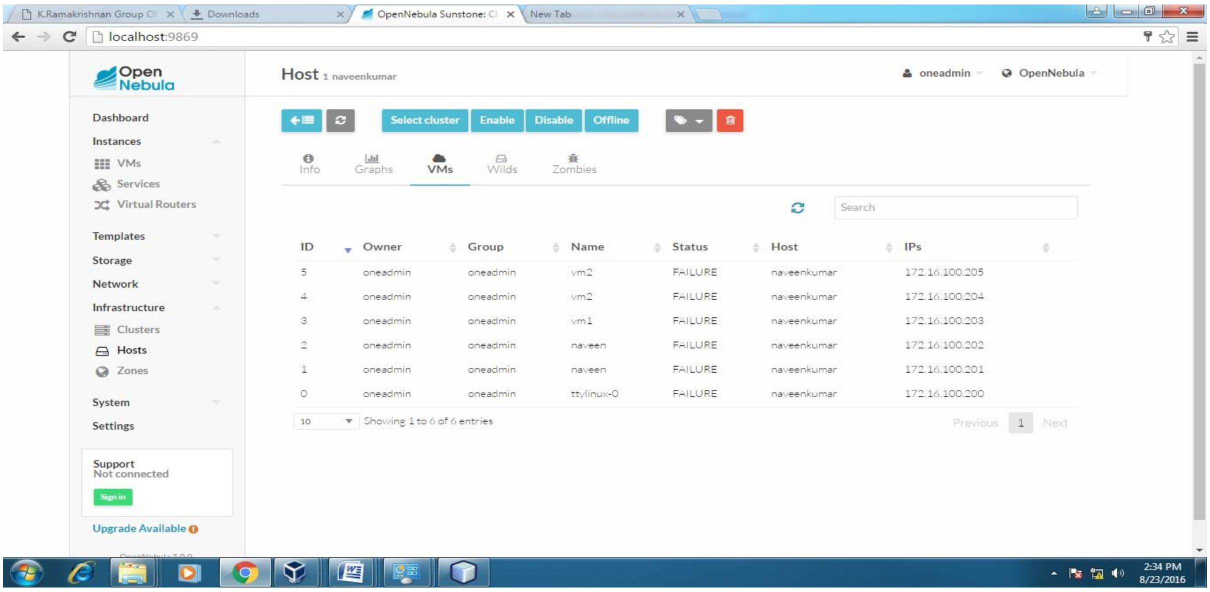
to Generate once SSH Keys to copy files on/from a remote machine without writing password anymore. In this way it functions bash auto-completion remotely too!

PROCEDURE:

Steps:

1. Open Browser, type localhost:9869
2. Login using username: oneadmin, password: opennebula
3. Then follow the steps to migrate VMs.
    a. Click on infrastructure.
    b. Select clusters and enter the cluster name.
    c. Then select host tab and select all host.
    d. Then select Vnets tab, and select all vnet
    e. Then select the datastores tab and select all datastores.
    f. And then choose host under infrastructure tab.
    g. Click on + symbol to add new host, name the host then click on create.
4. on instances, select VMs to migrate then follow the steps
    a. Click on the 8th icon ,the drop-down list display.
    b. Select migrate on that ,the popup window display.
    c. On that select the target host to migrate then click on migrate.

Before Migration

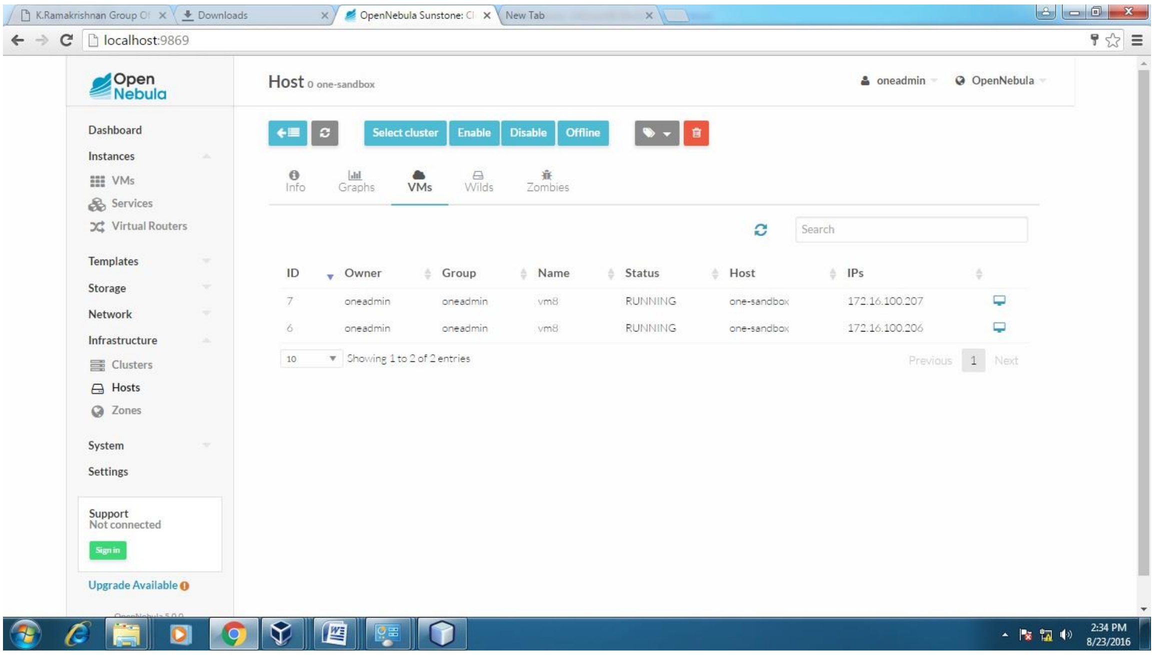


Host 1 naveenkumar

ID	Owner	Group	Name	Status	Host	IPs
5	oneadmin	oneadmin	vm2	FAILURE	naveenkumar	172.16.100.205
4	oneadmin	oneadmin	vm2	FAILURE	naveenkumar	172.16.100.204
3	oneadmin	oneadmin	vm1	FAILURE	naveenkumar	172.16.100.203
2	oneadmin	oneadmin	naveen	FAILURE	naveenkumar	172.16.100.202
1	oneadmin	oneadmin	naveen	FAILURE	naveenkumar	172.16.100.201
0	oneadmin	oneadmin	ttylinux-0	FAILURE	naveenkumar	172.16.100.200

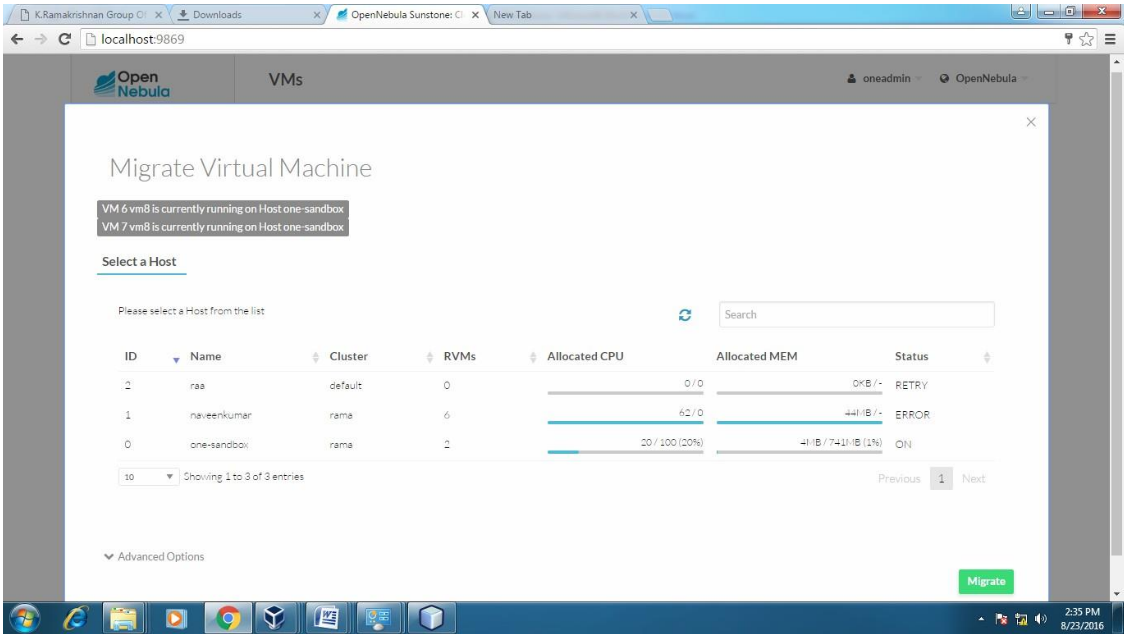
Showing 1 to 6 of 6 entries

Host: one_sandbox



ID	Owner	Group	Name	Status	Host	IPs
7	oneadmin	oneadmin	vm8	RUNNING	one-sandbox	172.16.100.207
6	oneadmin	oneadmin	vm8	RUNNING	one-sandbox	172.16.100.206

Showing 1 to 2 of 2 entries

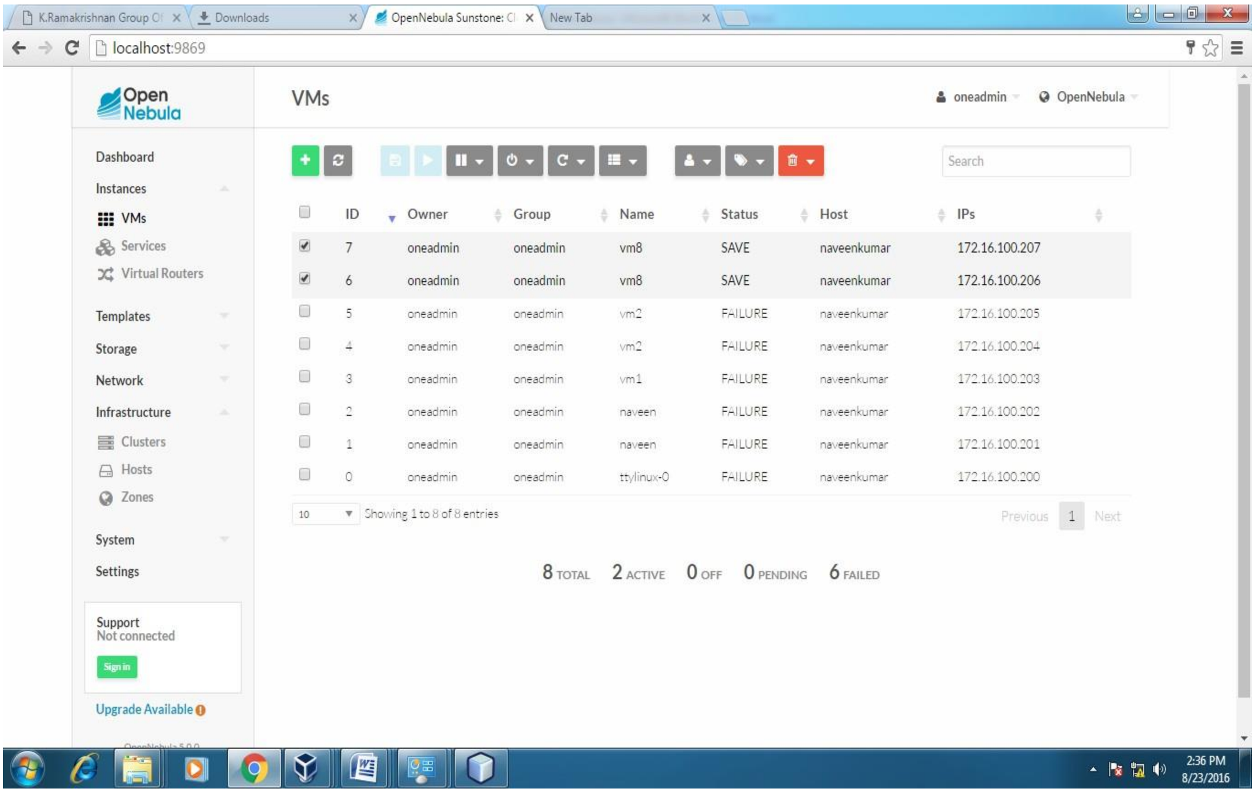


VM 6 vm8 is currently running on Host one-sandbox
VM 7 vm8 is currently running on Host one-sandbox

Please select a Host from the list

ID	Name	Cluster	RVMs	Allocated CPU	Allocated MEM	Status
2	raa	default	0	0 / 0	0KB / -	RETRY
1	naveenkumar	rama	6	62 / 0	44MB / -	ERROR
0	one-sandbox	rama	2	20 / 100 (20%)	4MB / 741MB (1%)	ON

Showing 1 to 3 of 3 entries



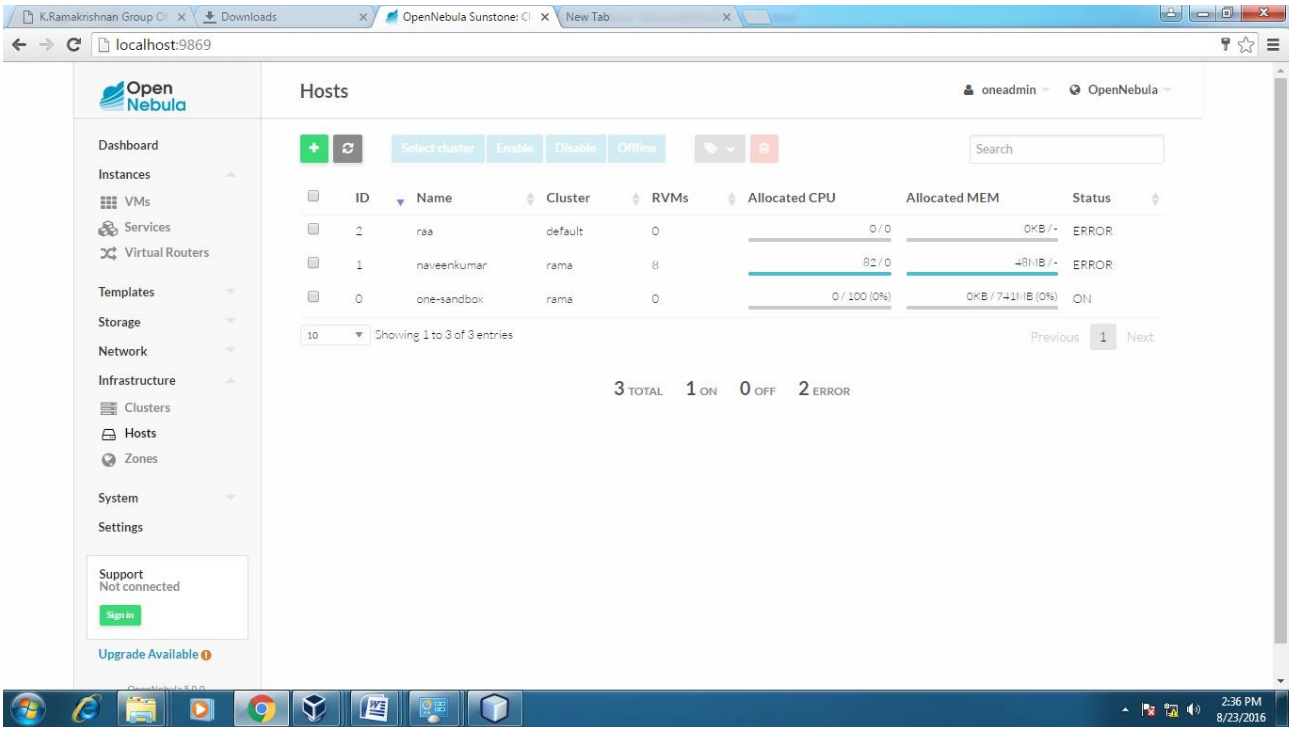
VMs

☐	ID	Owner	Group	Name	Status	Host	IPs
☑	7	oneadmin	oneadmin	vm8	SAVE	naveenkumar	172.16.100.207
☑	6	oneadmin	oneadmin	vm8	SAVE	naveenkumar	172.16.100.206
☐	5	oneadmin	oneadmin	vm2	FAILURE	naveenkumar	172.16.100.205
☐	4	oneadmin	oneadmin	vm2	FAILURE	naveenkumar	172.16.100.204
☐	3	oneadmin	oneadmin	vm1	FAILURE	naveenkumar	172.16.100.203
☐	2	oneadmin	oneadmin	naveen	FAILURE	naveenkumar	172.16.100.202
☐	1	oneadmin	oneadmin	naveen	FAILURE	naveenkumar	172.16.100.201
☐	0	oneadmin	oneadmin	ttylinux-0	FAILURE	naveenkumar	172.16.100.200

