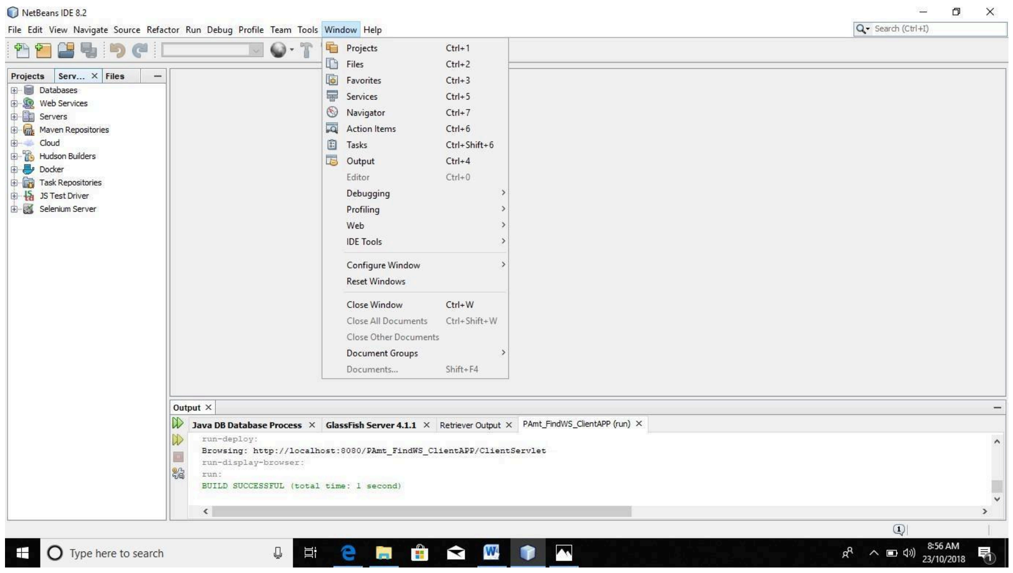
Practical-3

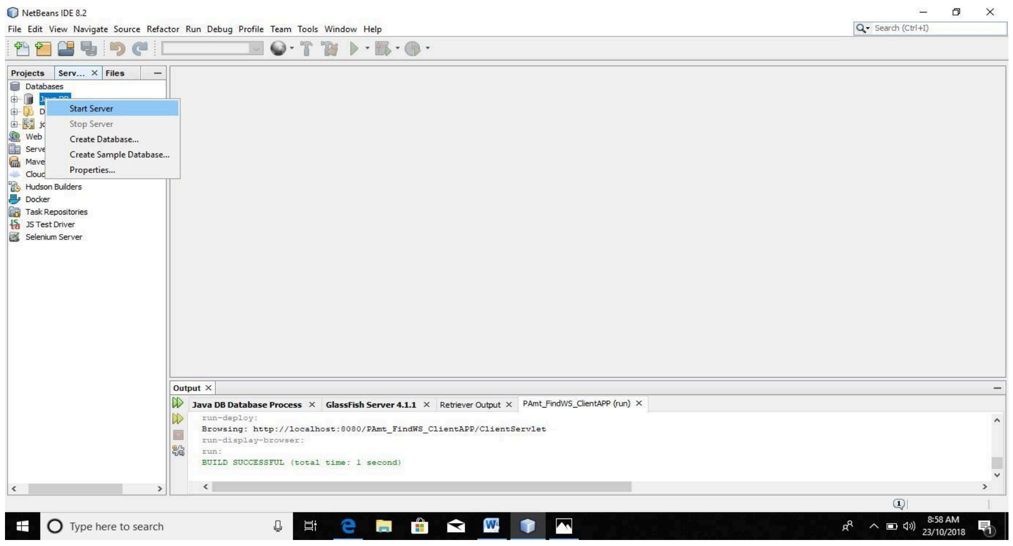
Create a Simple REST Service to demonstrate CRUD operations with "Student" database

1. **Click on Window menu and click on Projects, Files & Services to open it.**



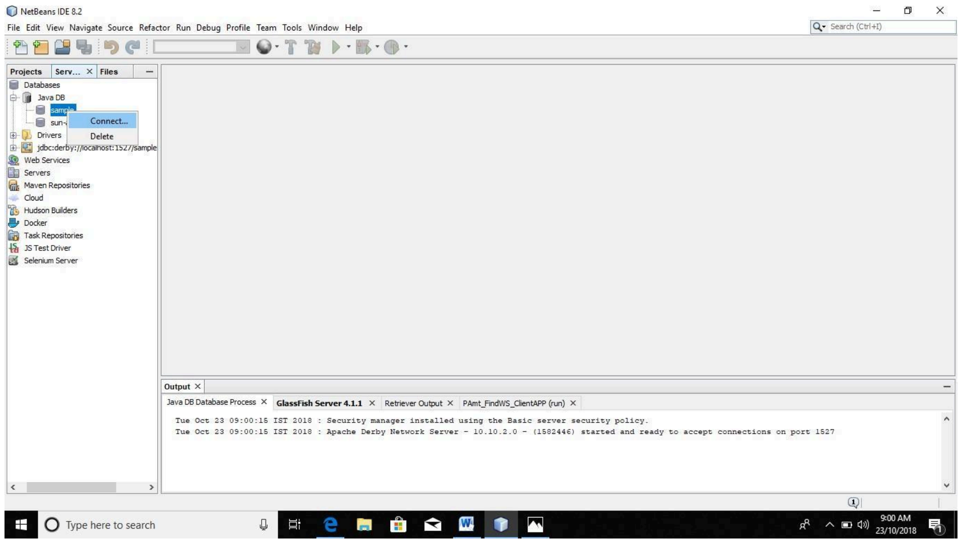
```
run-deploy:
Browsing: http://localhost:8080/PAmt_FindWS_ClientAPP/ClientServlet
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 1 second)
```

2. **Right click on Java DB and then click on Start Server to start the server .**



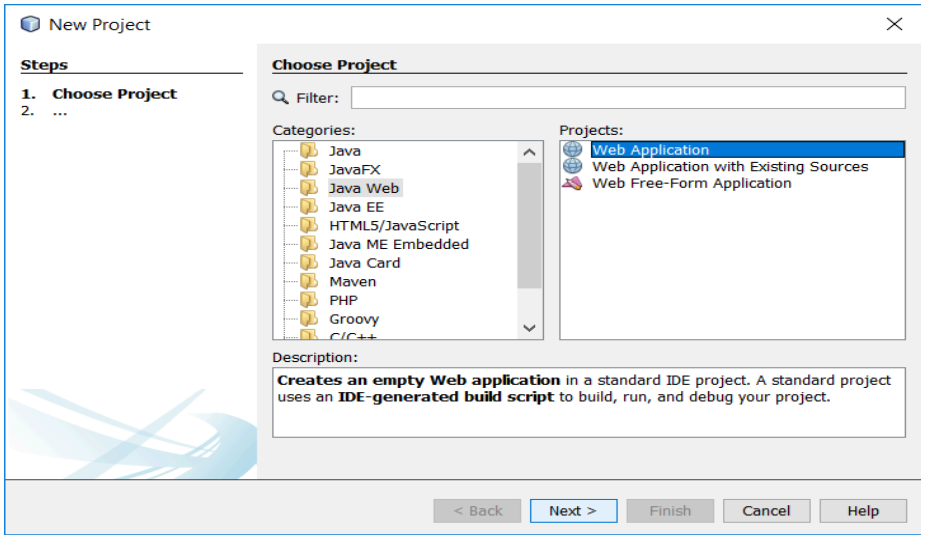
```
run-deploy:
Browsing: http://localhost:8080/PAmt_FindWS_ClientAPP/ClientServlet
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 1 second)
```

3. **Now expand Java DB and right click on sample and then click on connect to connect the sample database with server.**

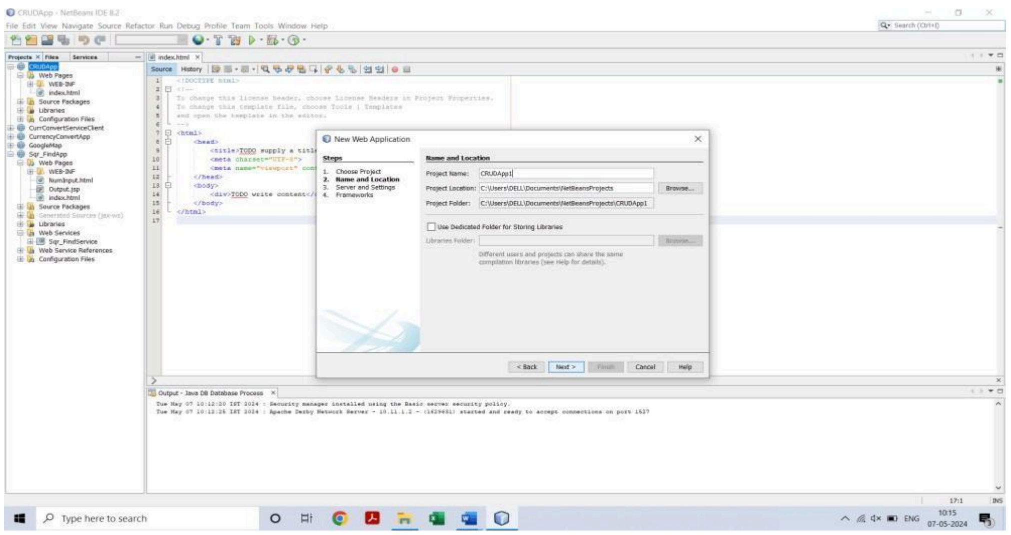


```
Tue Oct 23 09:00:15 IST 2018 : Security manager installed using the Basic server security policy.
Tue Oct 23 09:00:15 IST 2018 : Apache Derby Network Server - 10.10.2.0 - (1582446) started and ready to accept connections on port 1527
```

4. **Now create a web application with the name CRUD_Operation. A window will open like following pic.**



Creates an empty Web application in a standard IDE project. A standard project uses an **IDE-generated build script** to build, run, and debug your project.



```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
    <head>
        <title>TODO supply a titl
        <meta charset="UTF-8">
        <meta name="viewport" cont
    </head>
    <body>
        <div>TODO write content</
    </body>
</html>
```

Project Name: CRUDApp1

Project Location: C:\Users\DELL\Documents\NetBeansProjects

Project Folder: C:\Users\DELL\Documents\NetBeansProjects\CRUDApp1

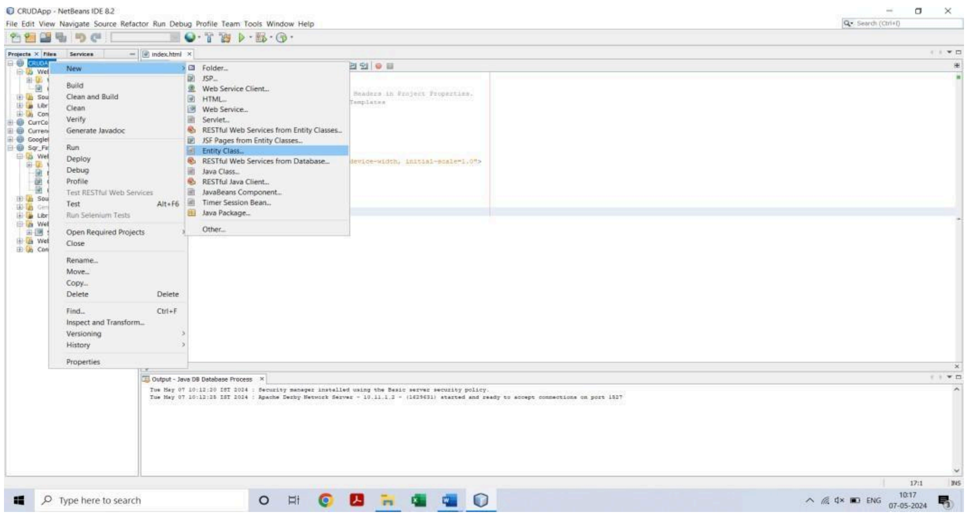
☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

```
Tue May 07 10:12:20 IST 2024 : Security manager installed using the Basic server security policy.
Tue May 07 10:12:26 IST 2024 : Apache Derby Network Server - 10.11.1.2 - (1629631) started and ready to accept connections on port 1527
```

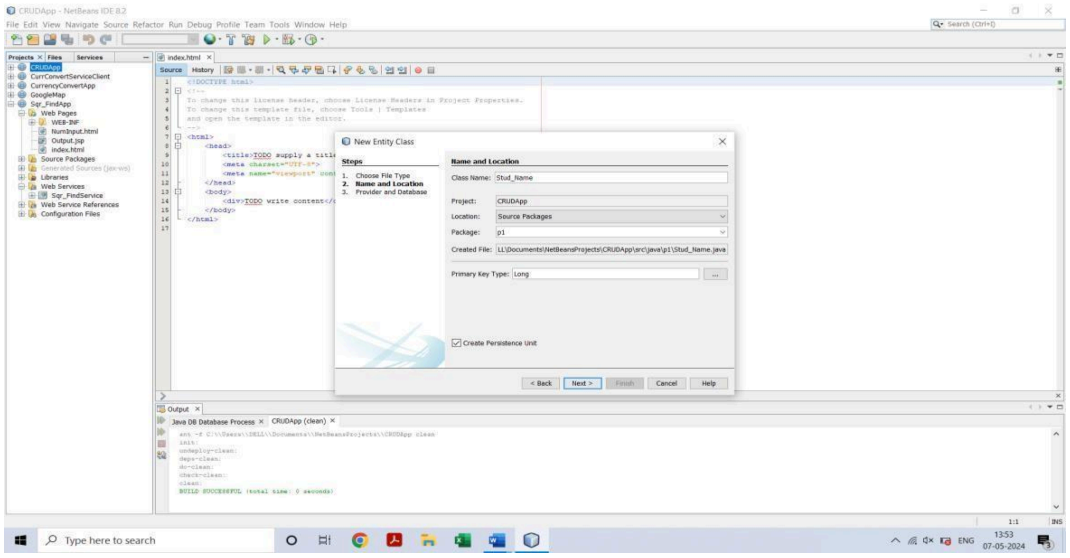
5. **Create an entity class. Right click on project name -> New -> Entity Class.**



Output - Java DB Database Process

```
Tue May 07 10:12:20 IST 2024 : Security manager installed using the Basic server security policy.
Tue May 07 10:12:25 IST 2024 : Apache Derby Network Server - 10.11.1.2 - (1629631) started and ready to accept connections on port 1527
```

