

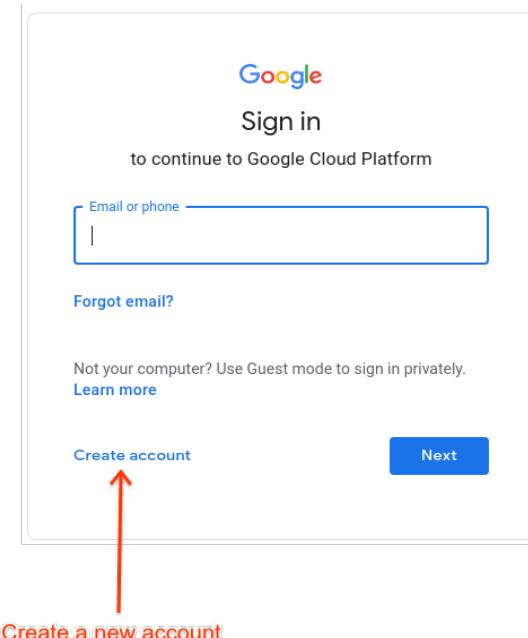
## EXP NO. 1

### CREATE A GOOGLE CLOUD ACCOUNT & PROJECT

To create a Google Cloud account:

Open Google Cloud console in a browser.

When prompted to sign in, create a new account by clicking Create account:



1. Follow the instructions to register your corporate email address as a Google Cloud. Alternatively, you can use a Gmail account or other Google account.
2. Continue to the Google Cloud console and accept the Google Cloud Platform terms presented.

**To create a new Google Cloud project:**

1. Open the Google Cloud console and log in with the account you created in Step:-1 Create a Google Account.
2. Click the **Project Selector** at the top of the view.

The **Project Selector** dialog box displays:

Name	ID
My Project	my-project
Some Project	other-project
Some Other Project	some-other-project
Yet Another Project	yet-another-project

3. Click **New Project** in the upper right.

The **New Project** view displays.

4. Enter a friendly name for your project in the **Project name** field.

Google Cloud Platform generates a corresponding project ID below the **Project name** field, as the following example shows:

You can change the generated project ID by clicking **Edit**, but the project ID must be unique.

Take note of the **project ID**. You will use it later in this setup process.

**NOTE:** After you create the project, you cannot change the project ID.

5. In the **Billing account** field, select the account that you will use to pay for your project.
6. In the **Location** field, select a folder for your project. Folders are an optional part of the Google Cloud Platform project resource tree.
7. Click **Create**.

Google Cloud Platform creates your new project.

## **EXP NO. 2 :**

### **INSTALL VIRTUALBOX / VMWARE WORKSTATION WITH DIFFERENT FLAVOURS OF LINUX OR WINDOWS OS ON TOP OF WINDOWS7 OR 8.**

**Aim:**

To Install Virtualbox / VMware Workstation with different flavours of linux or windows OS on top of windows7 or 8.

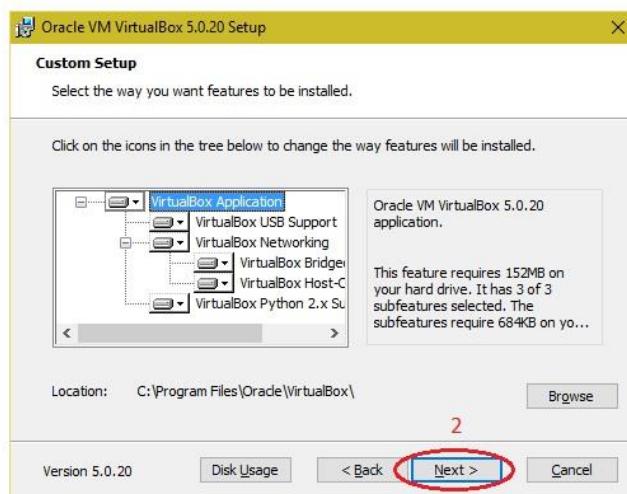
**PROCEDURE:**

Steps to install Virtual Box:

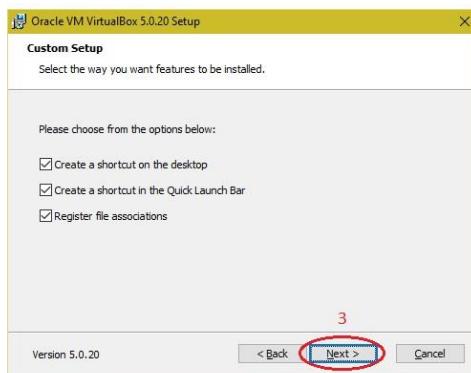
1. Download the Virtual box exe and click the exe file...and select next button..



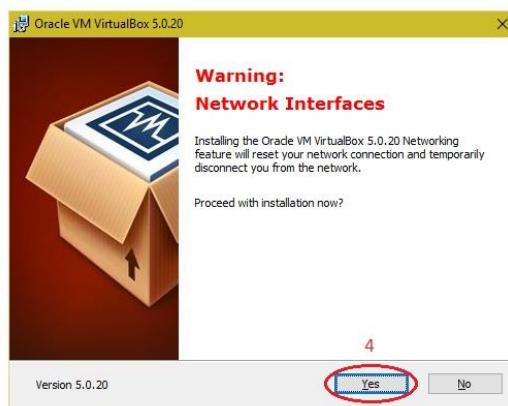
Click the next button..



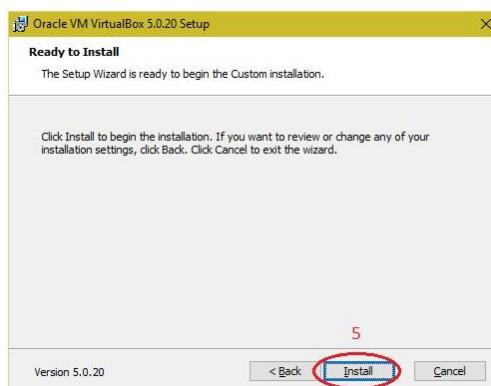
Click the next button



4. Click the YES button..



5. Click the install button...

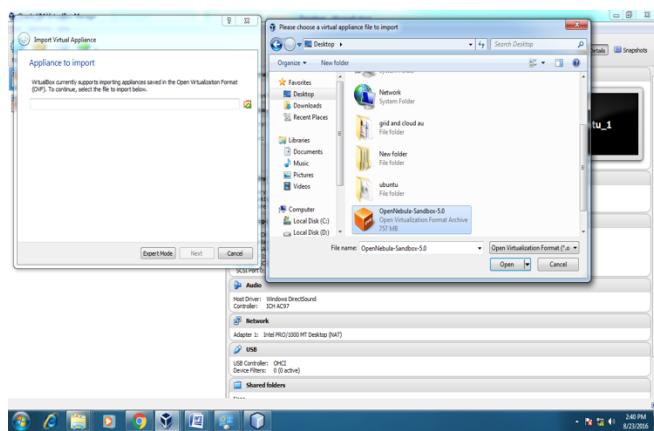
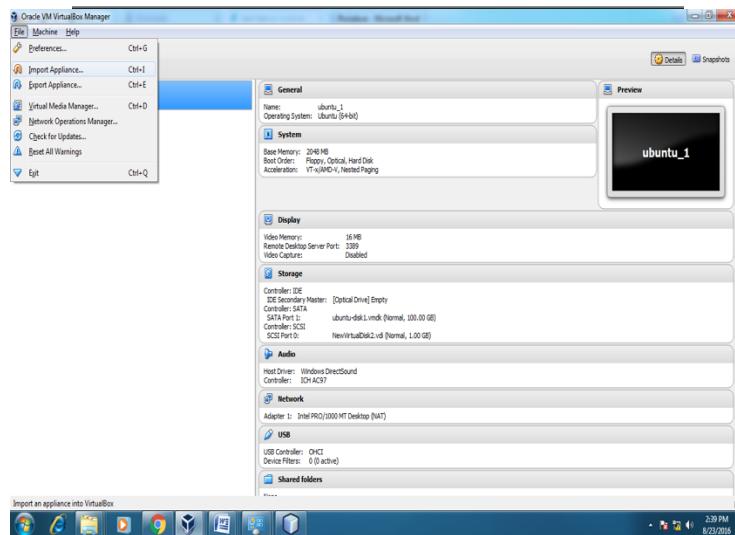


6. Then installation was completed. The show virtual box icon on desktop screen....



Steps to import Open nebula sandbox:

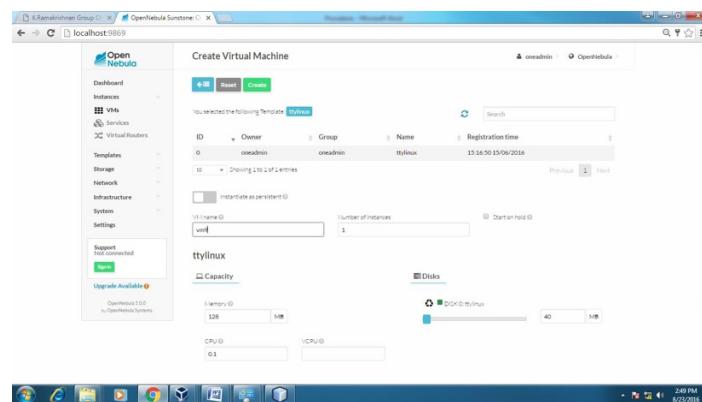
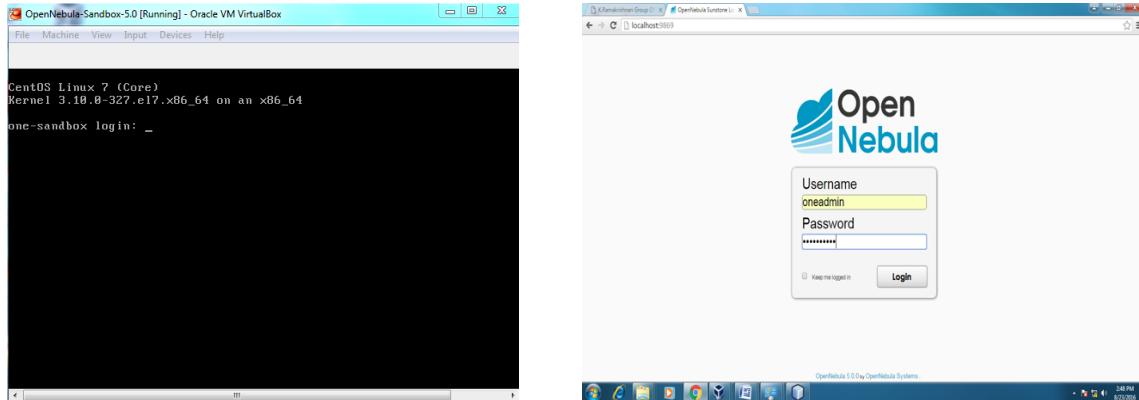
1. Open Virtual box
2. File ▾ Import Appliance
3. Browse OpenNebula-Sandbox-5.0.ova file
4. Then go to setting, select Usb and choose USB 1.1
5. Then Start the Open Nebula
6. Login using username: root, password:opennebula



Steps to create Virtual Machine through opennebula

1. Open Browser, type localhost:9869
2. Login using username: oneadmin, password: opennebula
3. Click on instances, select VMs then follow the steps to create Virtaul machine

- Expand the + symbol
- Select user oneadmin
- Then enter the VM name,no.of instance, cpu.
- Then click on create button.
- Repeat the steps the C,D for creating more than one VMs.



## APPLICATIONS:

There are various applications of cloud computing in today's network world. Many search engines and social websites are using the concept of cloud computing like [www.amazon.com](http://www.amazon.com), [hotmail.com](http://hotmail.com), [facebook.com](http://facebook.com), [linkedin.com](http://linkedin.com) etc. the advantages of cloud computing in context to scalability is like reduced risk , low cost testing ,ability to segment the customer base and auto-scaling based on application load.

## RESULT:

Thus the procedure to run the virtual machine of different configuration.

## **EXP NO. 3:**

### **INSTALL A C COMPILER IN THE VIRTUAL MACHINE CREATED USING VIRTUAL BOX AND EXECUTE SIMPLE PROGRAMS**

Aim:

To Install a C compiler in the virtual machine created using 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: dinesh, password:99425.

virtual box and

Aim:

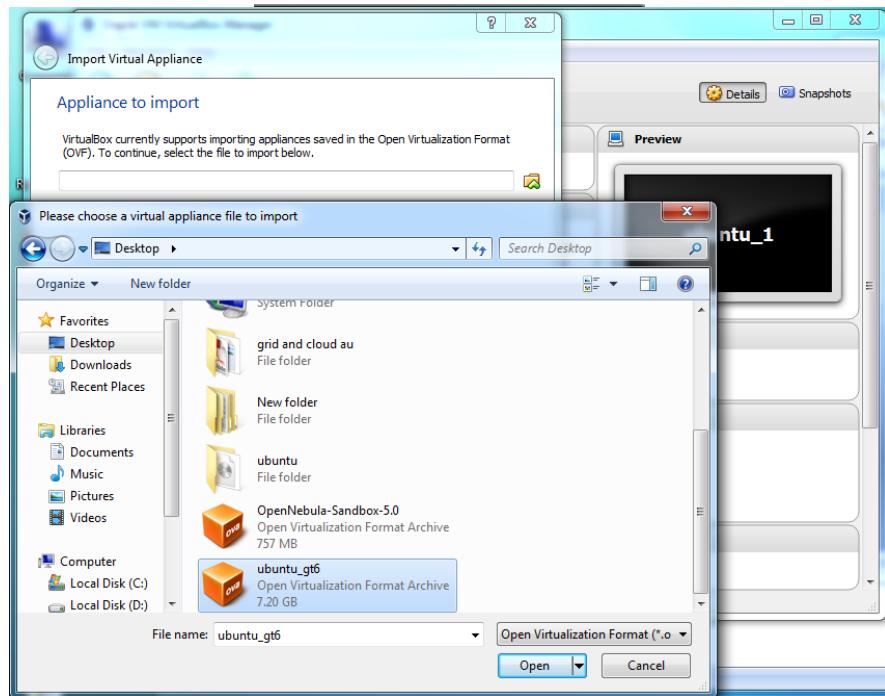
To Install a C compiler in the virtual machine created using execute Simple Programs`

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: dinesh, password:99425.

virtual box and



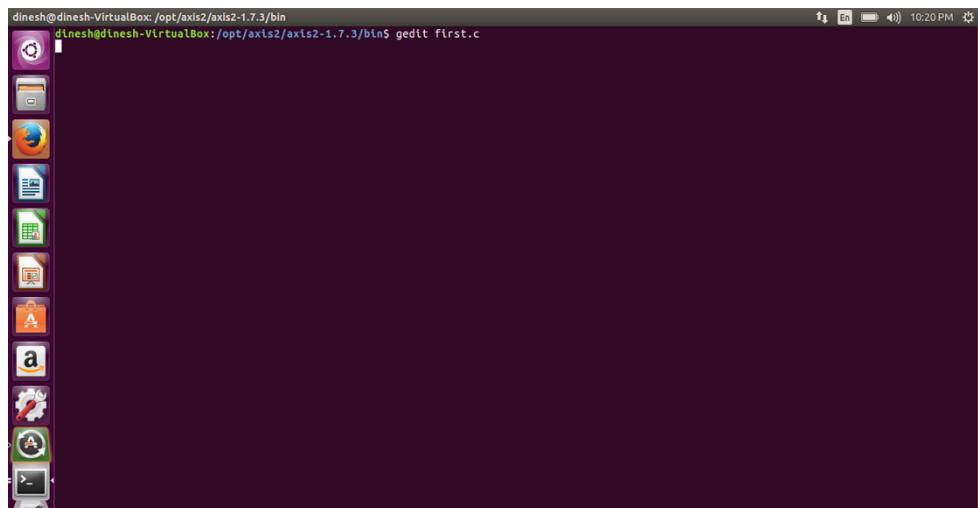
## Steps to run c program:

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
  5. ./a.out
1. Type cd /opt/axis2/axis2-1.7.3/bin then press enter

r

```
dinesh@dinesh-VirtualBox: /opt/axis2/axis2-1.7.3/bin
dinesh@dinesh-VirtualBox:-5 cd /opt/axis2/axis2-1.7.3/bin
dinesh@dinesh-VirtualBox:/opt/axis2/axis2-1.7.3/bin$
```

2. Type gedit first.c



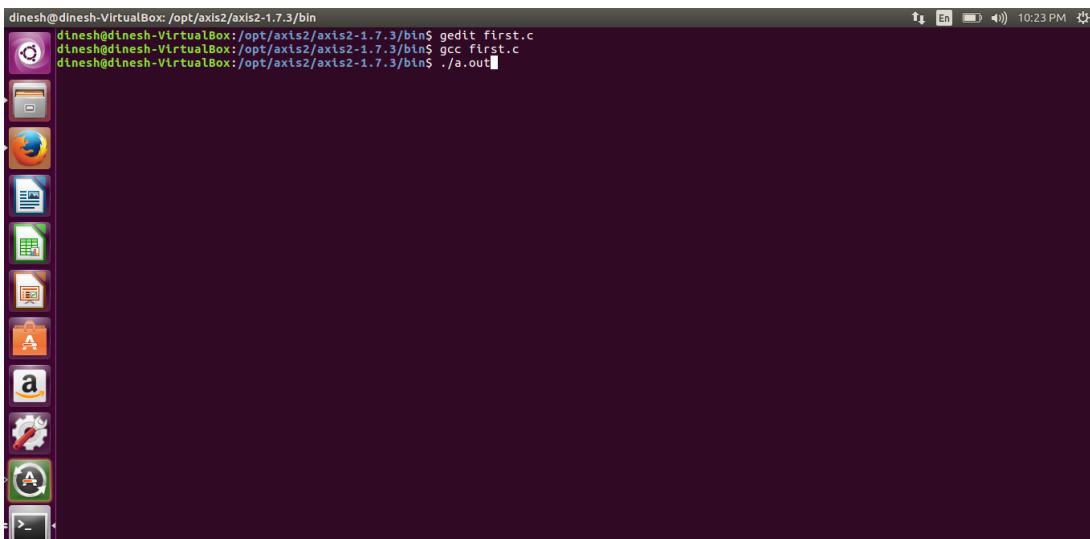
### 3. Type the c program

A screenshot of the gedit text editor. The title bar says 'first.c (~/) - gedit'. The code in the editor is:

```
#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");
}
```

The status bar at the bottom shows 'Saving file '/home/dinesh/first.c'...' and other editor settings like 'Tab Width: 8' and 'Ln 13, Col 2'.