Showing 1 to 8 of 8 entries

8 TOTAL 2 ACTIVE 0 OFF 0 PENDING 6 FAILED

After Migration



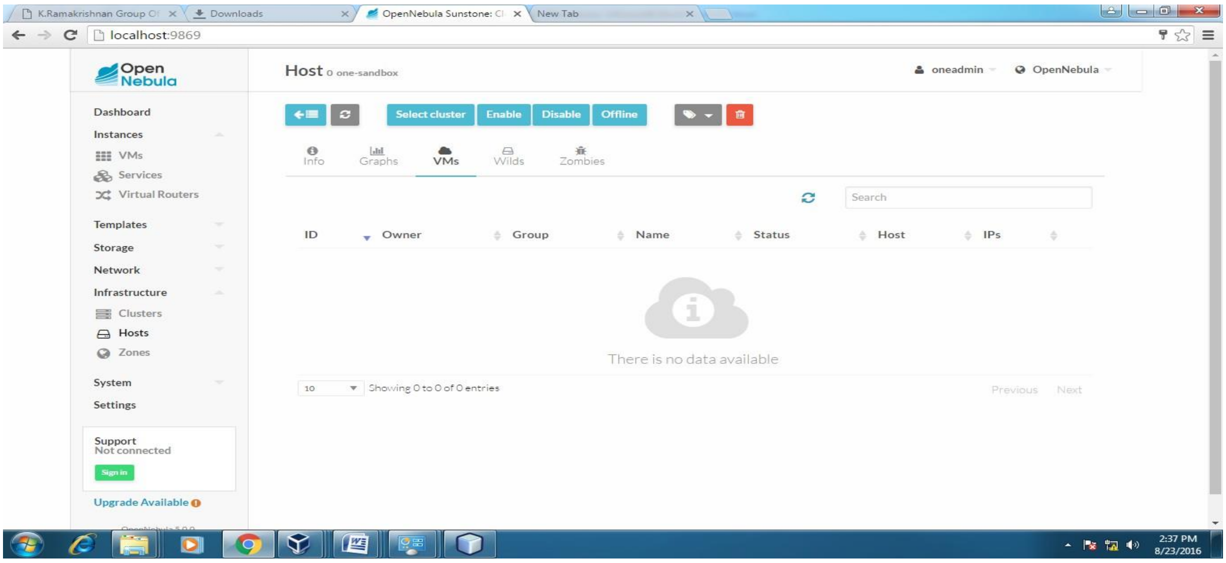
Hosts

☐	ID	Name	Cluster	RVMs	Allocated CPU	Allocated MEM	Status
☐	2	raa	default	0	0 / 0	0KB / -	ERROR
☐	1	naveenkumar	rama	8	82 / 0	48MB / -	ERROR
☐	0	one-sandbox	rama	0	0 / 100 (0%)	0KB / 741MB (0%)	ON

Showing 1 to 3 of 3 entries

3 TOTAL 1 ON 0 OFF 2 ERROR

Host: one_sandbox



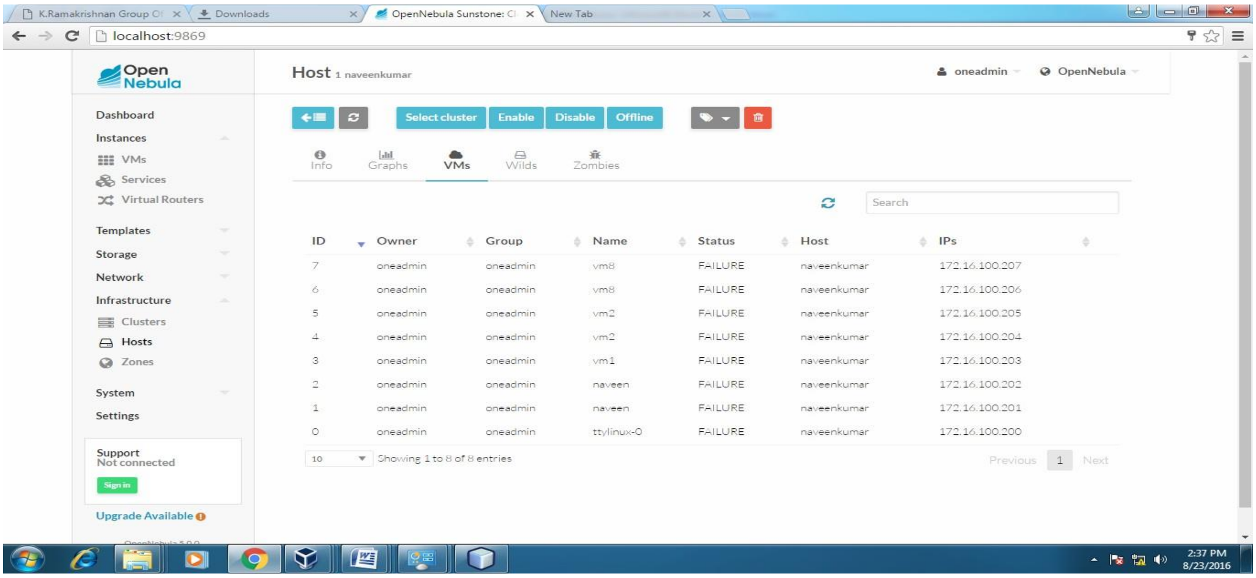
Host 0 one-sandbox

ID	Owner	Group	Name	Status	Host	IPs

There is no data available

10 Showing 0 to 0 of 0 entries

Host:SACET



Host 1 naveenkumar

ID	Owner	Group	Name	Status	Host	IPs
7	oneadmin	oneadmin	vm8	FAILURE	naveenkumar	172.16.100.207
6	oneadmin	oneadmin	vm8	FAILURE	naveenkumar	172.16.100.206
5	oneadmin	oneadmin	vm2	FAILURE	naveenkumar	172.16.100.205
4	oneadmin	oneadmin	vm2	FAILURE	naveenkumar	172.16.100.204
3	oneadmin	oneadmin	vm1	FAILURE	naveenkumar	172.16.100.203
2	oneadmin	oneadmin	naveen	FAILURE	naveenkumar	172.16.100.202
1	oneadmin	oneadmin	naveen	FAILURE	naveenkumar	172.16.100.201
0	oneadmin	oneadmin	ttylinux-0	FAILURE	naveenkumar	172.16.100.200

10 Showing 1 to 8 of 8 entries

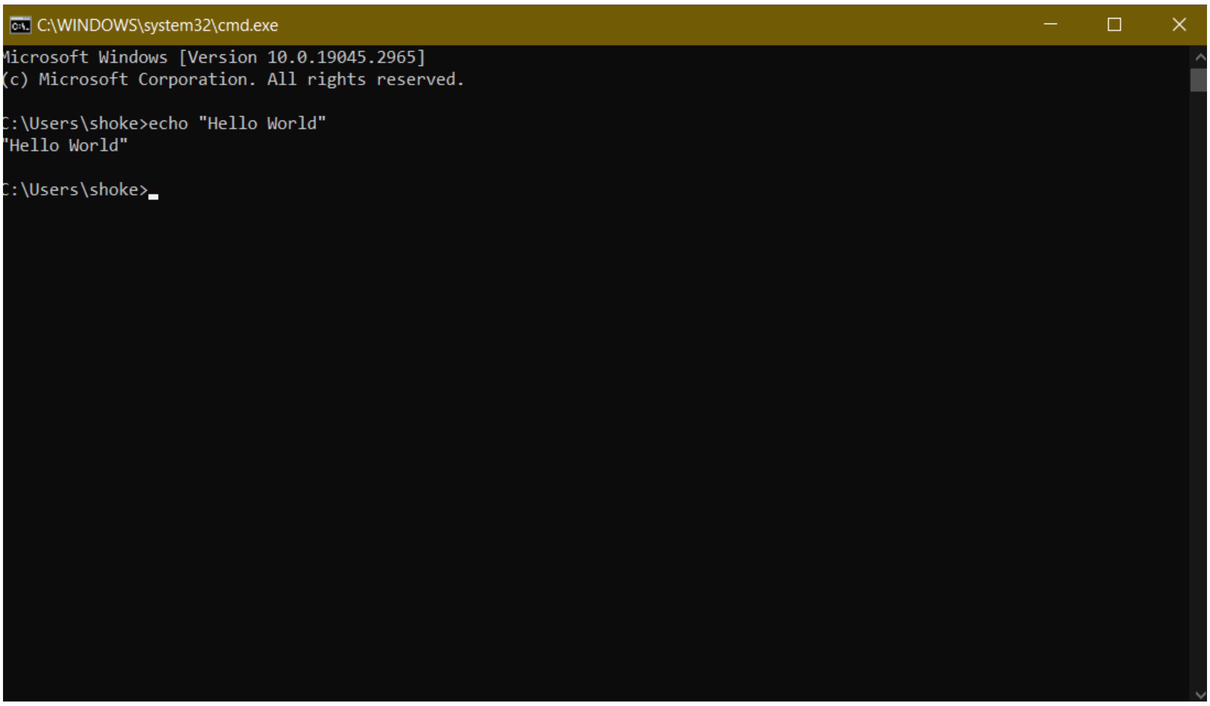
Result:

Thus, the file transfer between VM was successfully completed

Q7. Bash Command Execution

Here is a list of some of the most used bash commands:

cd: Change the directory to a different location.
ls: List the contents of the current directory.
mkdir: Create a new directory.
touch: Create a new file.
rm: Remove a file or directory.
cp: Copy a file or directory.
mv: Move or rename a file or directory.
echo: Print text to the terminal.
cat: Concatenate and print the contents of a file.
grep: Search for a pattern in a file.
chmod: Change the permissions of a file or directory.
sudo: Run a command with administrative privileges.
df: Display the amount of disk space available.
history: Show a list of previously executed commands.
ps: Display information about running processes.

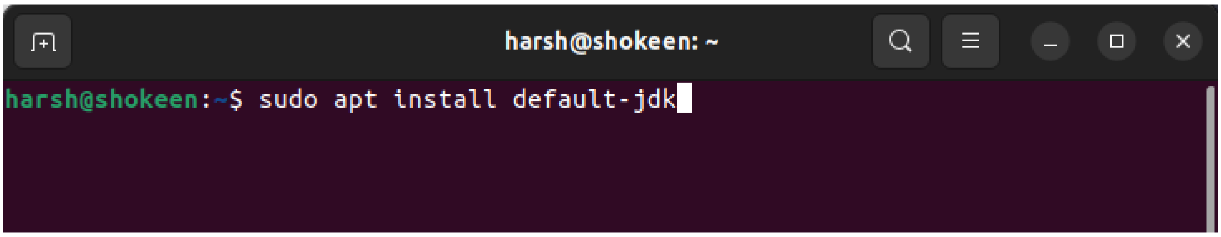


```
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Users\shoke>echo "Hello World"
"Hello World"

C:\Users\shoke>
```

Q8. Install Hadoop single node cluster and run simple applications like wordcount.

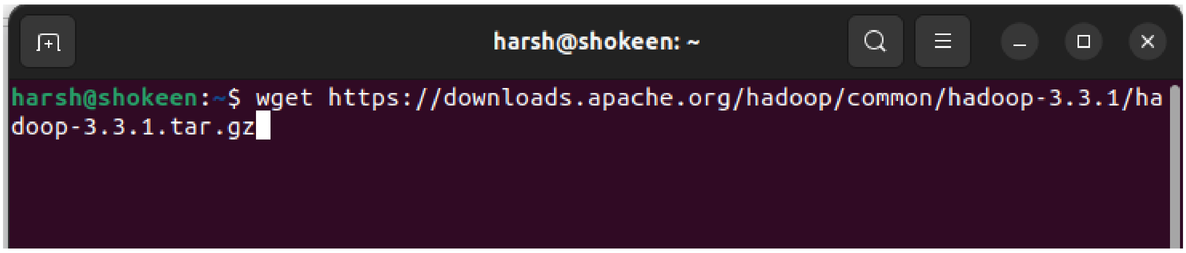
Procedure:

Step 1: Install JDK first in terminal by writing the following command: **sudo apt install openjdk-8-jdk**



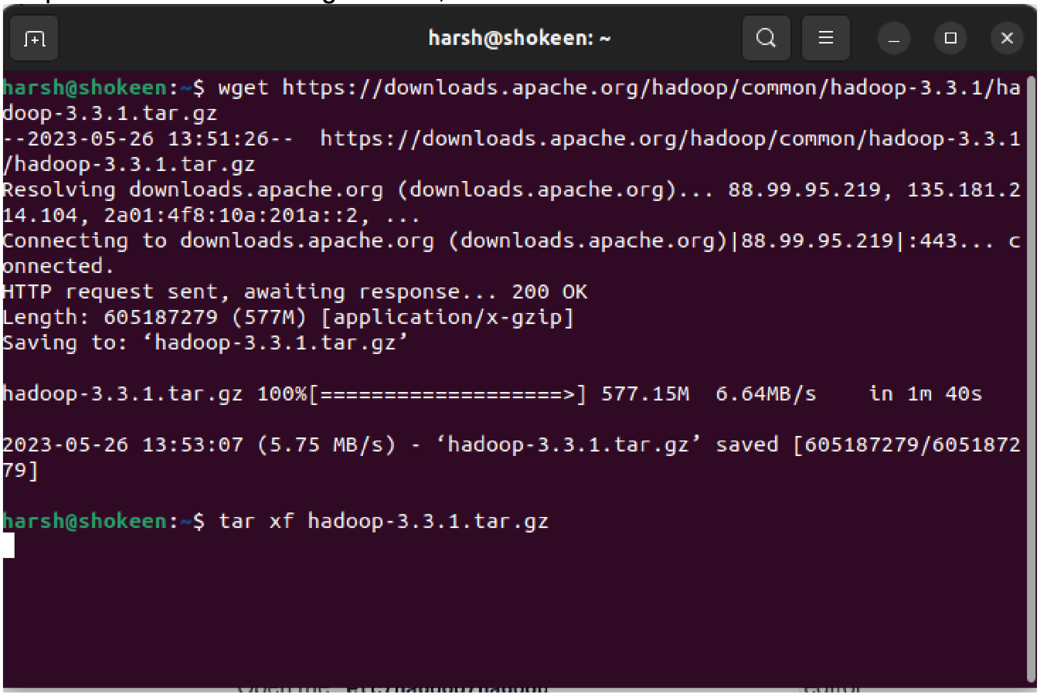
```
harsh@shokeen:~$ sudo apt install default-jdk
```

Step 2: Download Hadoop to your system using this command.