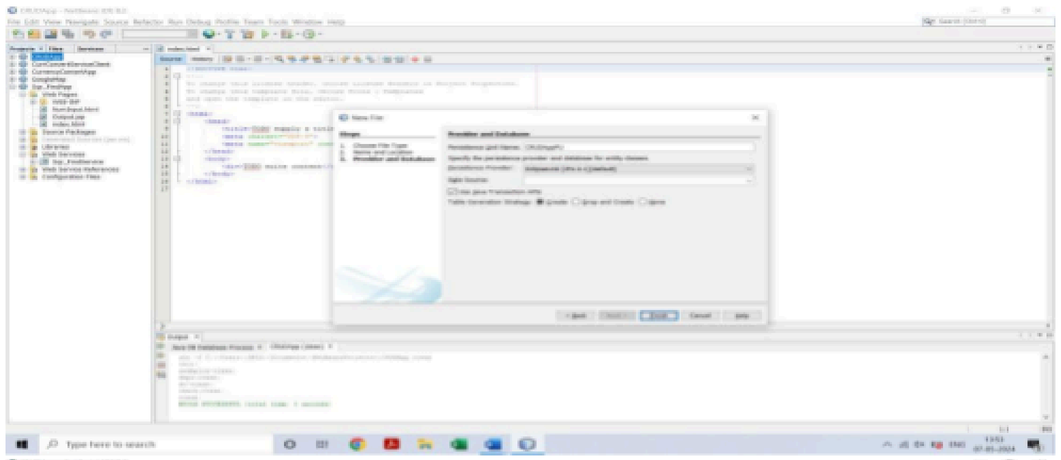
6. **A window will appear like bellow pic. Enter following data and click on Next ….**
   Class Name ->Stud_Name Package Name -> p1
7. **Click on finish.**

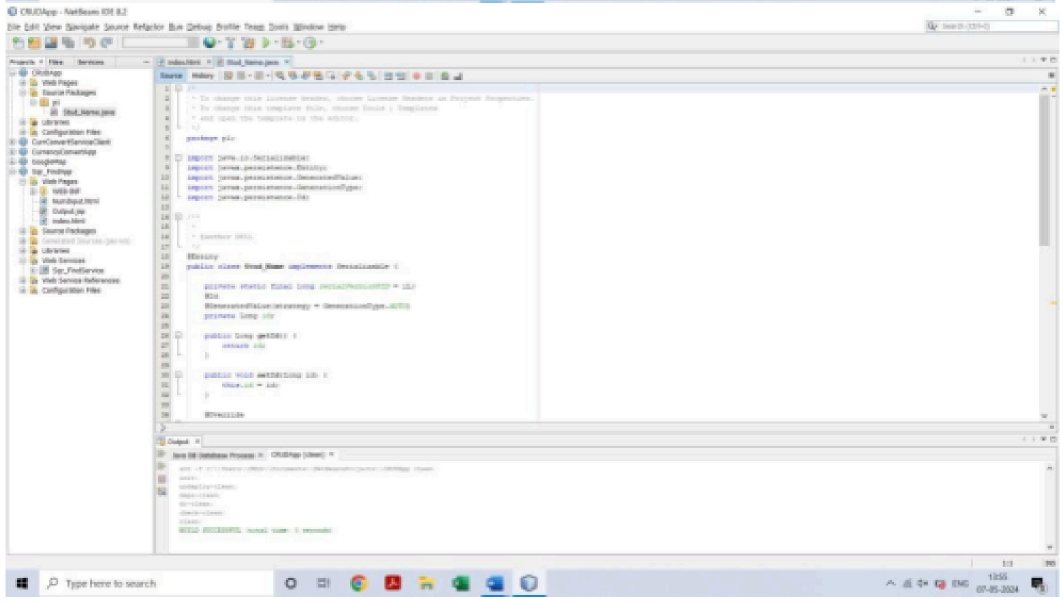


```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
    <head>
        <title>TODO supply a title
        <meta charset="UTF-8">
        <meta name="viewport" cont
    </head>
    <body>
        <div>TODO write content</
    </body>
</html>
```

Class Name: Stud_Name
Project: CRUDApp
Location: Source Packages
Package: p1
Created File: LL\Documents\NetBeansProjects\CRUDApp\src\java\p1\Stud_Name.java
Primary Key Type: Long
[✓] Create Persistence Unit

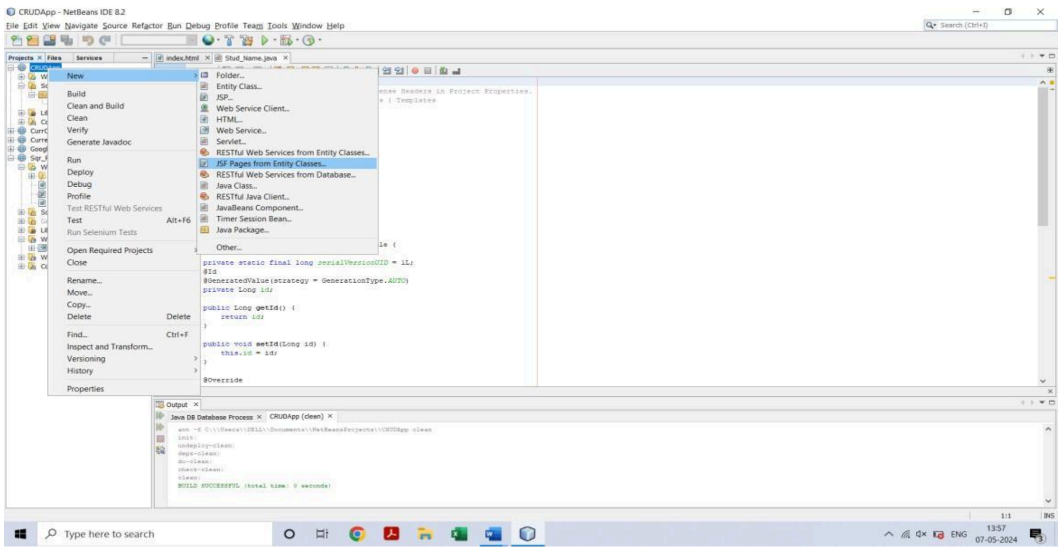
```
ant -f C:\\Users\\DELL\\Documents\\NetBeansProjects\\CRUDApp clean
init:
undeploy-clean:
deps-clean:
do-clean:
check-clean:
clean:
BUILD SUCCESSFUL (total time: 0 seconds)
```





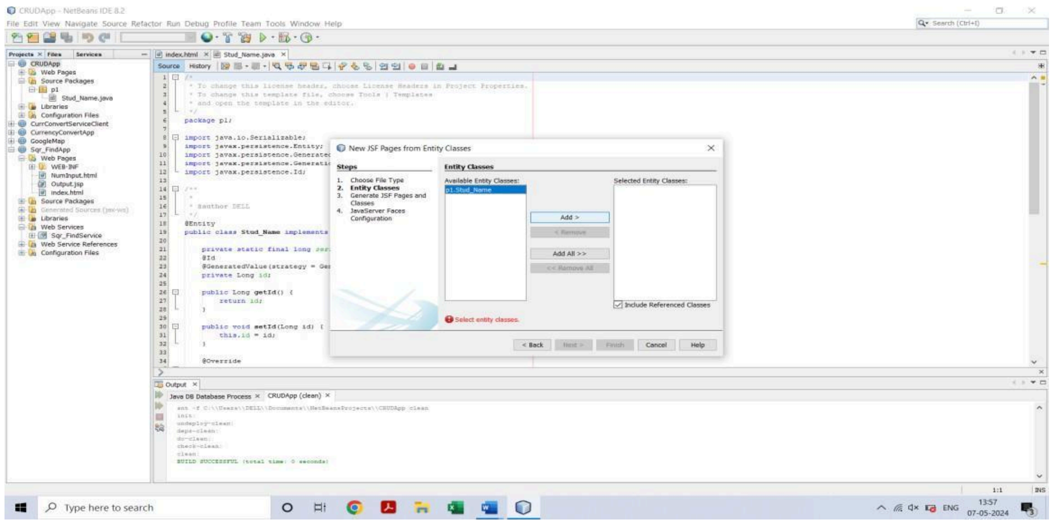
```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package pk;

import java.io.Serializable;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

/**
 *
 * @author DELL
 */
@Entity
public class Stud_Name implements Serializable {

    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    @Override
```

8. **Right click on project name and create JSF Pages from Entity Classes.**
9. **Select and click on Add button and then Next button on below**



```
    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    @Override
```

```
ant -f C:\\Users\\DELL\\Documents\\NetBeansProjects\\CRUDApp clean
init:
undeploy-clean:
deps-clean:
do-clean:
check-clean:
clean:
BUILD SUCCESSFUL (total time: 0 seconds)
```



New JSF Pages from Entity Classes

Steps
1. Choose File Type
2. **Entity Classes**
3. Generate JSF Pages and Classes
4. JavaServer Faces Configuration

Entity Classes

Available Entity Classes: p1.Stud_Name

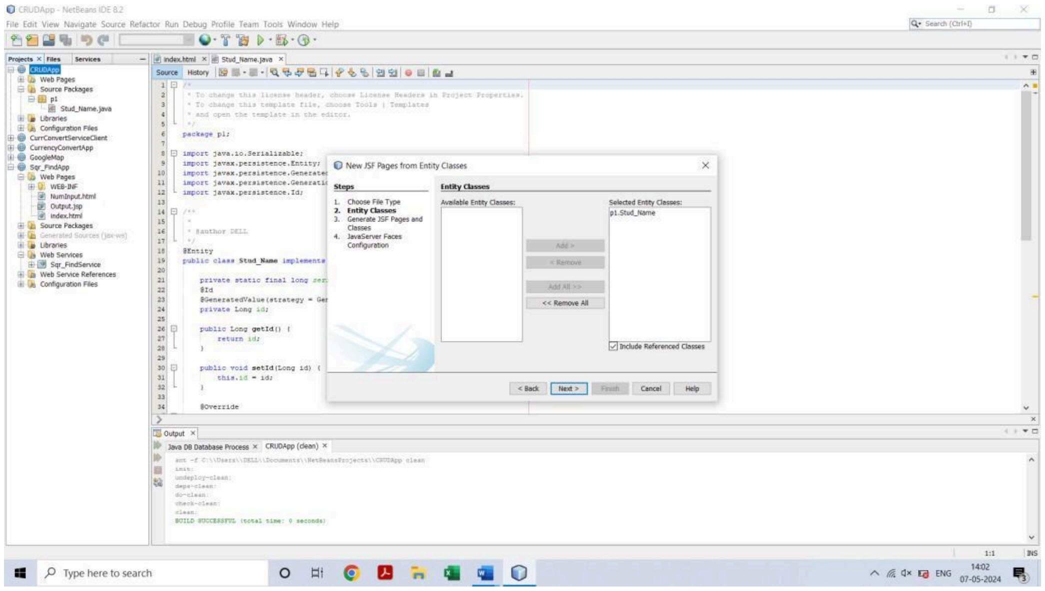
Selected Entity Classes:

☑ Include Referenced Classes

Select entity classes.

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package p1;

import java.io.Serializable;
import javax.persistence.Entity;
import javax.persistence.Generatec
import javax.persistence.Generatic
import javax.persistence.Id;

/**
 *
 * @author DELL
 */
@Entity
public class Stud_Name implements

    private static final long ser
    @Id
    @GeneratedValue(strategy = Ger
    private Long id;

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    @Override
```

```
ant -f C:\\Users\\DELL\\Documents\\NetBeansProjects\\CRUDApp clean
init:
undeploy-clean:
deps-clean:
do-clean:
check-clean:
clean:
BUILD SUCCESSFUL (total time: 0 seconds)
```



New JSF Pages from Entity Classes

Steps
1. Choose File Type
2. **Entity Classes**
3. Generate JSF Pages and Classes
4. JavaServer Faces Configuration

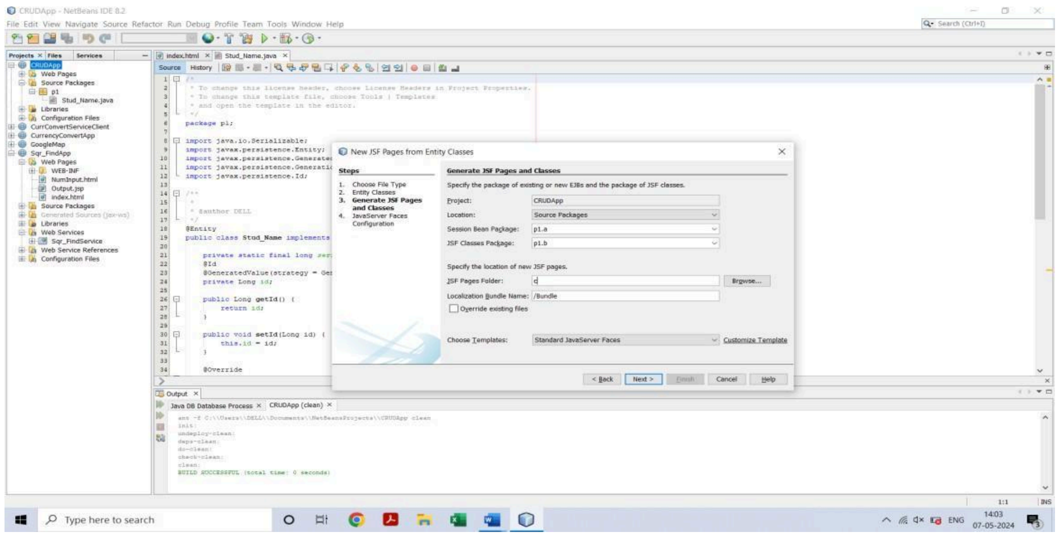
Entity Classes

Available Entity Classes:

Selected Entity Classes: p1.Stud_Name

☑ Include Referenced Classes

10. **A window like below will appear on the screen. Enter the data into that window as entered in below pic and click on Next button.**



```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package p1;

import java.io.Serializable;
import javax.persistence.Entity;
import javax.persistence.Generate
import javax.persistence.Generati
import javax.persistence.Id;

/**
 *
 * @author DELL
 */
@Entity
public class Stud_Name implements

    private static final long ser
    @Id
    @GeneratedValue(strategy = Ge
    private Long id;

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    @Override
```

New JSF Pages from Entity Classes

Steps
1. Choose File Type
2. Entity Classes
3. **Generate JSF Pages and Classes**
4. JavaServer Faces Configuration

Generate JSF Pages and Classes

Specify the package of existing or new EJBs and the package of JSF classes.

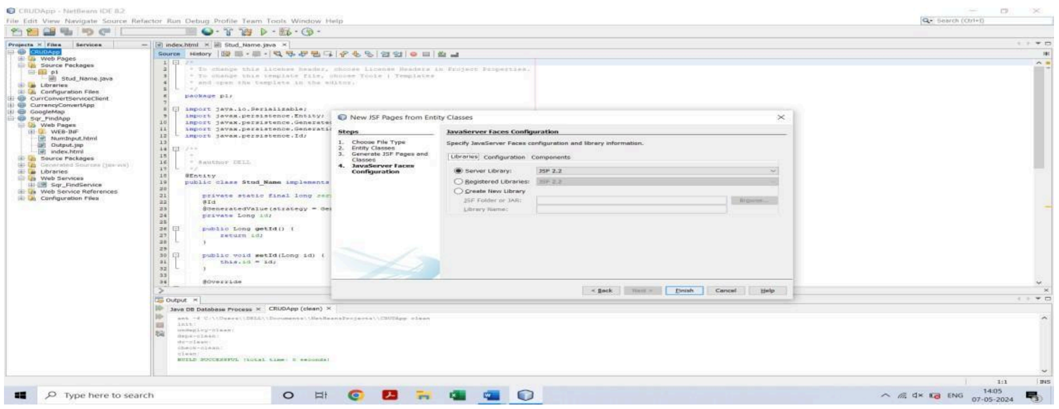
Project: CRUDApp
Location: Source Packages
Session Bean Package: p1.a
JSF Classes Package: p1.b

Specify the location of new JSF pages.

JSF Pages Folder: c
Localization Bundle Name: /Bundle
Override existing files
Choose Templates: Standard JavaServer Faces

