**Practical 1**

**Aim :-Create a Simple SOAP service.**

**Steps:**

1] Open the NetBeans , and you will get the following screen. Close the start page.

A screenshot of a computer

Description automatically generated

2] Now click on the file tab and click on new project you will get the following screen :

A screenshot of a computer

Description automatically generated

3] In Categories select Java Web and in Projects , Select Web Application .After selecting click onnext . You will get the following window:

A screenshot of a computer

Description automatically generated

4] Now Give name to the Project Name: , and click on next . you will get the following windowthen . again click o

A screenshot of a computer

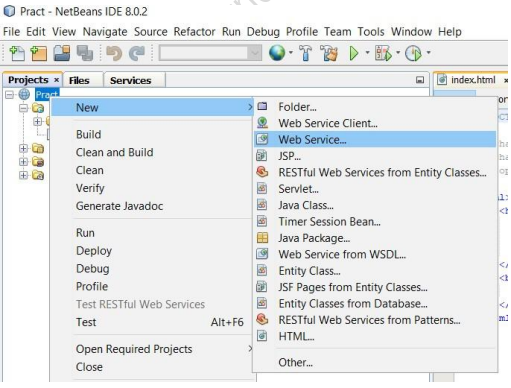
Description automatically generated

5] You will get the following screen now carefully see in projects section your recently created project appears double click to expand

A screenshot of a computer

Description automatically generated

6] Now Right click on the project and select new and then select Web Service ,As sho



7] after clicking Web Service following window should appear , now give name to web serviceand give package as “server”. As shown in the image:

A screenshot of a computer

Description automatically generated

After clicking finish you should get the following window erase the mentioned code :

A screenshot of a computer program

Description automatically generated

8] Now right click any where and click on insert code and select Add Web Service Operation :

A screenshot of a computer

Description automatically generated

9] the following window would appear :

Just give name to the method or operation and click on add button to add parameters to the method as here we are converting dollar to rupees we should need only one param

10] give the name to the variable select data type as double and click on ok as shown below:

A screenshot of a computer

Description automatically generated

public class currency {

@WebMethod(operationName = "IntroDoller")

public String IntroDoller(@WebParam(name = "a") double a) {

//TODO write your implementation code here:

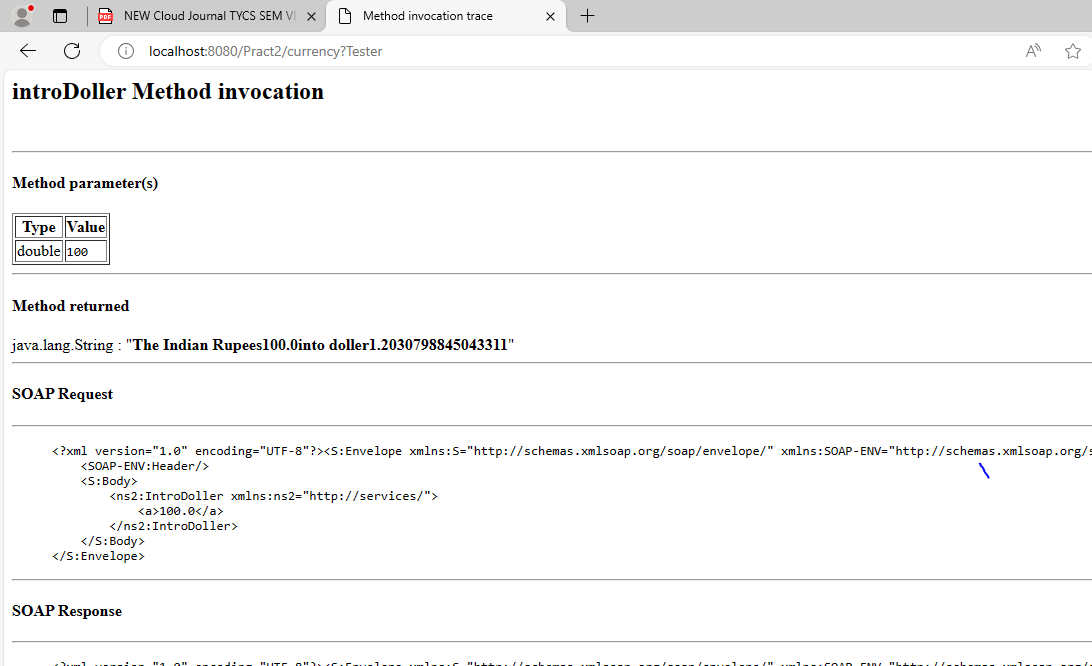
return "The Indian Rupees"+a+"into doller"+(a/83.12);