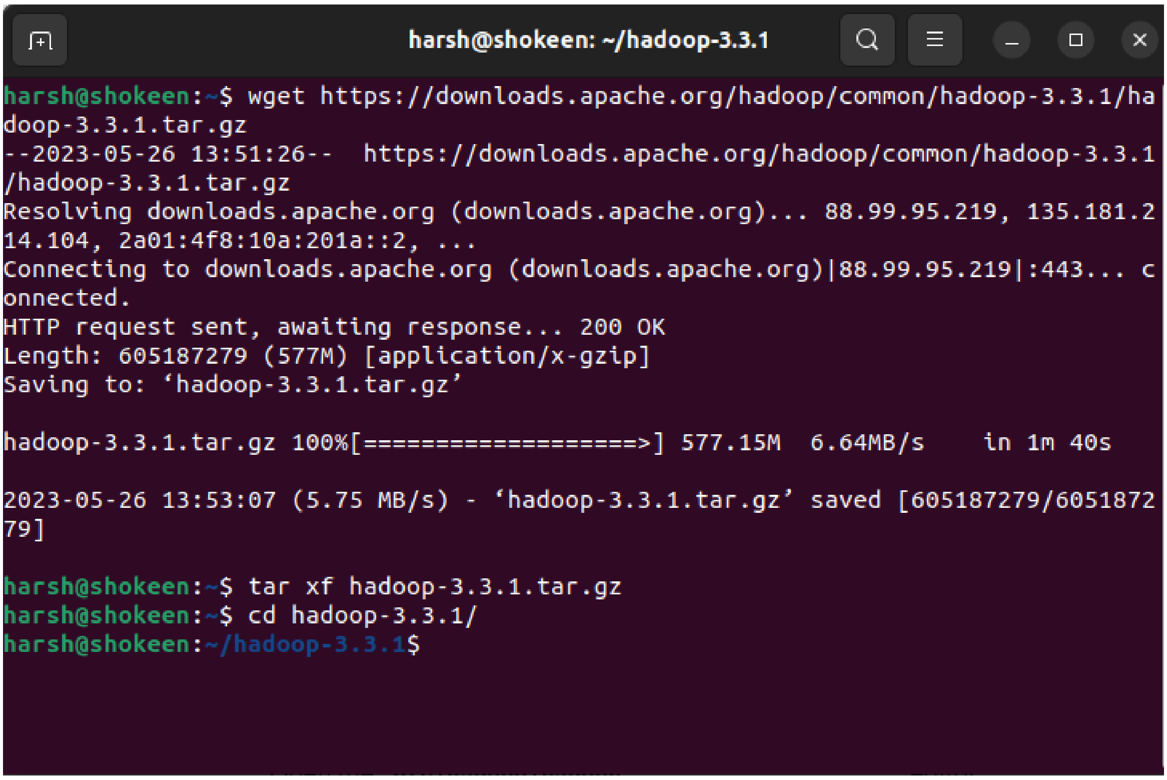
```
harsh@shokeen:~$ wget https://downloads.apache.org/hadoop/common/hadoop-3.3.1/ha
doop-3.3.1.tar.gz
```

Step 3: After Downloading the File, install and extract the file



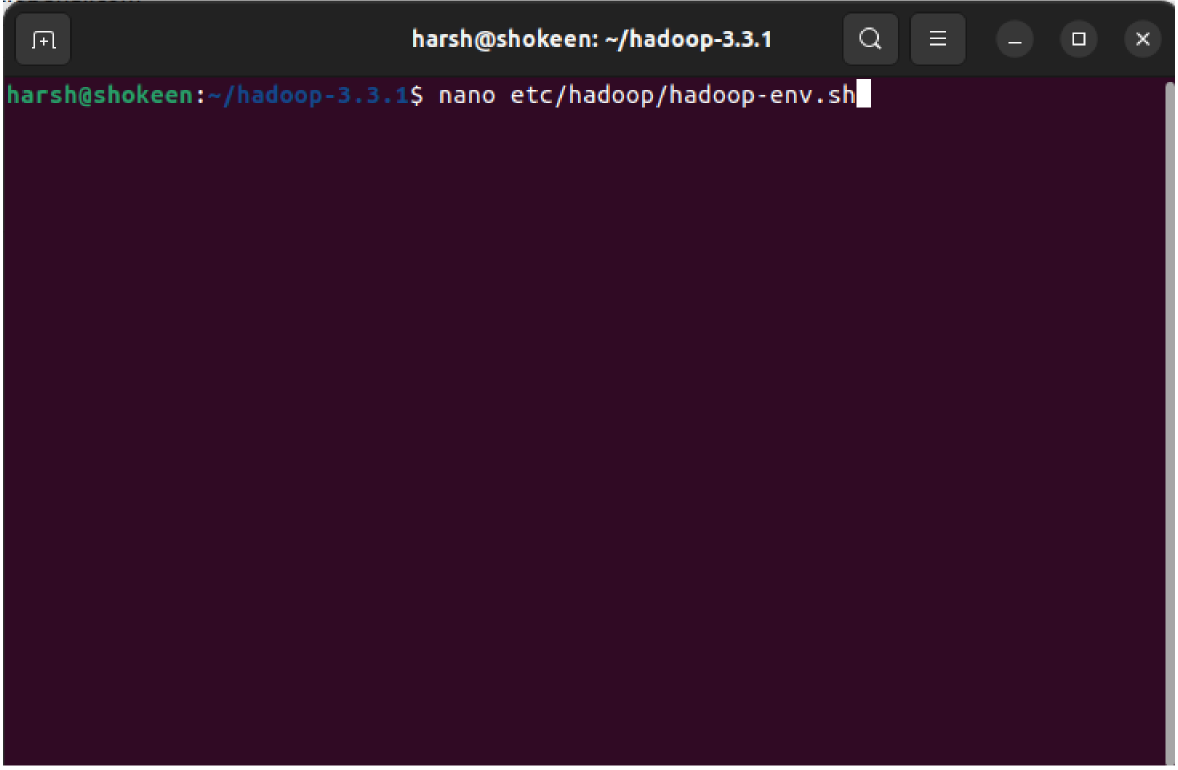
```
harsh@shokeen:~$ wget https://downloads.apache.org/hadoop/common/hadoop-3.3.1/ha
doop-3.3.1.tar.gz
--2023-05-26 13:51:26--  https://downloads.apache.org/hadoop/common/hadoop-3.3.1
/hadoop-3.3.1.tar.gz
Resolving downloads.apache.org (downloads.apache.org)... 88.99.95.219, 135.181.2
14.104, 2a01:4f8:10a:201a::2, ...
Connecting to downloads.apache.org (downloads.apache.org)|88.99.95.219|:443... c
onnected.
HTTP request sent, awaiting response... 200 OK
Length: 605187279 (577M) [application/x-gzip]
Saving to: ‘hadoop-3.3.1.tar.gz’

hadoop-3.3.1.tar.gz 100%[===================>] 577.15M  6.64MB/s    in 1m 40s

2023-05-26 13:53:07 (5.75 MB/s) - ‘hadoop-3.3.1.tar.gz’ saved [605187279/6051872
79]

harsh@shokeen:~$ tar xf hadoop-3.3.1.tar.gz
```

Step 4: Open the file location using cd to navigate to that folder.



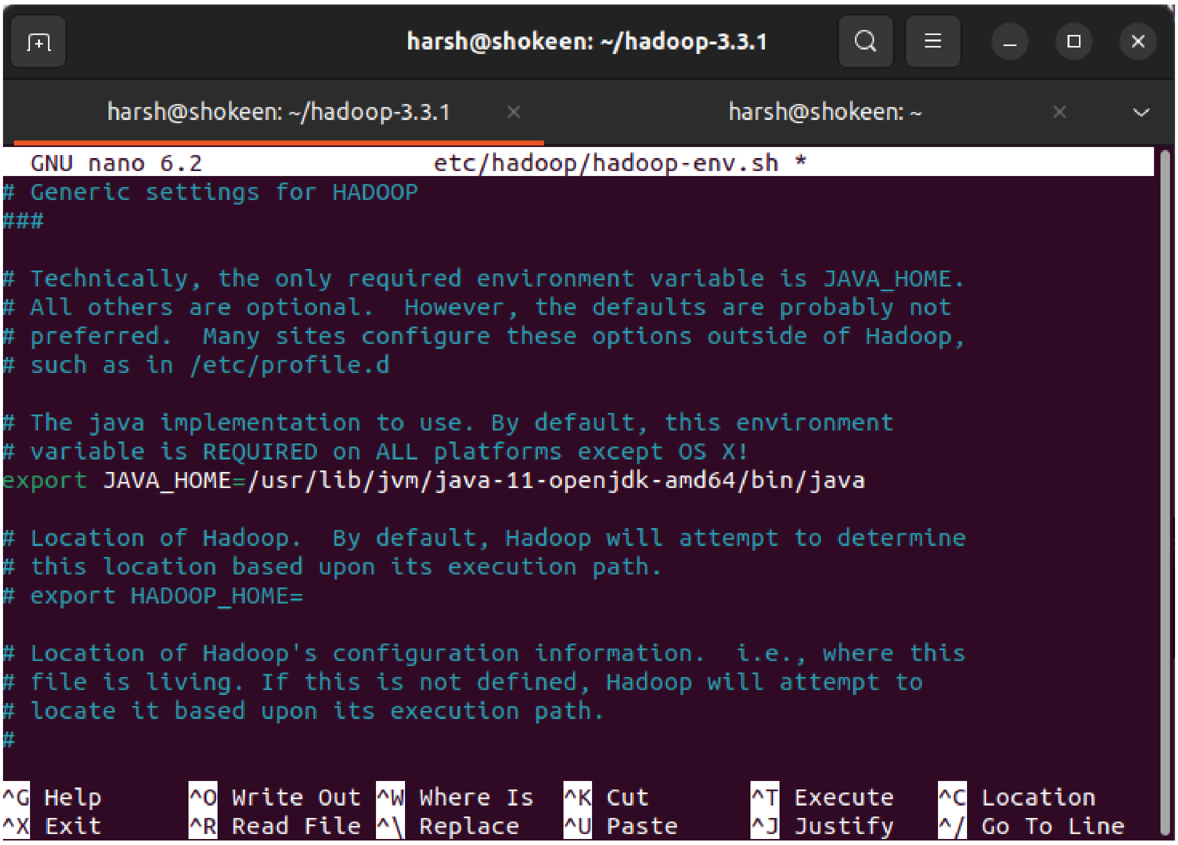
```
harsh@shokeen:~$ wget https://downloads.apache.org/hadoop/common/hadoop-3.3.1/ha
doop-3.3.1.tar.gz
--2023-05-26 13:51:26--  https://downloads.apache.org/hadoop/common/hadoop-3.3.1
/hadoop-3.3.1.tar.gz
Resolving downloads.apache.org (downloads.apache.org)... 88.99.95.219, 135.181.2
14.104, 2a01:4f8:10a:201a::2, ...
Connecting to downloads.apache.org (downloads.apache.org)|88.99.95.219|:443... c
onnected.
HTTP request sent, awaiting response... 200 OK
Length: 605187279 (577M) [application/x-gzip]
Saving to: ‘hadoop-3.3.1.tar.gz’

hadoop-3.3.1.tar.gz 100%[===================>] 577.15M  6.64MB/s    in 1m 40s

2023-05-26 13:53:07 (5.75 MB/s) - ‘hadoop-3.3.1.tar.gz’ saved [605187279/6051872
79]

harsh@shokeen:~$ tar xf hadoop-3.3.1.tar.gz
harsh@shokeen:~$ cd hadoop-3.3.1/
harsh@shokeen:~/hadoop-3.3.1$
```

Step 5 Configure Hadoop



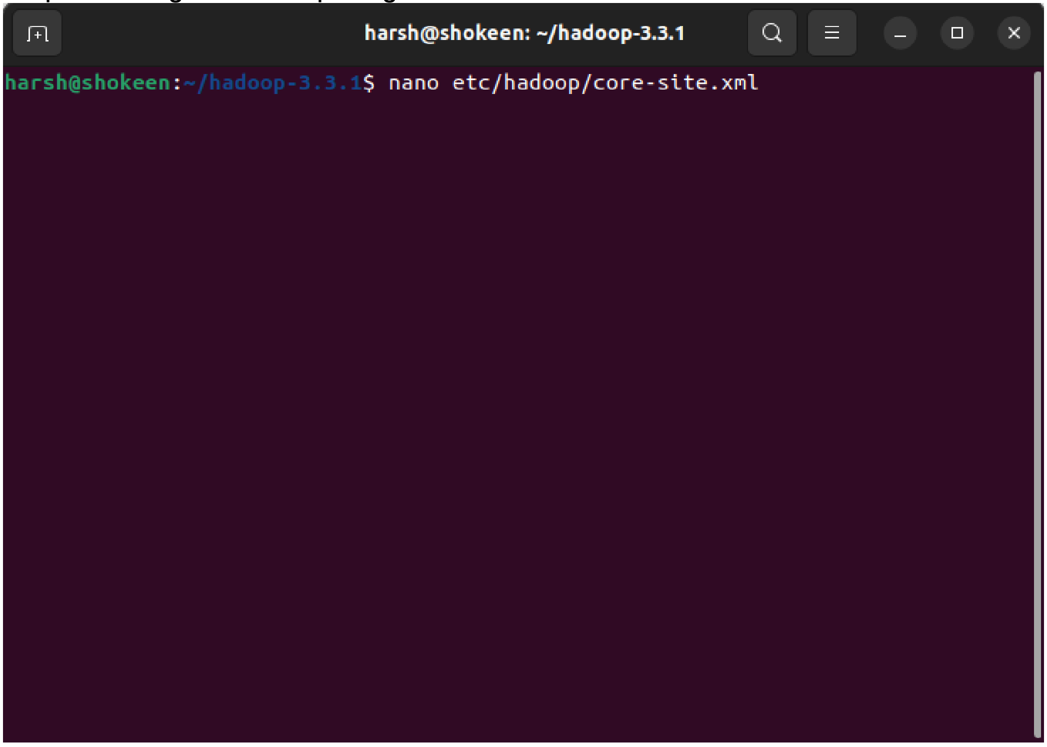
```
harsh@shokeen:~/hadoop-3.3.1$ nano etc/hadoop/hadoop-env.sh
```

Step 6: Uncomment the line that sets the JAVA_HOME variable and provide the path to your JDK installation. Save the file and exit the text editor. This is to enable the path for JDK to function in Hadoop.