```
ant -f C:\\Users\\DELL\\Documents\\NetBeansProjects\\CRUDApp clean
init:
undeploy-clean:
deps-clean:
do-clean:
check-clean:
clean:
BUILD SUCCESSFUL (total time: 0 seconds)
```

11. **Now click on Finish.**



New JSF Pages from Entity Classes

Steps
1. Choose File Type
2. Entity Classes
3. Generate JSF Pages and Classes
4. **JavaServer Faces Configuration**

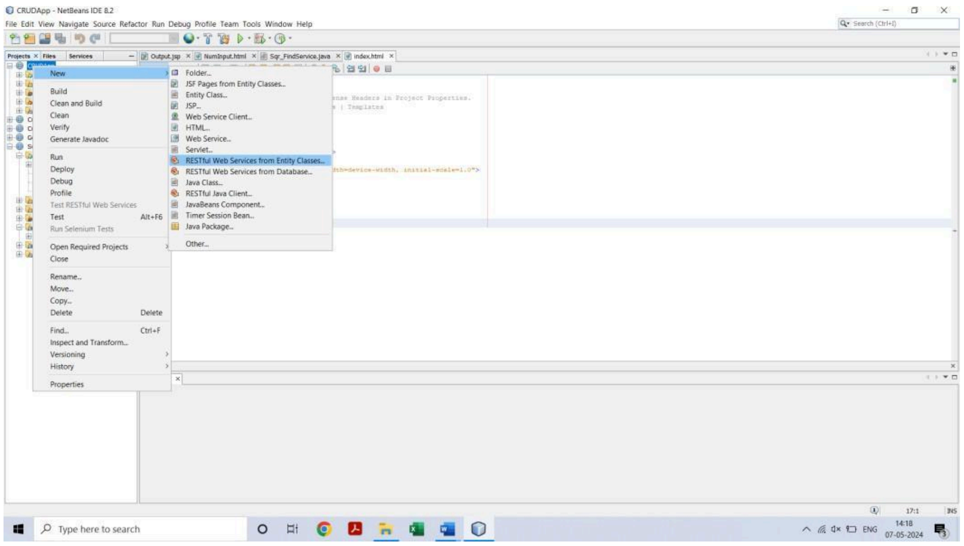
JavaServer Faces Configuration

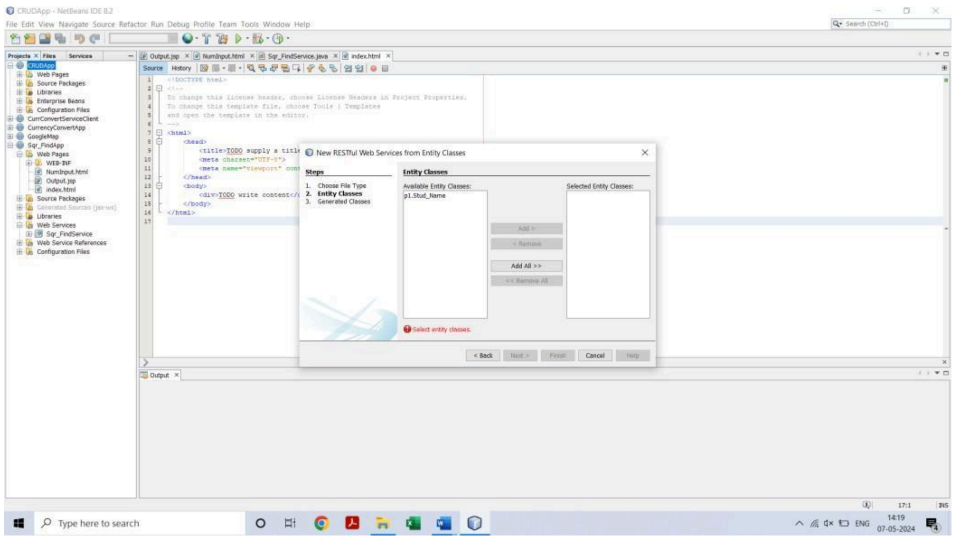
Specify JavaServer Faces configuration and library information.

Libraries | Configuration | Components

[✓ Server Library:] JSF 2.2
Registered Libraries: JSF 2.2
Create New Library
JSF Folder or JAR:
Library Name:

12. **Right click on project name and create RESTful Web Services from Entity Classes.**




New RESTful Web Services from Entity Classes

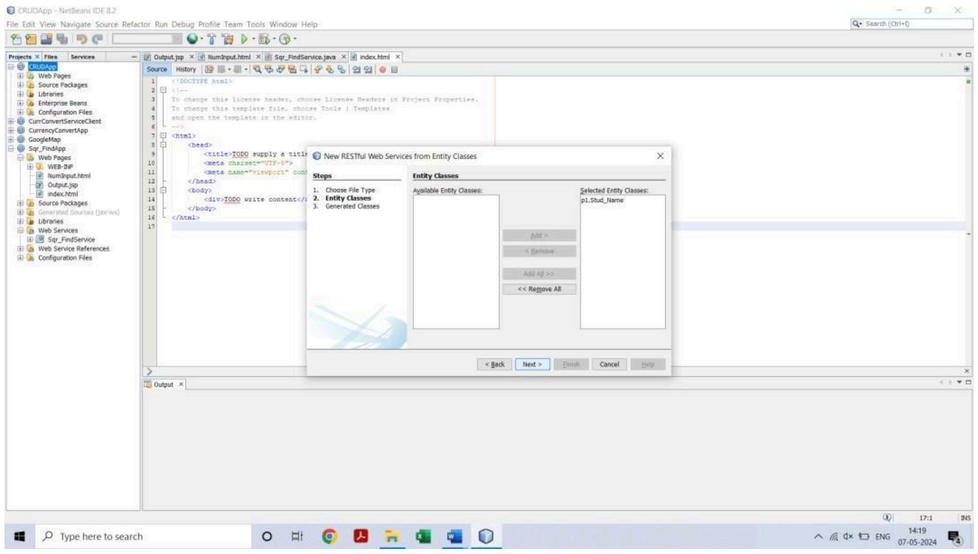
Steps
1. Choose File Type
2. **Entity Classes**
3. Generated Classes

Entity Classes

Available Entity Classes: p1.Stud_Name

Selected Entity Classes:

Select entity classes.


New RESTful Web Services from Entity Classes

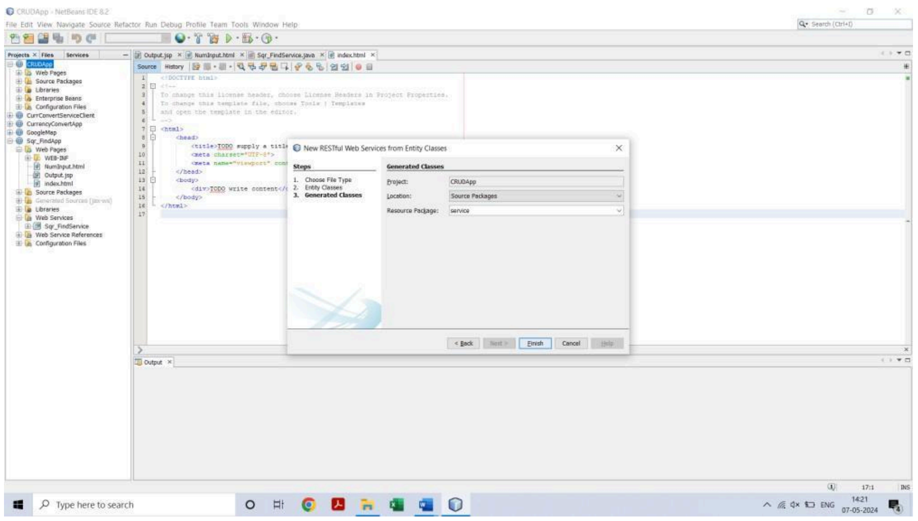
Steps
1. Choose File Type
2. **Entity Classes**
3. Generated Classes

Entity Classes

Available Entity Classes:

Selected Entity Classes: p1.Stud_Name

13. Repeat step 9 and then it will go on next page. Then enter the p1.service in Resource Package and then click on Finish button.


New RESTful Web Services from Entity Classes

Steps
1. Choose File Type
2. Entity Classes
3. **Generated Classes**

Generated Classes

Project: CRUDApp

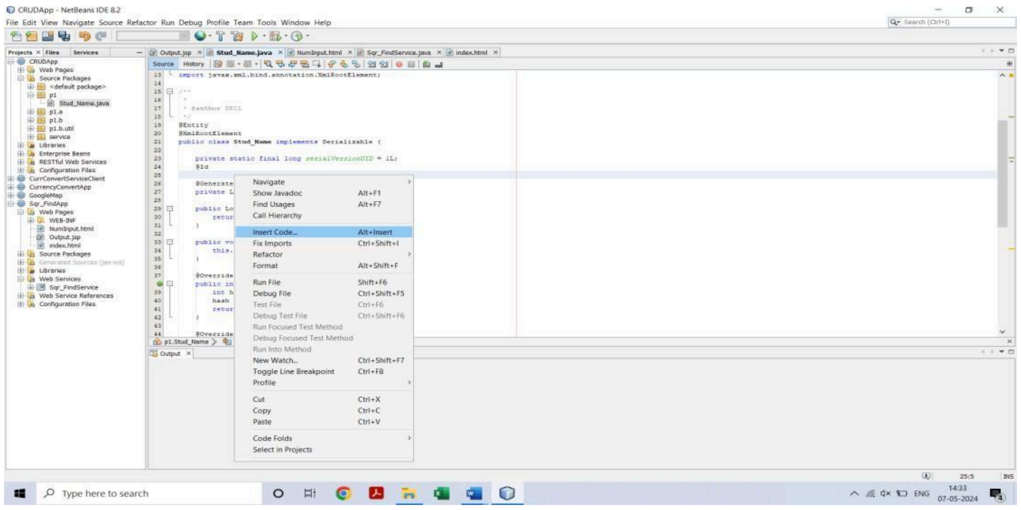
Location: Source Packages

Resource Package: service

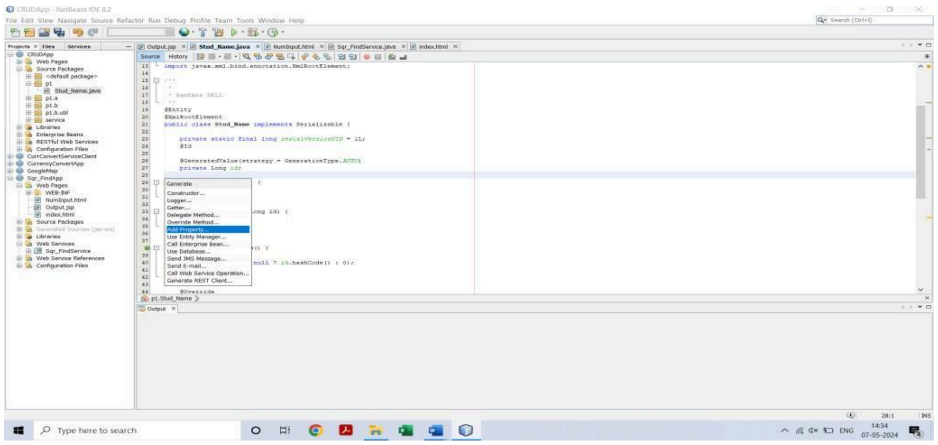
14. Now open Stud_Name.java file under p1 package.

15. In this file at line number 24, do the right click and select Insert Code.

16. A new list will appear. Click on Add Property

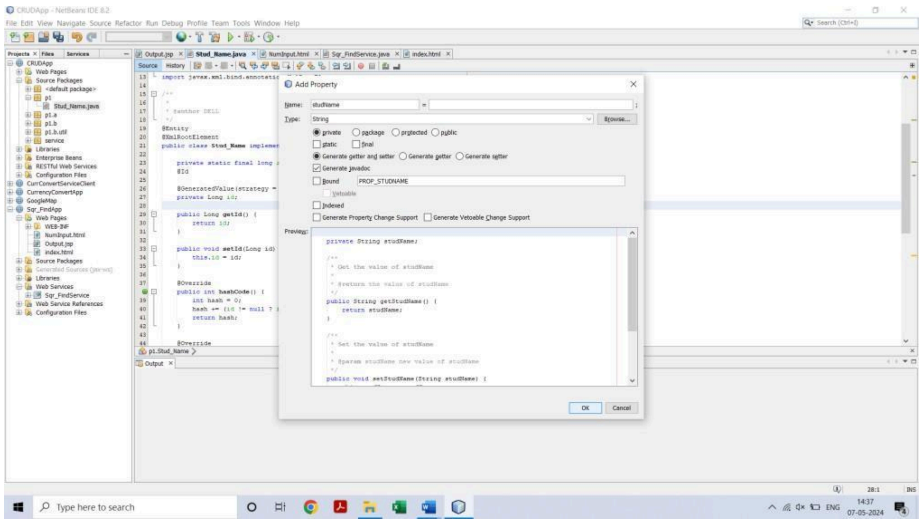


```
13  import javax.xml.bind.annotation.XmlRootElement;
14
15  /**
16   *
17   * @author DELL
18   */
19  @Entity
20  @XmlRootElement
21  public class Stud_Name implements Serializable {
22
23      private static final long serialVersionUID = 1L;
24      @Id
25
26      @Generate
27      private L
```



