

Practical 1

To develop a program for multi-client chat server using Socket

```
// RPC
//ChatClient.java
import java.net.*;
import java.io.*;

public class ChatClient {
    Socket soc;

    BufferedReader br, br1;
    PrintWriter out;

    String str;

    public ChatClient() // constructor
    {
        try {
            soc = new Socket(InetAddress.getLocalHost(), 9999);

            br = new BufferedReader(new InputStreamReader(System.in));
            out = new PrintWriter(soc.getOutputStream(), true);
            System.out.println("Chat Client Started");

            while (true) {
                str = br.readLine();
                out.println(str);
                new InnerClient();
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    class InnerClient extends Thread // inside Chatclient class
    {
        String str1;

        InnerClient() {
            try {
```

```
        br1 = new BufferedReader(new InputStreamReader(soc.getInputStream()));
        start();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public void run() {
    try {
        while (true) {
            str1 = br1.readLine();
            System.out.println("Server says : " + str1);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
} // end of InnerClient class

public static void main(String args[]) // method of outer class ChatClient
{
    new ChatClient();
}
}
```

```
//RPC
//ChatServer.java

import java.net.*;
import java.io.*;

public class ChatServer extends Thread {
    ServerSocket ss;
    Socket soc;

    BufferedReader br, br1;
    PrintWriter out;

    String str;

    public ChatServer() {
        try {
            ss = new ServerSocket(9999);
            soc = ss.accept();
            br = new BufferedReader(new InputStreamReader(soc.getInputStream()));

            // InputStreamReader ir=new InputStreamReader(soc.getInputStream());
            // br = new BufferedReader(ir);

            System.out.println("Chat Server Started");
            start();
            new InnerServer();
        } catch (Exception e) {
            e.printStackTrace();
        }
    } // end of ChatServer()

    public void run() {
        try {
            while (true) // the data coming client is displayed
            {
                str = br.readLine();
                System.out.println("Client says : " + str);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    } // end of run()

    class InnerServer // inside ChatServer class
```

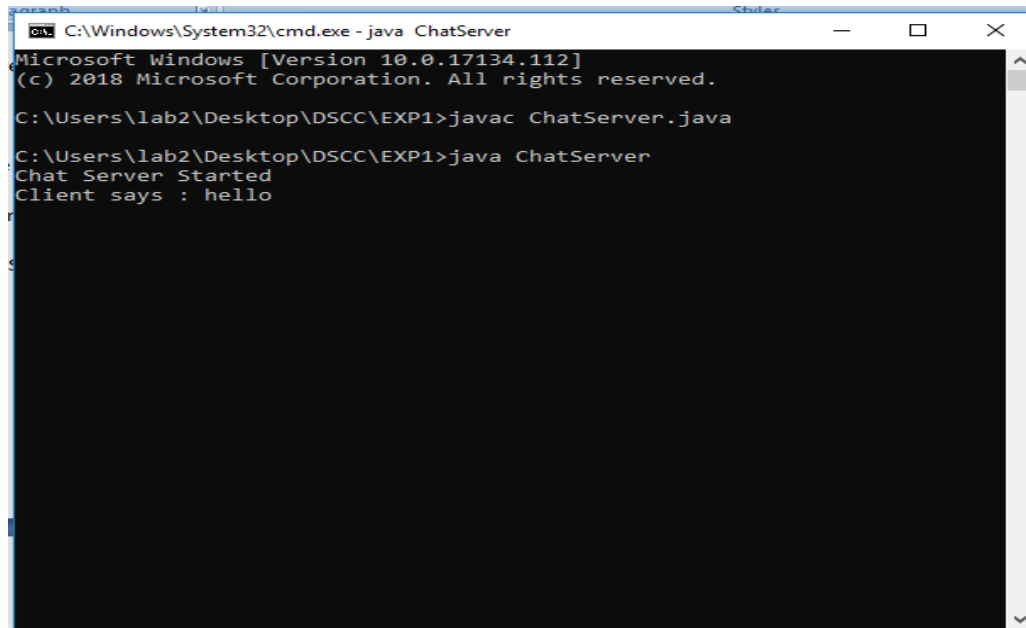
```
// the data from server will be read from console and sent to client.
{
    String str1;

    InnerServer() // constructor
    {
        try {
            br1 = new BufferedReader(new InputStreamReader(System.in));
            out = new PrintWriter(soc.getOutputStream(), true);
            while (true) // read the data from br1 and put in 'out'
            {
                // str1 = br1.readLine();
                // out.println(str1);

                out.println(br1.readLine());
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
} // end of InnerServer class

public static void main(String args[]) // method of chatServer class
{
    new ChatServer();
}
}
```

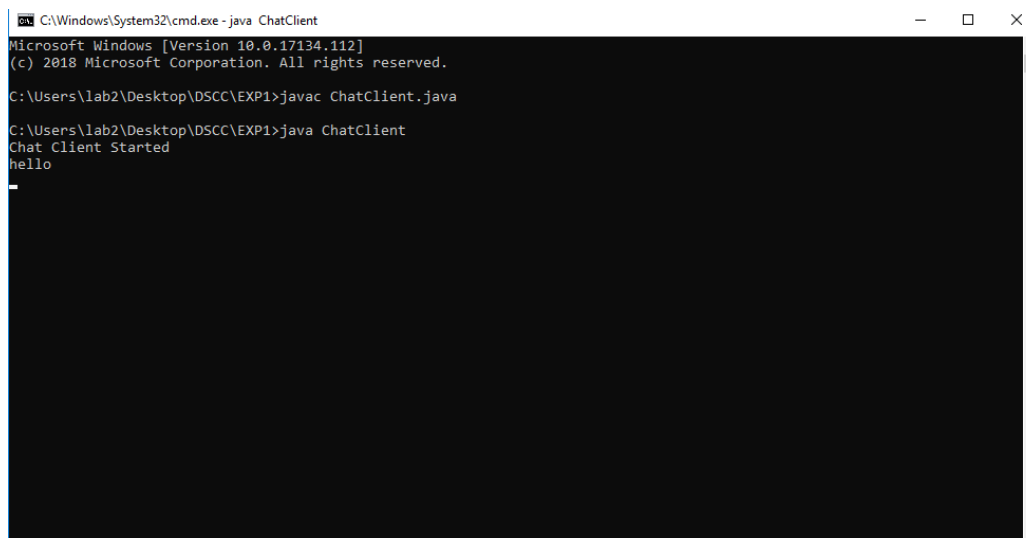
Output:-



```
C:\Windows\System32\cmd.exe - java ChatServer
Microsoft Windows [Version 10.0.17134.112]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\lab2\Desktop\DSCC\EXP1>javac ChatServer.java

C:\Users\lab2\Desktop\DSCC\EXP1>java ChatServer
Chat Server Started
Client says : hello
```



```
C:\Windows\System32\cmd.exe - java ChatClient
Microsoft Windows [Version 10.0.17134.112]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\lab2\Desktop\DSCC\EXP1>javac ChatClient.java

C:\Users\lab2\Desktop\DSCC\EXP1>java ChatClient
Chat Client Started
hello
```

Practical 2

To Implement a server calculator using RPC concept

```
//RPC Server
//To Implement a server calculator containing ADD(),MUL(),SUB()

import java.util.*;
import java.net.*;

class RPCServer {
    DatagramSocket ds;
    DatagramPacket dp;

    String str, methodName, result;

    int val1, val2;

    RPCServer() {
        try {
            ds = new DatagramSocket(1200);
            byte b[] = new byte[4096];

            // int a[]=new int[100];

            while (true) {
                dp = new DatagramPacket(b, b.length);
                ds.receive(dp);
                str = new String(dp.getData(), 0, dp.getLength()); // add 10 20

                if (str.equalsIgnoreCase("q")) {
                    System.exit(1);
                } else {
                    StringTokenizer st = new StringTokenizer(str, " ");
                    int i = 0;
                    while (st.hasMoreTokens()) {
                        // String token=st.nextToken();
                        methodName = st.nextToken(); // add
                        val1 = Integer.parseInt(st.nextToken()); // 10
                        val2 = Integer.parseInt(st.nextToken()); // 20
                    }
                }
            }
        }
    }
}
```

```
System.out.println(str); // print add 10 20
InetAddress ia = InetAddress.getLocalHost();

if (methodName.equalsIgnoreCase("add"))
    result = "" + add(val1, val2);
else if (methodName.equalsIgnoreCase("sub"))
    result = "" + sub(val1, val2);
else if (methodName.equalsIgnoreCase("mul"))
    result = "" + mul(val1, val2);
else if (methodName.equalsIgnoreCase("div"))
    result = "" + div(val1, val2);

byte b1[] = result.getBytes(); // convert string data into byte
DatagramSocket ds1 = new DatagramSocket();
DatagramPacket dp1 = new DatagramPacket(b1, b1.length,
InetAddress.getLocalHost(), 1300);
System.out.println("result : " + result + "\n");
ds1.send(dp1);
}
} catch (Exception e) {
    e.printStackTrace();
}
}

public int add(int val1, int val2) {
    return val1 + val2;
}

public int sub(int val3, int val4) {
    return val3 - val4;
}

public int mul(int val3, int val4) {
    return val3 * val4;
}

public int div(int val3, int val4) {
    return val3 / val4;
}

public static void main(String[] args) {
    new RPCServer();
}
}
```

```
//RPC Client
import java.io.*;
import java.net.*;

class RPCClient {
    RPCClient() {
        try {
            InetAddress ia = InetAddress.getLocalHost();

            DatagramSocket ds = new DatagramSocket();
            DatagramSocket ds1 = new DatagramSocket(1300);

            System.out.println("\nRPC Client\n");
            System.out.println("Enter method name and parameter like add 10 20\n");
            System.out.println("Press \'q\' to quit \n");

            while (true) {
                BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
                String str = br.readLine();
                byte b[] = str.getBytes();
                DatagramPacket dp = new DatagramPacket(b, b.length, ia, 1200);
                ds.send(dp);

                dp = new DatagramPacket(b, b.length);
                ds1.receive(dp);
                String s = new String(dp.getData(), 0, dp.getLength());
                System.out.println("\nResult = " + s + "\n");
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        new RPCClient();
    }
}
```


Output:-

```
C:\Windows\System32\cmd.exe - java RPCServer
Microsoft Windows [Version 10.0.17134.112]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\lab2\Desktop\DSCC\Expt 2>javac RPCServer.java

C:\Users\lab2\Desktop\DSCC\Expt 2>java ChatServer
Error: Could not find or load main class ChatServer

C:\Users\lab2\Desktop\DSCC\Expt 2>java RPCServer
add 10 20
result : 30
```

```
C:\Windows\System32\cmd.exe - java RPCClient
Microsoft Windows [Version 10.0.17134.112]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\lab2\Desktop\DSCC\Expt 2>javac RPCClient.java

C:\Users\lab2\Desktop\DSCC\Expt 2>java RPCClient
RPC Client
Enter method name and parameter like add 10 20
Press 'q' to quit
add 10 20
Result = 30
```

Practical 3

To implement a Date Time Server using RPC concept.(Make use of datagram)