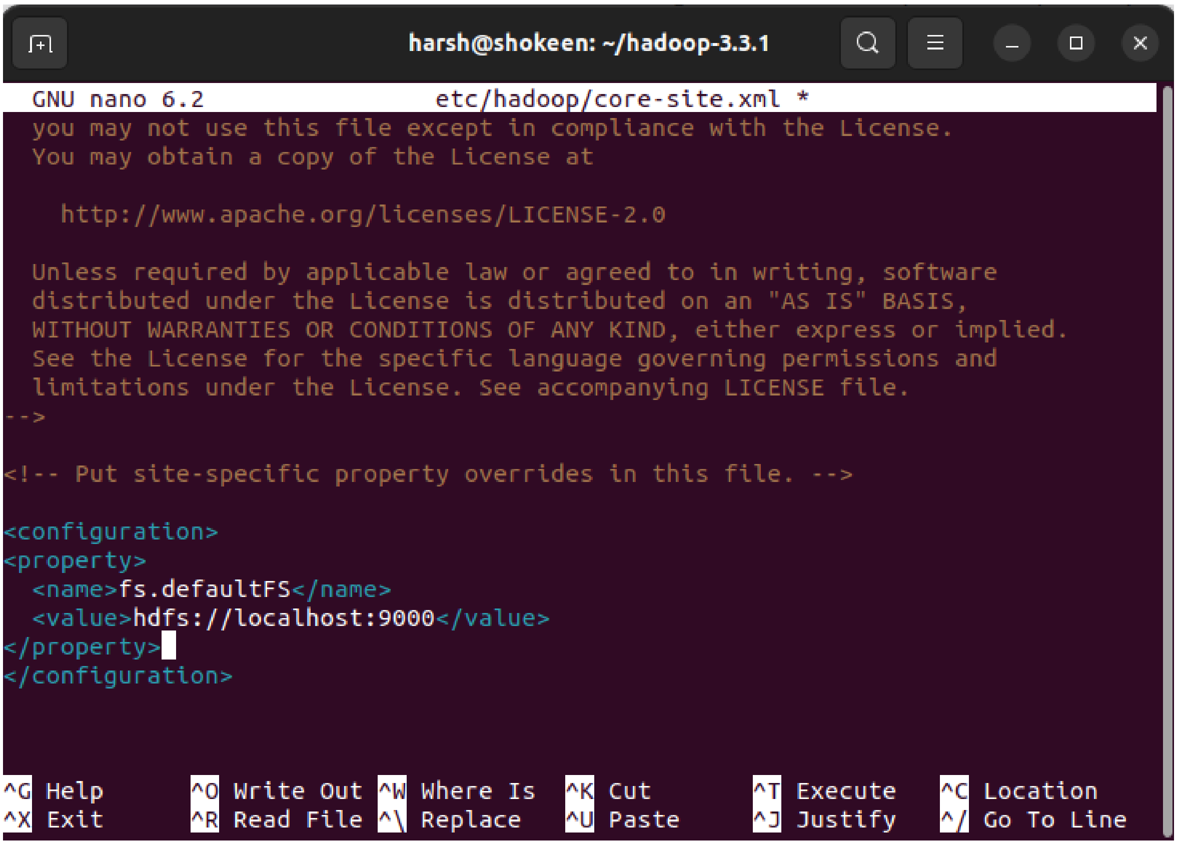
```
  GNU nano 6.2                etc/hadoop/hadoop-env.sh *
# Generic settings for HADOOP
###

# Technically, the only required environment variable is JAVA_HOME.
# All others are optional.  However, the defaults are probably not
# preferred.  Many sites configure these options outside of Hadoop,
# such as in /etc/profile.d

# The java implementation to use. By default, this environment
# variable is REQUIRED on ALL platforms except OS X!
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64/bin/java

# Location of Hadoop.  By default, Hadoop will attempt to determine
# this location based upon its execution path.
# export HADOOP_HOME=

# Location of Hadoop's configuration information.  i.e., where this
# file is living. If this is not defined, Hadoop will attempt to
# locate it based upon its execution path.
#

^G Help       ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit       ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

Step 7: Configure Hadoop Single-Node Cluster



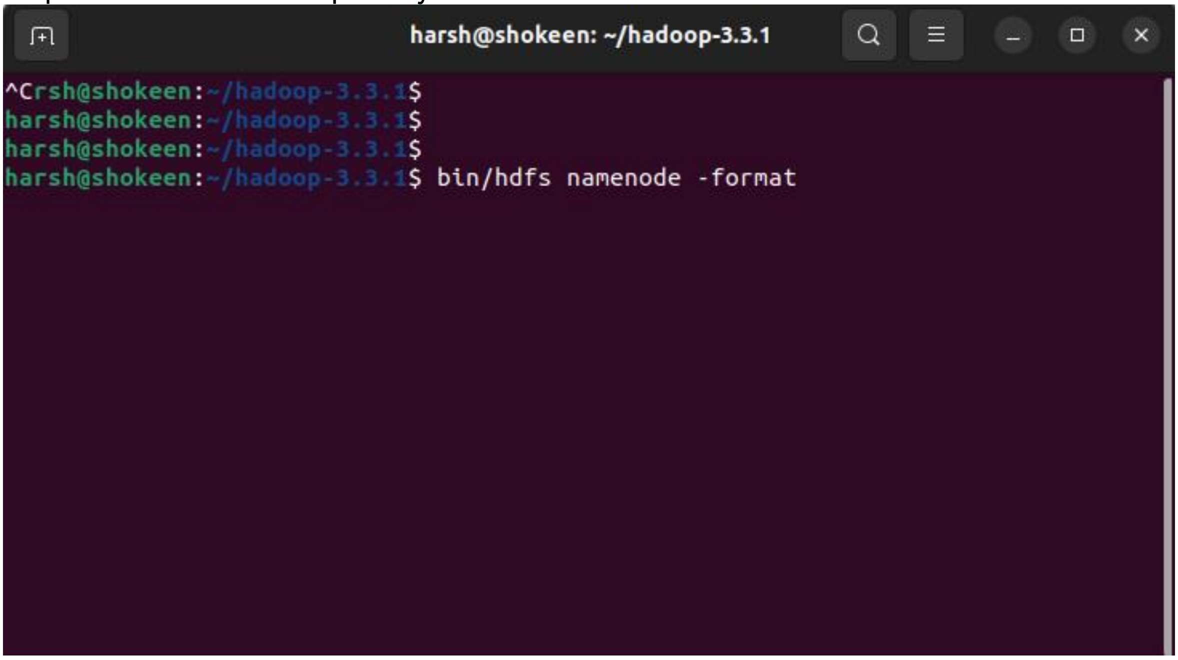
```
harsh@shokeen:~/hadoop-3.3.1$ nano etc/hadoop/core-site.xml
```

Step 8: Add the following configuration inside the <configuration> section.



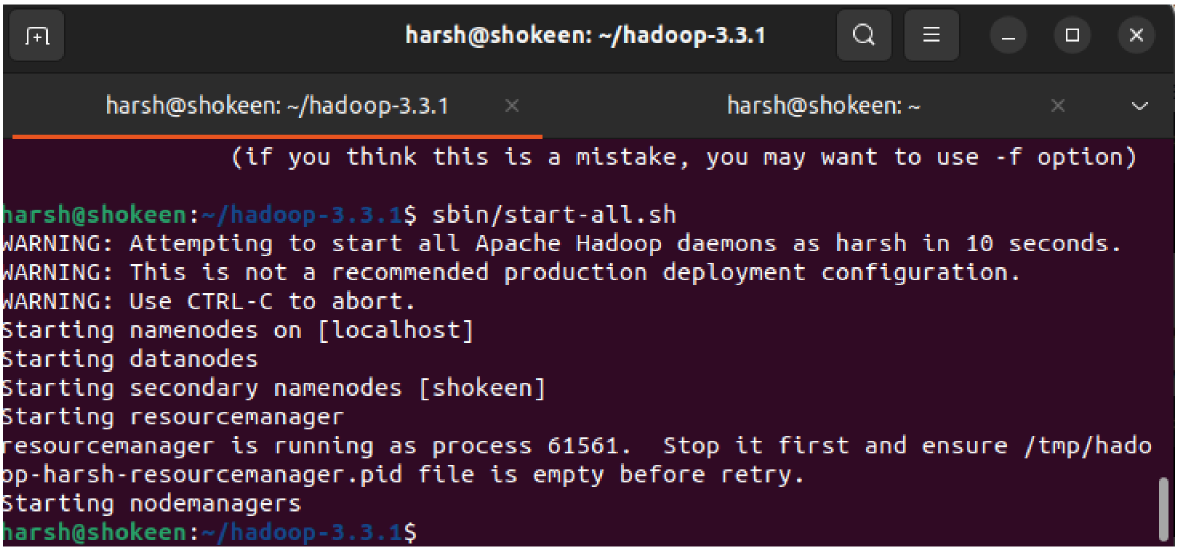
```
  GNU nano 6.2                  etc/hadoop/core-site.xml *
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
  <name>fs.defaultFS</name>
  <value>hdfs://localhost:9000</value>
</property>
</configuration>


^G Help         ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit         ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line
```

Step 9: Format the Hadoop file system



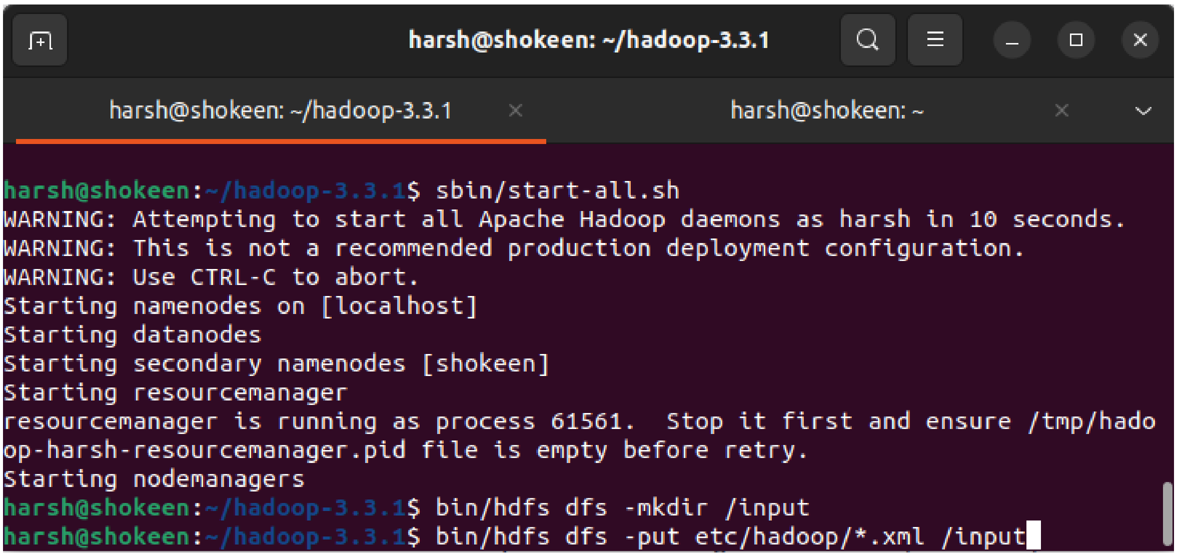
```
^Crsh@shokeen:~/hadoop-3.3.1$
harsh@shokeen:~/hadoop-3.3.1$
harsh@shokeen:~/hadoop-3.3.1$
harsh@shokeen:~/hadoop-3.3.1$ bin/hdfs namenode -format
```

Step 10: Start the Hadoop cluster. This will start the Hadoop services including the NameNode, DataNode, ResourceManager, and NodeManager.



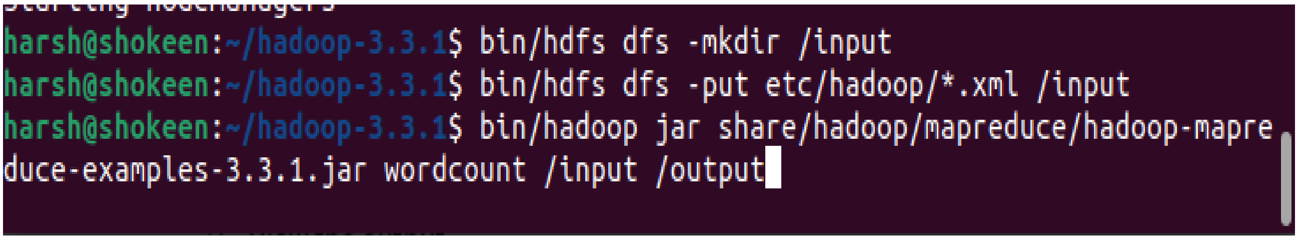
```
              (if you think this is a mistake, you may want to use -f option)

harsh@shokeen:~/hadoop-3.3.1$ sbin/start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as harsh in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [shokeen]
Starting resourcemanager
resourcemanager is running as process 61561.  Stop it first and ensure /tmp/hado
op-harsh-resourcemanager.pid file is empty before retry.
Starting nodemanagers
harsh@shokeen:~/hadoop-3.3.1$
```

Step 11: Create input directory and copy input files to HDFS:



```
harsh@shokeen:~/hadoop-3.3.1$ sbin/start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as harsh in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [shokeen]
Starting resourcemanager
resourcemanager is running as process 61561.  Stop it first and ensure /tmp/hado
op-harsh-resourcemanager.pid file is empty before retry.
Starting nodemanagers
harsh@shokeen:~/hadoop-3.3.1$ bin/hdfs dfs -mkdir /input
harsh@shokeen:~/hadoop-3.3.1$ bin/hdfs dfs -put etc/hadoop/*.xml /input
```

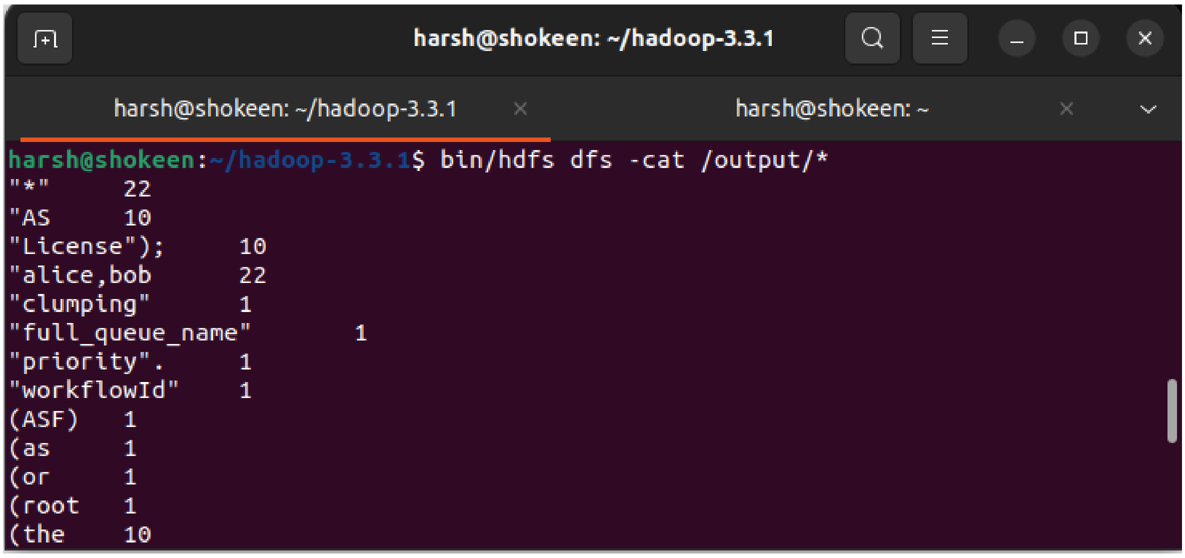
Step 12: Run the Word Count example using this command: **bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.1.jar wordcount /input /output**