```
@GeneratedValue(strategy = GenerationType.AUTO)
private Long id;
```

17. A new window will open. Enter name as studName. Make sure name should be exact same as of mine and then click on OK button. Actually we are setting getter and setter method for studName.



Add Property

Name: studName =

Type: String

[✓ private] package protected public

static final

[✓ Generate getter and setter] Generate getter Generate setter

[✓] Generate javadoc

Bound PROP_STUDNAME

Vetoable

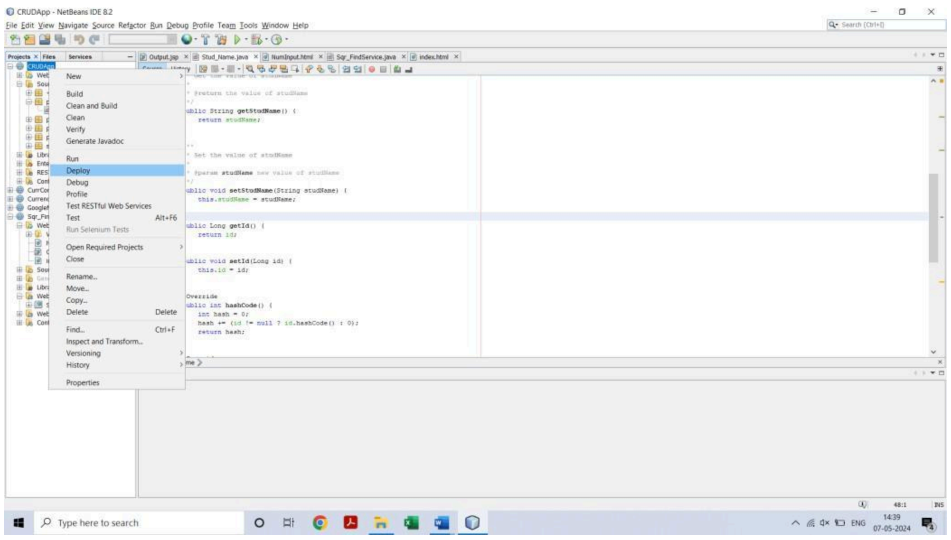
Indexed

Generate Property Change Support Generate Vetoable Change Support

Preview:

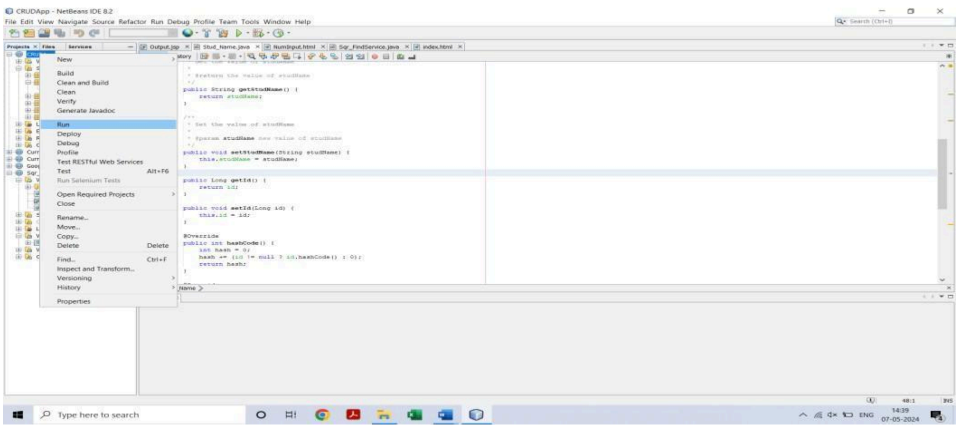
```
private String studName;

/**
 * Get the value of studName
 *
 * @return the value of studName
 */
public String getStudName() {
    return studName;
}

/**
 * Set the value of studName
 *
 * @param studName new value of studName
 */
public void setStudName(String studName) {
```

18. Now right click on web application name and Deploy it.



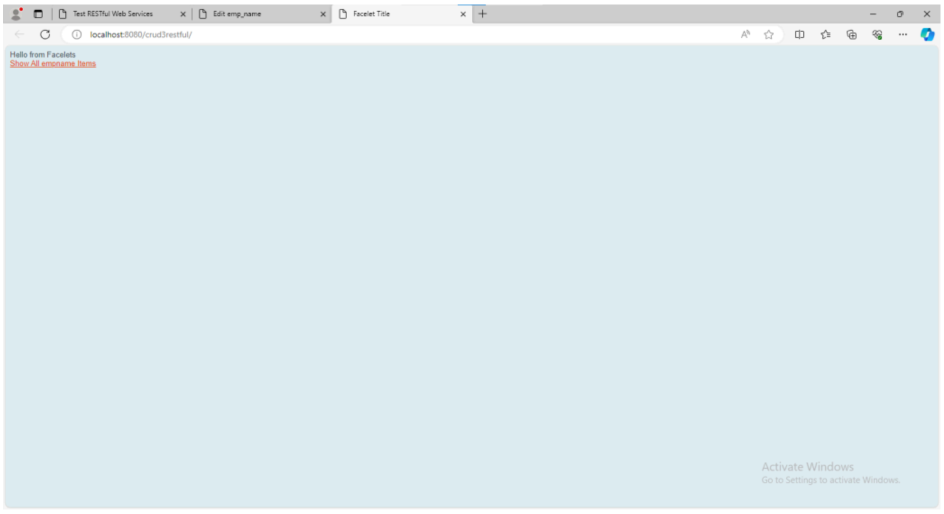
```
 * @return the value of studName
 */
ublic String getStudName() {
   return studName;


**
 * Set the value of studName
 *
 * @param studName new value of studName
 */
ublic void setStudName(String studName) {
   this.studName = studName;


ublic Long getId() {
   return id;


ublic void setId(Long id) {
   this.id = id;


Override
ublic int hashCode() {
   int hash = 0;
   hash += (id != null ? id.hashCode() : 0);
   return hash;
```

19. Now right click on project name and run it

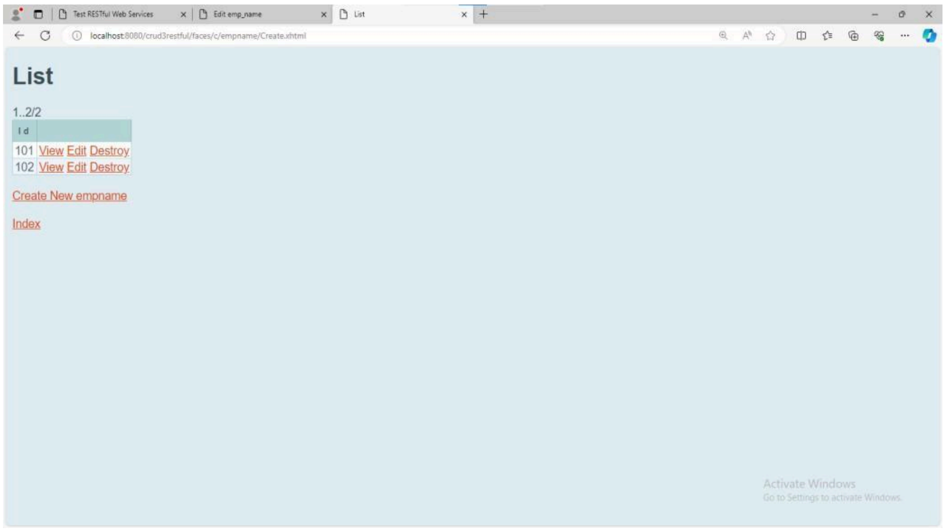


```
 * @return the value of studName
 */
public String getStudName() {
    return studName;
}

/**
 * Set the value of studName
 *
 * @param studName new value of studName
 */
public void setStudName(String studName) {
    this.studName = studName;
}

public Long getId() {
    return id;
}

public void setId(Long id) {
    this.id = id;
}

@Override
public int hashCode() {
    int hash = 0;
    hash += (id != null ? id.hashCode() : 0);
    return hash;
}
```

20. A window will open in browser like below….


Test RESTful Web Services
Edit emp_name
Facelet Title
localhost:8080/crud3restful/
Activate Windows
Go to Settings to activate Windows.

Hello from Facelets
Show All empname Items


Test RESTful Web Services
Edit emp_name
List
localhost:8080/crud3restful/faces/c/empname/Create.xhtml
Activate Windows
Go to Settings to activate Windows.

List

1..2/2

I d	
101	View Edit Destroy
102	View Edit Destroy

Create New empname

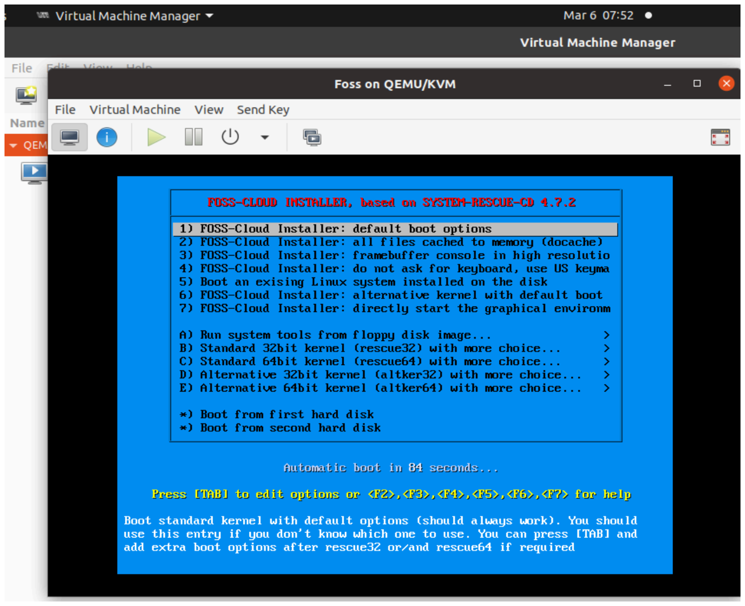
Index

Practical-7

Implement FOSS-Cloud Functionality VSI (Virtual Server Infrastructure) Infrastructure as a Service (IaaS), Creating Virtual Machine or Storage

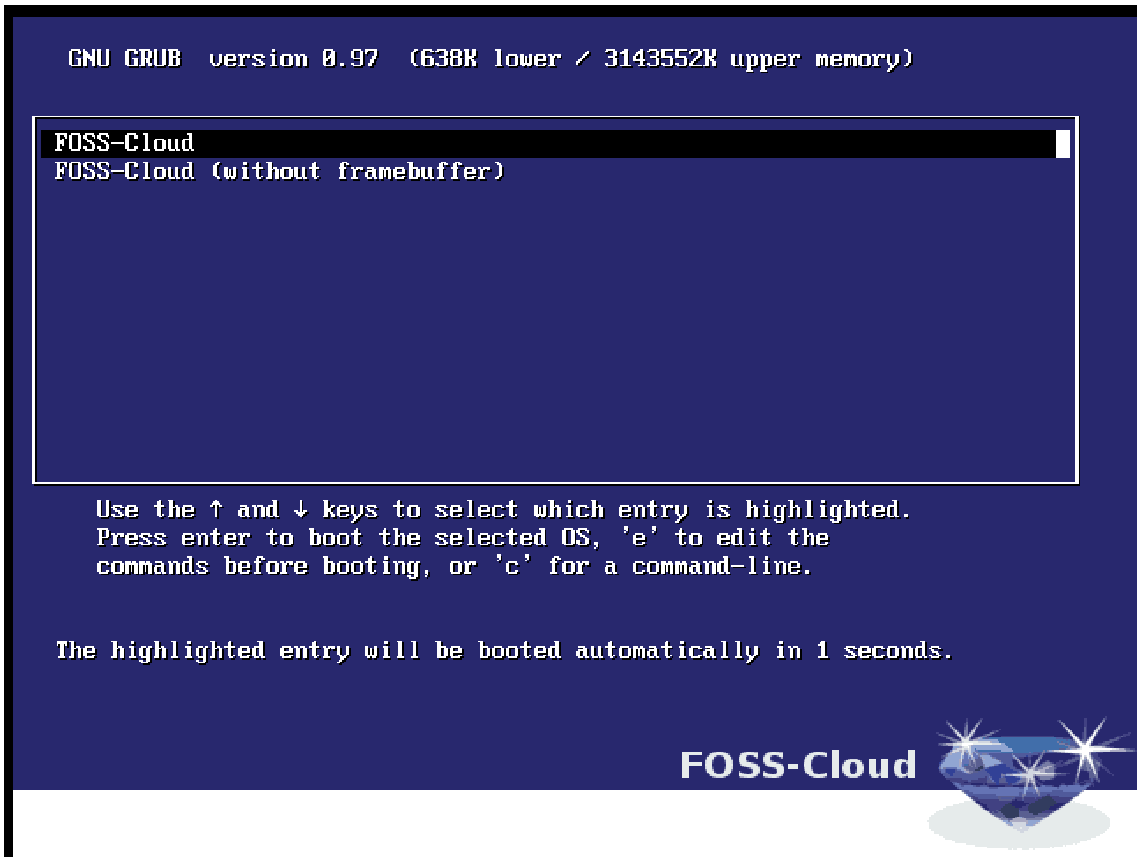
Steps:

1. Start VMWARE workstation player 15 □ select Open virtual machine then□ Browse a foss cloud iso file□ then select Operating System: Other Version: Other, Name: FOSS.
2. Then click on edit virtual machine and select Processor: 2 or 3, Memory: expand to 165 GM.
3. Once the foss Cloud is launched select "Foss-Cloud installer :default boot options" and press Enter



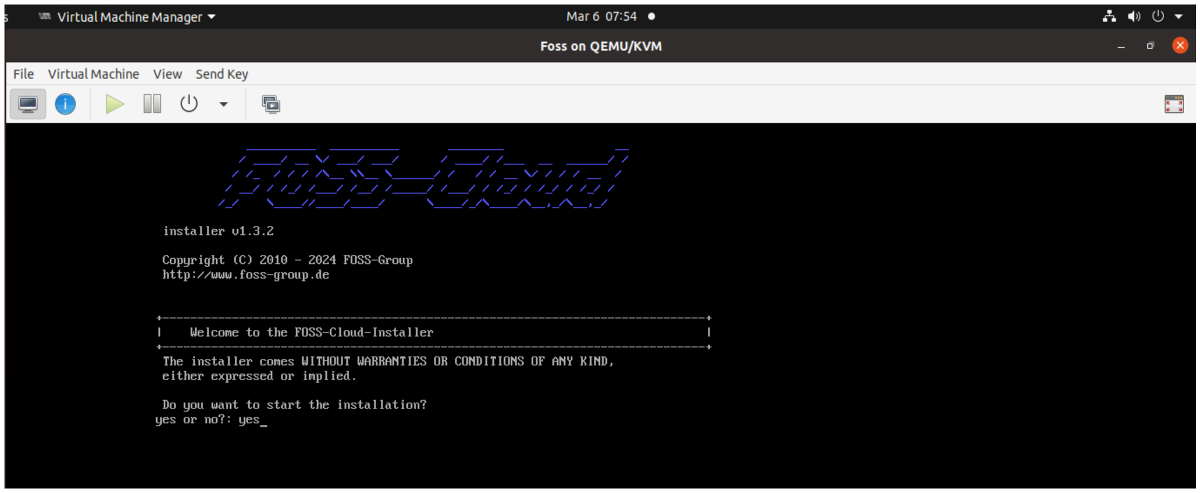
```
FOSS-CLOUD INSTALLER, based on SYSTEM-RESCUE-CD 4.7.2

1) FOSS-Cloud Installer: default boot options
2) FOSS-Cloud Installer: all files cached to memory (docache)
3) FOSS-Cloud Installer: framebuffer console in high resolutio
4) FOSS-Cloud Installer: do not ask for keyboard, use US keyma
5) Boot an exising Linux system installed on the disk
6) FOSS-Cloud Installer: alternative kernel with default boot
7) FOSS-Cloud Installer: directly start the graphical environm

A) Run system tools from floppy disk image...                >
B) Standard 32bit kernel (rescue32) with more choice...      >
C) Standard 64bit kernel (rescue64) with more choice...      >
D) Alternative 32bit kernel (altker32) with more choice...   >
E) Alternative 64bit kernel (altker64) with more choice...   >

*) Boot from first hard disk
*) Boot from second hard disk

Automatic boot in 84 seconds...