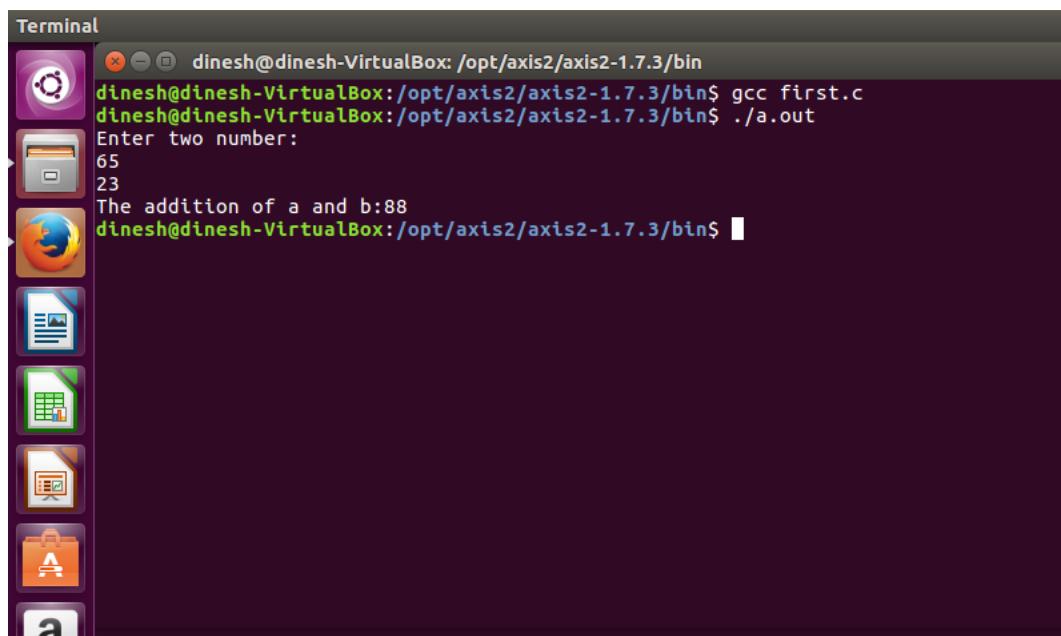
### 4. Running the C program



A screenshot of a Linux desktop environment. On the left is a vertical dock containing icons for various applications: Dash, Home, File Manager, Terminal, Mail, Photos, Videos, Calendar, Activities, App Center, and Help. The main window is a terminal window titled 'dinesh@dinesh-VirtualBox'. The terminal shows the following command-line session:

```
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
```

## 5. Display the output:



A screenshot of a Linux desktop environment. On the left is a vertical dock containing icons for various applications: Dash, Home, File Manager, Terminal, Mail, Photos, Videos, Calendar, Activities, App Center, and Help. The main window is a terminal window titled 'Terminal' (indicated by the icon). The terminal shows the following command-line session:

```
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$
```

## APPLICATIONS:

Simply running all programs in grid environment.

## RESULT:

Thus the simple C programs executed successfully.

## EXP NO. 4

### INSTALL GOOGLE APP ENGINE. CREATE HELLO WORLD APP AND OTHER SIMPLE WEB APPLICATIONS USING PYTHON/JAVA.

Aim:

To Install Google App Engine. Create hello world app and other simple web applications using python/java.

Procedure:

#### 1. Install Google Plugin for Eclipse

Read this guide – [how to install Google Plugin for Eclipse](#). If you install the Google App Engine

Java SDK together with “Google Plugin for Eclipse“, then go to step 2, Otherwise, get the [Goo App Engine Java SDK](#) and extract it.

#### 2. Create New Web Application Project

In Eclipse toolbar, click on the Google icon, and select “New Web Application Project...” Figure – New Web Application Project

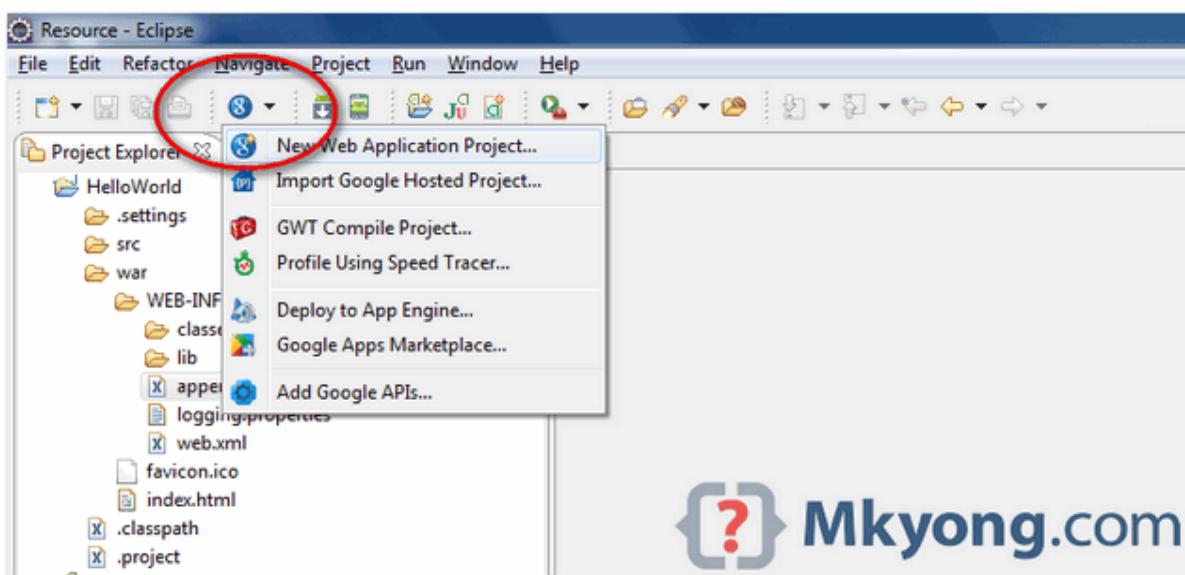
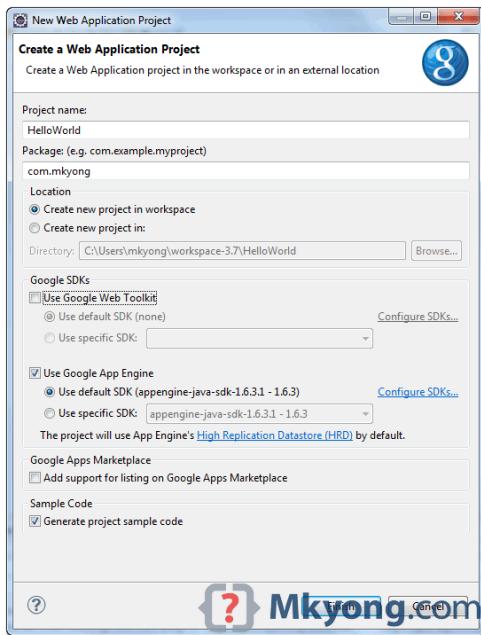


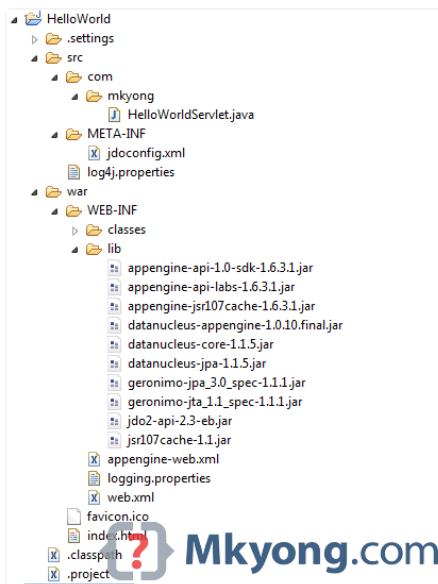
Figure – Deselect the “Google Web ToolKit“, and link your GAE Java SDK via the “configure SDK” link.



Click finished, Google Plugin for Eclipse will generate a sample project automatically.

### 3. Hello World

Review the generated project directory.



Nothing special, a standard Java web project structure.

HelloWorld/ src/

...Java source code... META-INF/

...other configuration... war/  
...JSPs, images, data files... WEB-INF/  
...app configuration... lib/  
...JARs for libraries... classes/  
...compiled classes...

Copy

The extra is this file “appengine-web.xml“, Google App Engine need this to run and deploy the application.