}

}

13] now right click on webservice and click on test web service you will get the following output:

So this is how we created our web service and deployed it.



14] now our web service is successfully deployed. Right click on web pages and select new and select jsp as shown below

A screenshot of a computer

Description automatically generated

15] give name and click on finish as shown below do it 2 times on for input and one foroutput:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

16] In input.jsp create a form for taking user input as shown below:

Input.jsp :-

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

</head>

<body>

<form action="output.jsp">

<pre>

Enter the currency in rupees:<input type="text" name="t1">

<input type="submit"> <input type="reset">

</pre>

</form>

</body>

</html>

In this code set action = the jsp file where u want output and give name to textbox input. 17] now we have to create web service client, for same right click on project and select new then select web service client you will get the following screen

A screenshot of a computer

Description automatically generated

At this stage copy the url and paste it in notepad for future. Give package name as “Client”.

Now click on finish.

18] now in project section you can se a new folder is been created named “web servicerefrence”. double click to expand it you should get the following :

A screenshot of a computer

Description automatically generated

Hold the operation or your operation name and drag it into the output.jsp in body tag as shown : You will get the following auto generated c

Output.jsp :-

<body>

<h1>Hello World!</h1>

<%-- start web service invocation --%><hr/>

<%

try {

client.Currency\_Service service = new client.Currency\_Service();

client.Currency port = service.getCurrencyPort();

// TODO initialize WS operation arguments here

double a = Double.parseDouble(request.getParameter("t1"));

// TODO process result here

java.lang.String result = port.introDoller(a);

out.println("Result = "+result);

} catch (Exception ex) {

// TODO handle custom exceptions here

}

%>

<%-- end web service invocation --%><hr/>

</body>

20] now Deploy our project again and right click on input.jsp and click run file you will redirect to a browser page as shown :

A screenshot of a computer

Description automatically generated

**Practical 2**

**Aim: Create a simple SOAP web service**

**steps :**

File -> New Project->JAVA Web ->Web Application->Project Name (Practical2)->Next->Finish

Right click on project name (Practical2)->New->Other->Web Services->Web Service->Web Service Name(calcy)->package(dev.services)->Finish

Right Click on hello->insert code->add web service operation->addition -> int->add->n1 int->add n2 int-> finish

code :

public String addition(@WebParam(name = "n1") int n1, @WebParam(name = "n2") int n2) {

//TODO write your implementation code here:

return "Sum is " + (n1+n2);

right click Project(Practical2) clean & build ,deploy

right click web service (calcy) ->test web service

calcy.java

Code :-

/\*

 \* 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 dev.services;

import javax.jws.WebService;

import javax.jws.WebMethod;

import javax.jws.WebParam;

/\*\*

 \*

 \* @author Admin

 \*/

@WebService(serviceName = "calcy")

public class calcy {

    /\*\*

     \* This is a sample web service operation

     \* @param txt

     \* @return

     \*/

    @WebMethod(operationName = "hello")

    public String hello(@WebParam(name = "name") String txt) {

        return "Hello " + txt + " !";

    }

    /\*\*

     \* Web service operation

     \* @param a

     \* @param b

     \* @return

     \*/

    @WebMethod(operationName = "addition")

    public String addition(@WebParam(name = "a") int a, @WebParam(name = "b") int b) {

        //TODO write your implementation code here:

        return "Sum is "+(a+b);