Press [TAB] to edit options or <F2>,<F3>,<F4>,<F5>,<F6>,<F7> for help

Boot standard kernel with default options (should always work). You should
use this entry if you don't know which one to use. You can press [TAB] and
add extra boot options after rescue32 or/and rescue64 if required
```

4. Reboot your system and select foss cloud



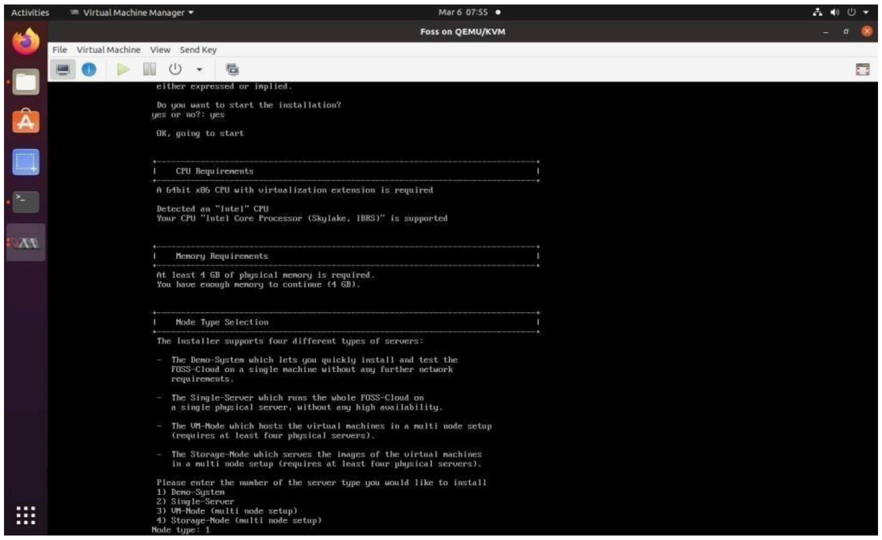
```
  GNU GRUB  version 0.97  (638K lower / 3143552K upper memory)

 FOSS-Cloud
 FOSS-Cloud (without framebuffer)

    Use the ↑ and ↓ keys to select which entry is highlighted.
    Press enter to boot the selected OS, 'e' to edit the
    commands before booting, or 'c' for a command-line.

 The highlighted entry will be booted automatically in 1 seconds.
```

5. Types yes to start the installation



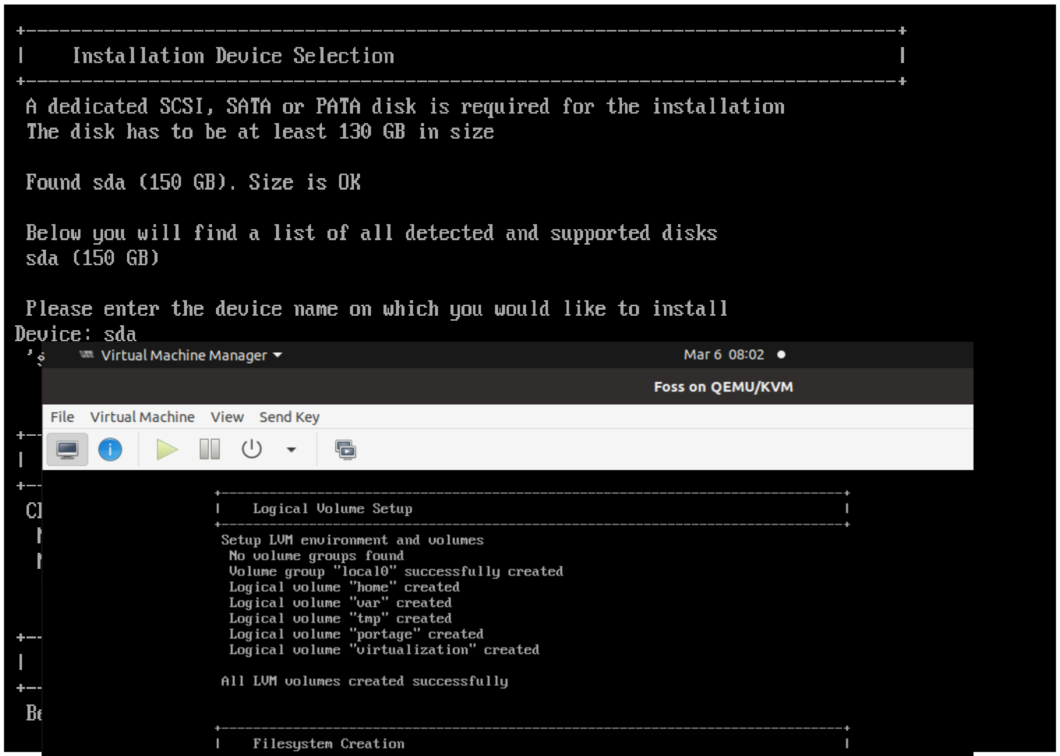
```
 installer v1.3.2

 Copyright (C) 2010 - 2024 FOSS-Group
 http://www.foss-group.de


+----------------------------------------------------------------------------+
|    Welcome to the FOSS-Cloud-Installer                                     |
+----------------------------------------------------------------------------+
 The installer comes WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
 either expressed or implied.

 Do you want to start the installation?
yes or no?: yes_
```

6. Select “Demo System” and press Enter

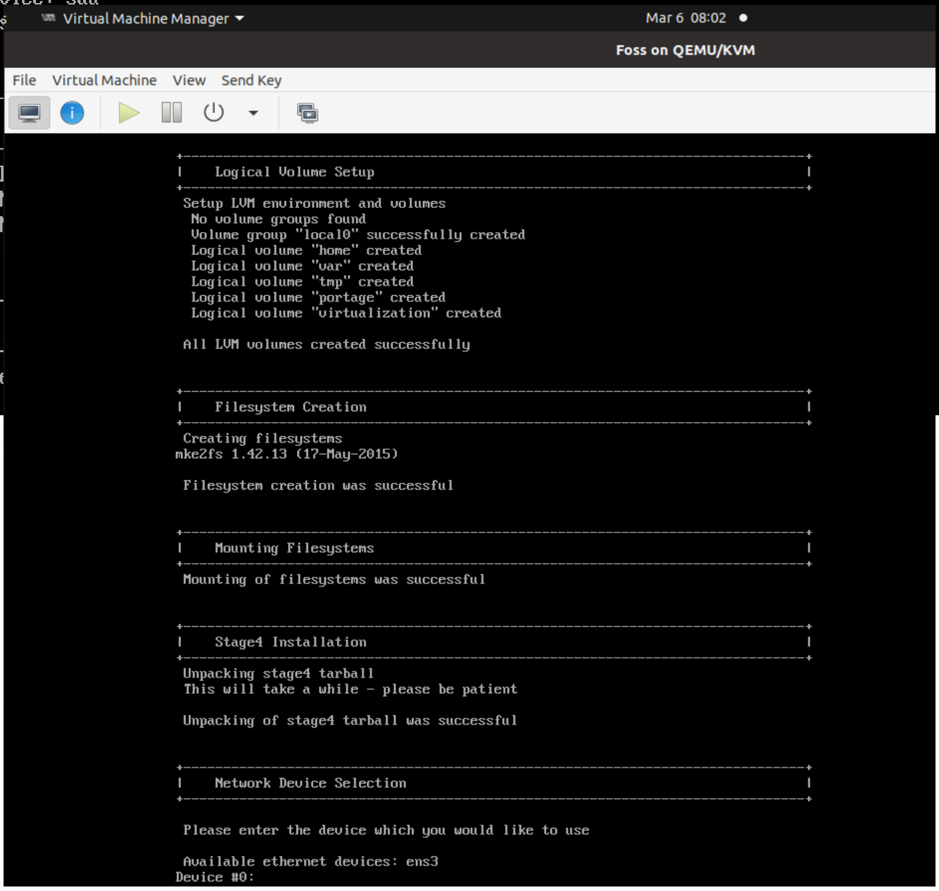


```
 either expressed or implied.

 Do you want to start the installation?
yes or no?: yes

 OK, going to start


+------------------------------------------------------------------------------+
|    CPU Requirements                                                          |
+------------------------------------------------------------------------------+
 A 64bit x86 CPU with virtualization extension is required

 Detected an "Intel" CPU
 Your CPU "Intel Core Processor (Skylake, IBRS)" is supported


+------------------------------------------------------------------------------+
|    Memory Requirements                                                       |
+------------------------------------------------------------------------------+
 At least 4 GB of physical memory is required.
 You have enough memory to continue (4 GB).


+------------------------------------------------------------------------------+
|    Node Type Selection                                                       |
+------------------------------------------------------------------------------+
 The Installer supports four different types of servers:

 -  The Demo-System which lets you quickly install and test the
    FOSS-Cloud on a single machine without any further network
    requirements.

 -  The Single-Server which runs the whole FOSS-Cloud on
    a single physical server, without any high availability.

 -  The VM-Node which hosts the virtual machines in a multi node setup
    (requires at least four physical servers).

 -  The Storage-Node which serves the images of the virtual machines
    in a multi node setup (requires at least four physical servers).

 Please enter the number of the server type you would like to install
 1) Demo-System
 2) Single-Server
 3) VM-Node (multi node setup)
 4) Storage-Node (multi node setup)
Node type: 1
```

7. Enter the Device name as "sda" and press enter

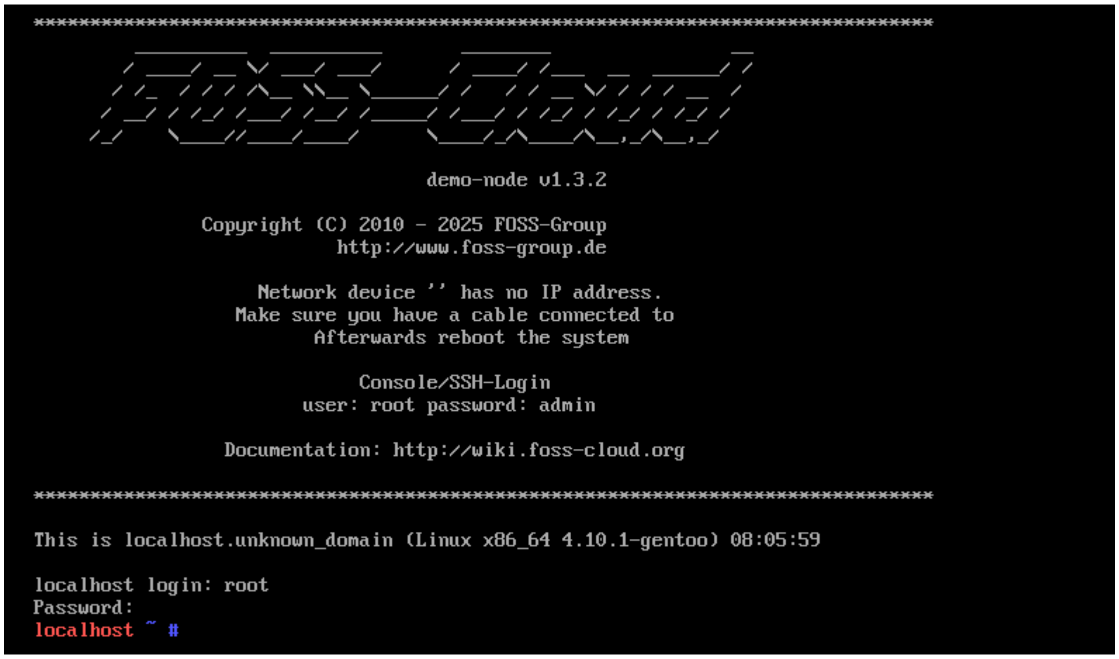


```
+------------------------------------------------------------------------------+
|    Installation Device Selection                                             |
+------------------------------------------------------------------------------+
 A dedicated SCSI, SATA or PATA disk is required for the installation
 The disk has to be at least 130 GB in size

 Found sda (150 GB). Size is OK

 Below you will find a list of all detected and supported disks
 sda (150 GB)

 Please enter the device name on which you would like to install
Device: sda
```



```
+------------------------------------------------------------------------------+
|    Logical Volume Setup                                                      |
+------------------------------------------------------------------------------+
 Setup LVM environment and volumes
  No volume groups found
  Volume group "local0" successfully created
  Logical volume "home" created
  Logical volume "var" created
  Logical volume "tmp" created
  Logical volume "portage" created
  Logical volume "virtualization" created

 All LVM volumes created successfully


+------------------------------------------------------------------------------+
|    Filesystem Creation                                                       |
+------------------------------------------------------------------------------+
 Creating filesystems
mke2fs 1.42.13 (17-May-2015)

 Filesystem creation was successful


+------------------------------------------------------------------------------+
|    Mounting Filesystems                                                      |
+------------------------------------------------------------------------------+
 Mounting of filesystems was successful


+------------------------------------------------------------------------------+
|    Stage4 Installation                                                       |
+------------------------------------------------------------------------------+
 Unpacking stage4 tarball
 This will take a while - please be patient

 Unpacking of stage4 tarball was successful


+------------------------------------------------------------------------------+
|    Network Device Selection                                                  |
+------------------------------------------------------------------------------+

 Please enter the device which you would like to use

 Available ethernet devices: ens3
Device #0:
```

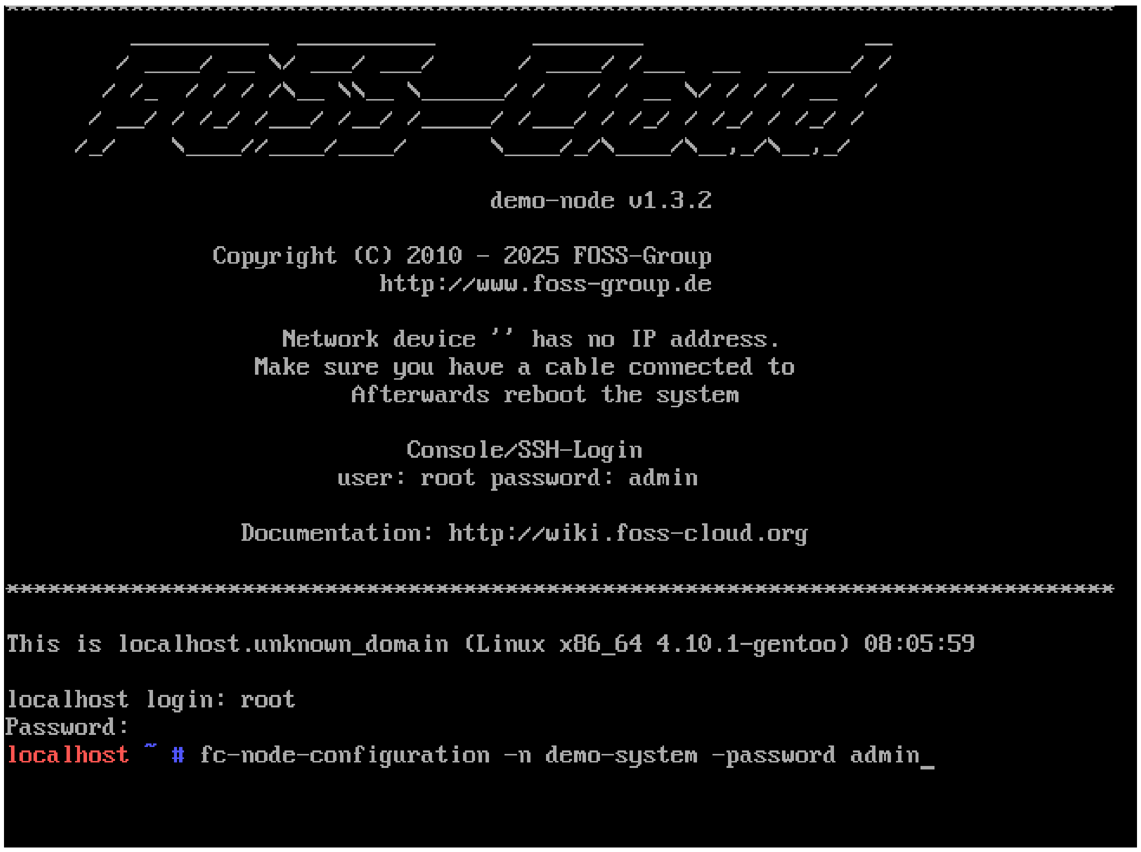
8. Put credential of foss cloud and start foss cloud