File : appengine-web.xml

```
<?xml version="1.0" encoding="utf-8"?>

<appengine-web-app xmlns="http://appengine.google.com/ns/1.0">
<application></application>
<version>1</version>

<!-- Configure java.util.logging -->

<system-properties>
<property name="java.util.logging.config.file" value="WEB-
INF/logging.properties"/>

</system-properties>

</appengine-web-app>
```

Copy

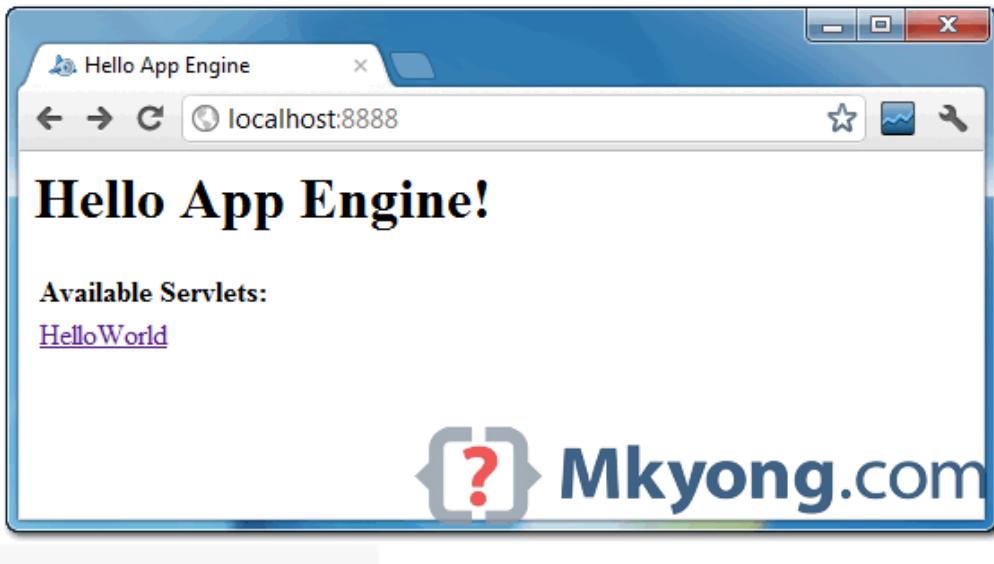
4. Run it local

Right click on the project and run as “Web Application“. Eclipse console :

```
//...
INFO: The server is running at http://localhost:8888/
30 Mac 2012 11:13:01 PM
com.google.appengine.tools.development.DevAppServerImpl start INFO: The
admin console is running at http://localhost:8888/_ah/admin
```

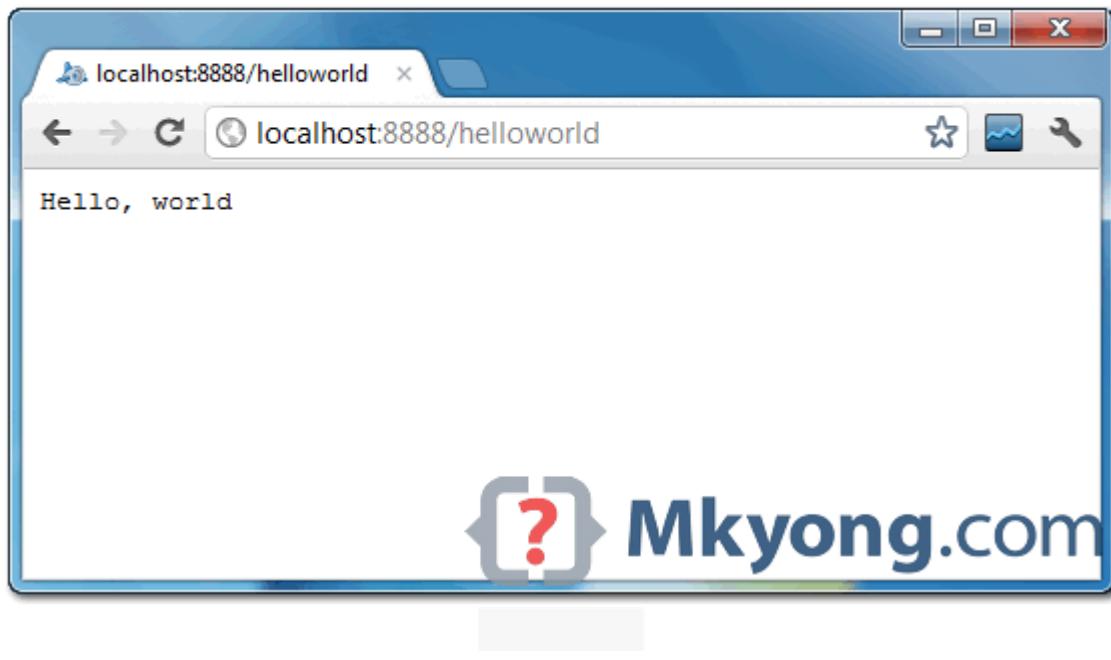
Copy

Access URL <http://localhost:8888/>, see output



and also the hello world servlet –

<http://localhost:8888/helloworld>



web.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<appengine-web-app xmlns="http://appengine.google.com/ns/1.0">  
<application>mkyong123</application>
```

```

<version>1</version>

<!-- Configure java.util.logging -->

<system-properties>
<property name="java.util.logging.config.file" value="WEB-
INF/logging.properties"/>