//To Implement a Date time server containing date() and time() procedure.

```
import java.util.*;
import java.net.*;
import java.text.SimpleDateFormat;

class RPCServer1 {
    DatagramSocket ds;
    DatagramPacket dp;
    String str, methodName, result;
    int val1, val2;

    RPCServer1() {
        try {
            ds = new DatagramSocket(1200);
            byte b[] = new byte[4096];

            while (true) {
                dp = new DatagramPacket(b, b.length);
                ds.receive(dp);
                str = new String(dp.getData(), 0, dp.getLength());

                if (str.equalsIgnoreCase("q")) {
                    System.exit(1);
                } else {
                    StringTokenizer st = new StringTokenizer(str, " ");
                    int i = 0;

                    while (st.hasMoreTokens()) {
                        methodName = st.nextToken();
                    }
                }

                Calendar c = Calendar.getInstance();
                SimpleDateFormat dateFormat = new SimpleDateFormat("dd/MM/yyyy");

                Date d = c.getTime();
```

```
InetAddress ia = InetAddress.getLocalHost();

if (methodName.equalsIgnoreCase("date"))
    result = "" + dateFormat.format(d);
else if (methodName.equalsIgnoreCase("time"))
    result = "" + d.getHours() + " : " + d.getMinutes() + " : " +
d.getSeconds();

byte b1[] = result.getBytes();
DatagramSocket ds1 = new DatagramSocket();
DatagramPacket dp1 = new DatagramPacket(b1, b1.length,
InetAddress.getLocalHost(), 1300);
System.out.println("result : " + result + "\n");
ds1.send(dp1);
}
} catch (Exception e) {
    e.printStackTrace();
}
}

public static void main(String[] args) {
    new RPCServer1();
}
}
```

```
//RPC client1
import java.io.*;
import java.net.*;

class RPCClient1 {
    RPCClient1() {
        try {
            InetAddress ia = InetAddress.getLocalHost();
            DatagramSocket ds = new DatagramSocket();
            byte b1[] = new byte[50];
            DatagramSocket ds1 = new DatagramSocket(1300);
            System.out.println("\nRPC Client\n");

            while (true) {
                BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
                String str = br.readLine();
                byte b[] = str.getBytes();
                DatagramPacket dp = new DatagramPacket(b, b.length, ia, 1200);
                ds.send(dp);
                if (str.equalsIgnoreCase("q")) {
                    System.out.println("server exited...");
                    System.exit(1);
                }

                dp = new DatagramPacket(b1, b1.length);
                ds1.receive(dp);

                String s = new String(dp.getData(), 0, dp.getLength());
                System.out.println("\n Result = " + s);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        new RPCClient1();
    }
}
```

Output:-

```
C:\Windows\System32\cmd.exe - java RPCServer1
Microsoft Windows [Version 10.0.17134.112]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\lab2\Desktop\DSCC\Expt 3>
C:\Users\lab2\Desktop\DSCC\Expt 3>java RPCServer1
result : 30/07/2024

result : 15 : 6 : 52
```

```
C:\Windows\System32\cmd.exe - java RPCClient1
Microsoft Windows [Version 10.0.17134.112]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\lab2\Desktop\DSCC\Expt 3>javac RPCClient1.java
C:\Users\lab2\Desktop\DSCC\Expt 3>java RPCClient1

RPC Client

date

Result = 30/07/2024
time

Result = 15 : 6 : 52
```

Practical 4

To retrieve day, time and date function from server to client. This program should display server day, time and date.(use concept of JDBC and RMI for accessing multiple data access objects)

```
// DateClient.java

import java.rmi.*;

public class DateClient {
    public static void main(String[] args) {
        try {
            String url = "rmi://127.0.0.1/DServer";
            IDate intf = (IDate) Naming.lookup(url);

            // IDate intf=(IDate)Naming.lookup("rmi://127.0.0.1/DateServer");

            System.out.println("The Date On Server is: " + intf.getDate());
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
// DateServer.java
```

```
import java.rmi.*;
```

```
public class DateServer {  
    public static void main(String[] args) {  
        try {  
            DateImpl di = new DateImpl();  
            Naming.rebind("DServer", di);  
  
            System.out.println("Date Server is Ready");  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

```
// IDate.java
```

```
import java.rmi.*;
```

```
public interface IDate extends Remote {  
    String getDate() throws RemoteException;  
}
```

```
// DateImpl.java
```

```
import java.rmi.*;
```

```
import java.rmi.server.*;
```

```
import java.util.*;
```

```
public class DateImpl extends UnicastRemoteObject implements IDate {  
    public DateImpl() throws RemoteException {  
    }  
  
    public String getDate() {  
        Date d = new Date();  
        return (d.toString());  
    }  
}
```

Output:-

```
C:\Windows\System32\cmd.exe - rmiregistry
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\Users\lab2\Desktop\DSCC\Expt 4>set path="C:\Program Files\Java\jdk1.7.0_51\bin"
C:\Users\lab2\Desktop\DSCC\Expt 4>rmiregistry
```

```
C:\Windows\System32\cmd.exe - java DateServer
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\Users\lab2\Desktop\DSCC\Expt 4>set path="C:\Program Files\Java\jdk1.7.0_51\bin"
C:\Users\lab2\Desktop\DSCC\Expt 4>javac DateServer.java
C:\Users\lab2\Desktop\DSCC\Expt 4>java DateServer
Date Server is Ready
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\Users\lab2\Desktop\DSCC\Expt 4>set path="C:\Program Files\Java\jdk1.7.0_51\bin"
C:\Users\lab2\Desktop\DSCC\Expt 4>javac DateClient.java
C:\Users\lab2\Desktop\DSCC\Expt 4>java DateClient
The Date On Server is: Sat Aug 31 10:41:12 IST 2024
C:\Users\lab2\Desktop\DSCC\Expt 4>
```


Practical 5

The client should provide an equation to the server through an interface. The server will solve the expression given by the client.

```
// clientEqSolve.java
import java.io.*;
import java.net.*;
import java.rmi.*;

public class clientEqSolve {
    public static void main(String[] args) {
        try {
            int num1, num2, res = 0, choice;
            intfEqSolve object = (intfEqSolve) Naming.lookup("hello");

            BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));

            System.out.println("Equations:-");
            System.out.println("1.(a-b)²");
            System.out.println("2.(a+b)²");
            System.out.println("3.(a-b)³");
            System.out.println("4.(a+b)³");

            System.out.println("Choose the equation:");
            choice = Integer.parseInt(br.readLine());

            System.out.println("Enter the value of a and b");
            num1 = Integer.parseInt(br.readLine());
            num2 = Integer.parseInt(br.readLine());

            switch (choice) {
                case 1:
                    res = object.solveEq1(num1, num2);
                    break;
                case 2:
                    res = object.solveEq2(num1, num2);
                    break;
                case 3:
                    res = object.solveEq3(num1, num2);
```

```
        break;
    case 4:
        res = object.solveEq4(num1, num2);
        break;
    default:
        System.out.println("Invalid option");
        break;
    }
    System.out.println("the answer is" + res);
} catch (Exception e) {
    System.out.println("Exception:" + e);
}
}
```

```
// serverEqSolve.java
```

```
import java.io.*;
import java.net.*;
import java.rmi.*;

public class serverEqSolve {
    public static void main(String[] args) {
        try {
            implEqSolve obj = new implEqSolve();
            Naming.rebind("hello", obj);
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

```
// intfEqSolve.java
```

```
import java.rmi.*;

public interface intfEqSolve extends Remote {
    public int solveEq1(int a, int b) throws RemoteException;

    public int solveEq2(int a, int b) throws RemoteException;

    public int solveEq3(int a, int b) throws RemoteException;

    public int solveEq4(int a, int b) throws RemoteException;
}
```

```
// implEqSolve.java
import java.rmi.*;
import java.rmi.server.*;

public class implEqSolve extends UnicastRemoteObject implements intfEqSolve {
    public implEqSolve() throws RemoteException {
    }

    public int solveEq1(int a, int b) throws RemoteException {
        int ans = (a * a) - (2 * a * b) + (b * b);
        return ans;
    }

    public int solveEq2(int a, int b) throws RemoteException {
        int ans = (a * a) + (2 * a * b) + (b * b);
        return ans;
    }

    public int solveEq3(int a, int b) throws RemoteException {
        int ans = (a * a * a) - (3 * a * a * b) + (3 * a * b * b) - (b * b * b);
        return ans;
    }

    public int solveEq4(int a, int b) throws RemoteException {
        int ans = (a * a * a) + (3 * a * a * b) + (3 * a * b * b) + (b * b * b);
        return ans;
    }
}
```

Output:-

```
C:\Windows\System32\cmd.exe - rmiregistry
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\Users\lab2\Desktop\DSCC\Expt 5>rmiregistry
```

```
C:\Windows\System32\cmd.exe - java serverEqSolve
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\Users\lab2\Desktop\DSCC\Expt 5>javac serverEqSolve.java
C:\Users\lab2\Desktop\DSCC\Expt 5>java serverEqSolve
```

```
C:\Users\lab2\Desktop\DSCC\Expt 5>javac clientEqSolve.java
C:\Users\lab2\Desktop\DSCC\Expt 5>java clientEqSolve
Equations:-
1.(a-b)2
2.(a+b)2
3.(a-b)3
4.(a+b)3
Choose the equation:
2
Enter the value of a and b
4
5
the answer is81
C:\Users\lab2\Desktop\DSCC\Expt 5>
```

Practical 6

Using MySQL create Library database. Create table Book(Book_id,Book_name,Book_author)and retrieve the book information from Library database using Remote Object Communication concept.

//DBClient.java for Practical 6 (Using Only one .accdb file)(same for mysql odbc connection)

```
import java.rmi.*;
import java.io.*;

public class DBClient_P6 {
    public static void main(String[] args) {
        String sql = "", res = "";
        try
        {
            BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
            System.out.println("\n*** Books Table ***");
            sql = "select * from Book";
            String url = "rmi://127.0.0.1/DBServer";
            IDb id = (IDb) Naming.lookup(url);
            res = id.getData(sql, "LibraryDB");
            System.out.println("\n-----");
            System.out.print(res);
            System.out.println("-----");
        } // end of try
        catch (Exception e)
        {
            e.printStackTrace();
        }
    }
}
```

```
//DBServer.java
```

```
import java.rmi.*;
```

```
public class DBServer {  
    public static void main(String[] args) {  
        try {  
            DBImpl di = new DBImpl();  
            Naming.rebind("rmi://127.0.0.1/DBServer", di);  
            System.out.println("Server Registered.");  
        } catch (Exception e1) {  
            e1.printStackTrace();  
        }  
    }  
}
```

```
//IDb.java
```

```
import java.rmi.*;
```

```
public interface IDb extends Remote {  
    public String getData(String s, String db) throws RemoteException;  
}
```

```
//DBImpl.java

import java.rmi.*;
import java.rmi.server.*;
import java.sql.*;

public class DBImpl extends UnicastRemoteObject implements IDb {
    String str, str1;

    public DBImpl() throws RemoteException {
    }

    public String getData(String sql, String dsn) {
        String URL = "jdbc:odbc:" + dsn;
        try {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            Connection con = DriverManager.getConnection(URL);
            Statement s = con.createStatement();
            ResultSet rs = s.executeQuery(sql);
            ResultSetMetaData rsmd = rs.getMetaData();
            str = "";
            str1 = "";

            for (int i = 1; i <= rsmd.getColumnCount(); i++) {
                str1 = str1 + rsmd.getColumnName(i) + "\t";
            }
            System.out.println();

            while (rs.next()) {
                for (int i = 1; i <= rsmd.getColumnCount(); i++) {
                    str = str + rs.getString(i) + "\t";
                }
                str = str + "\n";
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return (str1 + "\n" + str);
    }
}
```