```
                          demo-node v1.3.2

              Copyright (C) 2010 - 2025 FOSS-Group
                          http://www.foss-group.de

                  Network device '' has no IP address.
                Make sure you have a cable connected to
                       Afterwards reboot the system

                           Console/SSH-Login
                      user: root password: admin

               Documentation: http://wiki.foss-cloud.org

********************************************************************************

This is localhost.unknown_domain (Linux x86_64 4.10.1-gentoo) 08:05:59

localhost login: root
Password:
localhost ~ #
```

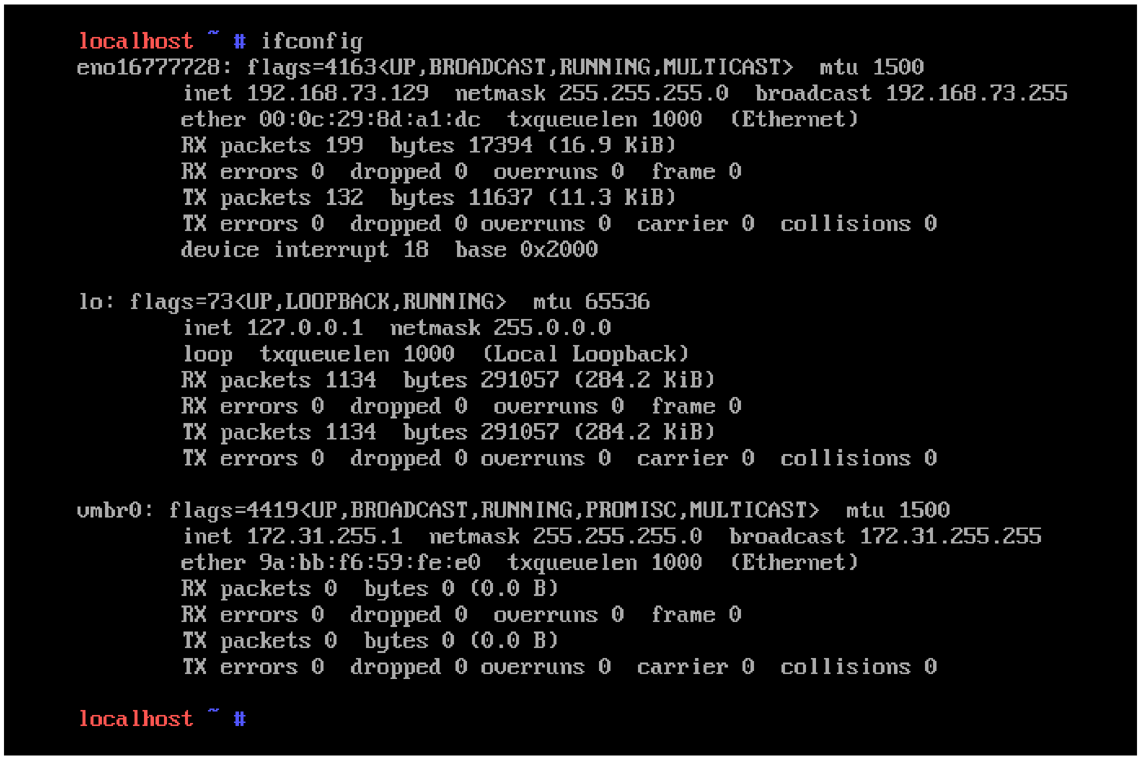
9. Then Execute below mention command

   Fc-node-configuration –n demo-system –password admin



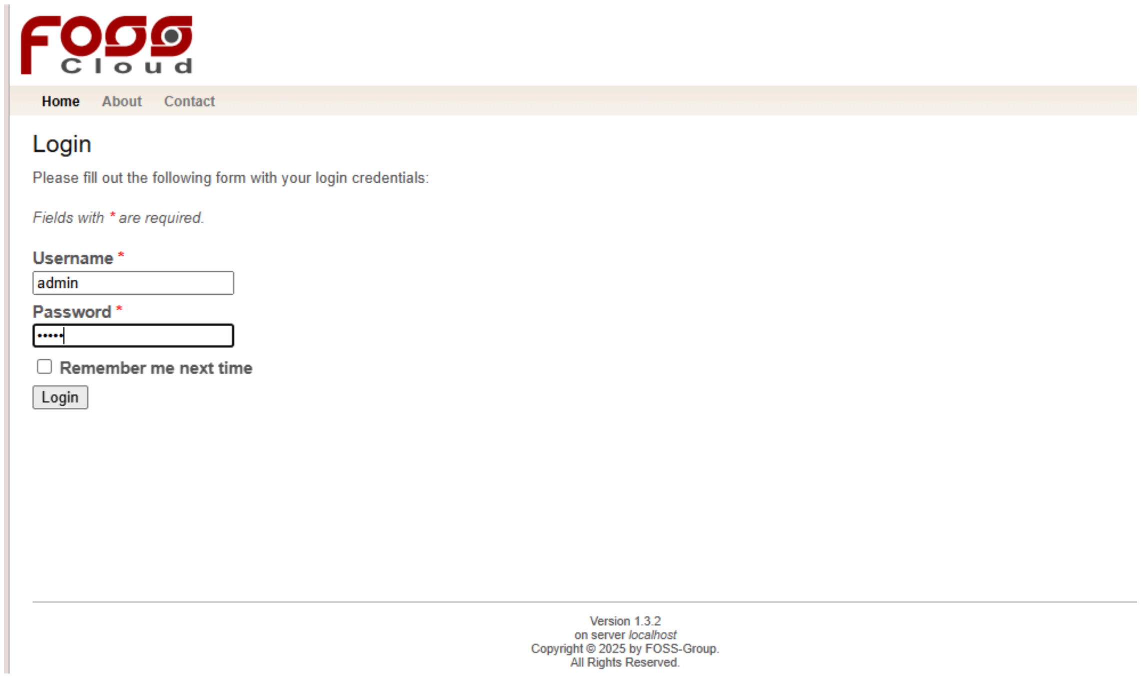
```
                          demo-node v1.3.2

              Copyright (C) 2010 - 2025 FOSS-Group
                          http://www.foss-group.de

                  Network device '' has no IP address.
                Make sure you have a cable connected to
                       Afterwards reboot the system

                           Console/SSH-Login
                      user: root password: admin

               Documentation: http://wiki.foss-cloud.org

********************************************************************************

This is localhost.unknown_domain (Linux x86_64 4.10.1-gentoo) 08:05:59

localhost login: root
Password:
localhost ~ # fc-node-configuration -n demo-system -password admin_
```

10. Then type ifconfig command to get the ip address



```
localhost ~ # ifconfig
eno16777728: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.73.129  netmask 255.255.255.0  broadcast 192.168.73.255
        ether 00:0c:29:8d:a1:dc  txqueuelen 1000  (Ethernet)
        RX packets 199  bytes 17394 (16.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 132  bytes 11637 (11.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 18  base 0x2000

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 1134  bytes 291057 (284.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1134  bytes 291057 (284.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

vmbr0: flags=4419<UP,BROADCAST,RUNNING,PROMISC,MULTICAST>  mtu 1500
        inet 172.31.255.1  netmask 255.255.255.0  broadcast 172.31.255.255
        ether 9a:bb:f6:59:fe:e0  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

localhost ~ #
```

11. Then go to any web browser in windows OS and type the IP Address FOSS cloud web page will open and you will find login details.



Login

Please fill out the following form with your login credentials:

*Fields with * are required.*

Username *
admin

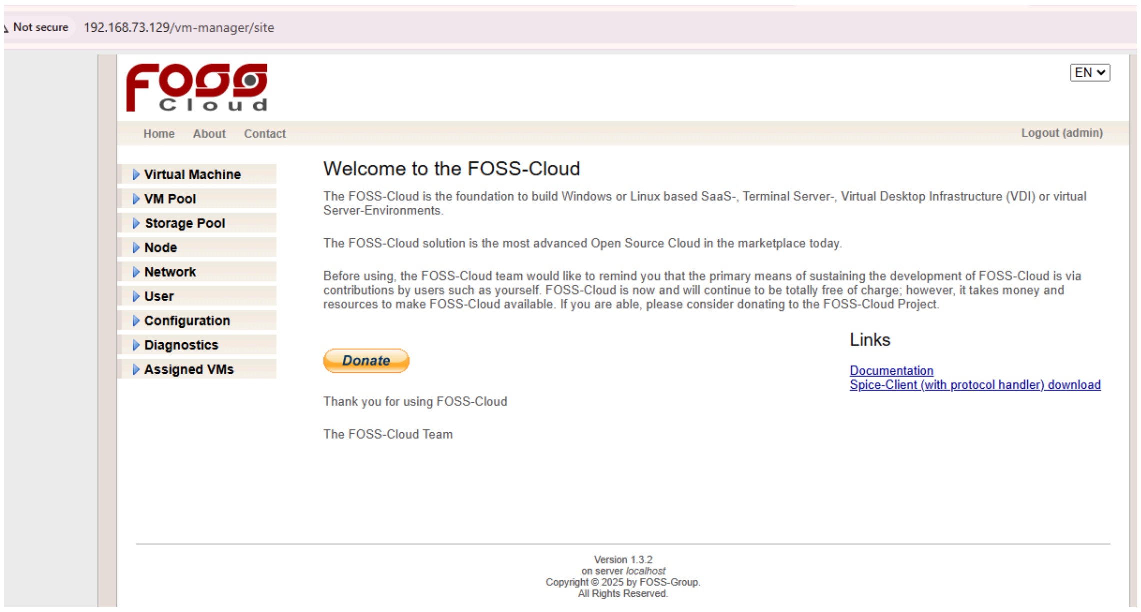
Password *
•••••

Remember me next time

Login

Version 1.3.2
on server *localhost*
Copyright © 2025 by FOSS-Group.
All Rights Reserved.

12. After putting the login details you will get the below mention page



Welcome to the FOSS-Cloud

The FOSS-Cloud is the foundation to build Windows or Linux based SaaS-, Terminal Server-, Virtual Desktop Infrastructure (VDI) or virtual Server-Environments.

The FOSS-Cloud solution is the most advanced Open Source Cloud in the marketplace today.

Before using, the FOSS-Cloud team would like to remind you that the primary means of sustaining the development of FOSS-Cloud is via contributions by users such as yourself. FOSS-Cloud is now and will continue to be totally free of charge; however, it takes money and resources to make FOSS-Cloud available. If you are able, please consider donating to the FOSS-Cloud Project.

Thank you for using FOSS-Cloud

The FOSS-Cloud Team

Links

Documentation
Spice-Client (with protocol handler) download

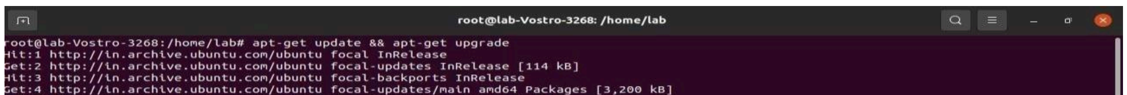
Version 1.3.2
on server *localhost*
Copyright © 2025 by FOSS-Group.
All Rights Reserved.

Practical-9

Using AWS Flow Framework develop application that includes a simple workflow. Workflow calls an activity to print hello world to the console. It must define the basic usage of AWS Flow Framework, including defining contracts, implementation of activities and workflow coordination logic and worker programs to host them

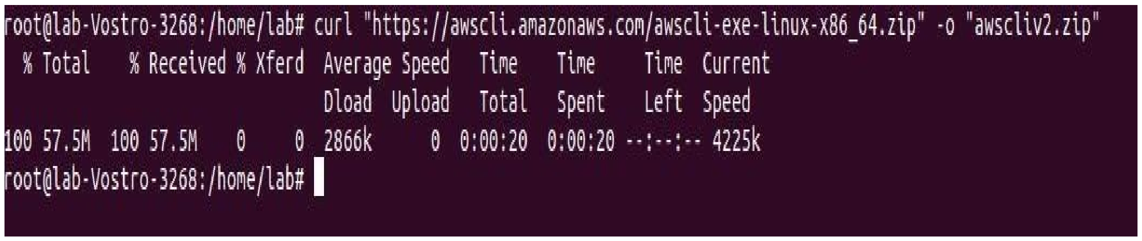
Step 1: Open Terminal and Update and Upgrade your system by command

sudo apt-get update && sudo apt-get upgrade



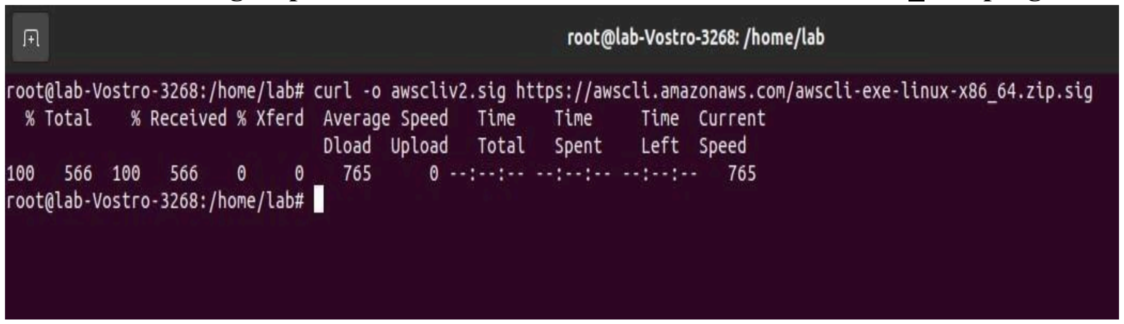
```
root@lab-Vostro-3268:/home/lab# apt-get update && apt-get upgrade
Hit:1 http://in.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Hit:3 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease
Get:4 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [3,200 kB]
```

Step 2: Download awscliv2.zip with command
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"