</system-properties> </appengine-web-app>

```

Copy

appengine-

## 5. Deploy to Google App Engine

Register an account on <https://appengine.google.com/>, and create an application ID for your web application.

In this demonstration, I created an application ID, named “mkyong123”, and put it in .

File : appengine-web.xml

To deploy, see following steps:

Figure 1.1 – Click on GAE deploy button on the toolbar.

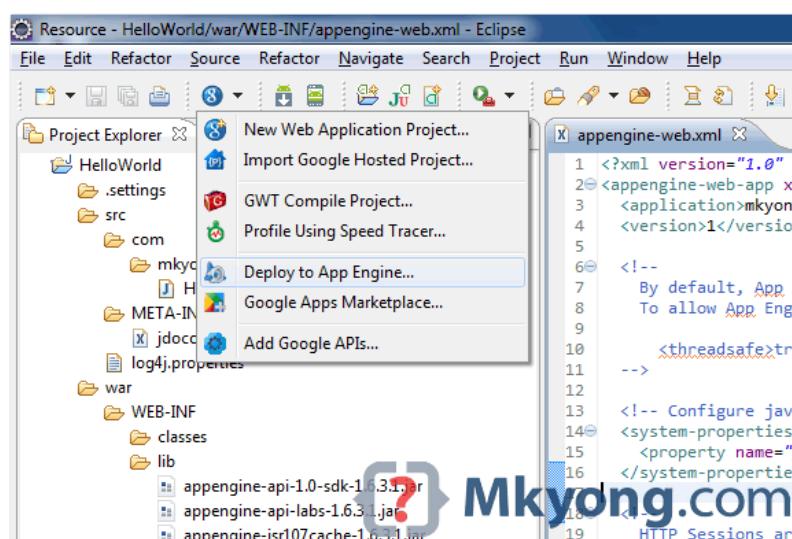
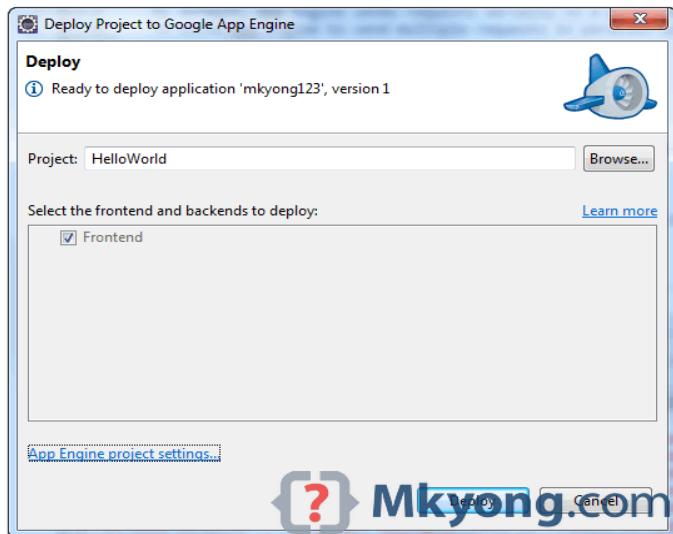


Figure 1.2 – Sign in with your Google account and click on the Deploy button.



Result:

Thus the simple application was created successfully.

## EXP NO-5

### FIND A PROCEDURE TO TRANSFER THE FILES FROM ONE VIRTUAL MACHINE TO ANOTHER VIRTUAL MACHINE.

PROCEDURE: Steps:

1. Open Browser, type localhost:9869
2. Login using username: oneadmin, password: opennebula 3. Then follow the steps to migrate VMs
  1. Click on infrastructure
  2. Select clusters and enter the cluster name
  3. Then select host tab, and select all hosts
  4. Then select Vnets tab, and select all vnet
  5. Then select datastores tab, and select all datastores
  6. And then choose host under infrastructure tab
  7. 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 8<sup>th</sup> 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:SACET

ID	Owner	Group	Name	Status	Host	IPs
5	oneadmin	oneadmin	vm2	FAILURE	navneekumar	172.16.100.205
4	oneadmin	oneadmin	vm2	FAILURE	navneekumar	172.16.100.204
3	oneadmin	oneadmin	vm1	FAILURE	navneekumar	172.16.100.203
2	oneadmin	oneadmin	naveen	FAILURE	navneekumar	172.16.100.202
1	oneadmin	oneadmin	naveen	FAILURE	navneekumar	172.16.100.201
0	oneadmin	oneadmin	ttlinu0	FAILURE	navneekumar	172.16.100.200

Host:one-sandbox

Host 0 one-sandbox

ID	Owner	Group	Name	Status	Host	IPs
7	oneadmin	oneadmin	vm8	RUNNING	one-sandbox	172.16.100.207
6	oneadmin	oneadmin	vm9	RUNNING	one-sandbox	172.16.100.206

Migrate Virtual Machine

VM 6 vm6 is currently running on Host one-sandbox  
VM 7 vm7 is currently running on Host one-sandbox

Select a Host

ID	Name	Cluster	RVMs	Allocated CPU	Allocated MEM	Status
2	rea	default	0	0/0	0KB/1...	PETRY
1	naveenkumar	rama	6	42/0	441MB/...	ERROR
0	one-sandbox	rama	2	20/100 [20%]	418/742 [55%]	ON

**Migrate**

VMs

ID	Owner	Group	Name	Status	Host	IPs
7	oneadmin	oneadmin	vm8	SAVE	naveenkumar	172.16.100.207
6	oneadmin	oneadmin	vm9	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	naseen	FAILURE	naveenkumar	172.16.100.202
1	oneadmin	oneadmin	naseen	FAILURE	naveenkumar	172.16.100.201
0	oneadmin	oneadmin	tsliun0	FAILURE	naveenkumar	172.16.100.200

After Migration:

ID	Name	Cluster	RVMS	Allocated CPU	Allocated MEM	Status
1	na-veekumar	rama	8	82 / 0	0 / 100 (0%)	ERROR
0	one-sandbox	rama	0	0 / 0	0 / 742 (0%)	ON

3 TOTAL 1 ON 0 OFF 2 ERROR

Host:one-sandbox

ID	Owner	Group	Name	Status	Host	IPs
----	-------	-------	------	--------	------	-----

There is no data available

Host:SACET

ID	Owner	Group	Name	Status	Host	IPs
7	oneadmin	oneadmin	vms1	FAILURE	na-veekumar	172.16.100.207
6	oneadmin	oneadmin	vms2	FAILURE	na-veekumar	172.16.100.206
5	oneadmin	oneadmin	vms3	FAILURE	na-veekumar	172.16.100.205
4	oneadmin	oneadmin	vms4	FAILURE	na-veekumar	172.16.100.204
3	oneadmin	oneadmin	vms5	FAILURE	na-veekumar	172.16.100.203
2	oneadmin	oneadmin	vms6	FAILURE	na-veekumar	172.16.100.202
1	oneadmin	oneadmin	vms7	FAILURE	na-veekumar	172.16.100.201
0	oneadmin	oneadmin	thimmo1	FAILURE	na-veekumar	172.16.100.200

## APPLICATIONS:

Easily migrate your virtual machine from one pc to another.

Result:

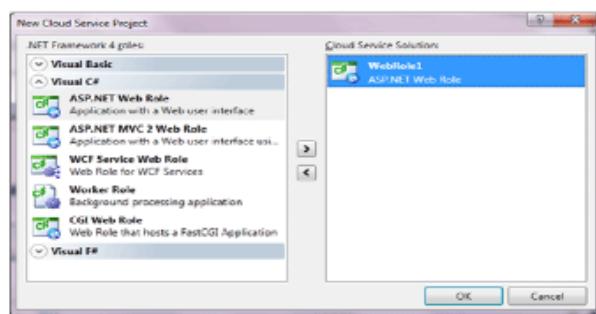
Thus the file transfer between VM was successfully completed.....

## EXP NO. 6

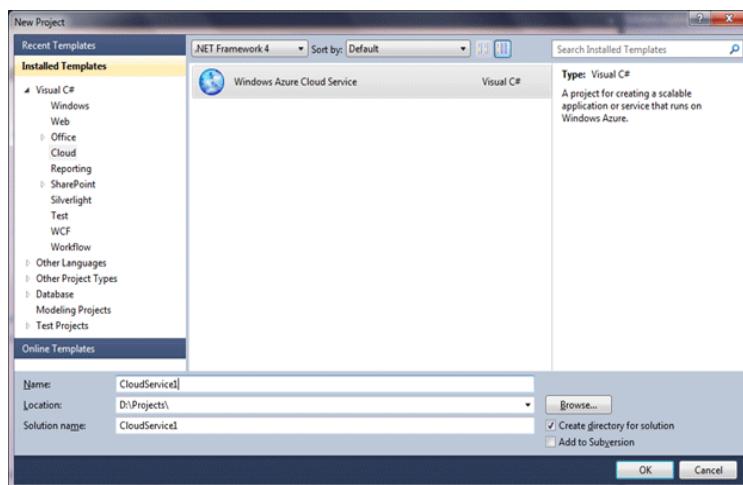
### DEVELOP A WINDOWS AZURE HELLO WORLD APPLICATION

In this experiment, we will see how to create a "Hello World!!!" application in Azure using .Net. I used Visual Studio 2010 along with the Azure SDK 1.3 installed. Start Visual Studio and select a new project. In the cloud template select Windows Azure Cloud Service.

Once you click OK, it asks for selecting a Role. Here we are adding an ASP.Net Web role.

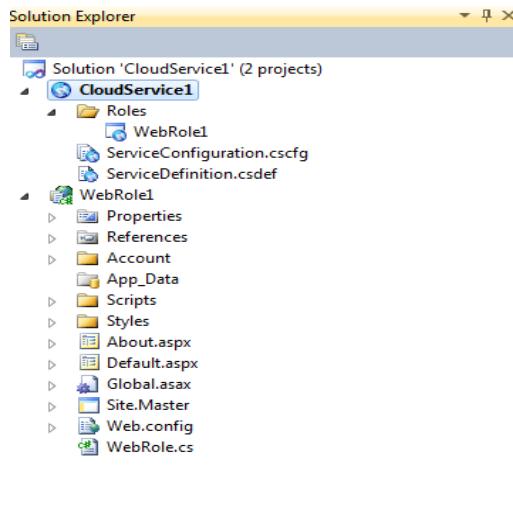


It will create two projects, CloudService1 (Azure Service Project) and WebRole1



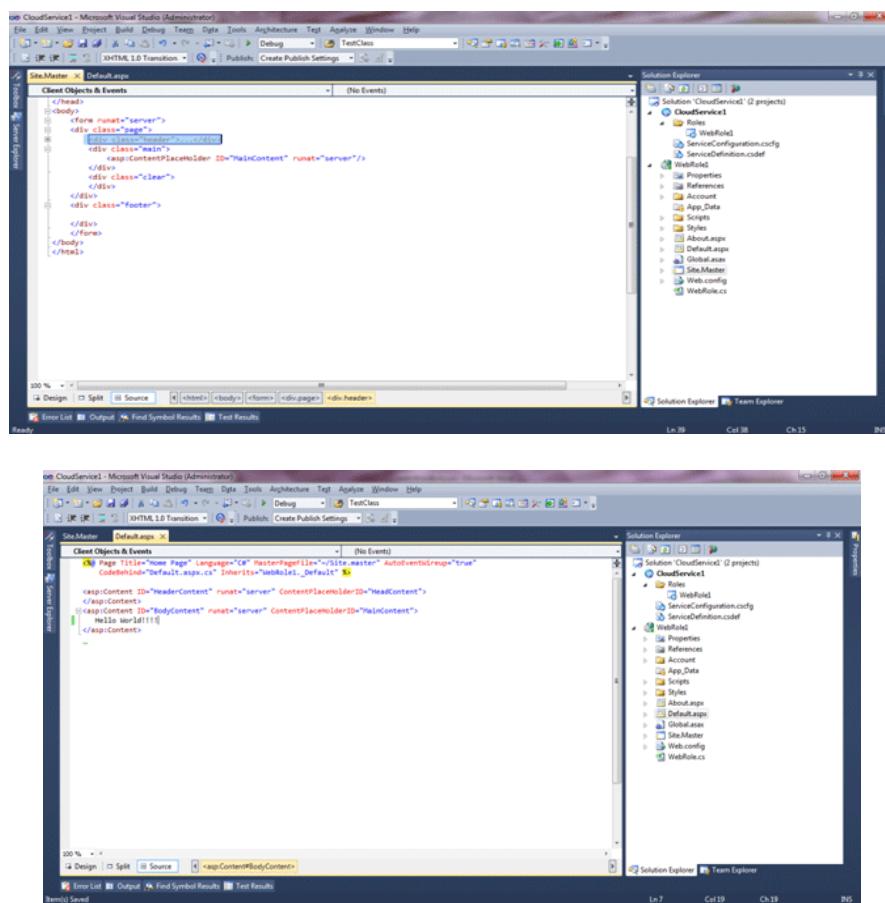
to create a deployment package. The ASP.Net project is like a normal web project.

Here i removed header div from the SiteMaster.

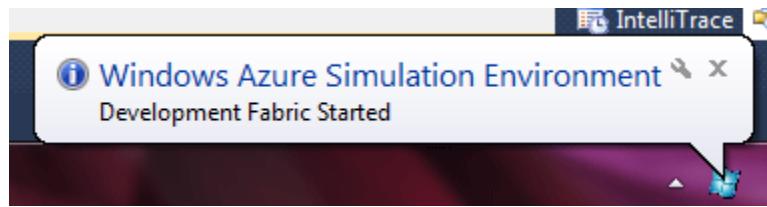


(ASP.Net Project). The Azure service project is used to configure the application, and

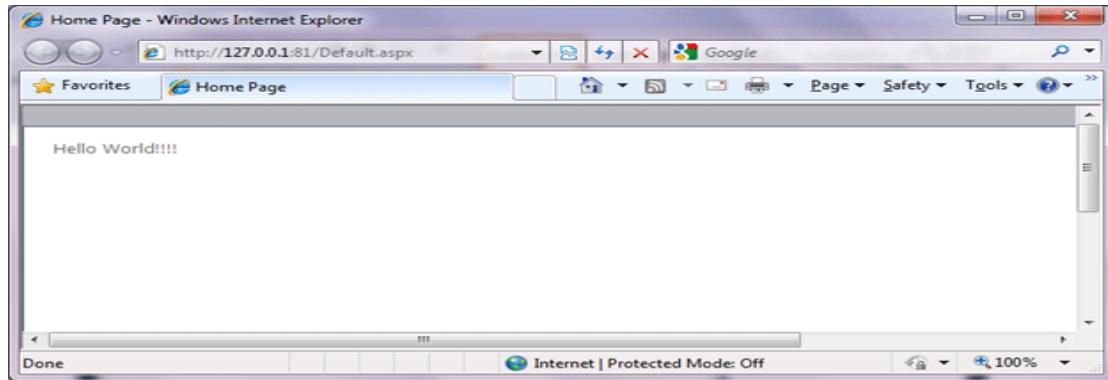
And add a text "Hello World!!!!" in the default.aspx



And we are done. Now Select Cloud Service as your startup project and run it. When you run it, the Azure simulation environment is initialized, and it starts development storage and development fabric.



And here you can see your Hello World!!!

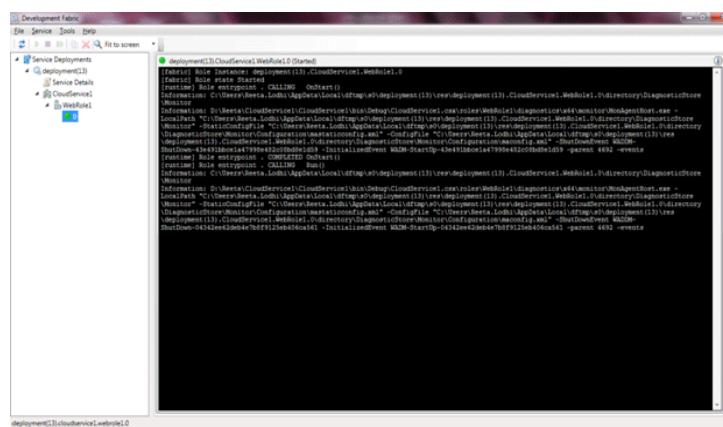


If you select WebRole1 as your startup project and run the solution, it will run as normal ASP.NET project and hosted locally on localhost.

If you click on the Azure Simulation and select "Show development Fabric UI" then it will show the Azure services running on your machine.



Here My CloudService1 is running having a single instance of WebRole.



## **EXP NO. 7**

### **ESTABLISH AN AWS ACCOUNT. USE THE AWS MANAGEMENT CONSOLE TO LAUNCH AN EC2 INSTANCE AND CONNECT TO IT.**

#### **Tasks**

To complete this tutorial, perform the following tasks:

1. Launch an Instance
2. Connect to Your Instance
3. Clean Up Your Instance

Before you begin, be sure that you've completed the steps in Setting Up with Amazon EC2.

#### **Step 1: Launch an Instance**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the console dashboard, choose Launch Instance.
3. The Choose an Amazon Machine Image (AMI) page displays a list of basic

Notice that these AMIs are marked "Free tier eligible."

#### **Step 2: Connect to Your Instance**

There are several ways to connect to your Linux instance. For more information, see Connect to Your Linux Instance.

#### **Step 3: Clean Up Your Instance**

1. In the navigation pane, choose Instances. In the list of instances, select the instance.
2. Choose Actions, Instance State, Terminate.
3. Choose Yes, Terminate when prompted for confirmation.

**EXP NO-8**  
**SENDING AND RECEIVING DATA WITH MPI**

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    string name;
    cout << "Enter name:";
    getline (cin, name);
    cout << "Hello " << name;
    return 0;
}
```