Output:-

```
C:\Windows\System32\cmd.exe - rmiregistry
Microsoft Windows [Version 10.0.19045.4651]
(c) Microsoft Corporation. All rights reserved.

C:\Users\lab2\Desktop\DSCC\Expt 6 and 7 [Final]\Practical 6>set path="C:\Program Files\Java\jdk1.7.0_51\bin"
C:\Users\lab2\Desktop\DSCC\Expt 6 and 7 [Final]\Practical 6>rmiregistry
```

```
C:\Windows\System32\cmd.exe - java DBServer
Microsoft Windows [Version 10.0.19045.4651]
(c) Microsoft Corporation. All rights reserved.

C:\Users\lab2\Desktop\DSCC\Expt 6 and 7 [Final]\Practical 6>set path="C:\Program Files\Java\jdk1.7.0_51\bin"
C:\Users\lab2\Desktop\DSCC\Expt 6 and 7 [Final]\Practical 6>javac DBServer.java
C:\Users\lab2\Desktop\DSCC\Expt 6 and 7 [Final]\Practical 6>java DBServer
Server Registered.
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.4651]
(c) Microsoft Corporation. All rights reserved.

C:\Users\lab2\Desktop\DSCC\Expt 6 and 7 [Final]\Practical 6>javac DBClient_P6.java
C:\Users\lab2\Desktop\DSCC\Expt 6 and 7 [Final]\Practical 6>java DBClient_P6

*** Books Table ***

-----
BookID  Book_Name      Author  Price  No_Of_Copies
1       C++      Balagurusamy  200    20
2       Java     Herbert      400    30
3       Let us C   Kanetkar    300    10
-----

C:\Users\lab2\Desktop\DSCC\Expt 6 and 7 [Final]\Practical 6>
```


Practical 7

Using MySQL create Elecrtic_Bill database. Create table Bill (consumer_name, bill_due_date, bill_amount) and retrieve the Bill information from the Elecrtic_Bill database using Remote Object Communication concept.

//DBClient.java for Practical 7 (Using Only one .acddb file)(same for mysql odbc connection)

```
import java.rmi.*;
import java.io.*;
public class DBClient_P7 {
    public static void main(String[] args) {
        String sql = "", res = "";
        try
        {
            BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
            System.out.println("\n*** Bill Table ***");
            sql = "select * from Bill";
            String url = "rmi://127.0.0.1/DBServer";
            IDb id = (IDb) Naming.lookup(url);
            res = id.getData(sql, "MtnlDB");
            System.out.println("\n-----");
            System.out.print(res);
            System.out.println("-----");
        } // end of try
        catch (Exception e)
        {
            e.printStackTrace();
        }
    }
}
```

// DBServer.java

```
import java.rmi.*;
public class DBServer {
    public static void main(String[] args) {
```

```
    try {  
        DBImpl di = new DBImpl();  
        Naming.rebind("rmi://127.0.0.1/DBServer", di);  
        System.out.println("Server Registered.");  
    } catch (Exception e1) {  
        e1.printStackTrace();  
    }  
}  
}
```

//IDb.java

```
import java.rmi.*;  
  
public interface IDb extends Remote {  
    public String getData(String s, String db) throws RemoteException;  
}
```

```
//DBImpl.java
```

```
import java.rmi.*;
import java.rmi.server.*;
import java.sql.*;

public class DBImpl extends UnicastRemoteObject implements IDb {
    String str, str1;

    public DBImpl() throws RemoteException {
    }

    public String getData(String sql, String dsn) {
        String URL = "jdbc:odbc:" + dsn;
        try {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            Connection con = DriverManager.getConnection(URL);
            Statement s = con.createStatement();
            ResultSet rs = s.executeQuery(sql);
            ResultSetMetaData rsmd = rs.getMetaData();
            str = "";
            str1 = "";

            for (int i = 1; i <= rsmd.getColumnCount(); i++) {
                str1 = str1 + rsmd.getColumnName(i) + "\t";
            }
            System.out.println();

            while (rs.next()) {
                for (int i = 1; i <= rsmd.getColumnCount(); i++) {
                    str = str + rs.getString(i) + "\t";
                }
                str = str + "\n";
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return (str1 + "\n" + str);
    }
}
```

Output:-

```
cmd C:\Windows\System32\cmd.exe - rmiregistry
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\Users\lab2\Desktop\DSCC\Expt 6 and 7 [Final]\Practical 7>set path="C:\Program Files\Java\jdk1.7.0_51\bin"

C:\Users\lab2\Desktop\DSCC\Expt 6 and 7 [Final]\Practical 7>rmiregistry
```

```
cmd C:\Windows\System32\cmd.exe - java DBServer
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\Users\lab2\Desktop\DSCC\Expt 6 and 7 [Final]\Practical 7>set path="C:\Program Files\Java\jdk1.7.0_51\bin"

C:\Users\lab2\Desktop\DSCC\Expt 6 and 7 [Final]\Practical 7>javac DBServer.java

C:\Users\lab2\Desktop\DSCC\Expt 6 and 7 [Final]\Practical 7>java DBServer
Server Registered.
```

```
cmd C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\Users\lab2\Desktop\DSCC\Expt 6 and 7 [Final]\Practical 7>set path="C:\Program Files\Java\jdk1.7.0_51\bin"

C:\Users\lab2\Desktop\DSCC\Expt 6 and 7 [Final]\Practical 7>javac DBClient_P7.java

C:\Users\lab2\Desktop\DSCC\Expt 6 and 7 [Final]\Practical 7>java DBClient_P7

*** Bill Table ***

-----
CustID  Customer_Name  Bill_Due_Date  Bill_Amount
1       Kirtee      2022-09-21     200
2       Komal       2022-09-22     400
3       Poonam      2022-09-23     300
-----

C:\Users\lab2\Desktop\DSCC\Expt 6 and 7 [Final]\Practical 7>_
```

Practical 8

Implementation of mutual exclusion using Token ring algorithm

```
//TokenServer
import java.net.*;
import java.io.*;

class TokenServer {
    public static DatagramSocket ds;
    public static DatagramPacket dp;

    public static void main(String[] args) throws Exception {
        try {
            ds = new DatagramSocket(1000); // ds is the object of DatagramSocket at
            port 1000
            while (true) {
                byte buff[] = new byte[1024];
                // data coming from client is in byte form. This data is stored in buff
                array
                // this data is received at server through ds
                dp = new DatagramPacket(buff, buff.length);
                ds.receive(dp);

                // data in byte form is converted in string form
                String str = new String(dp.getData(), 0, dp.getLength());

                // the data received from client is display at server
                System.out.println("Message from " + str);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
//TokenClient1
import java.net.*;
import java.io.*;

class TokenClient1 {
    public static DatagramSocket ds;
    public static DatagramPacket dp;
    public static BufferedReader br;

    static int cp = 100;

    public static void main(String[] args) throws Exception {
        boolean hasToken;
        try {
            ds = new DatagramSocket(100);
        } catch (Exception e) {
            e.printStackTrace();
        }

        hasToken = true;

        while (true) {
            if (hasToken == true) {
                System.out.println("Do you want to enter data...(yes/no):");

                br = new BufferedReader(new InputStreamReader(System.in));
                String ans = br.readLine();

                if (ans.equalsIgnoreCase("yes")) {
                    System.out.println("ready to send.....");
                    System.out.println("sending.....");
                    System.out.println("Enter the data          ");

                    br = new BufferedReader(new InputStreamReader(System.in));
                    String str = "Client-1==> " + br.readLine();

                    byte buff[] = new byte[1024];
                    buff = str.getBytes();

                    ds.send(new DatagramPacket(buff, buff.length,
InetAddress.getLocalHost(),
                        1000));
                    System.out.println("now sending.....");
                } else if (ans.equalsIgnoreCase("no")) {
```

```
        System.out.println("I am in busy state now.....");

        // sending msg to client-2
        String msg = "Token";
        byte bf1[] = new byte[1024];
        bf1 = msg.getBytes();

        ds.send(new DatagramPacket(bf1, bf1.length,
                                   InetAddress.getLocalHost(), 200));
        hasToken = false;

        // receiving msg from client-2
        byte bf2[] = new byte[1024];
        ds.receive(dp = new DatagramPacket(bf2, bf2.length));

        String clientmsg = new String(dp.getData(), 0, dp.getLength());

        System.out.println("The data passed from client2 is " + clientmsg);

        if (clientmsg.equals("Token"))
            hasToken = true;

        System.out.println("I am leaving busy state");
    }
} else {
    System.out.println("Entering in receive mode.");

    byte bf[] = new byte[1024];
    ds.receive(dp = new DatagramPacket(bf, bf.length));

    String clientmsg1 = new String(dp.getData(), 0, dp.getLength());

    System.out.println("The data passed from Client1 is " + clientmsg1);

    if (clientmsg1.equals("Token"))
    {
        ;
        {
            hasToken = true;
        }
    }
}
}
```

```
//Token client 2:
import java.net.*;
import java.io.*;

class TokenClient2 {
    static DatagramSocket ds;
    static DatagramPacket dp;
    static BufferedReader br;

    public static void main(String[] args) throws Exception {
        try {
            ds = new DatagramSocket(2000);
        } catch (Exception e) {
            e.printStackTrace();
        }
        boolean hasToken = true;
        while (true) {
            // System.out.println("Entering if");
            if (hasToken == true) {
                System.out.println("Do you want to enter data(Yes/No):");
                br = new BufferedReader(new InputStreamReader(System.in));
                String str = br.readLine();
                if (str.equalsIgnoreCase("yes")) {
                    System.out.println("Enter Data; ");
                    br = new BufferedReader(new InputStreamReader(System.in));
                    String msg = "Client-2==>" + br.readLine();
                    byte bf1[] = new byte[1024];
                    bf1 = msg.getBytes();
                    ds.send(new DatagramPacket(bf1, bf1.length, InetAddress.getLocalHost(),
1000));
                    System.out.println("Data sent");
                } else {
                    // send to client 1.
                    String clientmsg = "Token";
                    byte bf2[] = new byte[1024];
                    bf2 = clientmsg.getBytes();
                    ds.send(new DatagramPacket(bf2, bf2.length, InetAddress.getLocalHost(),
100));
                    hasToken = false;
                }
            } else {
                try {
                    byte buff[] = new byte[1024];
                    System.out.println("Entering in waiting/receiving mode.");
                }
            }
        }
    }
}
```



```
        ds.receive(dp = new DatagramPacket(buff, buff.length));
        String clientmsg1 = new String(dp.getData(), 0, dp.getLength());
        System.out.println("The data is " + clientmsg1);
        if (clientmsg1.equals("Token"))
            hasToken = true;
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}
```

Output:-

```
C:\Windows\System32\cmd.exe - java TokenServer
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\Users\lab2\Desktop\DSCC\Expt 8>javac TokenServer.java

C:\Users\lab2\Desktop\DSCC\Expt 8>java TokenServer
Message from Client-1==> Hello
Message from Client-2==>hii
```

```
C:\Windows\System32\cmd.exe - java TokenClient1
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\Users\lab2\Desktop\DSCC\Expt 8>javac TokenClient1.java

C:\Users\lab2\Desktop\DSCC\Expt 8>java TokenClient1
Do you want to enter data...(yes/no):
yes
ready to send???.
sending?????..
Enter the data
Hello
now sending???.
Do you want to enter data...(yes/no):
no
I am in busy state now??..
```

```
C:\Windows\System32\cmd.exe - java TokenClient2
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\Users\lab2\Desktop\DSCC\Expt 8>javac TokenClient2.java

C:\Users\lab2\Desktop\DSCC\Expt 8>java TokenClient2
Do you want to enter data(Yes/No):
yes
Enter Data;
hii
Data sent
Do you want to enter data(Yes/No):
```

Practical 9

Implementation of Storage as a Service using Google Docs.