```
root@lab-Vostro-3268:/home/lab# curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 57.5M  100 57.5M    0     0  2866k      0  0:00:20  0:00:20 --:--:-- 4225k
root@lab-Vostro-3268:/home/lab# 
```

Step 3: Download awscliv2.sig file with command
curl -o awscliv2.sig https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip.sig

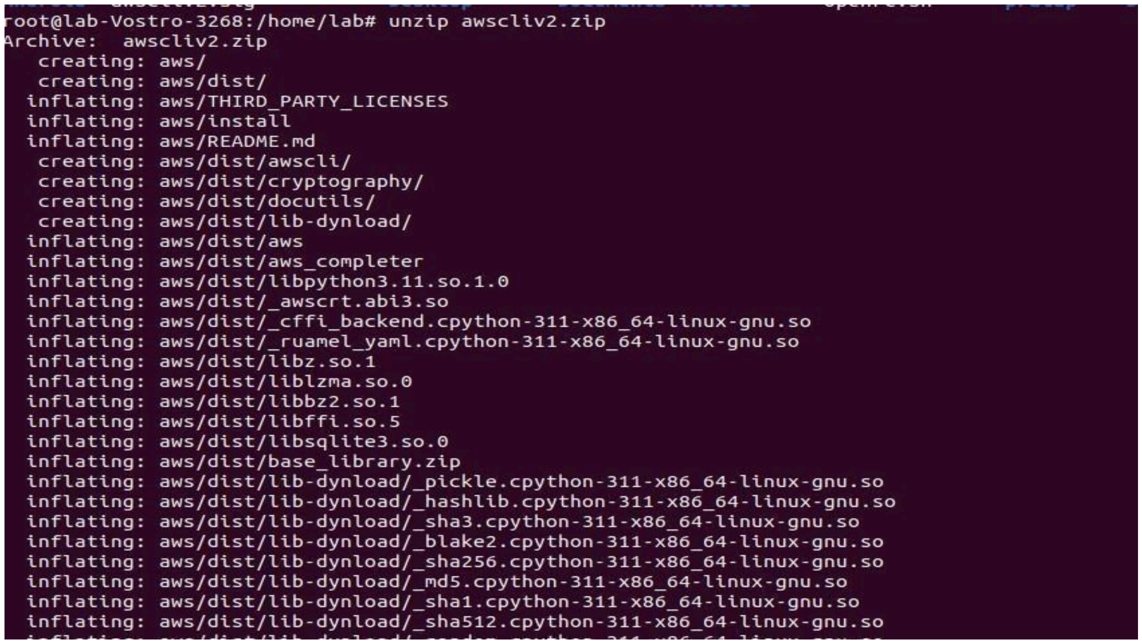


root@lab-Vostro-3268: /home/lab

```
root@lab-Vostro-3268:/home/lab# curl -o awscliv2.sig https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip.sig
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   566  100   566    0     0    765      0 --:--:-- --:--:-- --:--:--   765
root@lab-Vostro-3268:/home/lab# 
```

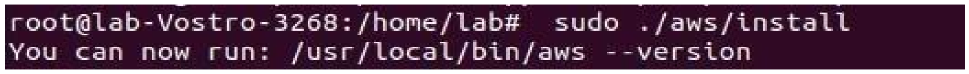
Step 4: unzip awscliv2.zip with command

unzip awscliv2.zip



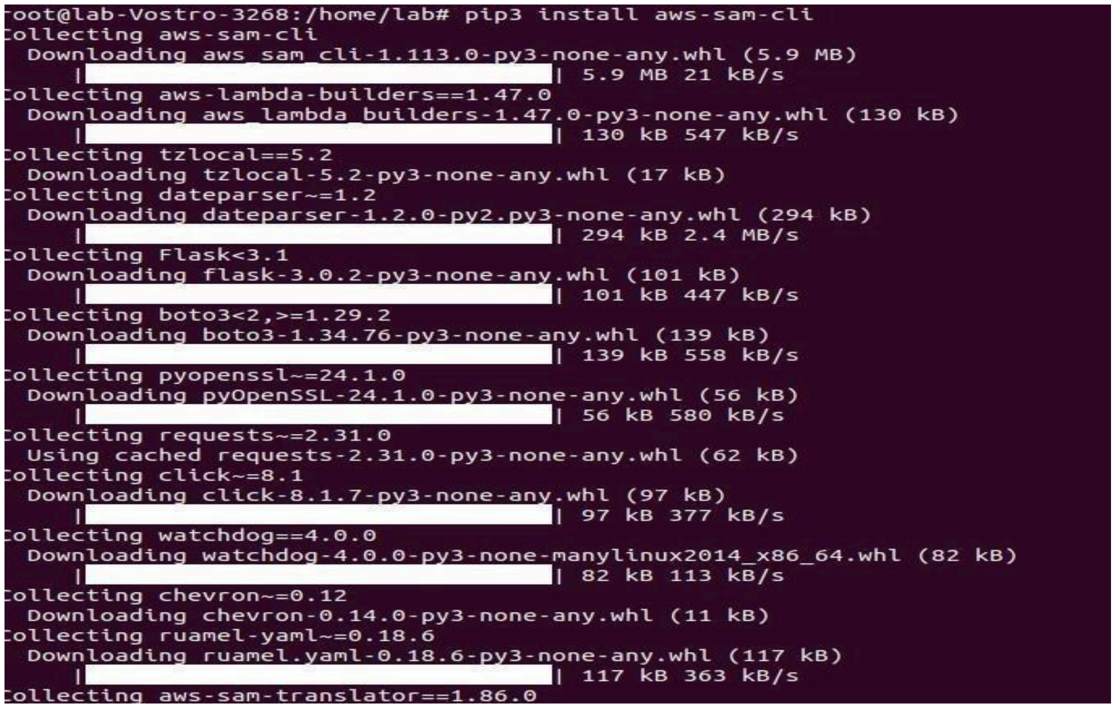
```
root@lab-Vostro-3268:/home/lab# unzip awscliv2.zip
Archive:  awscliv2.zip
   creating: aws/
   creating: aws/dist/
  inflating: aws/THIRD_PARTY_LICENSES
  inflating: aws/install
  inflating: aws/README.md
   creating: aws/dist/awscli/
   creating: aws/dist/cryptography/
   creating: aws/dist/docutils/
   creating: aws/dist/lib-dynload/
  inflating: aws/dist/aws
  inflating: aws/dist/aws_completer
  inflating: aws/dist/libpython3.11.so.1.0
  inflating: aws/dist/_awscrt.abi3.so
  inflating: aws/dist/_cffi_backend.cpython-311-x86_64-linux-gnu.so
  inflating: aws/dist/_ruamel_yaml.cpython-311-x86_64-linux-gnu.so
  inflating: aws/dist/libz.so.1
  inflating: aws/dist/liblzma.so.0
  inflating: aws/dist/libbz2.so.1
  inflating: aws/dist/libffi.so.5
  inflating: aws/dist/libsqlite3.so.0
  inflating: aws/dist/base_library.zip
  inflating: aws/dist/lib-dynload/_pickle.cpython-311-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_hashlib.cpython-311-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_sha3.cpython-311-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_blake2.cpython-311-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_sha256.cpython-311-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_md5.cpython-311-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_sha1.cpython-311-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_sha512.cpython-311-x86_64-linux-gnu.so
```

Step 5: Run command
sudo ./aws/install



```
root@lab-Vostro-3268:/home/lab#  sudo ./aws/install
You can now run: /usr/local/bin/aws --version
```

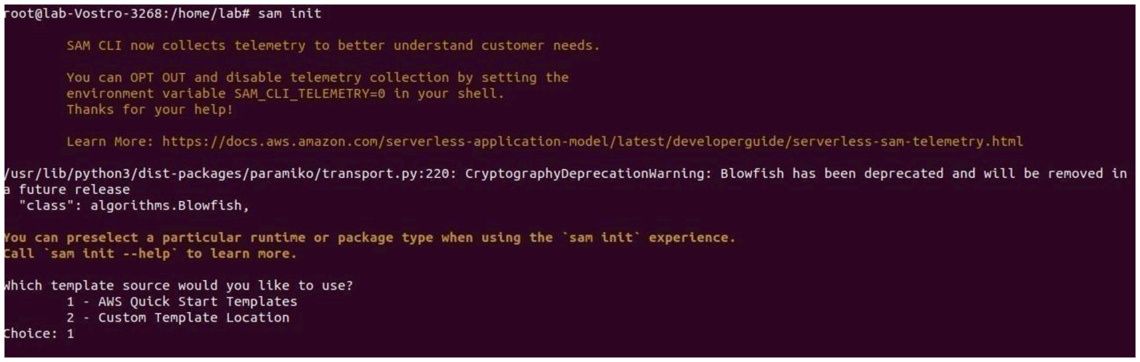
Step 6: Type command
pip3 install aws-sam-cli



```
oot@lab-Vostro-3268:/home/lab# pip3 install aws-sam-cli
ollecting aws-sam-cli
  Downloading aws_sam_cli-1.113.0-py3-none-any.whl (5.9 MB)
     |████████████████████████████████| 5.9 MB 21 kB/s
ollecting aws-lambda-builders==1.47.0
  Downloading aws_lambda_builders-1.47.0-py3-none-any.whl (130 kB)
     |████████████████████████████████| 130 kB 547 kB/s
ollecting tzlocal==5.2
  Downloading tzlocal-5.2-py3-none-any.whl (17 kB)
ollecting dateparser~=1.2
  Downloading dateparser-1.2.0-py2.py3-none-any.whl (294 kB)
     |████████████████████████████████| 294 kB 2.4 MB/s
ollecting Flask<3.1
  Downloading flask-3.0.2-py3-none-any.whl (101 kB)
     |████████████████████████████████| 101 kB 447 kB/s
ollecting boto3<2,>=1.29.2
  Downloading boto3-1.34.76-py3-none-any.whl (139 kB)
     |████████████████████████████████| 139 kB 558 kB/s
ollecting pyopenssl~=24.1.0
  Downloading pyOpenSSL-24.1.0-py3-none-any.whl (56 kB)
     |████████████████████████████████| 56 kB 580 kB/s
ollecting requests~=2.31.0
  Using cached requests-2.31.0-py3-none-any.whl (62 kB)
ollecting click~=8.1
  Downloading click-8.1.7-py3-none-any.whl (97 kB)
     |████████████████████████████████| 97 kB 377 kB/s
ollecting watchdog==4.0.0
  Downloading watchdog-4.0.0-py3-none-manylinux2014_x86_64.whl (82 kB)
     |████████████████████████████████| 82 kB 113 kB/s
ollecting chevron~=0.12
  Downloading chevron-0.14.0-py3-none-any.whl (11 kB)
ollecting ruamel-yaml~=0.18.6
  Downloading ruamel.yaml-0.18.6-py3-none-any.whl (117 kB)
     |████████████████████████████████| 117 kB 363 kB/s
ollecting aws-sam-translator==1.86.0
```

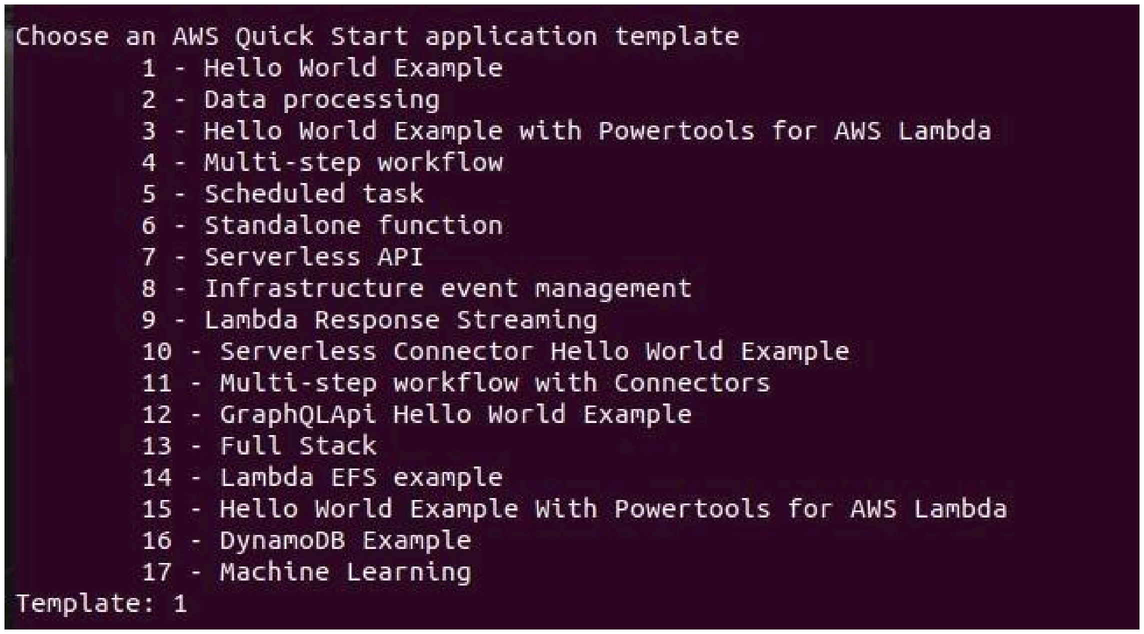
Step 7: Type command sam init in terminal to launch Sam

CLISelect 1st option to use AWS Quick Ttart Templates



```
root@lab-Vostro-3268:/home/lab# sam init

        SAM CLI now collects telemetry to better understand customer needs.

        You can OPT OUT and disable telemetry collection by setting the
        environment variable SAM_CLI_TELEMETRY=0 in your shell.
        Thanks for your help!

        Learn More: https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/serverless-sam-telemetry.html

/usr/lib/python3/dist-packages/paramiko/transport.py:220: CryptographyDeprecationWarning: Blowfish has been deprecated and will be removed in
a future release
  "class": algorithms.Blowfish,

You can preselect a particular runtime or package type when using the `sam init` experience.
Call `sam init --help` to learn more.