```
harsh@shokeen:~/hadoop-3.3.1$ bin/hdfs dfs -mkdir /input
harsh@shokeen:~/hadoop-3.3.1$ bin/hdfs dfs -put etc/hadoop/*.xml /input
harsh@shokeen:~/hadoop-3.3.1$ bin/hadoop jar share/hadoop/mapreduce/hadoop-mapre
duce-examples-3.3.1.jar wordcount /input /output
```

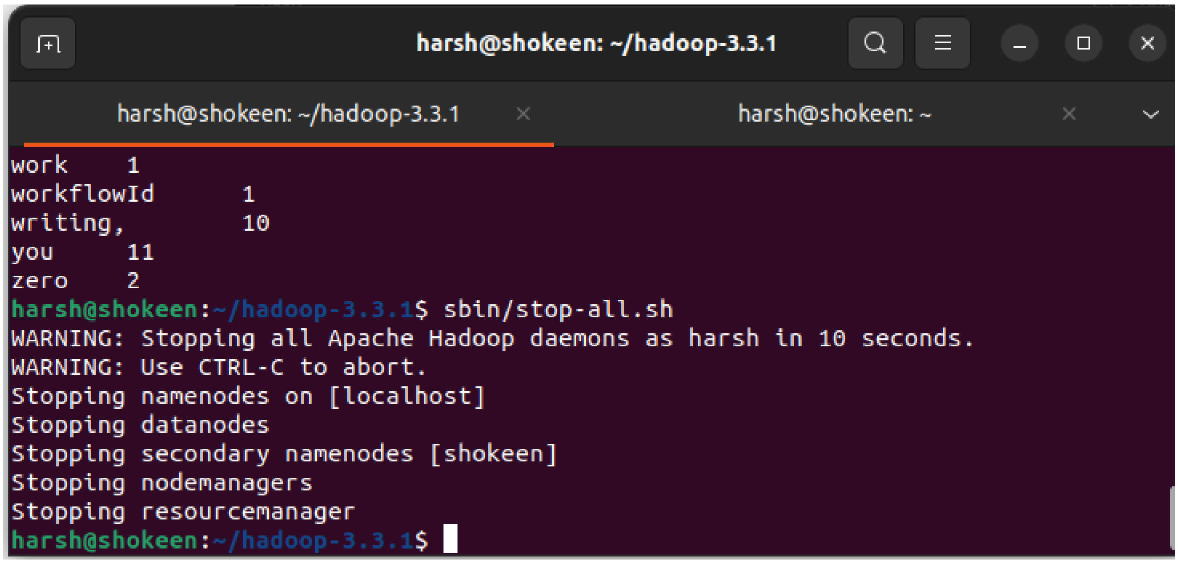
View the output:

bin/hdfs dfs -cat /output/* - This will display the word count output on the console.