Theory:

Storage as a service (STaaS) Storage as a service (STaaS) is a cloud business model in which a company leases or rents its storage infrastructure to another company or individuals to store data. Small companies and individuals often find this to be a convenient methodology for managing backups, and providing cost savings in personnel, hardware and physical space. As an alternative to storing magnetic tapes offsite in a vault, IT administrators are meeting their storage and backup needs by service level agreements (SLAs) with an SaaS provider, usually on a cost-per-gigabyte-stored and cost-per-data-transferred basis. The client transfers the data meant for storage to the service provider on a set schedule over the SaaS provider's wide area network or over the Internet

Advantages of STaaS

1. Storage costs Personnel, hardware and physical storage space expenses are reduced.
2. Disaster recovery Having multiple copies of data stored in different locations can better enable disaster recovery measures.
3. Scalability With most public cloud services, users only pay for the resources that they use.
4. Syncing Files can be automatically synced across multiple devices.
5. Security can be both an advantage and a disadvantage, as security methods may change per vendor. Data tends to be encrypted during transmission and while at rest.

Disadvantages of STaaS

1. Security Users may end up transferring business-sensitive or mission-critical data to the cloud, which makes it important to choose a service provider that's reliable.
2. Potential storage costs If bandwidth limitations are exceeded, these could be expensive.
3. Potential downtimes Vendors may go through periods of downtime where the service is not available, which can be trouble for mission-critical data.
4. Limited customization Since the cloud infrastructure is owned and managed by the service provider, it is less customizable.
5. Potential for vendor lock-in It may be difficult to migrate from one service to another

Software As Service using Google Drive

Agenda

- Introduction to Google Drive
- Share with Google Drive
- My Drive
- Shared Google Drives
- Google Drive tools and options

Introduction to Google Drive

Google Drive is Google Software as a Service as part of productivity suite Google Drive helps you sharing files safely with your peers, collaborate real-time and organize your files quickly and easily.

With Google Drive you can:

- Create, add, or upload a file with a single button.
- Find and add files shared with you more easily.
- Drag-and-drop files and folders just like you do on your desktop.
- Access your files from any of your devices.

Common Use Cases for Google Drive

- Access and edit your work files anywhere
- Share files with a team or department
- Store and search all of your company's files
- Collaborate with co-workers and partners in real-time

My Drive

- My drive represents content that you own & content that you have permission to access
- When content is created in my drive the creator owns the content

Owners can

- Move files and folders
- Add collaborators with edit comment or view permission
- Share a link to a file or folder d
- Delete content and restore it from the trash

Share Google Drive

- Share drives are shared spaces in Google drive for sharing content with a team instead
- The content in a shared drive is owned by the team
- As people join and leave a team the content remains accessible in the shared drive
- Share drives have more collaborator roles and more options for granular access to files and folders

Common Use Cases for Shared Drive

- Project file repositories
- Employee onboarding
- Training resources
- Scenarios where the entire company needs access to the same file

Share Google Drive - Permissions

Shared drive permissions are more granular than the options available when sharing files or folders from my drive

Permissions for Shared Driver

- Manager: have full access to the shared drive and can change its settings along with managing members and deletion of content
- Content managers can perform most file level tasks except permanently delete content
- Contributors can create have permission to create new files and folders but not organize content
- Commenters can only provide comments
- Viewers can only view content

Each shared drive has its own trash and content is permanently removed from the trash after 30 days to restore a deleted file or folder

Google Drive tools and options

Google drive offers several additional tools add-ons and options that can be enabled to help meet your needs

Offline viewing drive: it's possible to work in Google drive without an active internet connection to make your Google docs sheets and slides available offline on your device. This is done by installing a Google drive offline extension to the browser. All the selected files can available for offline and any changes made to the file while offline will be sync when the internet is restored.

Google drive file stream: allows you to work from your desktop with files stored in the cloud the files are available on demand which means they don't require space on your device's storage nor do they require an enormous amount of bandwidth or download time this feature allows you to choose which files and folders are stored for offline access there are drive file stream applications for Microsoft windows and mac os. Drive file stream also supports real-time presence when editing Microsoft office files

Google workspace marketplace: an online store for applications that work with Google workspace for example Google docs doesn't have a built-in mail merge feature but several apps in the Google workspace marketplace can enable mail merge in Google docs there are add-ons related to productivity communication accounting legal marketing task management and many more

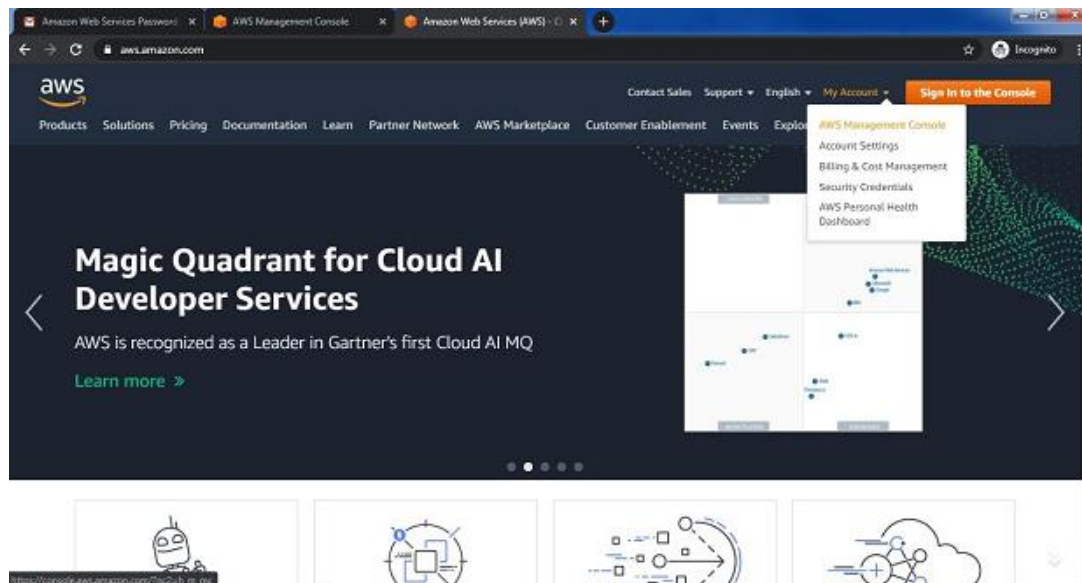
Practical 10

Aim: Study and implementation of Identity Management

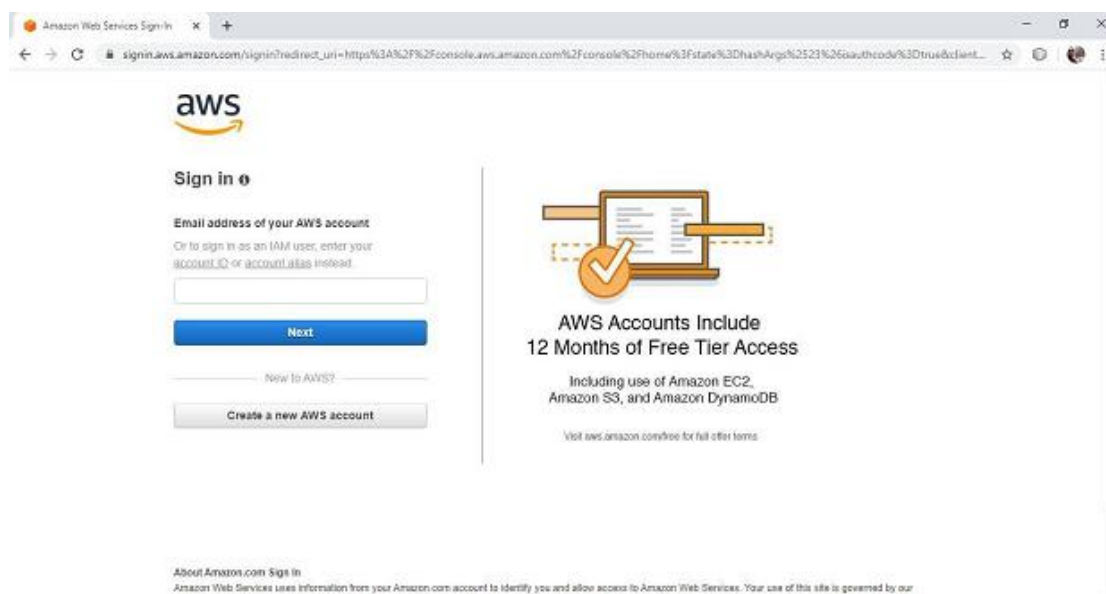
Procedure:

Step1: Open the following link <https://aws.amazon.com/>

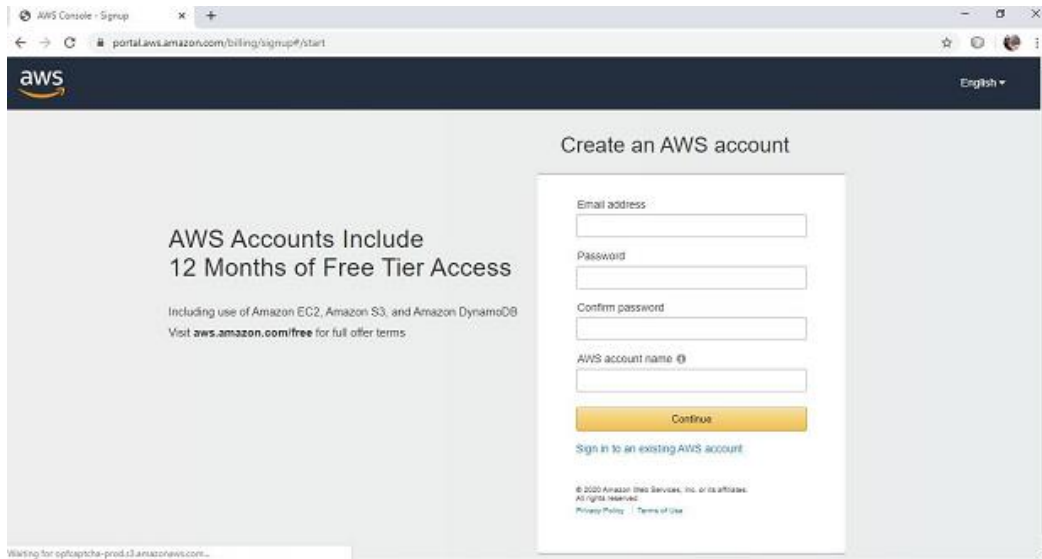
Step2: Go to my Account-> AWS management console