**OUTPUT:**

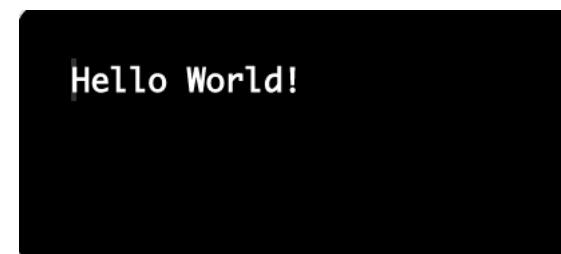
```
Enter name:
Error: Command failed: timeout 7 ./HelloWorld
```

**EXP NO. 9**

**SCALA COMPUTING HELLO WORLD**

```
object ScalaTest {  
    def main(args: Array[String]): Unit = {  
        println("Hello World!")  
    }  
}
```

**OUTPUT:**



## EXP NO. 10

### ARITHMETIC OPERATORS WITH SCALA

```
// Scala program to demonstrate
// the Arithmetic Operators

object Arithop
{

def main(args: Array[String])
{
    // variables
    var a = 50;
    var b = 30;

    // Addition
    println("Addition of a + b = " + (a + b));

    // Subtraction
    println("Subtraction of a - b = " + (a - b));

    // Multiplication
    println("Multiplication of a * b = " + (a * b));

    // Division
    println("Division of a / b = " + (a / b));

    // Modulus
    println("Modulus of a % b = " + (a % b));
}

}
```

#### **OUTPUT:**

```
Addition of a + b = 80
Subtraction of a - b = 20
Multiplication of a * b = 1500
Division of a / b = 1
Modulus of a % b = 20
```

## EXP NO. 11

### INHERITANCE WITH SCALA

```
// Scala program to illustrate the
// implementation of inheritance

// Base class
class Geeks1{
    var Name: String = "Ankita"
}

// Derived class
// Using extends keyword
class Geeks2 extends Geeks1
{
    var Article_no: Int = 130

    // Method
    def details()
    {
        println("Author name: " +Name);
        println("Total numbers of articles: " +Article_no);
    }
}

object Main
{

    // Driver code
    def main(args: Array[String])
    {

        // Creating object of derived class
        val ob = new Geeks2();
        ob.details();
    }
}
```

#### **OUTPUT:**

Author name: Ankita  
Total numbers of articles: 130

## EXP NO. 12

### APPLICATION OF APACHE SPARK

```
import org.apache.spark.SparkConf;
import org.apache.spark.SparkContext;
/**
 * Configure Apache Spark Application Name
 */
public class AppConfigureExample {
    public static void main(String[] args) {
        // configure spark
        SparkConf conf = new SparkConf().setMaster("local[2]");
        conf.set("spark.app.name", "SparkApplicationName");

        // start a spark context
        SparkContext sc = new SparkContext(conf);

        // print the configuration
        System.out.println(sc.getConf().toDebugString());

        // stop the spark context
        sc.stop();
    }
}
```

#### **OUTPUT:**

```
spark.app.id=local-1501222987079
spark.app.name=SparkApplicationName
spark.driver.host=192.168.1.100
spark.driver.port=44103
spark.executor.id=driver
spark.master=local[2]
```

## **EXP NO. 13**

### **ASYNCHRONOUS DYNAMIC LOAD BALANCER**

```
<!DOCTYPE html>

<html>

<body>

<h1>ASYNCHRONOUS</h1>

<h2>setInterval() with a Callback</h2>

<p>Using setInterval() to display the time every second (1000 milliseconds).</p>

<h1 id="demo"></h1>

<script>

setInterval(myFunction, 1000);

function myFunction() {

let d = new Date();

document.getElementById("demo").innerHTML=

d.getHours() + ":" +

d.getMinutes() + ":" +

d.getSeconds(); }

</script>

</body>

</html>
```

#### **OUTPUT:**

#### **ASYNCHRONOUS**

setInterval() with a Callback

Using setInterval() to display the time every second (1000 milliseconds).

15:20:00

# EXP NO. 14

## NETWORKING STORAGE IN AWS

```
<!DOCTYPE html>

<html>
<head>
<script>

function clickCounter() {
    if (typeof(Storage) !== "undefined") {
        if (sessionStorage.clickcount) {
            sessionStorage.clickcount = Number(sessionStorage.clickcount)+1;
        } else {
            sessionStorage.clickcount = 1;
        }
        document.getElementById("result").innerHTML = "You have clicked the button " +
        sessionStorage.clickcount + " time(s) in this session.";
    } else {
        document.getElementById("result").innerHTML = "Sorry, your browser does not support
        web storage...";
    }
}

</script>
</head>
<body>

<p><button onclick="clickCounter()" type="button">Click me!</button></p>
<div id="result"></div>
```

```
<p>Click the button to see the counter increase.</p>
<p>Close the browser tab (or window), and try again, and the counter is reset.</p>
</body>
</html>
```

## **OUTPUT:**

Click me!