```
harsh@shokeen:~/hadoop-3.3.1$ bin/hdfs dfs -cat /output/*
"*"     22
"AS     10
"License");     10
"alice,bob      22
"clumping"      1
"full_queue_name"       1
"priority".     1
"workflowId"    1
(ASF)   1
(as     1
(or     1
(root   1
(the    10
```

Step 14: Stop the Hadoop cluster



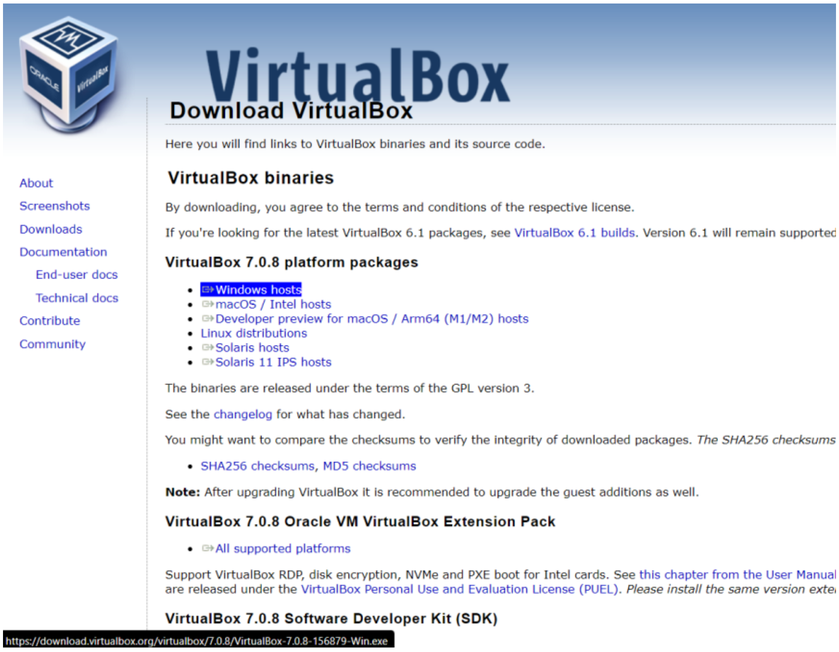
```
work    1
workflowId      1
writing,        10
you     11
zero    2
harsh@shokeen:~/hadoop-3.3.1$ sbin/stop-all.sh
WARNING: Stopping all Apache Hadoop daemons as harsh in 10 seconds.
WARNING: Use CTRL-C to abort.
Stopping namenodes on [localhost]
Stopping datanodes
Stopping secondary namenodes [shokeen]
Stopping nodemanagers
Stopping resourcemanager
harsh@shokeen:~/hadoop-3.3.1$ 
```

These steps should guide you through setting up a Hadoop single-node cluster and running the Word Count

Q9. Install VirtualBox/VMware Workstation with different flavors of Linux or windows OS on top of windows7 or 8.

Procedure:

Step 1: Download the Virtual box exe and click the exe file.



Download VirtualBox

Here you will find links to VirtualBox binaries and its source code.

VirtualBox binaries

By downloading, you agree to the terms and conditions of the respective license.

If you're looking for the latest VirtualBox 6.1 packages, see VirtualBox 6.1 builds. Version 6.1 will remain supported

VirtualBox 7.0.8 platform packages

- Windows hosts
- macOS / Intel hosts
- Developer preview for macOS / Arm64 (M1/M2) hosts
- Linux distributions
- Solaris hosts
- Solaris 11 IPS hosts

The binaries are released under the terms of the GPL version 3.

See the changelog for what has changed.

You might want to compare the checksums to verify the integrity of downloaded packages. *The SHA256 checksums*

- SHA256 checksums, MD5 checksums

Note: After upgrading VirtualBox it is recommended to upgrade the guest additions as well.

VirtualBox 7.0.8 Oracle VM VirtualBox Extension Pack

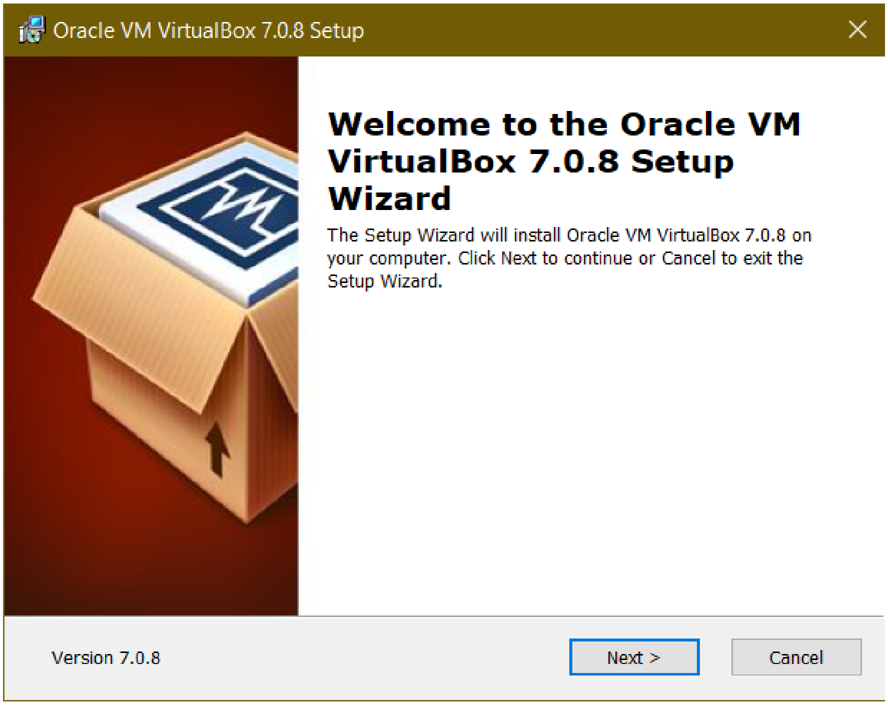
- All supported platforms

Support VirtualBox RDP, disk encryption, NVMe and PXE boot for Intel cards. See this chapter from the User Manual are released under the VirtualBox Personal Use and Evaluation License (PUEL). *Please install the same version exte*

VirtualBox 7.0.8 Software Developer Kit (SDK)

https://download.virtualbox.org/virtualbox/7.0.8/VirtualBox-7.0.8-156879-Win.exe

Step 2: Click the Next Button of the install Wizard



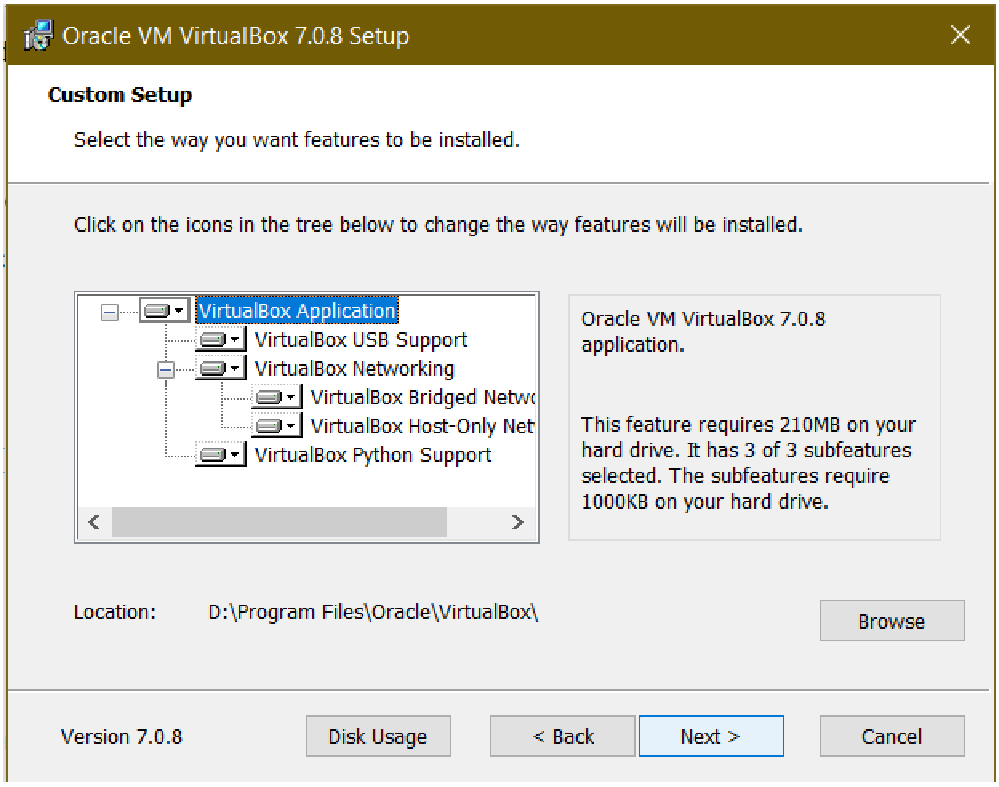
Oracle VM VirtualBox 7.0.8 Setup

Welcome to the Oracle VM VirtualBox 7.0.8 Setup Wizard

The Setup Wizard will install Oracle VM VirtualBox 7.0.8 on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

Version 7.0.8

Step 3: Click the next button.



Oracle VM VirtualBox 7.0.8 Setup

Custom Setup

Select the way you want features to be installed.

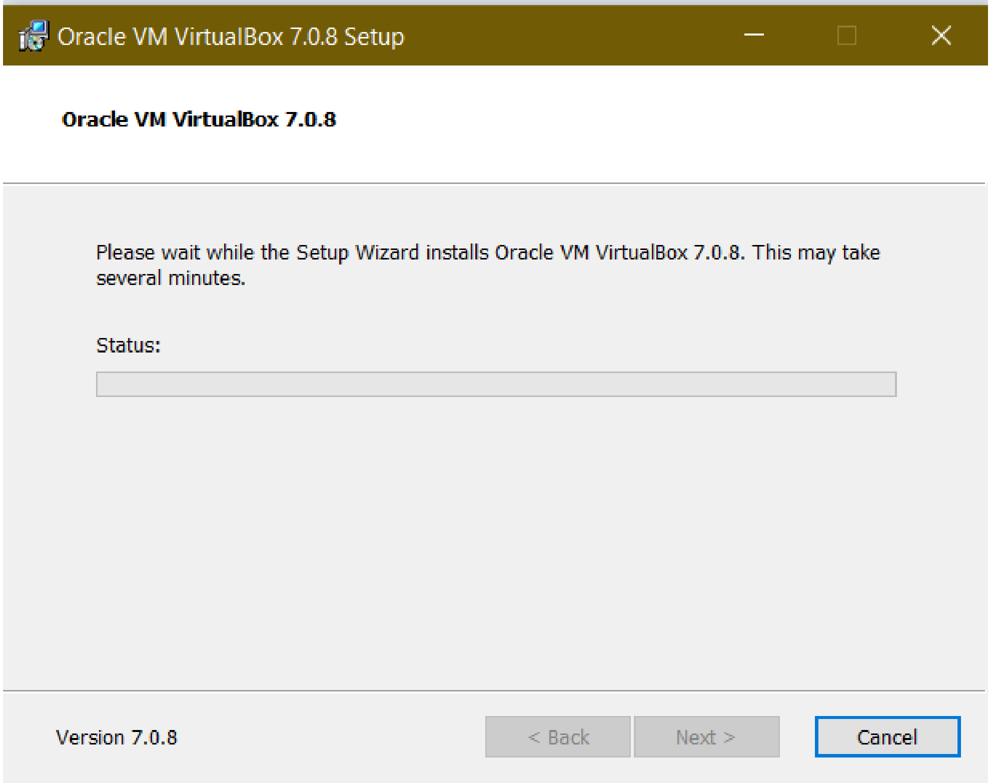
Click on the icons in the tree below to change the way features will be installed.

Oracle VM VirtualBox 7.0.8 application.

This feature requires 210MB on your hard drive. It has 3 of 3 subfeatures selected. The subfeatures require 1000KB on your hard drive.

Location: D:\Program Files\Oracle\VirtualBox\

Step 4: Wait for the installation to your desired location



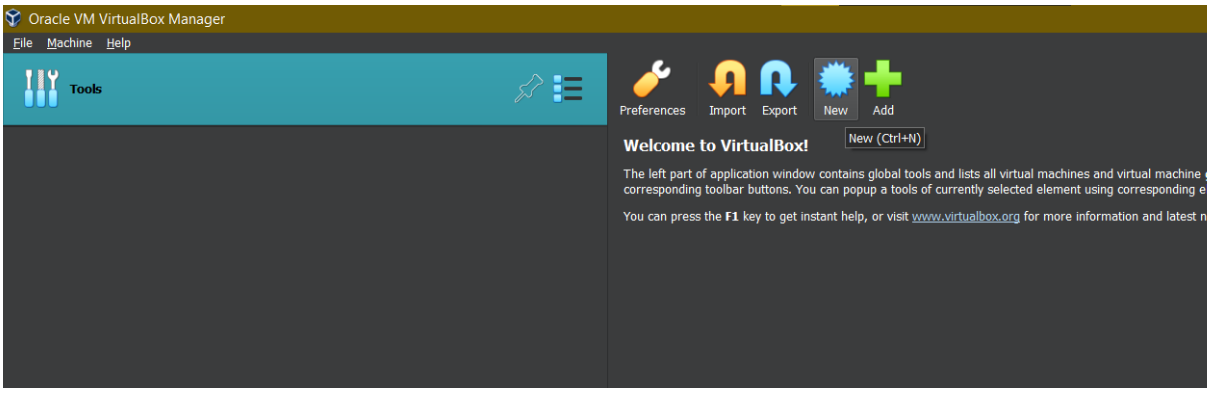
Oracle VM VirtualBox 7.0.8 Setup

Oracle VM VirtualBox 7.0.8

Please wait while the Setup Wizard installs Oracle VM VirtualBox 7.0.8. This may take several minutes.

Status:

Step 5: After the installation, the Virtualbox will open; click New button in blue colour to make a new workstation

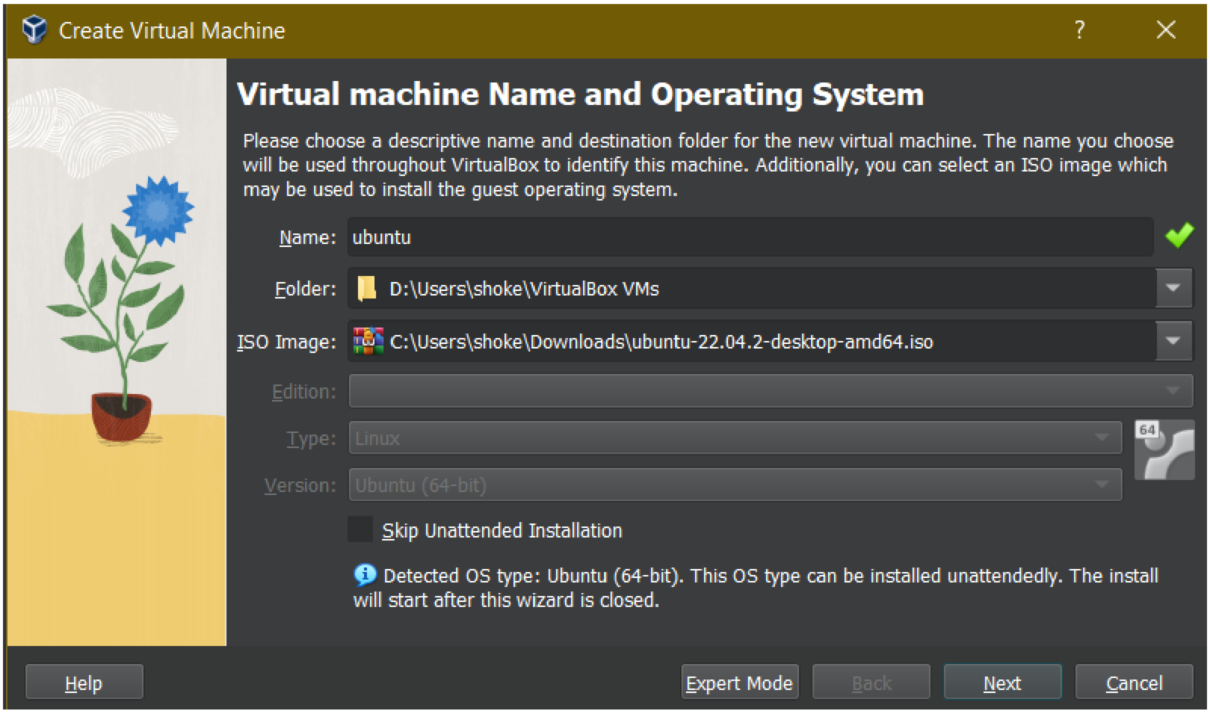


Welcome to VirtualBox!

The left part of application window contains global tools and lists all virtual machines and virtual machine corresponding toolbar buttons. You can popup a tools of currently selected element using corresponding e

You can press the **F1** key to get instant help, or visit www.virtualbox.org for more information and latest n

Step 6: After opening the installation Wizard selected the .iso file of your ubuntu downloaded and proceed to next steps.



Virtual machine Name and Operating System

Please choose a descriptive name and destination folder for the new virtual machine. The name you choose will be used throughout VirtualBox to identify this machine. Additionally, you can select an ISO image which may be used to install the guest operating system.

Name: ubuntu

Folder: D:\Users\shoke\VirtualBox VMs

ISO Image: C:\Users\shoke\Downloads\ubuntu-22.04.2-desktop-amd64.iso

Edition:

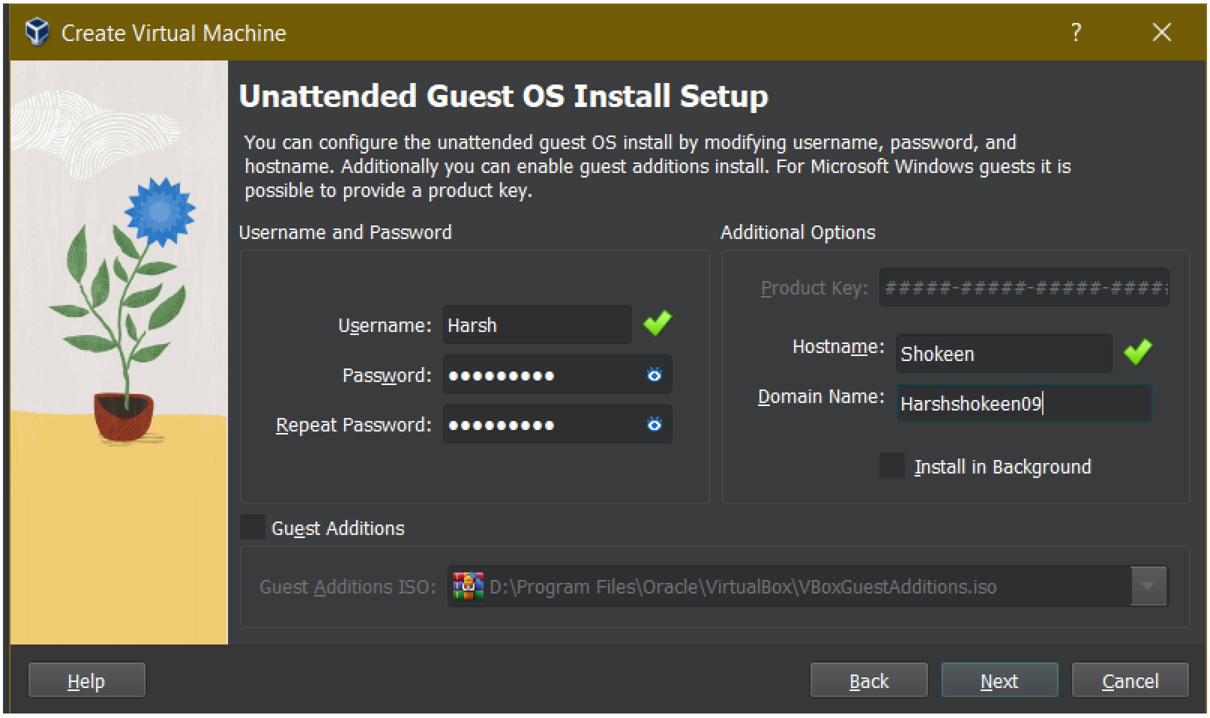
Type: Linux

Version: Ubuntu (64-bit)

□ Skip Unattended Installation

Detected OS type: Ubuntu (64-bit). This OS type can be installed unattendedly. The install will start after this wizard is closed.

Step 7: Give your name or the desired name to your system/workstation going to be created. Remember it is the same name as your localhost.



Create Virtual Machine

Unattended Guest OS Install Setup

You can configure the unattended guest OS install by modifying username, password, and hostname. Additionally you can enable guest additions install. For Microsoft Windows guests it is possible to provide a product key.

Username and Password

Username: Harsh ✓

Password: •••••••••

Repeat Password: •••••••••

Additional Options

Product Key: #####-#####-#####-####

Hostname: Shokeen ✓

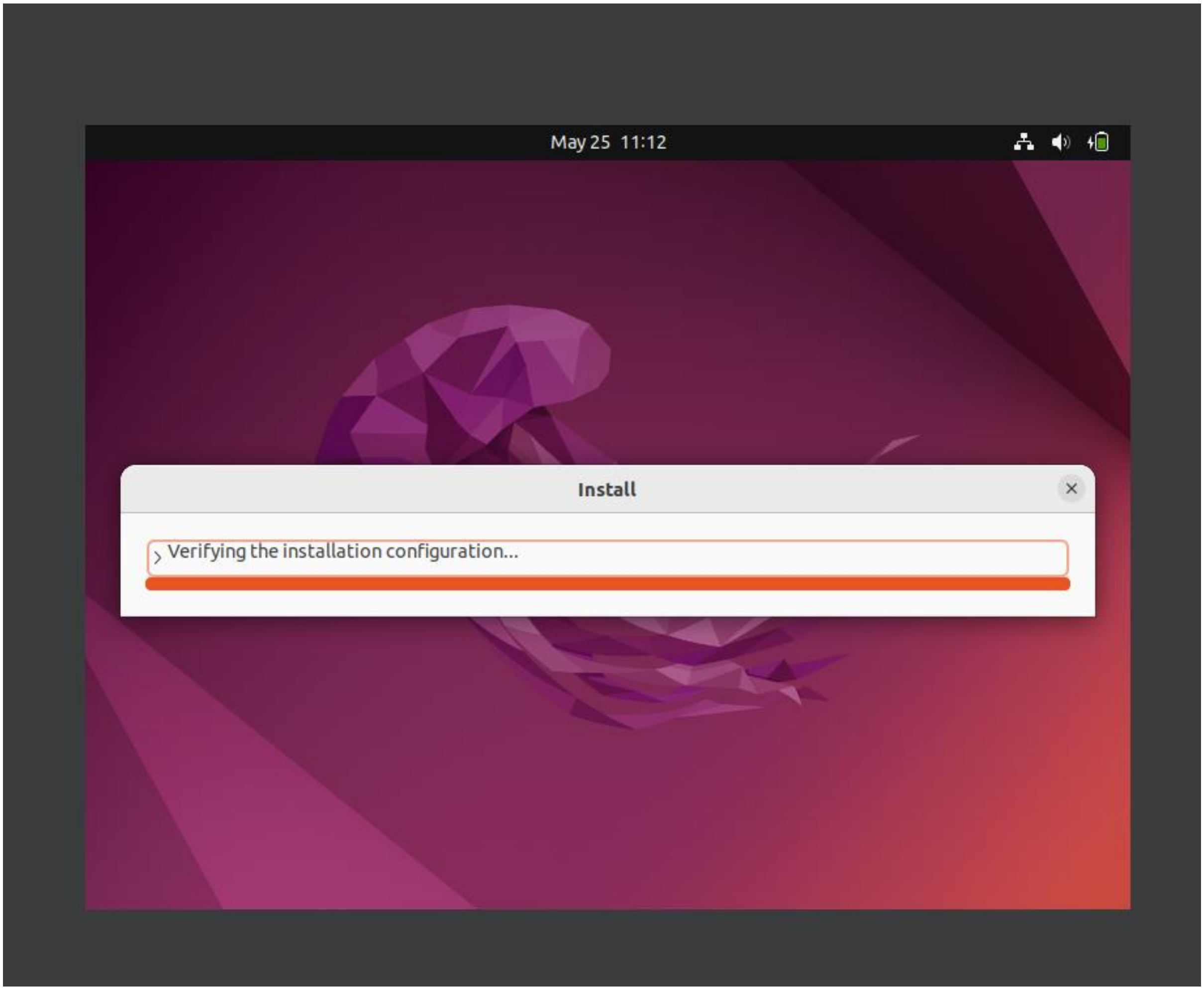
Domain Name: Harshshokeen09

□ Install in Background

□ Guest Additions

Guest Additions ISO: D:\Program Files\Oracle\VirtualBox\VBoxGuestAdditions.iso

Step 8: After clicking next your ubuntu will load and start installing in virtualbox.



May 25 11:12

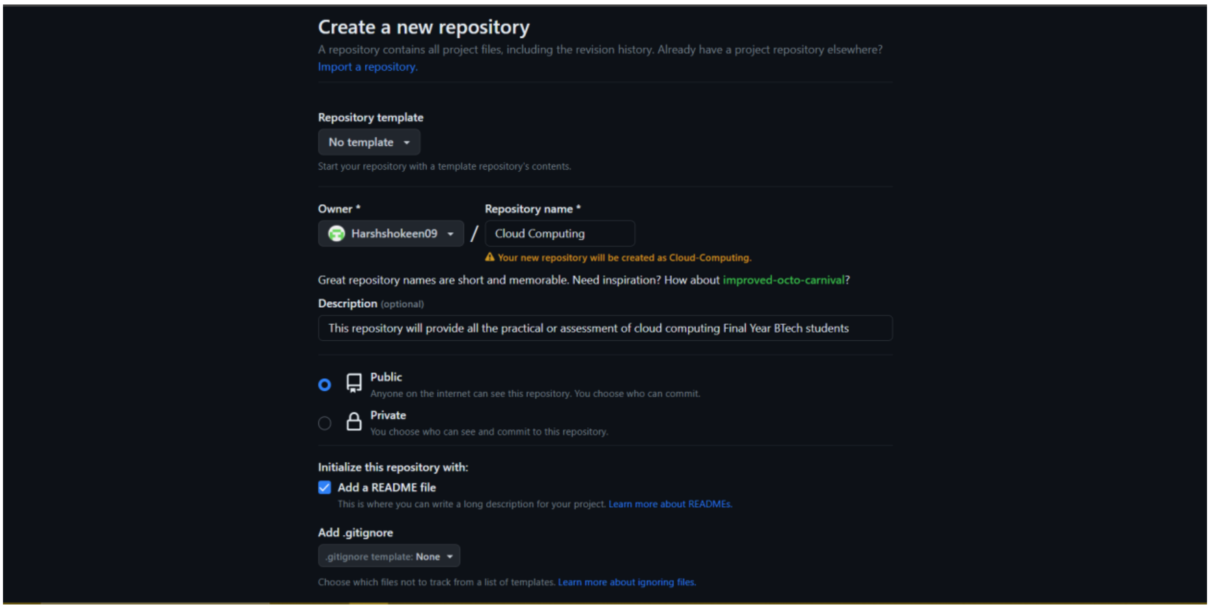
Install

Verifying the installation configuration...

Q10. GIT Commands Execution and GUI & CLI understanding.

Procedure

Step 1: Login to your GitHyb account and click on create repository.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Repository template: No template

Start your repository with a template repository's contents.

Owner *: Harshshokeen09 / Repository name *: Cloud Computing

⚠ Your new repository will be created as Cloud-Computing.

Great repository names are short and memorable. Need inspiration? How about improved-octo-carnival?

Description (optional): This repository will provide all the practical or assessment of cloud computing Final Year BTech students

● Public — Anyone on the internet can see this repository. You choose who can commit.

○ Private — You choose who can see and commit to this repository.

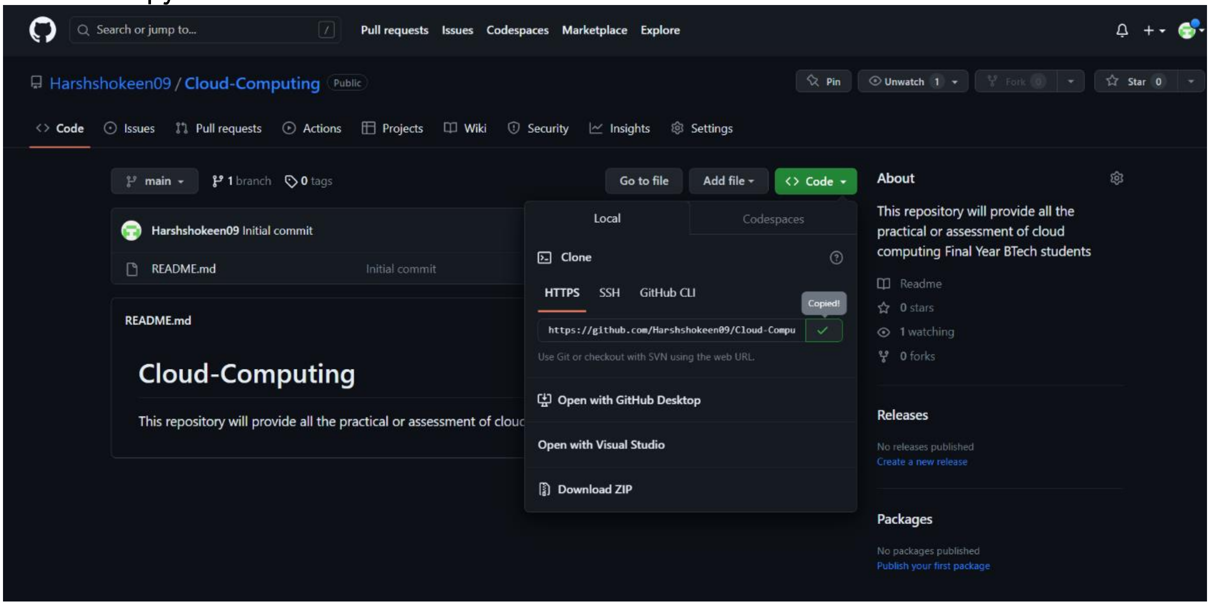
Initialize this repository with:

☑ Add a README file — This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore: .gitignore template: None

Choose which files not to track from a list of templates. Learn more about ignoring files.

Step 2: After creating the repository manually transfer a file or just add information in readme file. And copy the link in code section.



Harshshokeen09 Initial commit

README.md — Initial commit

README.md

Cloud-Computing

This repository will provide all the practical or assessment of clouc

https://github.com/Harshshokeen09/Cloud-Compu

Use Git or checkout with SVN using the web URL.

About

This repository will provide all the practical or assessment of cloud computing Final Year BTech students

Readme

0 stars

1 watching

0 forks

Releases

No releases published

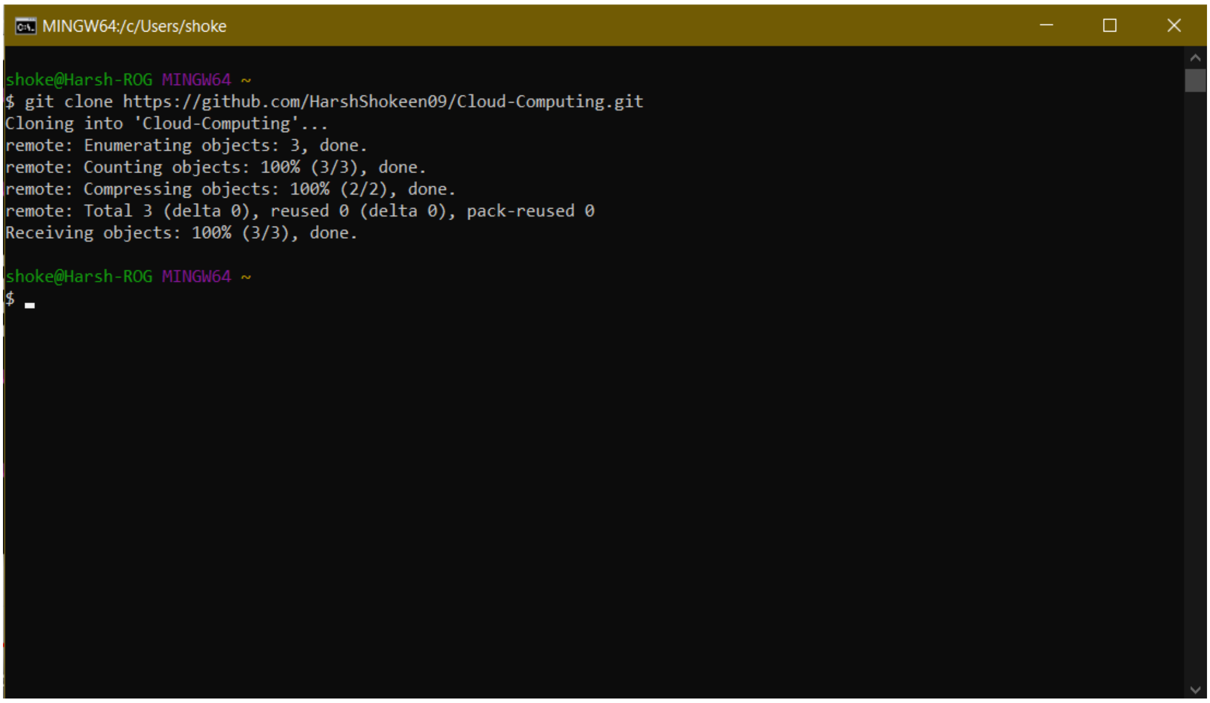
Create a new release

Packages

No packages published

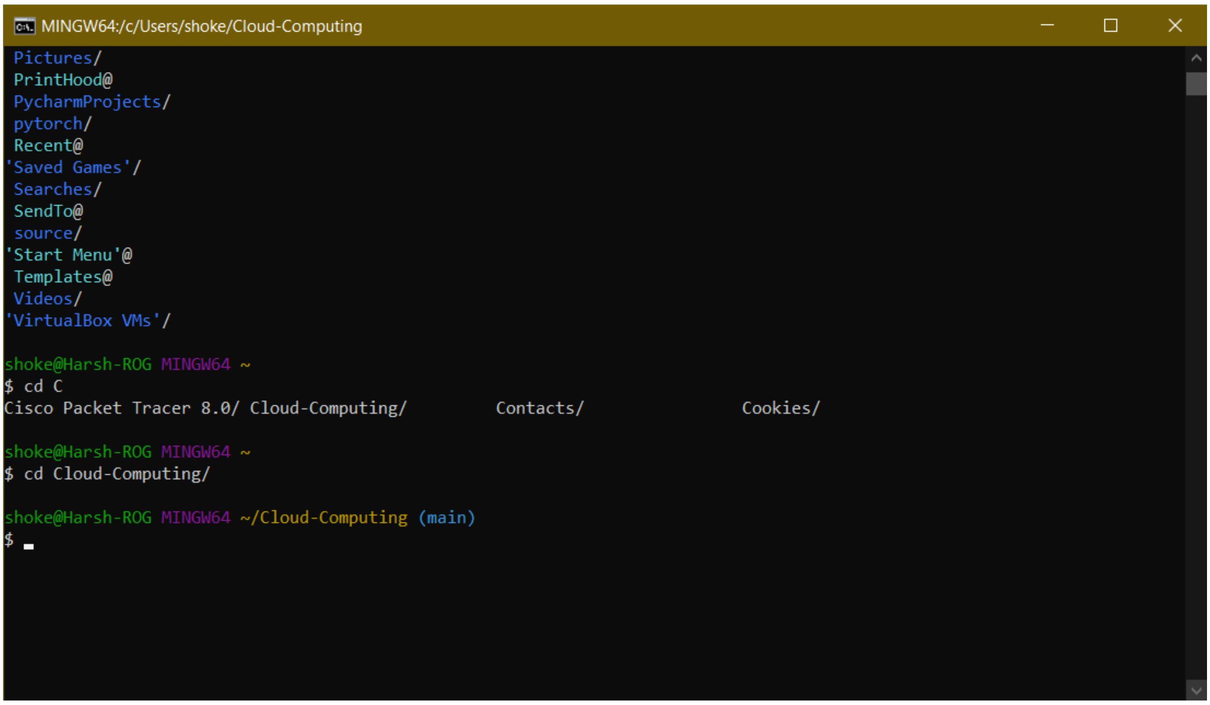
Publish your first package

Step 3: Open git-bash and clone the repository by pasting the link and writing the following command.



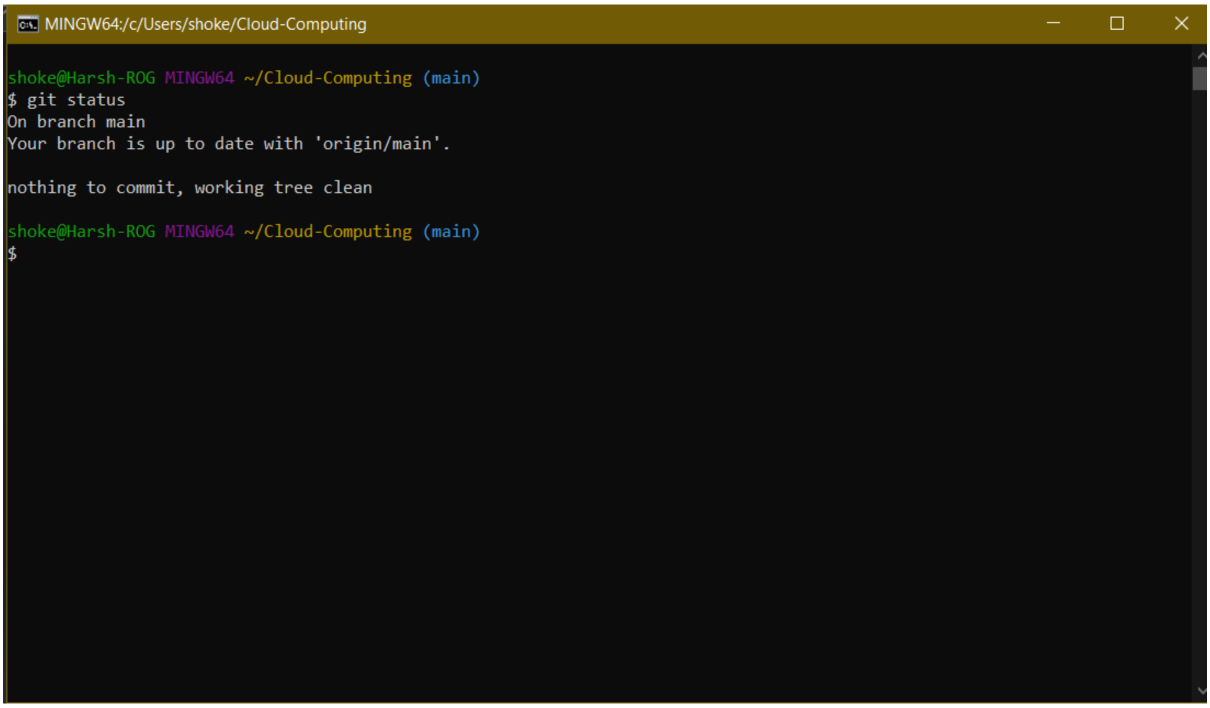
```
shoke@Harsh-ROG MINGW64 ~
$ git clone https://github.com/HarshShokeen09/Cloud-Computing.git
Cloning into 'Cloud-Computing'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