Step3: click on Create new user AWS account



Step4: Fill all the details and click on Continue



Step5: Fill your contact number and Home address and click on ***create account and continue***

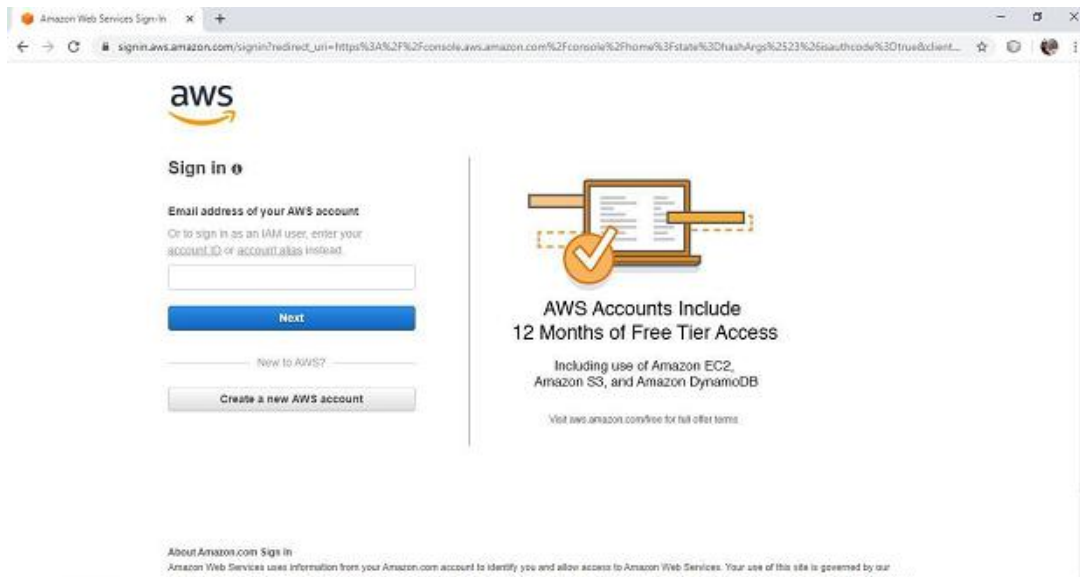
Step6: Now most curtail step AWS will ask for credit card and debit card details.

You have to close the browser

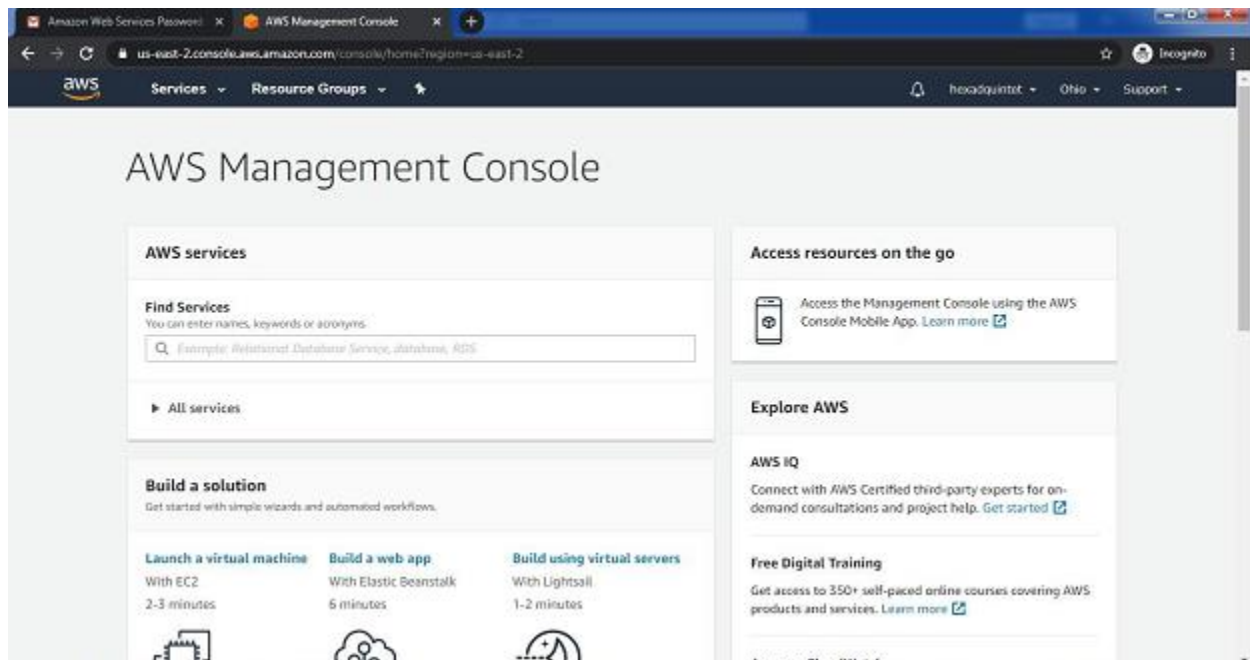
Step7: now again open the link <https://aws.amazon.com/>

Step8: Go to my Account->AWS Management console

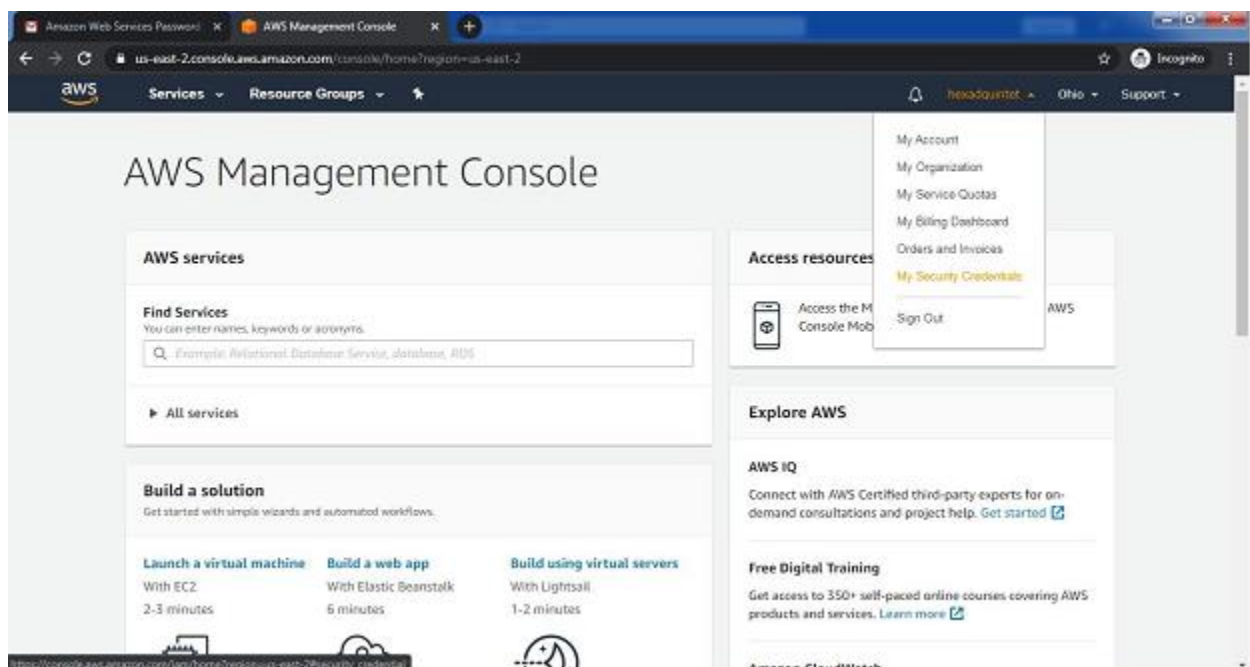
Enter your ID and click on next, After that enter password and click on sign in



Step 9: you will get the following screen



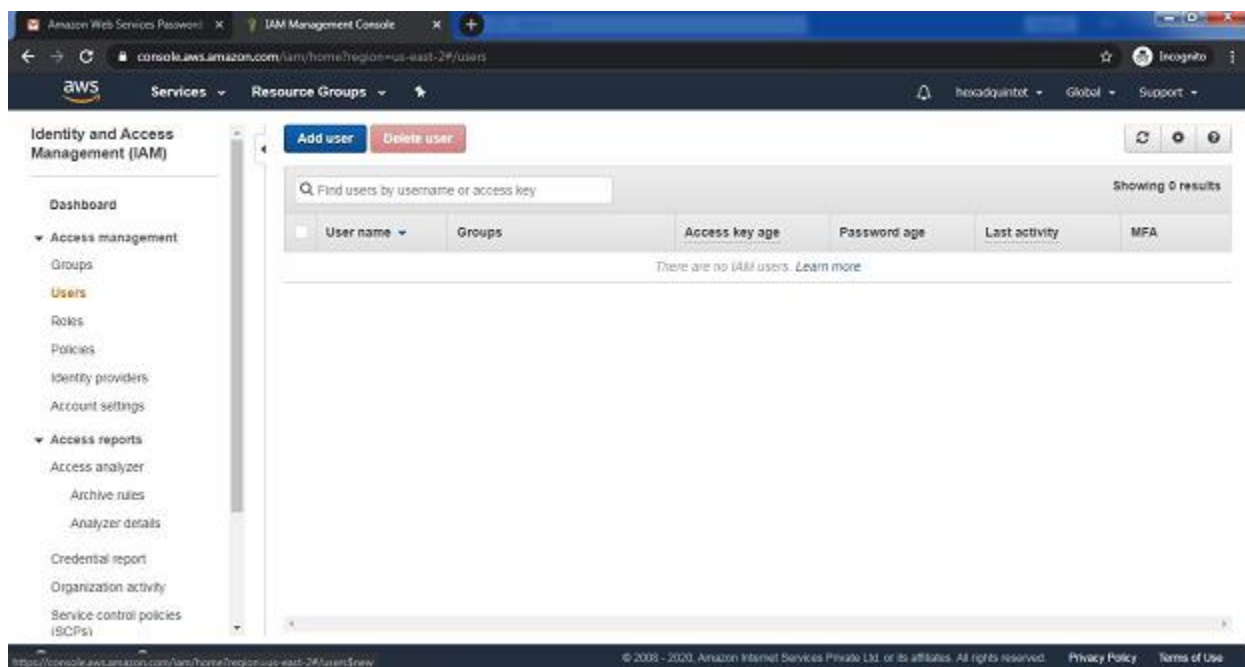
Step 10: Go to My Security credential



Step 11: now click on user



Step 12: Click on add user



Step 13: Provide the user name and check the check box in front of **programmatic access** and **AWS Management console Access** and enter the password for new user Click on **custom password** and click on next permission

User name* Admin

+ Add another user

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type*

- ☒ Programmatic access
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.
- ☒ AWS Management Console access
Enables a **password** that allows users to sign-in to the AWS Management Console.