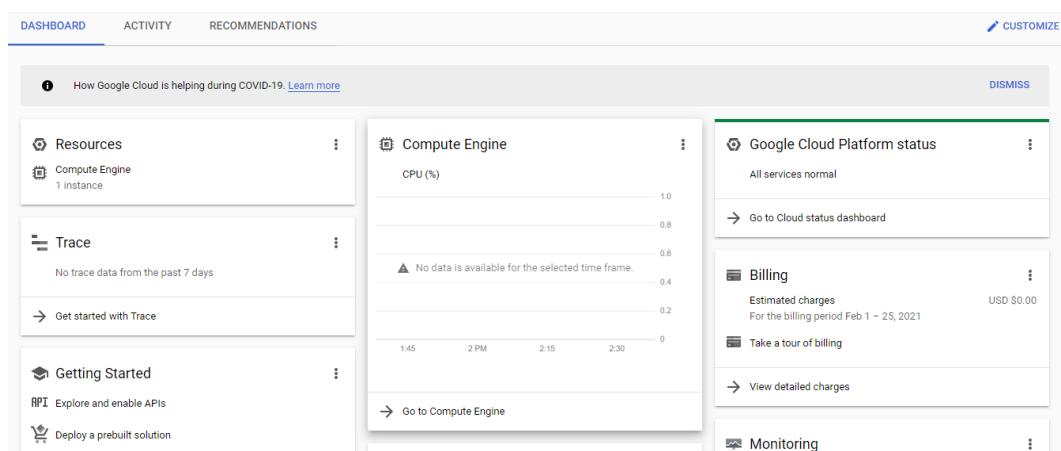
Click the button to see the counter increase.

Close the browser tab (or window), and try again, and the counter is reset.

## EXP NO. 15

# HOW TO LAUNCH A GPU INSTANCE ON GOOGLE CLOUD — FAST

Google Cloud is a great and easy way to use GPUs to train your models. The first step is to create a google cloud account. You can use your Gmail/google account to do this. Once you have your account set up, you should something like this.



Once you have done this, your screen should look something like this.

IAM & Admin		Quotas	EDIT QUOTAS	SHOW QUOTA PANEL	
+	IAM	Enter property name or value			
⊕	Identity & Organization				
🔍	Policy Troubleshooter				
📝	Policy Analyzer				
📋	Organization Policies				
✉️	Service Accounts				
🏷️	Labels				
⚙️	Settings				
🔒	Privacy & Security				
🌐	Identity-Aware Proxy				
👤	Roles				
Service		Limit name	Quota status ↑	Details	
Compute Engine API		GPUs (all regions)	1 of 1 quota is reaching limit	ALL QUOTAS	
Compute Engine API		NVIDIA K80 GPUs	1 of 99 quotas are reaching limit	ALL QUOTAS	
Compute Engine API		NVIDIA V100 GPUs	1 of 99 quotas are reaching limit	ALL QUOTAS	
Compute Engine API		CPUs	All 99 quotas are within limit	ALL QUOTAS	
Compute Engine API		CPUs (all regions)	One quota is within limit	ALL QUOTAS	
Compute Engine API		Subnetworks	One quota is within limit	ALL QUOTAS	
Compute Engine API		Networks	One quota is within limit	ALL QUOTAS	
Compute Engine API		VM instances	All 25 quotas are within limit	ALL QUOTAS	
Compute Engine API		In-use IP addresses	All 25 quotas are within limit	ALL QUOTAS	
Compute Engine API		Persistent Disk Standard (GB)	All 99 quotas are within limit	ALL QUOTAS	
Compute Engine API		Subnetwork ranges per VPC Network	All 2 quotas are within limit	ALL QUOTAS	
Compute Engine API		Firewall rules	One quota is within limit	ALL QUOTAS	
Cloud Deployment Manager V2 API		Queries per minute	One quota is within limit	ALL QUOTAS	

Now, use the filter button, the thing with the three horizontal bars on the upper left corner of the screen, to filter the results. First, select ‘Limit Name’ and then select ‘GPUs (all regions)’.

Click on the ‘ALL QUOTAS’ button, and you should have a page that shows the Global Quota.

Add a short description, and submit your request. Select the navigation menu and on the navigation menu, select AI Platform -> Notebooks.

Once you have done this, select Create New Instance. You will get the option to either use a pre-built notebook that Google has made, or you can create a custom instance.

AI Platform

Create a notebook instance

Instance name \* gpu-test-1

Region \* us-west1 (Oregon)

Zone \* us-west1-b

Environment

However, different regions may have different GPUs available, so make sure that you select a region that contains the GPU that you want to use.

AI Platform (Unified)

Create a notebook instance

+ ADD VARIABLE

Machine configuration

Machine type \* n1-standard-4 (4 vCPUs, 15 GB RAM)

GPU type NVIDIA Tesla V100 Number of GPUs 1

\$1,369.97 monthly estimate

That's about \$1.877 hourly

Pay for what you use: No upfront costs and per second billing

Networking cost also applies. [Learn more](#)

DETAILS

Select Create Instance. You now have a working notebook to use! In order to start the instance, select the notebook you want to use and start.

AI Platform

Notebooks

NEW INSTANCE REFRESH START STOP RESET UPGRADE DELETE SHOW INFO PANEL

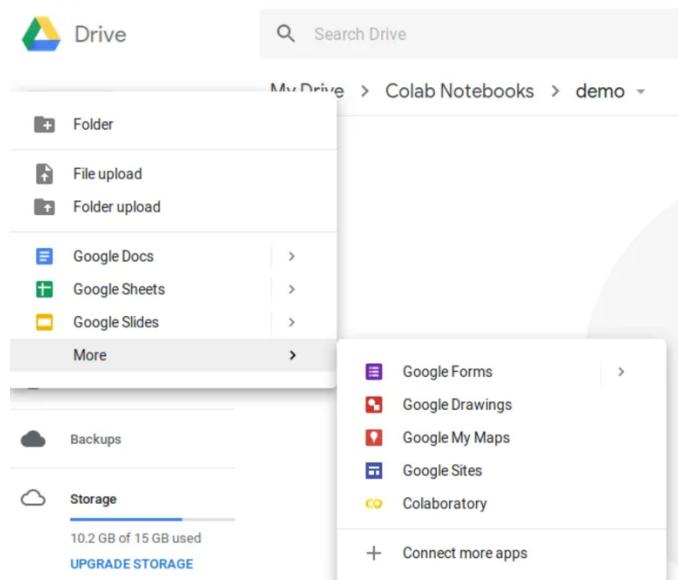
Instance name	Zone	Environment	Machine type	Gpus	Permission	Labels
gpu-test-1	us-west1-b	OPEN JUPYTERLAB	PyTorch:1.7	4 vCPUs, 15 GB RAM	NVIDIA Tesla V100 x 1	Service account

You can upload, download and save files, regularly. When you are done using the GPU, make sure to stop it, otherwise, you will continue being charged.

## EXP NO. 16

### PERFORM A STUDY ON GOOGLE GPU AND TPU OPTIONS

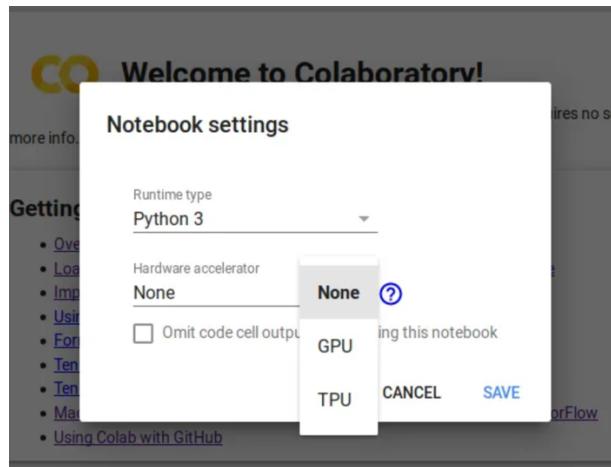
- Open GOOGLE DRIVE
- Create new folder
- Click on New -> More -> Colaboratory.



#### How to use GPU/ TPU?

Google Colab provides free GPU and TPU, but the default run-time type is CPU. To set it to GPU/TPU follow this steps:-

1. Click on **Runtime** from the top menu.
2. Select the **Change Runtime** option. It will open a small window like this -
3. Select your preferred Runtime type and click on **Save**.



Congratulations, you have created your first Colaboratory Notebook. I will be uploading new blogs regularly regarding how to use Colaboratory.

A GPU (graphics processing unit) is a specialized processor that works as a performance accelerator with the CPU.

TPUs stand for Tensor Processing Units, which are application-specific integrated circuits (ASICs). TPUs were designed from the ground up by Google; they started using TPUs in 2015 and made them public in 2018. TPUs are available as a cloud or smaller version of the chip.