    }

    /\*\*

     \* Web service operation

     \* @param a

     \* @param b

     \* @return

     \*/

    @WebMethod(operationName = "multi")

    public String multi(@WebParam(name = "a") int a, @WebParam(name = "b") int b) {

        //TODO write your implementation code here:

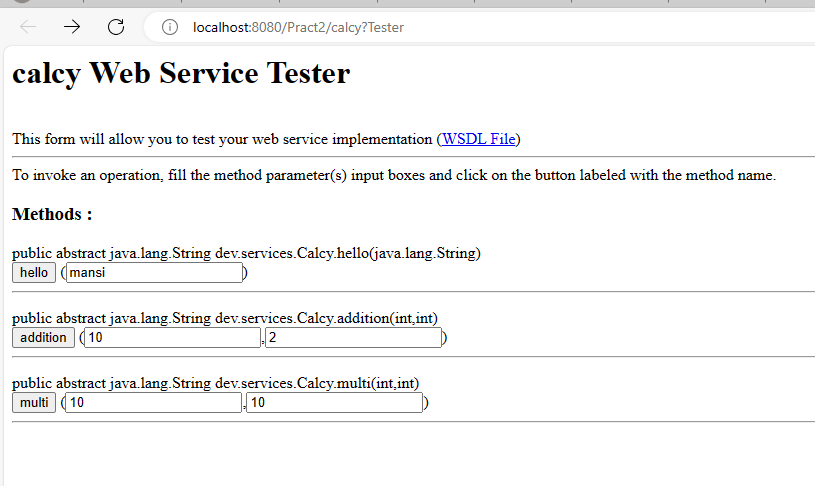
        return "multiplication  is "+(a\*b);

    }

}

**A screenshot of a computer

Description automatically generated**

****

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Practical 3**

**Aim: Create a Simple REST Service.**

NetBeans IDE

File - New Project - Java web - web application - next - project name - (hellorestservices)- finish

Right click - project name - (hellorestservices)- new - other - web services - restful web services from patterns - next - next -

Under specify resources classes

Resource package - helloworld

Path - helloworld

Class name - Helloworld

Mime type - text/ html

Finish

In helloworld.java file

import java.util.Calendar;

In Method

gethtml()

{

return “<html><body>time = “+Calendar.getInstance().getTime()+”</body></html>”;

}

To Run : Right click - project name - (hellorestservices)- test rest web service -> ok

**Code**

/\*

\* 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 helloworld;

import java.util.Calendar;

import javax.ws.rs.core.Context;

import javax.ws.rs.core.UriInfo;

import javax.ws.rs.Consumes;

import javax.ws.rs.Produces;

import javax.ws.rs.GET;

import javax.ws.rs.Path;

import javax.ws.rs.PUT;

import javax.ws.rs.core.MediaType;

/\*\*

\* REST Web Service

\*

\* @author computer lab

\*/

@Path("helloworld")

public class HelloWorld {

@Context

private UriInfo context;

/\*\*

\* Creates a new instance of HelloWorld

\*/

public HelloWorld() {

}

/\*\*

\* Retrieves representation of an instance of helloworld.HelloWorld

\* @return an instance of java.lang.String

\*/

@GET

@Produces(MediaType.TEXT\_HTML)

public String getHtml() {

//TODO return proper representation object

return "<html><body>time = "+Calendar.getInstance().getTime() +" </html></body>";