Which template source would you like to use?
        1 - AWS Quick Start Templates
        2 - Custom Template Location
Choice: 1
```

Step 8: Select Template no.1 Hello World Example

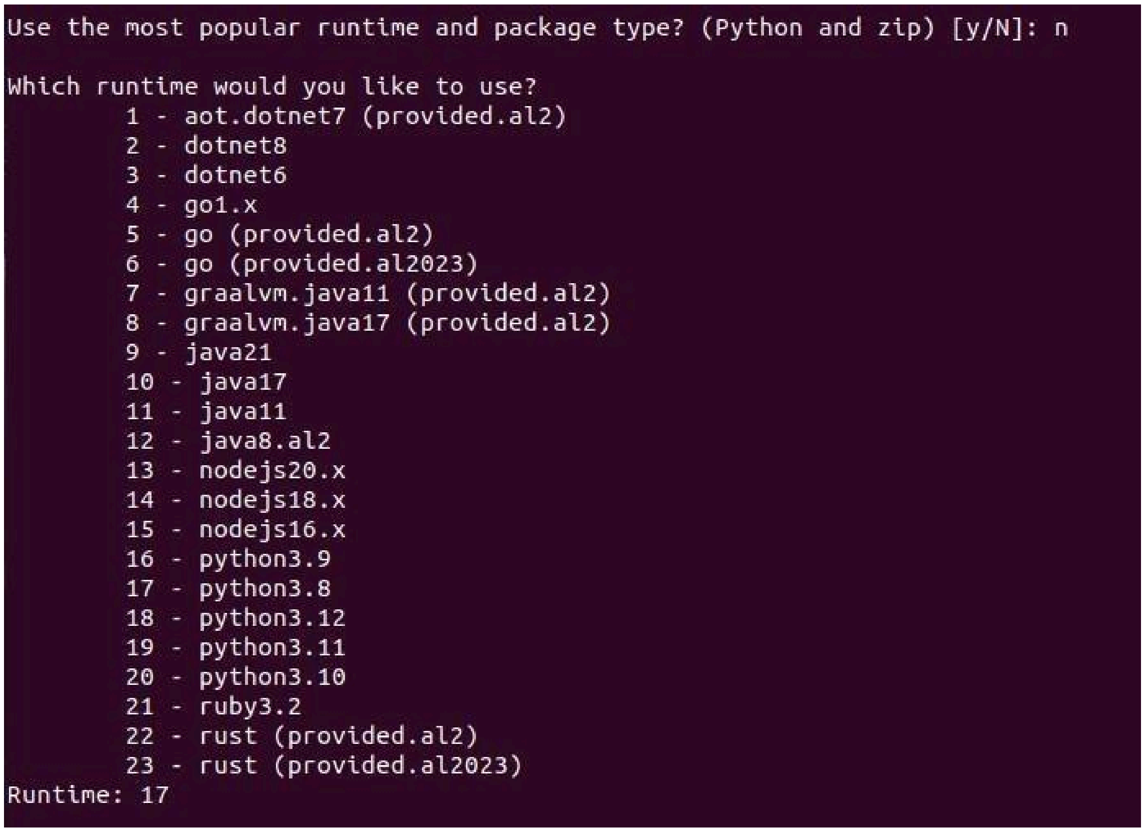


```
Choose an AWS Quick Start application template
        1 - Hello World Example
        2 - Data processing
        3 - Hello World Example with Powertools for AWS Lambda
        4 - Multi-step workflow
        5 - Scheduled task
        6 - Standalone function
        7 - Serverless API
        8 - Infrastructure event management
        9 - Lambda Response Streaming
        10 - Serverless Connector Hello World Example
        11 - Multi-step workflow with Connectors
        12 - GraphQLApi Hello World Example
        13 - Full Stack
        14 - Lambda EFS example
        15 - Hello World Example With Powertools for AWS Lambda
        16 - DynamoDB Example
        17 - Machine Learning
Template: 1
```

Step 9: Type "N" if it ask to use most popular runtime and package type

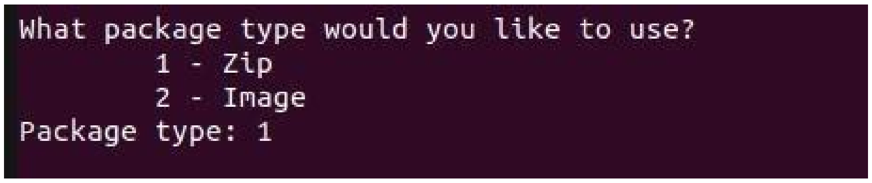
Open new terminal by pressing ctrl+shift+T and check for python version by command python –version

Select the option according to your python version in my case its option 19- python 3.11



```
Use the most popular runtime and package type? (Python and zip) [y/N]: n

Which runtime would you like to use?
        1 - aot.dotnet7 (provided.al2)
        2 - dotnet8
        3 - dotnet6
        4 - go1.x
        5 - go (provided.al2)
        6 - go (provided.al2023)
        7 - graalvm.java11 (provided.al2)
        8 - graalvm.java17 (provided.al2)
        9 - java21
        10 - java17
        11 - java11
        12 - java8.al2
        13 - nodejs20.x
        14 - nodejs18.x
        15 - nodejs16.x
        16 - python3.9
        17 - python3.8
        18 - python3.12
        19 - python3.11
        20 - python3.10
        21 - ruby3.2
        22 - rust (provided.al2)
        23 - rust (provided.al2023)
Runtime: 17
```

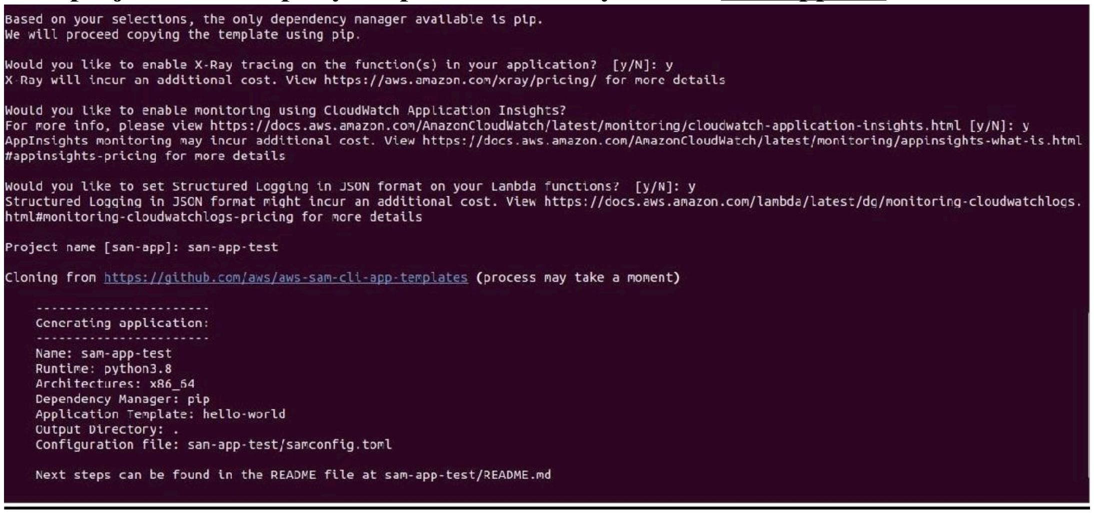
Step 10: Select package type as <u>Zip</u>



```
What package type would you like to use?
        1 - Zip
        2 - Image
Package type: 1
```

Step 11: Now choose Yes option everytime it ask .

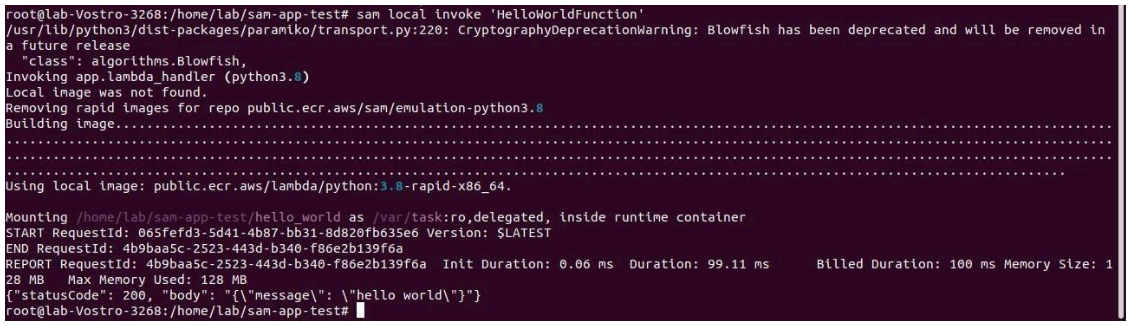
Give project name as per your preference in my case its <u>sam-app-test</u>



```
Based on your selections, the only dependency manager available is pip.
We will proceed copying the template using pip.

Would you like to enable X-Ray tracing on the function(s) in your application?  [y/N]: y
X-Ray will incur an additional cost. View https://aws.amazon.com/xray/pricing/ for more details

Would you like to enable monitoring using CloudWatch Application Insights?
For more info, please view https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/cloudwatch-application-insights.html [y/N]: y
AppInsights monitoring may incur additional cost. View https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/appinsights-what-is.html
#appinsights-pricing for more details

Would you like to set Structured Logging in JSON format on your Lambda functions?  [y/N]: y
Structured Logging in JSON format might incur an additional cost. View https://docs.aws.amazon.com/lambda/latest/dg/monitoring-cloudwatchlogs.
html#monitoring-cloudwatchlogs-pricing for more details

Project name [sam-app]: sam-app-test

Cloning from https://github.com/aws/aws-sam-cli-app-templates (process may take a moment)

    -----------------------
    Generating application:
    -----------------------
    Name: sam-app-test
    Runtime: python3.8
    Architectures: x86_64
    Dependency Manager: pip
    Application Template: hello-world
    Output Directory: .
    Configuration file: sam-app-test/samconfig.toml

    Next steps can be found in the README file at sam-app-test/README.md
```

Step 12: Now one folder will be created by your provided project name go into that folder by command cd (folder name)

After entering the project folder we will invoke the HelloWorldFunction by using commandsam local invoke „HelloWorldFunction"

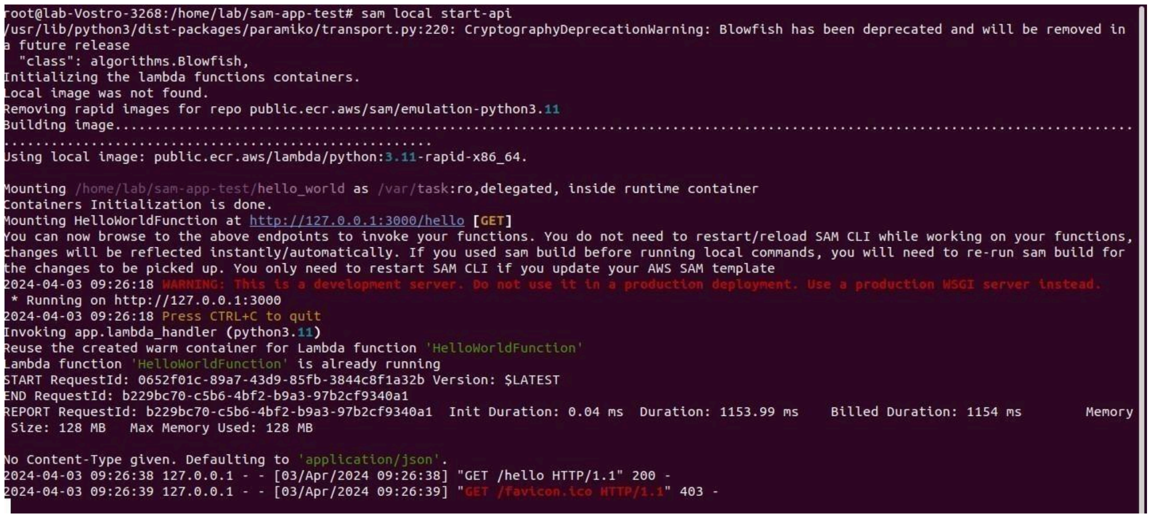


```
root@lab-Vostro-3268:/home/lab/sam-app-test# sam local invoke 'HelloWorldFunction'
/usr/lib/python3/dist-packages/paramiko/transport.py:220: CryptographyDeprecationWarning: Blowfish has been deprecated and will be removed in
a future release
  "class": algorithms.Blowfish,
Invoking app.lambda_handler (python3.8)
Local image was not found.
Removing rapid images for repo public.ecr.aws/sam/emulation-python3.8
Building image..........................................................................................................
........................................................................................................................
........................................................................................................................
...................................................................................................................
Using local image: public.ecr.aws/lambda/python:3.8-rapid-x86_64.

Mounting /home/lab/sam-app-test/hello_world as /var/task:ro,delegated, inside runtime container
START RequestId: 065fefd3-5d41-4b87-bb31-8d820fb635e6 Version: $LATEST
END RequestId: 4b9baa5c-2523-443d-b340-f86e2b139f6a
REPORT RequestId: 4b9baa5c-2523-443d-b340-f86e2b139f6a  Init Duration: 0.06 ms  Duration: 99.11 ms      Billed Duration: 100 ms Memory Size: 1
28 MB   Max Memory Used: 128 MB
{"statusCode": 200, "body": "{\"message\": \"hello world\"}"}
root@lab-Vostro-3268:/home/lab/sam-app-test#
```

It should give you StatusCode:200

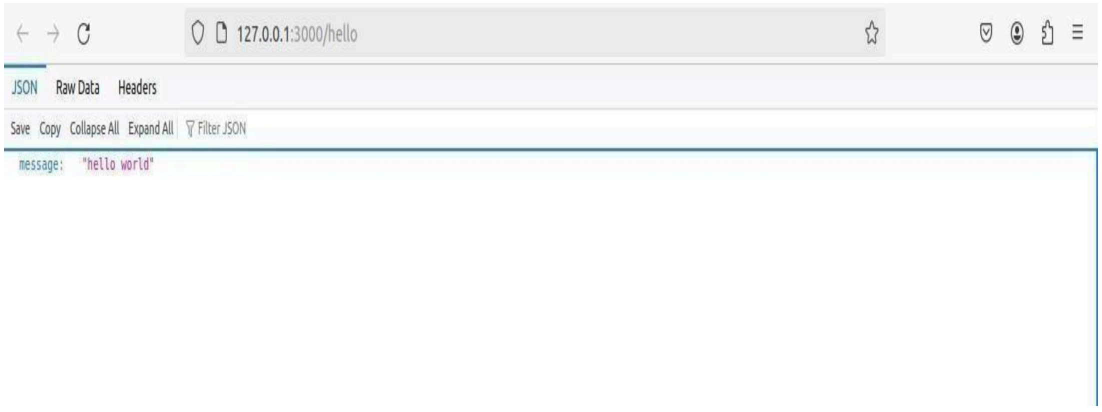
Use command sudo snap install docker if docker error occurs.

Step 13: Type command sam local start-api this will give you URL open it in any browser.



```
root@lab-Vostro-3268:/home/lab/sam-app-test# sam local start-api
/usr/lib/python3/dist-packages/paramiko/transport.py:220: CryptographyDeprecationWarning: Blowfish has been deprecated and will be removed in
a future release
  "class": algorithms.Blowfish,
Initializing the lambda functions containers.
Local image was not found.
Removing rapid images for repo public.ecr.aws/sam/emulation-python3.11
Building image..........................................................................................................
.....................................................
Using local image: public.ecr.aws/lambda/python:3.11-rapid-x86_64.

Mounting /home/lab/sam-app-test/hello_world as /var/task:ro,delegated, inside runtime container
Containers Initialization is done.
Mounting HelloWorldFunction at http://127.0.0.1:3000/hello [GET]
You can now browse to the above endpoints to invoke your functions. You do not need to restart/reload SAM CLI while working on your functions,
changes will be reflected instantly/automatically. If you used sam build before running local commands, you will need to re-run sam build for
the changes to be picked up. You only need to restart SAM CLI if you update your AWS SAM template
2024-04-03 09:26:18 WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:3000
2024-04-03 09:26:18 Press CTRL+C to quit
Invoking app.lambda_handler (python3.11)
Reuse the created warm container for Lambda function 'HelloWorldFunction'
Lambda function 'HelloWorldFunction' is already running
START RequestId: 0652f01c-89a7-43d9-85fb-3844c8f1a32b Version: $LATEST
END RequestId: b229bc70-c5b6-4bf2-b9a3-97b2cf9340a1
REPORT RequestId: b229bc70-c5b6-4bf2-b9a3-97b2cf9340a1  Init Duration: 0.04 ms  Duration: 1153.99 ms    Billed Duration: 1154 ms        Memory
 Size: 128 MB   Max Memory Used: 128 MB

No Content-Type given. Defaulting to 'application/json'.
2024-04-03 09:26:38 127.0.0.1 - - [03/Apr/2024 09:26:38] "GET /hello HTTP/1.1" 200 -
2024-04-03 09:26:39 127.0.0.1 - - [03/Apr/2024 09:26:39] "GET /favicon.ico HTTP/1.1" 403 -
```

Output:



```
message:  "hello world"
```