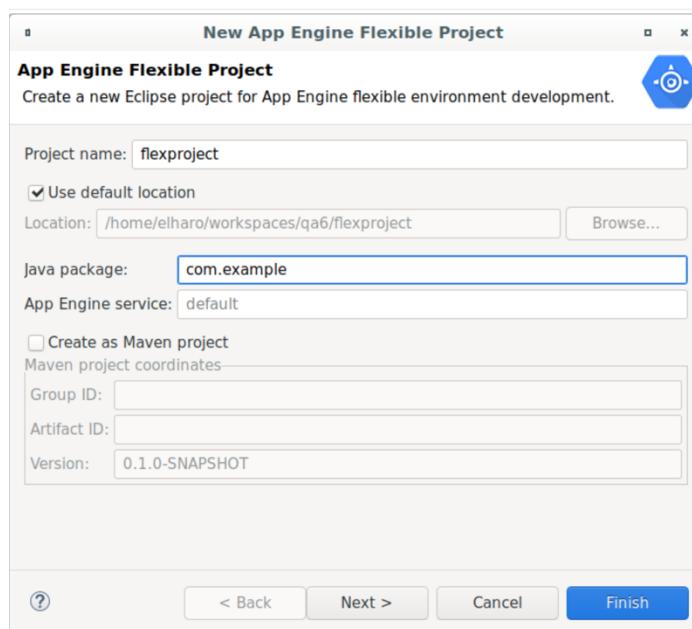
## EXP NO. 17

### CREATING AN APPLICATION IN THE APP ENGINE FLEXIBLE ENVIRONMENT

To create an application for the App Engine flexible environment:

1. Click the **Google Cloud Platform** toolbar button .
2. Select **Create New Project > Google App Engine Flexible Java Project....**

The **New App Engine Flexible Project** wizard displays.



1. Enter a **Project name** for your application.
2. If you want, enter values for the optional fields:
  - a. Enter a **Java package** name.
  - b. Select the **Create as Maven project** checkbox to generate a pom.xml and configure the Eclipse project to use the Maven builder.
  - c. Enter the **Group ID** and **Artifact ID** for your Maven project.
3. Click **Next**.
4. Select any libraries you need in the project.
5. Click **Finish**.

The new application appears in the **Eclipse Project Explorer**.

## EXP NO. 18

### DEPLOY AN APP TO A GKE CLUSTER- SIMPLE WEB SERVER APPLICATION

```
// [START
gke_hello_app]
    // [START container_hello_app]
    package main

    import (
        "fmt"
        "log"
        "net/http"
        "os"
    )

    func main() {
        // register hello function to handle all requests
        mux := http.NewServeMux()
        mux.HandleFunc("/", hello)

        // use PORT environment variable, or default to 8080
        port := os.Getenv("PORT")
        if port == "" {
            port = "8080"
        }

        // start the web server on port and accept requests
        log.Printf("Server listening on port %s", port)
        log.Fatal(http.ListenAndServe(": "+port, mux))
    }

    // hello responds to the request with a plain-text "Hello, world"
    // message.
    func hello(w http.ResponseWriter, r *http.Request) {
        log.Printf("Serving request: %s", r.URL.Path)
        host, _ := os.Hostname()
        fmt.Fprintf(w, "Hello, world!\n")
        fmt.Fprintf(w, "Version: 1.0.0\n")
        fmt.Fprintf(w, "Hostname: %s\n", host)
    }

    // [END container_hello_app]
    // [END gke_hello_app]
```

**Output:**

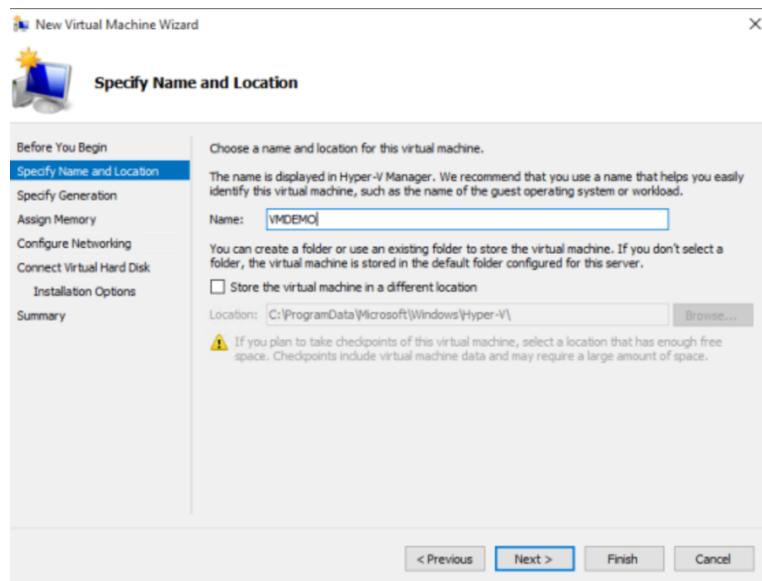
## EXP NO. 19

### CREATE A VIRTUAL MACHINE WITH HYPER-V MANAGER

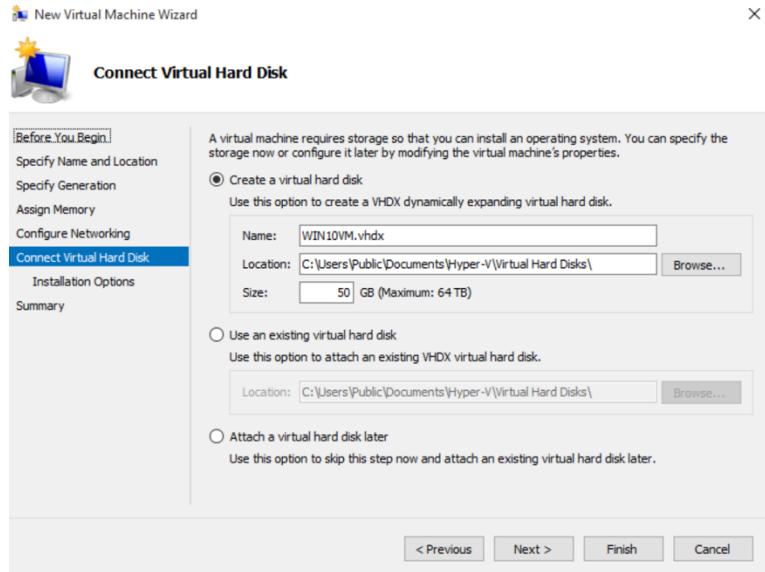
1. Open Hyper-V Manager by either pressing the Window's key and typing "Hyper-V Manager" or by finding Hyper-V Manager in your applications.
2. In Hyper-V Manager, click Action > New > Virtual Machine to bring up the New Virtual Machine Wizard.
3. Review the 'Before You Begin' content and click Next.
4. Give the virtual machine a name.

Note: This is the name Hyper-V uses for the virtual machine, not the computer name given to the guest operating system that will be deployed inside the virtual machine.

5. Choose a location where the virtual machine files will be stored such as c:\virtualmachine. You can also accept the default location. Click Next when done.



6. Select a generation for the machine and click Next.
7. Select 2048 MB for the Startup Memory value and leave Use Dynamic Memory selected. Click the Next button.
8. On the Configure Networking wizard, select a virtual switch for the virtual machine and click Next. For more information, see [Create a Virtual Switch](#).
9. Give the virtual hard drive a name, select a location or keep the default, and finally specify a size. Click Next when ready.



10. On the Installation Options wizard, select Install an operating system from a bootable image file and then select an operating system .iso file. Click Next once completed.
11. Review the virtual machine details and click Finish to complete the virtual machine creation.

## EXP NO. 20

### VIRTUAL MACHINE/ REGISTER VM IN C

```
# include <stdio.h>

# define NUM_REGS 4
unsigned regs[ NUM_REGS ];

unsigned program[] = { 0x1064, 0x11C8, 0x2201, 0x0000 };

/* program counter */
int pc = 0;

/* fetch the next word from the program */
int fetch()
{
    return program[ pc++ ];
}

/* instruction fields */
int instrNum = 0;
int reg1    = 0;
int reg2    = 0;
int reg3    = 0;
int imm     = 0;

/* decode a word */
void decode( int instr )
{
    instrNum = (instr & 0xF000) >> 12;
    reg1    = (instr & 0xF00 ) >> 8;
    reg2    = (instr & 0xF0 ) >> 4;
    reg3    = (instr & 0xF );
    imm     = (instr & 0xFF );
}

/* the VM runs until this flag becomes 0 */
int running = 1;

/* evaluate the last decoded instruction */
void eval()
{
    switch( instrNum )
    {
        case 0:
            /* halt */
            printf( "halt\n" );
    }
}
```

```

running = 0;
break;
case 1:
/* loadi */
printf( "loadi r%d #%d\n", reg1, imm );
regs[ reg1 ] = imm;
break;
case 2:
/* add */
printf( "add r%d r%d r%d\n", reg1, reg2, reg3 );
regs[ reg1 ] = regs[ reg2 ] + regs[ reg3 ];
break;
}

/*
 * display all registers as 4-digit hexadecimal words */
void showRegs()
{
int i;
printf( "regs = " );
for( i=0; i<NUM_REGS; i++ )
printf( "%04X ", regs[ i ] );
printf( "\n" );
}

void run()
{
while( running )
{
showRegs();
int instr = fetch();
decode( instr );
eval();
}
showRegs();
}

int main( int argc, const char * argv[] )
{
run();
return 0;
}

```

**Here is the output from the program:**

```

regs = 0000 0000 0000 0000
loadi r0 #100
regs = 0064 0000 0000 0000

```

```
loadi r1 #200
regs = 0064 00C8 0000 0000
add r2 r0 r1
regs = 0064 00C8 012C 0000
halt
regs = 0064 00C8 012C 0000
```