shoke@Harsh-ROG MINGW64 ~
$ 
```

Step 4: Navigate to the desired folder. Therefore, making it the main/master branch where all the actions would be reflected.



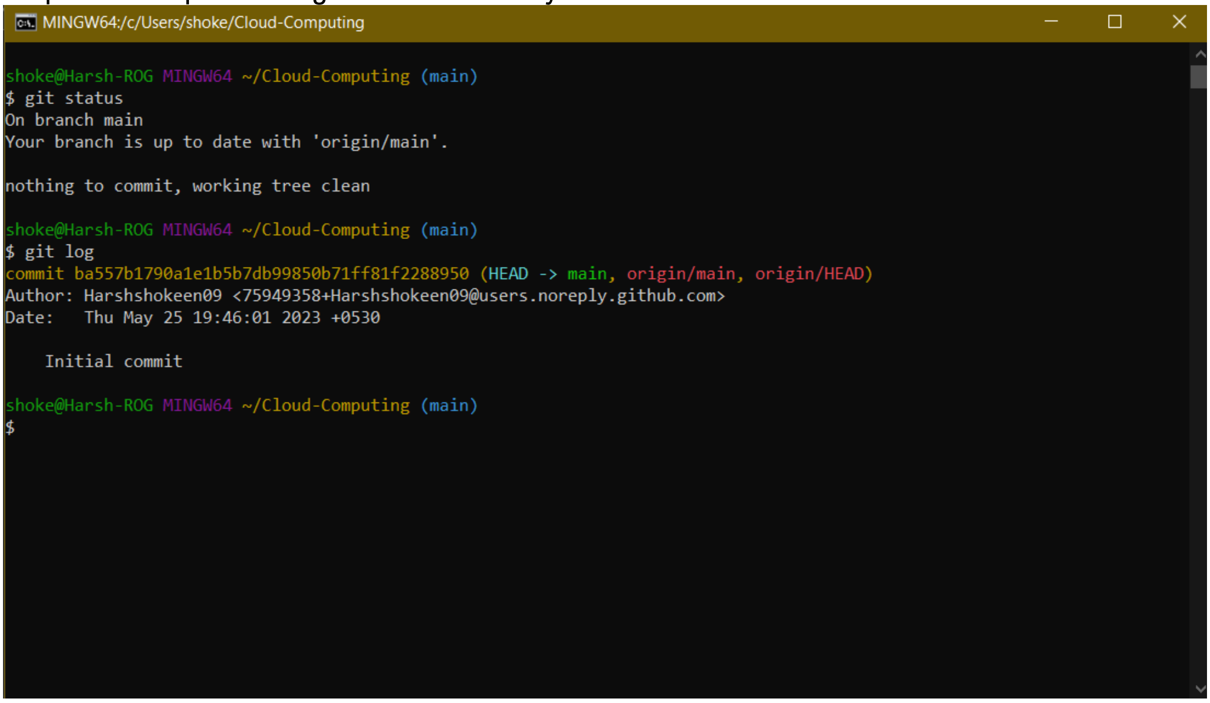
```
 Pictures/
 PrintHood@
 PycharmProjects/
 pytorch/
 Recent@
'Saved Games'/
 Searches/
 SendTo@
 source/
'Start Menu'@
 Templates@
 Videos/
'VirtualBox VMs'/

shoke@Harsh-ROG MINGW64 ~
$ cd C
Cisco Packet Tracer 8.0/ Cloud-Computing/         Contacts/                 Cookies/