}

/\*\*

\* PUT method for updating or creating an instance of HelloWorld

\* @param content representation for the resource

\*/

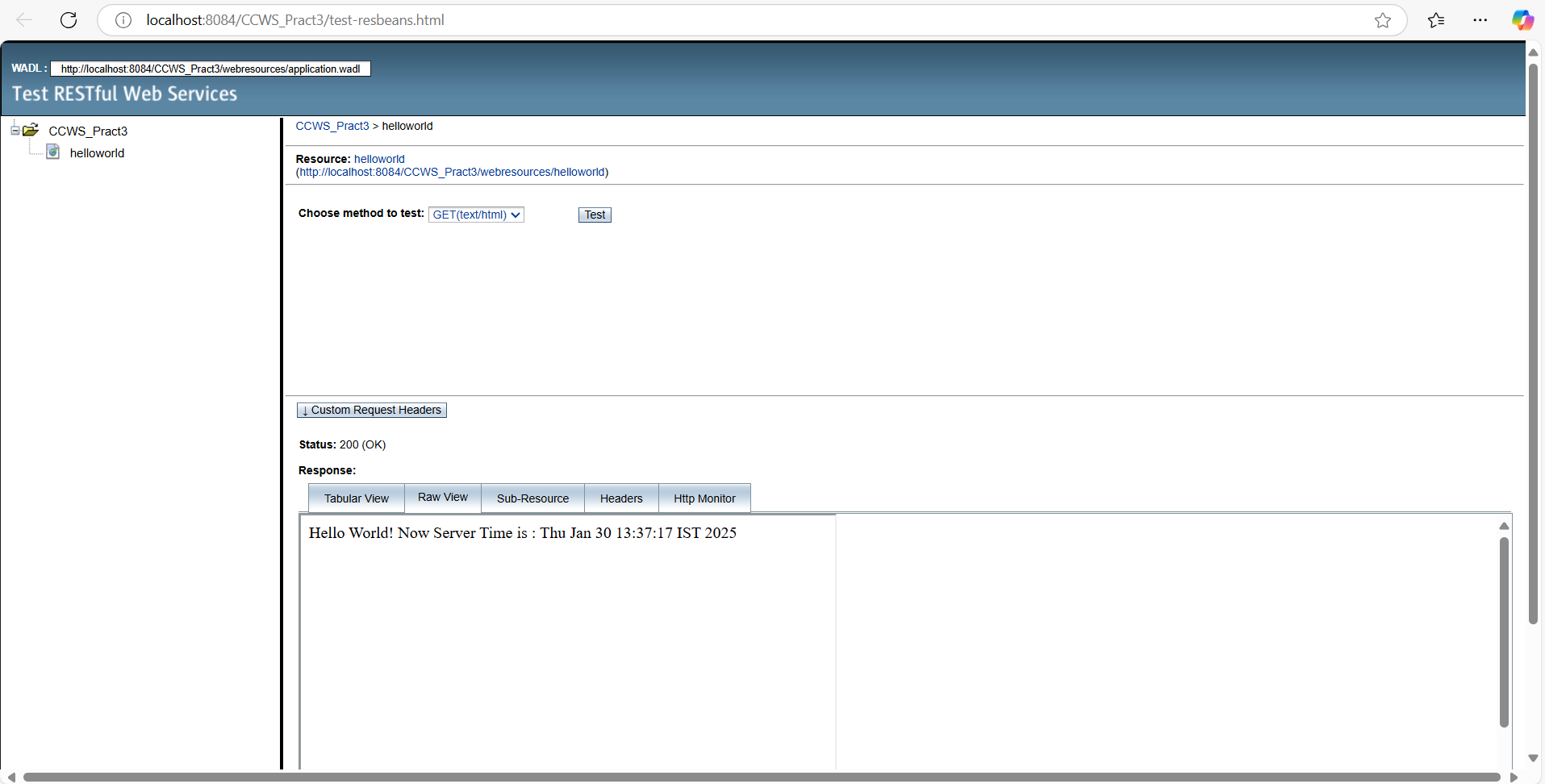
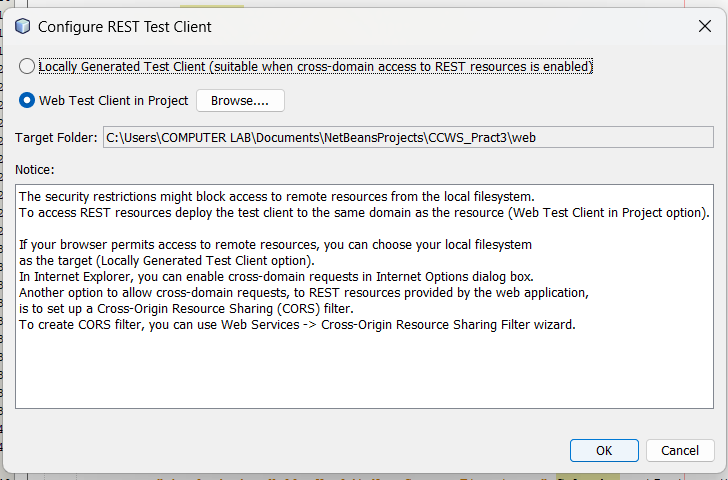
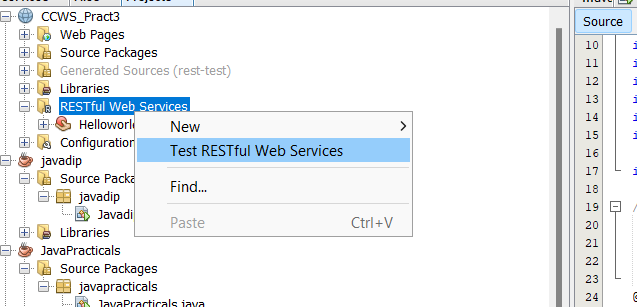
@PUT