Console password*

- ☐ Autogenerated password
- ☒ Custom password

☐ Show password

Require password reset* ☒ User must create a new password at next sign-in
Users automatically get the `IAMUserChangePassword` policy to allow them to change

* Required

Cancel Next: Permissions

Feedback English (US) © 2009 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Step 14: click on create Group

ADD USER

1 2 3 4 5

Set permissions

- Add user to group
- Copy permissions from existing user
- Attach existing policies directly

Get started with groups

You haven't created any groups yet. Using groups is a best-practice way to manage users' permissions by job functions, AWS service access, or your custom permissions. Get started by creating a group. [Learn more](#)

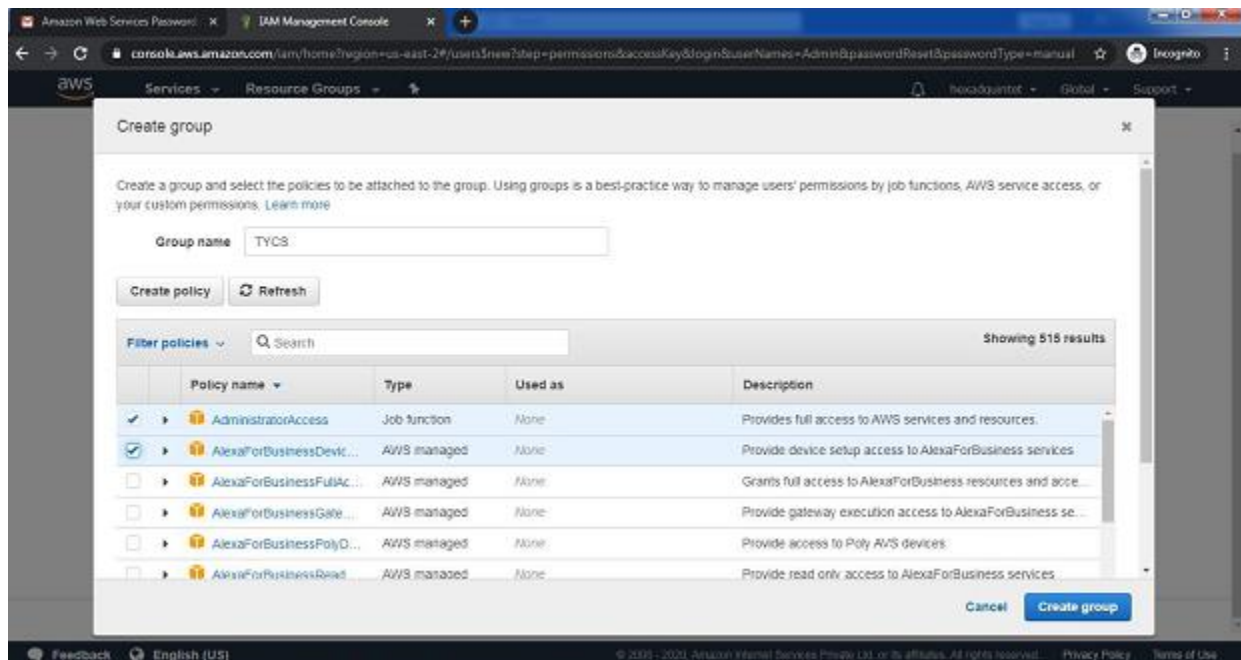
Create group

Set permissions boundary

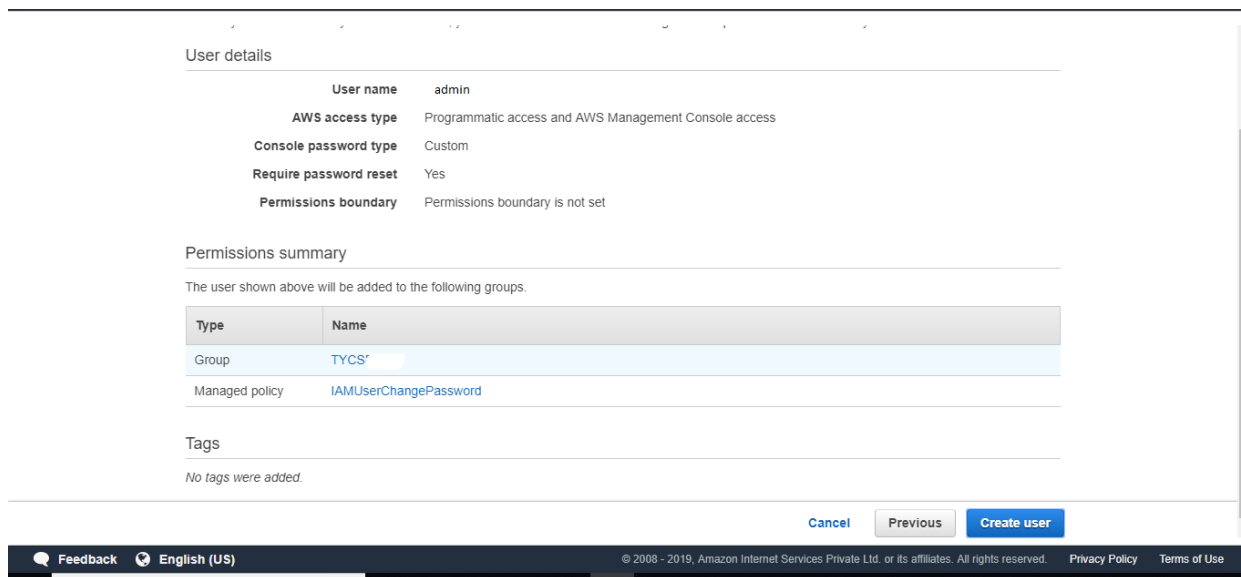
Cancel Previous Next: Tags

Feedback English (US) © 2009 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

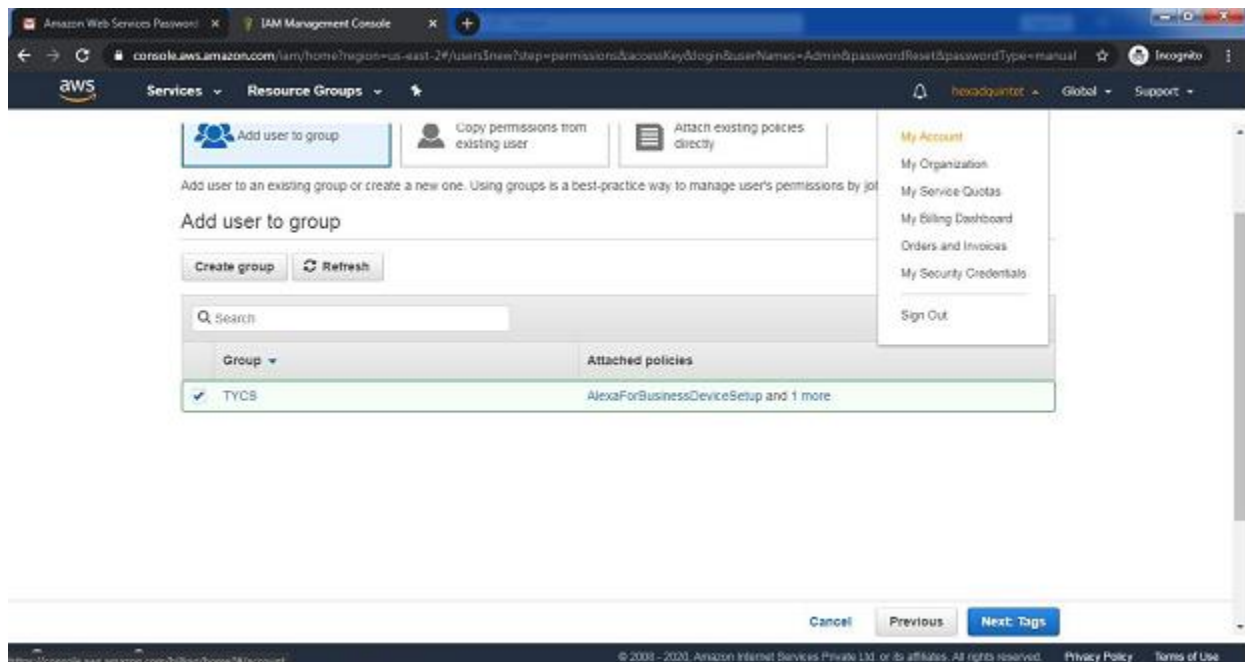
Step15:fill the information and click on Create Group



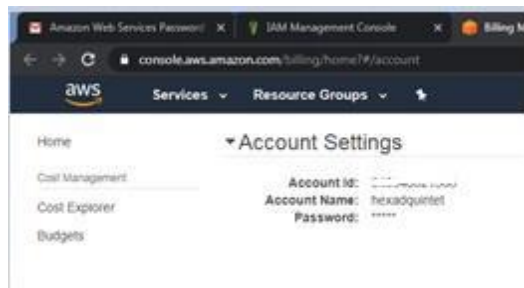
Step16:click on next tag leave blank , again click on next review leave as it is and click on create user