shoke@Harsh-ROG MINGW64 ~
$ cd Cloud-Computing/

shoke@Harsh-ROG MINGW64 ~/Cloud-Computing (main)
$ 
```

Step 5: Now check the status of your repository.



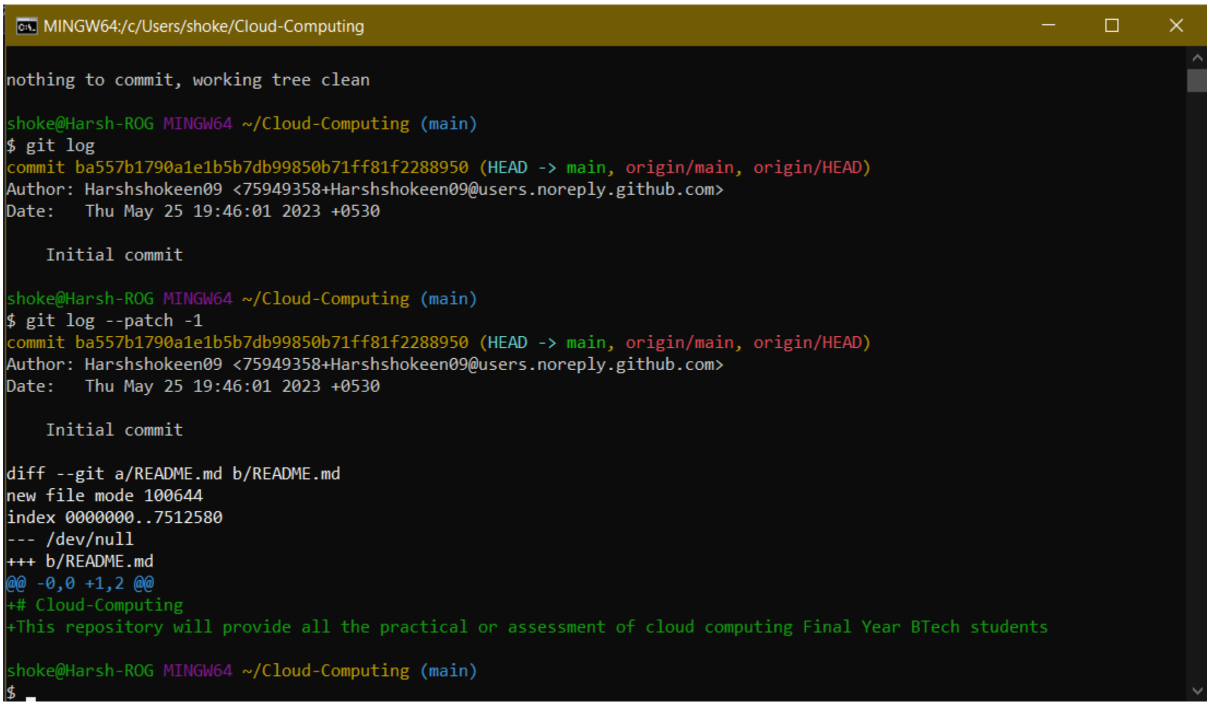
```
shoke@Harsh-ROG MINGW64 ~/Cloud-Computing (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

shoke@Harsh-ROG MINGW64 ~/Cloud-Computing (main)
$
```

Step 6: Now open the log to see the history of commits.



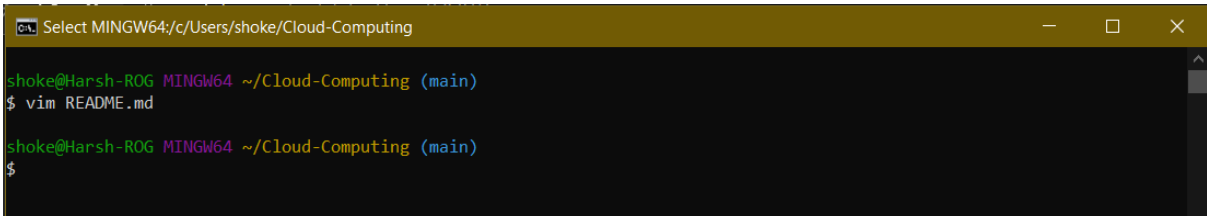
```
shoke@Harsh-ROG MINGW64 ~/Cloud-Computing (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

shoke@Harsh-ROG MINGW64 ~/Cloud-Computing (main)
$ git log
commit ba557b1790a1e1b5b7db99850b71ff81f2288950 (HEAD -> main, origin/main, origin/HEAD)
Author: Harshshokeen09 <75949358+Harshshokeen09@users.noreply.github.com>
Date:   Thu May 25 19:46:01 2023 +0530

    Initial commit

shoke@Harsh-ROG MINGW64 ~/Cloud-Computing (main)
$
```

Step 7: Now to see the history of commits content open its patch version



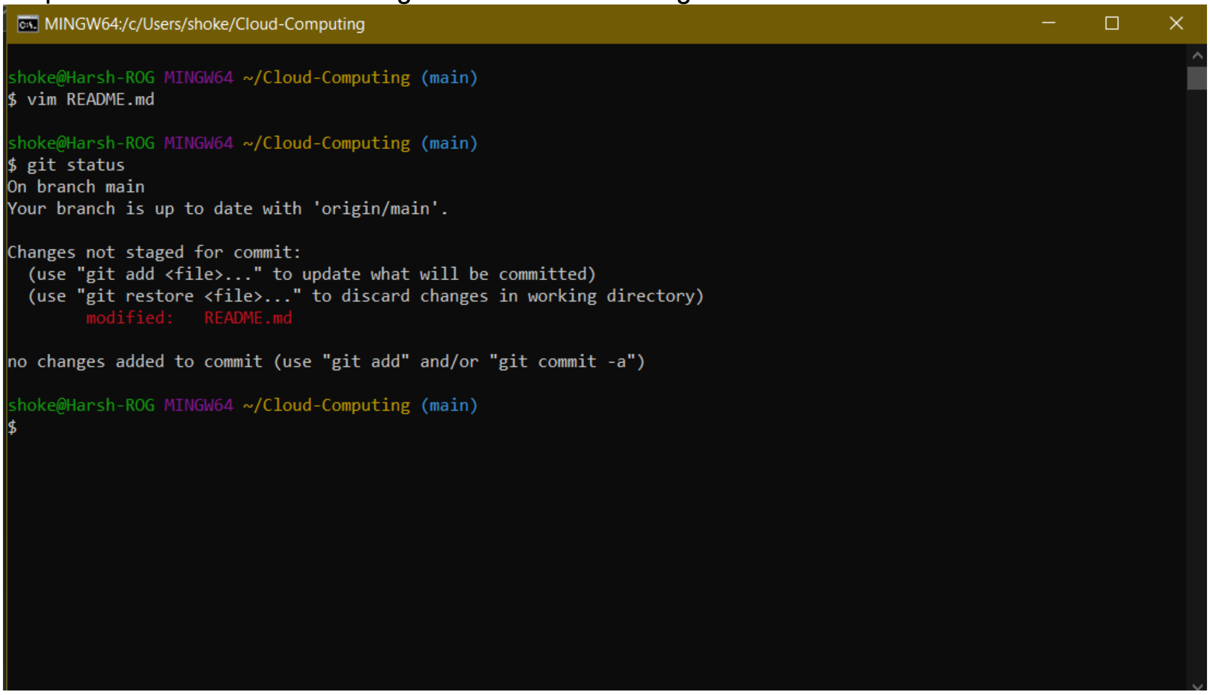
```
nothing to commit, working tree clean

shoke@Harsh-ROG MINGW64 ~/Cloud-Computing (main)
$ git log
commit ba557b1790a1e1b5b7db99850b71ff81f2288950 (HEAD -> main, origin/main, origin/HEAD)
Author: Harshshokeen09 <75949358+Harshshokeen09@users.noreply.github.com>
Date:   Thu May 25 19:46:01 2023 +0530

    Initial commit

shoke@Harsh-ROG MINGW64 ~/Cloud-Computing (main)
$ git log --patch -1
commit ba557b1790a1e1b5b7db99850b71ff81f2288950 (HEAD -> main, origin/main, origin/HEAD)
Author: Harshshokeen09 <75949358+Harshshokeen09@users.noreply.github.com>
Date:   Thu May 25 19:46:01 2023 +0530

    Initial commit

diff --git a/README.md b/README.md
new file mode 100644
index 0000000..7512580
--- /dev/null
+++ b/README.md
@@ -0,0 +1,2 @@
+# Cloud-Computing
+This repository will provide all the practical or assessment of cloud computing Final Year BTech students

shoke@Harsh-ROG MINGW64 ~/Cloud-Computing (main)
$ 
```

Step 8: Now open the readme file to make the first commit.



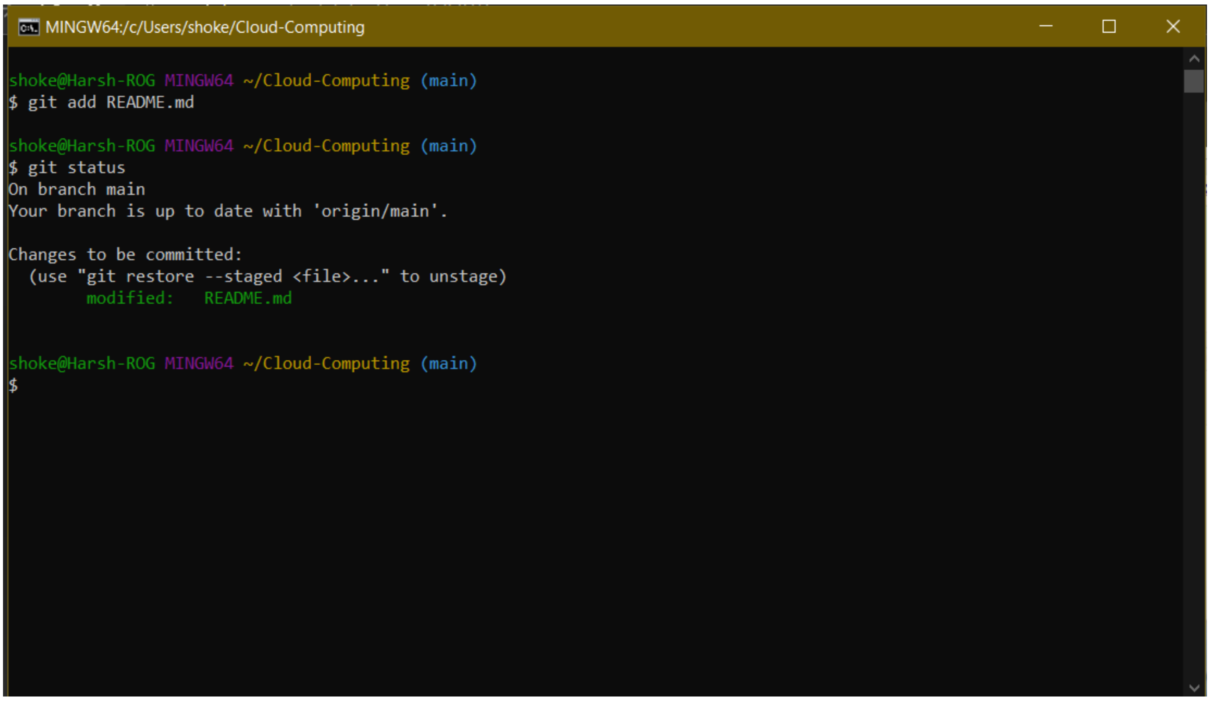
```
shoke@Harsh-ROG MINGW64 ~/Cloud-Computing (main)
$ vim README.md

shoke@Harsh-ROG MINGW64 ~/Cloud-Computing (main)
$
```

Step 9: now check the status again before committing.



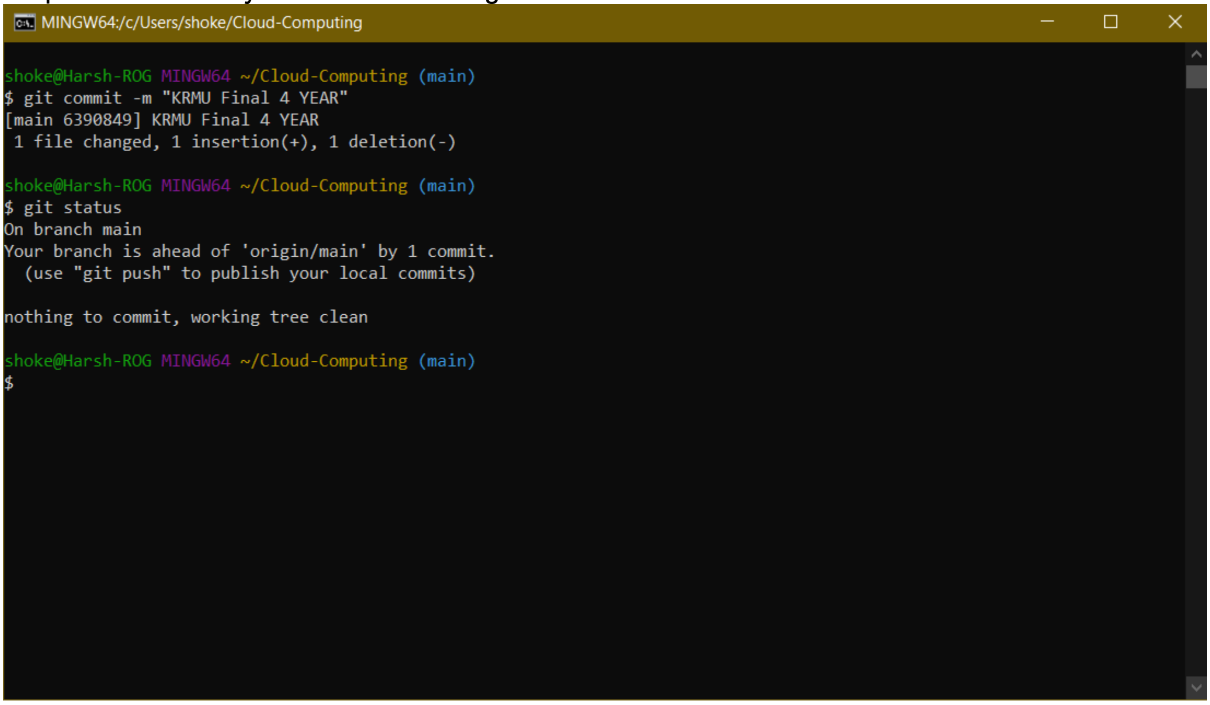
```
shoke@Harsh-ROG MINGW64 ~/Cloud-Computing (main)
$ vim README.md

shoke@Harsh-ROG MINGW64 ~/Cloud-Computing (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")

shoke@Harsh-ROG MINGW64 ~/Cloud-Computing (main)
$
```

Step 10: Now add changes in the readme.md file and save them using Crlt+O and exit with Crlt+X. Again, check the status to see the committed changes.



```
shoke@Harsh-ROG MINGW64 ~/Cloud-Computing (main)
$ git add README.md

shoke@Harsh-ROG MINGW64 ~/Cloud-Computing (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md


shoke@Harsh-ROG MINGW64 ~/Cloud-Computing (main)
$
```

Step 11: Now finally Commit the changes.



```
shoke@Harsh-ROG MINGW64 ~/Cloud-Computing (main)
$ git commit -m "KRMU Final 4 YEAR"
[main 6390849] KRMU Final 4 YEAR
 1 file changed, 1 insertion(+), 1 deletion(-)

shoke@Harsh-ROG MINGW64 ~/Cloud-Computing (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

shoke@Harsh-ROG MINGW64 ~/Cloud-Computing (main)
$
```