@Consumes(MediaType.TEXT\_HTML)

public void putHtml(String content) {

}

}



**Practical 4**

**Aim: Create a web service of using Google maps/google Search**

Right click on project name (Practical4)->New->JSP ->input.jsp

Right click on project name (Practical4)->New->JSP ->index.jsp

Code :-

input.jsp:-

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

</head>

<body>

<form action="index.jsp">

<pre>

Enter latitude:<input type="text" name="t1" />

Enter longitude:<input type="text" name="t2" />

<input type="submit" value="Show" />

</pre>

</form>

</body>

</html>

index.jsp:-

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<style>

    #map {

height: 400px;

width: 100%;

}

</style>

</head>

<body>

<%

double lati=Double.parseDouble(request.getParameter("t1"));

double longi=Double.parseDouble(request.getParameter("t2"));

%>

<h3> Google Maps </h3>

<div id="map"></div>

<script lang="javascript">

function initMap() {

var info={lat: <%=lati%>, lng: <%=longi%>};

var map = new google.maps.Map(document.getElementById('map'), {

zoom: 4, center: info

});

var marker = new google.maps.Marker({

position: info, map: map

});

}

</script>

<script src="https://maps.googleapis.com/maps/api/js?key=&callback=initMap" async defer></script>   // (get this link from google chrome by typing “get google map api key free”)

    </body>

</html>

**A screenshot of a computer

Description automatically generated**

**A map of the world

Description automatically generated**

**PRACTICAL 5**

**Aim: Installation and Configuration of virtualization using KVM.**

**Terminal:**

ddvsc@ddvsc-H410M-S2-V2:~$ **egrep -c '(vmx|svm)' /proc/cpuinfo**

16

ddvsc@ddvsc-H410M-S2-V2:~$ **sudo kvm-ok**

[sudo] password for ddvsc:

INFO: /dev/kvm exists

KVM acceleration can be used

ddvsc@ddvsc-H410M-S2-V2:~$ **sudo apt install cpu-checker**

Reading package lists... Done

Building dependency tree

Reading state information... Done

cpu-checker is already the newest version (0.7-0ubuntu7).

0 upgraded, 0 newly installed, 0 to remove and 675 not upgraded.

ddvsc@ddvsc-H410M-S2-V2:~$ **sudo apt install qemu-kvm libvirt-daemon-system libvirt-clients bridge-utils -y**

Reading package lists... Done

Building dependency tree

Reading state information... Done

bridge-utils is already the newest version (1.5-15ubuntu1).

libvirt-clients is already the newest version (4.0.0-1ubuntu8.21).

libvirt-daemon-system is already the newest version (4.0.0-1ubuntu8.21).

qemu-kvm is already the newest version (1:2.11+dfsg-1ubuntu7.42).