Step 17: click on close

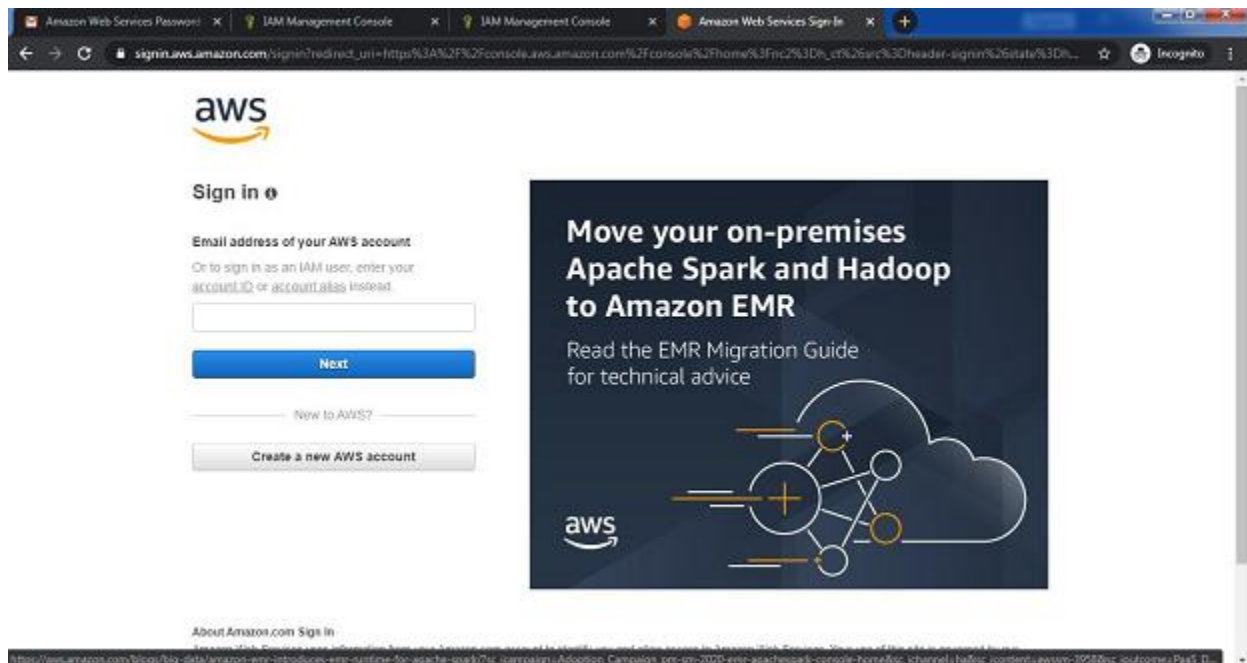


And COPY Account ID



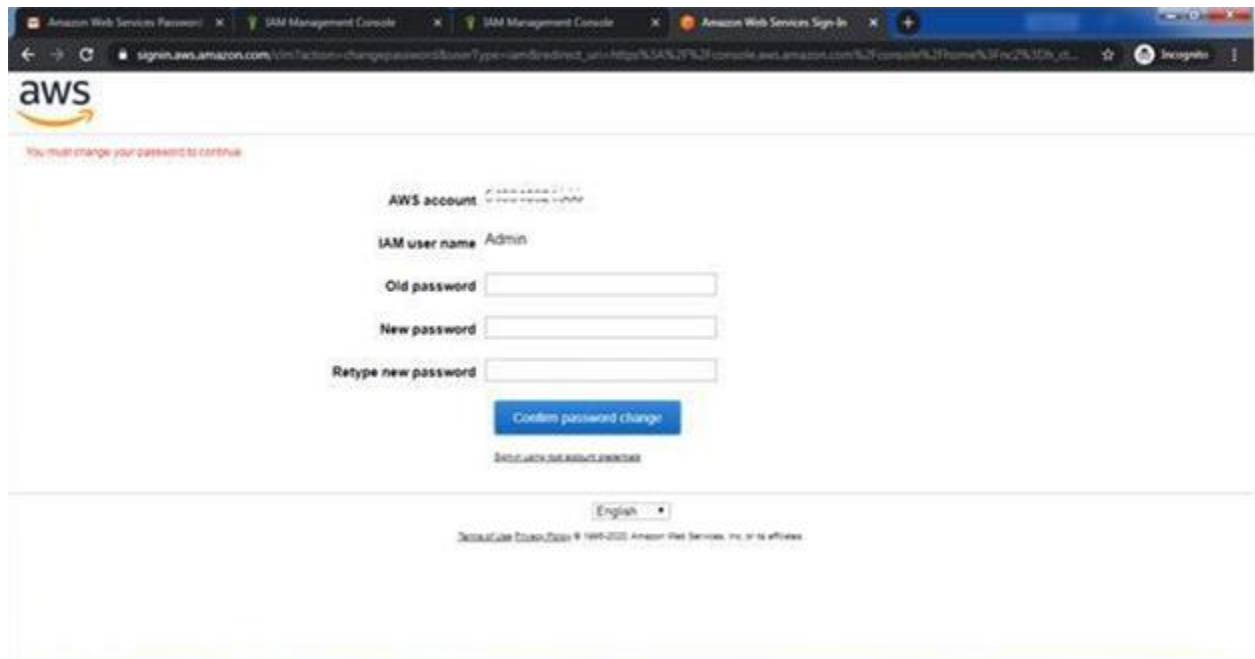
Now logout the admin account and try to login as user(newly created) .

Step 18: again Go to my Account->AWS Management console



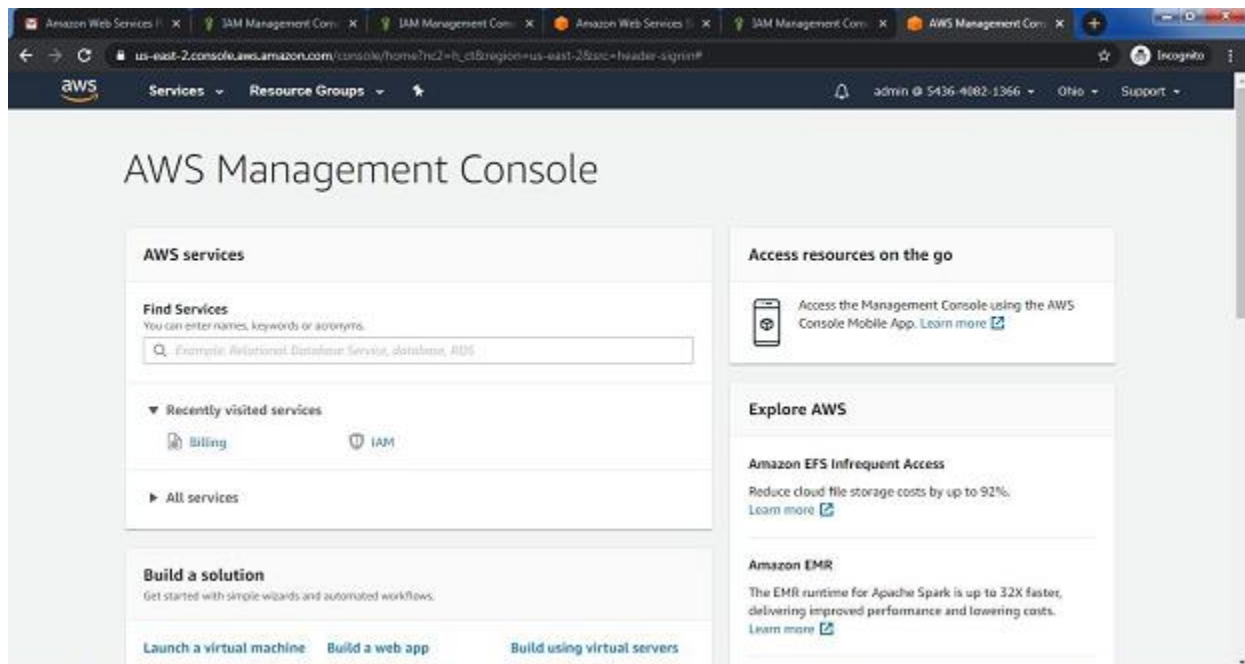
The screenshot shows the AWS Sign-in page in a web browser. The browser's address bar displays the URL: `signin.aws.amazon.com/signin?redirect_uri=https%3A%2F%2Fconsole.aws.amazon.com%2Fconsole%2Fhome%3Fnc2%3Dh_c%26nc%3Dheader-signin%26state%3Dh...`. The page features the AWS logo at the top left. Below it, the text "Sign in" is followed by a small icon. A section titled "Email address of your AWS account" includes a subtext: "Or to sign in as an IAM user, enter your account ID or account alias instead." Below this is a text input field and a blue "Next" button. A link "New to AWS?" is positioned below the "Next" button. At the bottom of the sign-in section is a button labeled "Create a new AWS account". To the right of the sign-in form is a large promotional banner with a dark blue background. The banner text reads: "Move your on-premises Apache Spark and Hadoop to Amazon EMR" and "Read the EMR Migration Guide for technical advice". The banner also features the AWS logo and a diagram of a cloud with nodes and arrows. At the bottom of the page, there is a small section titled "About Amazon.com Sign in" with a link to the Amazon.com sign-in page.

Click on next Provide the Account ID username and password and click on sign in
It will ask you to change the password which is been set by administrator



The screenshot shows the AWS Change Password page in a web browser. The browser's address bar displays the URL: `signin.aws.amazon.com/iam?action=change-password&userType=iam&redirect_uri=https%3A%2F%2Fconsole.aws.amazon.com%2Fconsole%2Fhome%3Fnc2%3Dh_c...`. The page features the AWS logo at the top left. Below it, a red message states: "You must change your password to continue". The page is titled "AWS account" and shows the "IAM user name" as "Admin". Below this are three text input fields: "Old password", "New password", and "Retype new password". A blue "Confirm password change" button is located below the input fields. Below the button is a link: "Get security best practices here". At the bottom of the page, there is a language selector set to "English" and a footer that reads: "Default Language: English © 1995-2022 Amazon Web Services, Inc. or its affiliates."

Yow will redirect to home screen



Conclusion: Hence we have studied the concept and implementation of identity management using amazon aws.

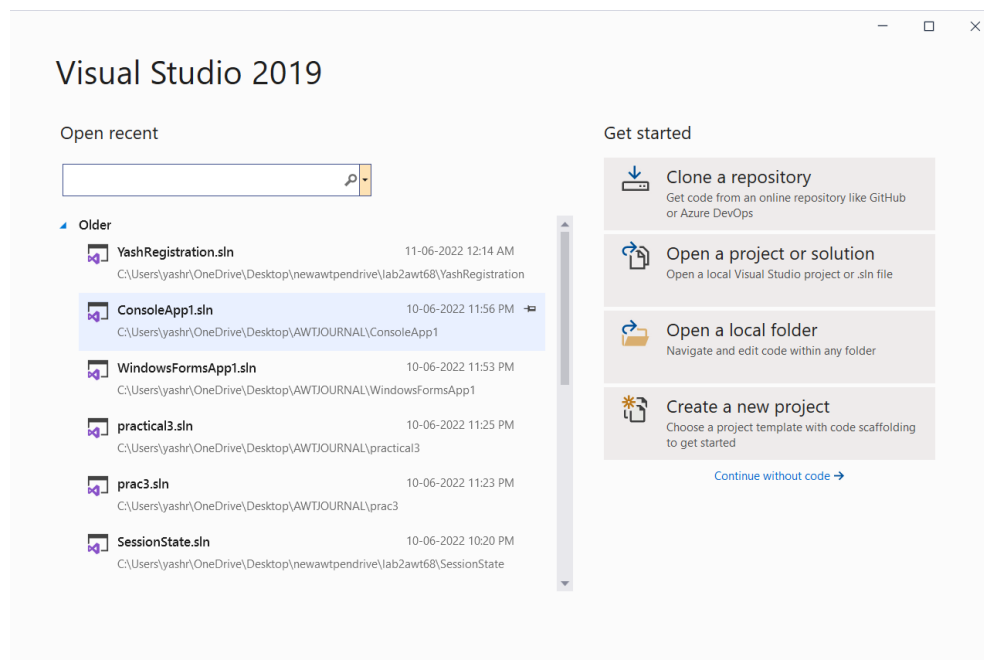
Practical 11

Develop application for Microsoft Azure

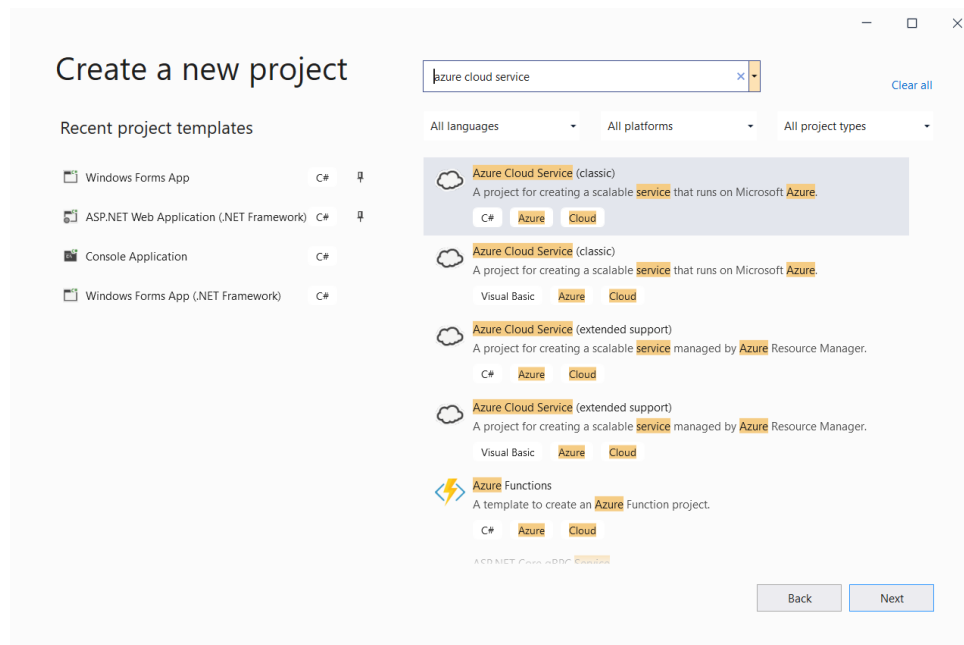
Step 1: To develop an application for Windows Azure on Visual Studio 2019, install the Visual Studio 2019.