0 upgraded, 0 newly installed, 0 to remove and 675 not upgraded.

ddvsc@ddvsc-H410M-S2-V2:~$ **sudo apt install qemu-kvm libvirt-daemon-system libvirt-clients bridge-utils -y**

Reading package lists... Done

Building dependency tree

Reading state information... Done

bridge-utils is already the newest version (1.5-15ubuntu1).

libvirt-clients is already the newest version (4.0.0-1ubuntu8.21).

libvirt-daemon-system is already the newest version (4.0.0-1ubuntu8.21).

qemu-kvm is already the newest version (1:2.11+dfsg-1ubuntu7.42).

0 upgraded, 0 newly installed, 0 to remove and 675 not upgraded.

ddvsc@ddvsc-H410M-S2-V2:~$ **sudo virsh list --all**

Id Name State

----------------------------------------------------

- ubuntu18.04 shut off

ddvsc@ddvsc-H410M-S2-V2:~$ **sudo systemctl status libvirtd**

● libvirtd.service - Virtualization daemon

Loaded: loaded (/lib/systemd/system/libvirtd.service; enabled; vendor preset: enabled)

Active: active (running) since Tue 2025-02-25 11:41:21 IST; 22min ago

Docs: man:libvirtd(8)

https://libvirt.org

Main PID: 934 (libvirtd)

Tasks: 19 (limit: 32768)

CGroup: /system.slice/libvirtd.service

├─ 934 /usr/sbin/libvirtd

├─1116 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default.conf --leasefile-ro --dhcp-script=/usr/lib/libvirt/libvirt\_le

└─1117 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default.conf --leasefile-ro --dhcp-script=/usr/lib/libvirt/libvirt\_le

Feb 25 11:51:48 ddvsc-H410M-S2-V2 dnsmasq[1116]: using nameserver 127.0.0.53#53

Feb 25 11:51:48 ddvsc-H410M-S2-V2 dnsmasq[1116]: reading /etc/resolv.conf

Feb 25 11:51:48 ddvsc-H410M-S2-V2 dnsmasq[1116]: using nameserver 127.0.0.53#53

Feb 25 11:51:48 ddvsc-H410M-S2-V2 dnsmasq[1116]: reading /etc/resolv.conf

Feb 25 11:51:48 ddvsc-H410M-S2-V2 dnsmasq[1116]: using nameserver 127.0.0.53#53

Feb 25 11:51:48 ddvsc-H410M-S2-V2 dnsmasq[1116]: reading /etc/resolv.conf

Feb 25 11:51:48 ddvsc-H410M-S2-V2 dnsmasq[1116]: using nameserver 127.0.0.53#53

Feb 25 11:54:19 ddvsc-H410M-S2-V2 libvirtd[934]: 2025-02-25 06:24:19.679+0000: 954: info : libvirt version: 4.0.0, package: 1ubuntu8.21 (Marc

Feb 25 11:54:19 ddvsc-H410M-S2-V2 libvirtd[934]: 2025-02-25 06:24:19.679+0000: 954: info : hostname: ddvsc-H410M-S2-V2

Feb 25 11:54:19 ddvsc-H410M-S2-V2 libvirtd[934]: 2025-02-25 06:24:19.679+0000: 954: error : virStoragePoolObjSourceFindDuplicate:1907 : operat

ddvsc@ddvsc-H410M-S2-V2:~$ **sudo systemctl enable --now libvirtd**

Synchronizing state of libvirtd.service with SysV service script with /lib/systemd/systemd-sysv-install.

Executing: /lib/systemd/systemd-sysv-install enable libvirtd

ddvsc@ddvsc-H410M-S2-V2:~$ **sudo apt install virt-manager -y**

Reading package lists... Done

Building dependency tree

Reading state information... Done

virt-manager is already the newest version (1:1.5.1-0ubuntu1.2).

0 upgraded, 0 newly installed, 0 to remove and 675 not upgraded.

ddvsc@ddvsc-H410M-S2-V2:~$ **sudo virt-manager**

ddvsc@ddvsc-H410M-S2-V2:~$ **ssh -Y username@hostname**

ssh: Could not resolve hostname hostname: Name or service not known