Step2:•Open Visual Studio 2019

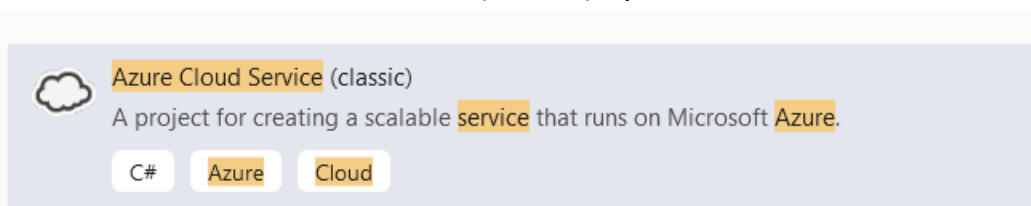
- Click on create new project



- After clicking on Create new project, a New Project Window will open type Azure cloud service in search box
Incase you don't get Azure cloud service(classic) go to installer and install Azure development



- In above window, choose Language as C#, Platform as Azure, Project as Cloud
- Then select Azure Cloud Service(classic) option



Click Next

Configure project window will appear

Configure your new project

Azure Cloud Service (classic) C# Azure Cloud

Project name

Location
 ...

Solution

Solution name ⓘ

☐ Place solution and project in the same directory

Framework

Back Create

Enter Project name(AzureCloudService1), then click on Create button.

New Microsoft Azure Cloud Service(Classic) window will appear as below. Select Visual C# -> ASP.net Web Role.

New Microsoft Azure Cloud Service (classic)

.NET Framework 4.7 roles:

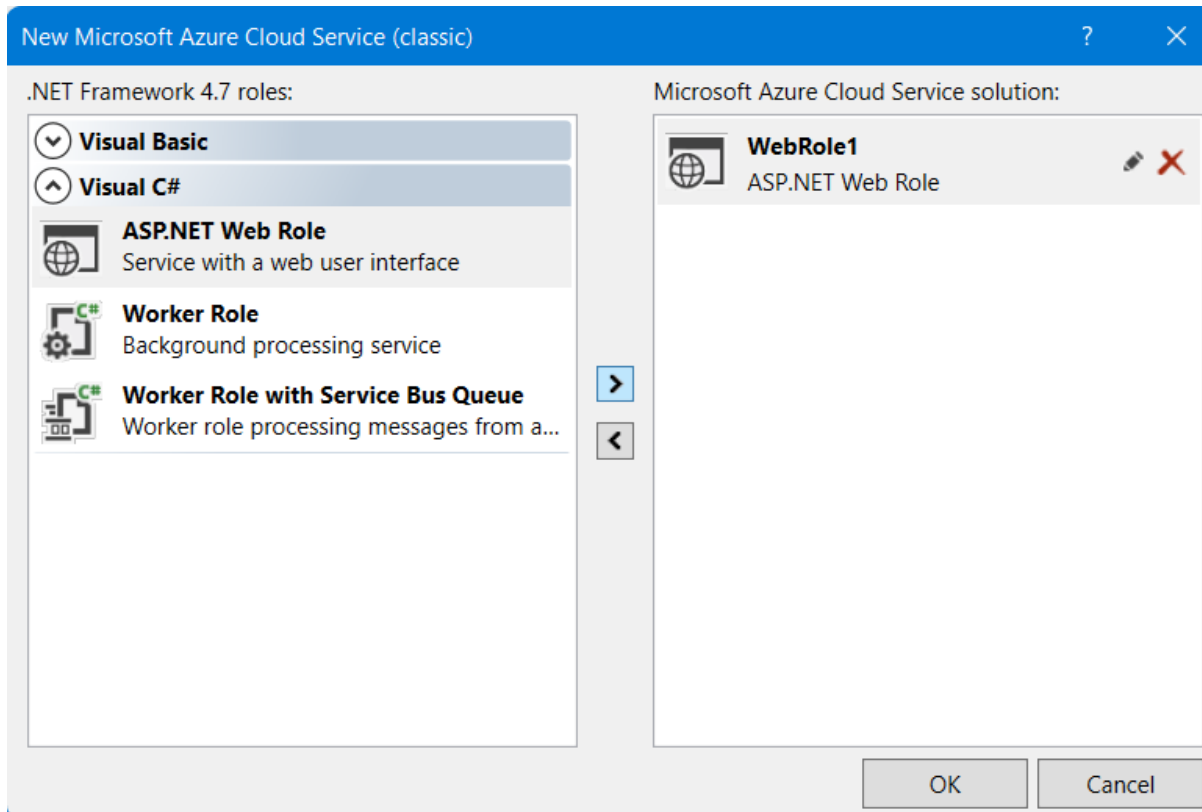
- Visual Basic
- Visual C#
- ASP.NET Web Role**
Service with a web user interface
- Worker Role
Background processing service
- Worker Role with Service Bus Queue
Worker role processing messages from a...

Microsoft Azure Cloud Service solution:

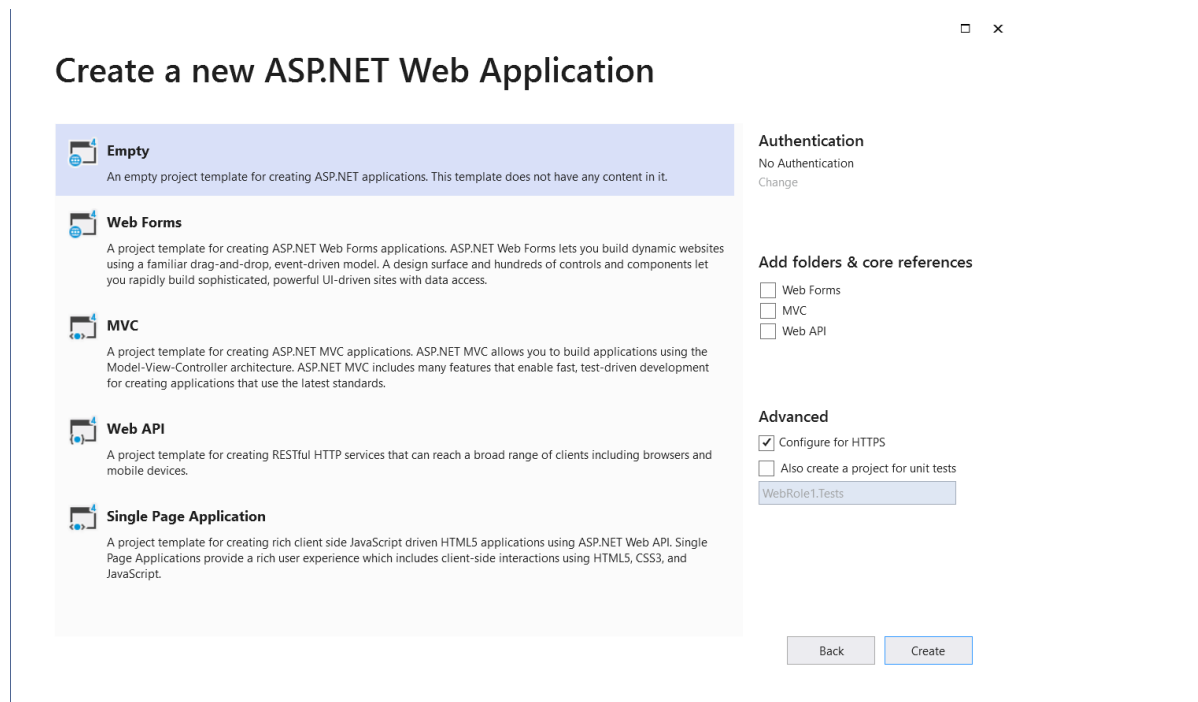
> <

OK Cancel

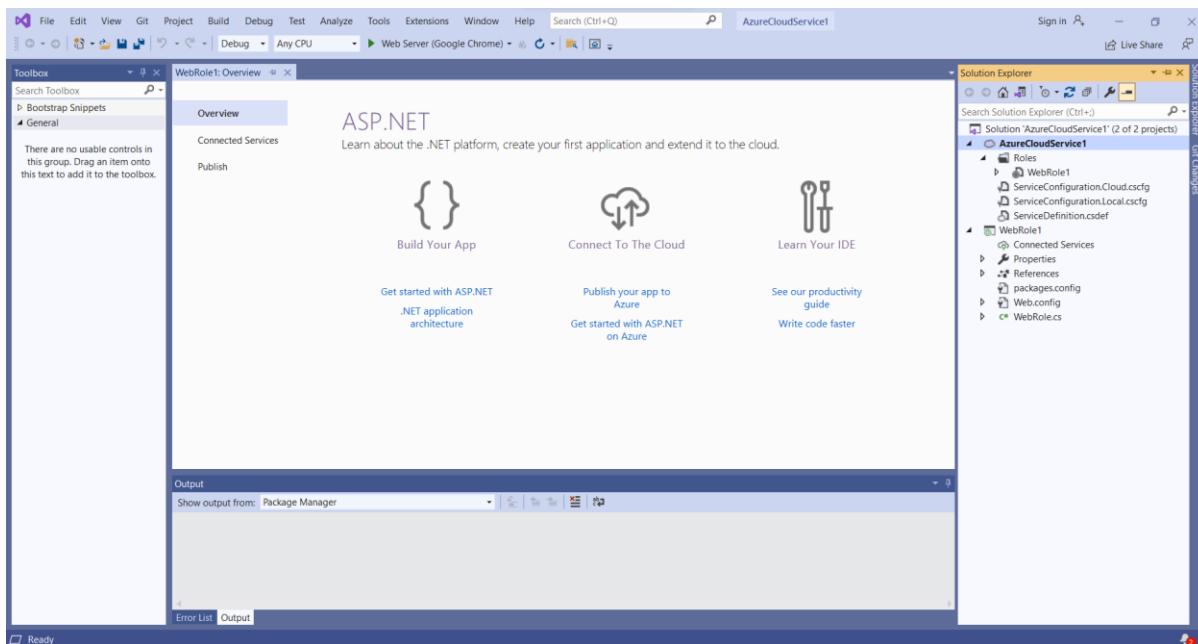
Click on > button for add webrole to solution. Below window will appear.



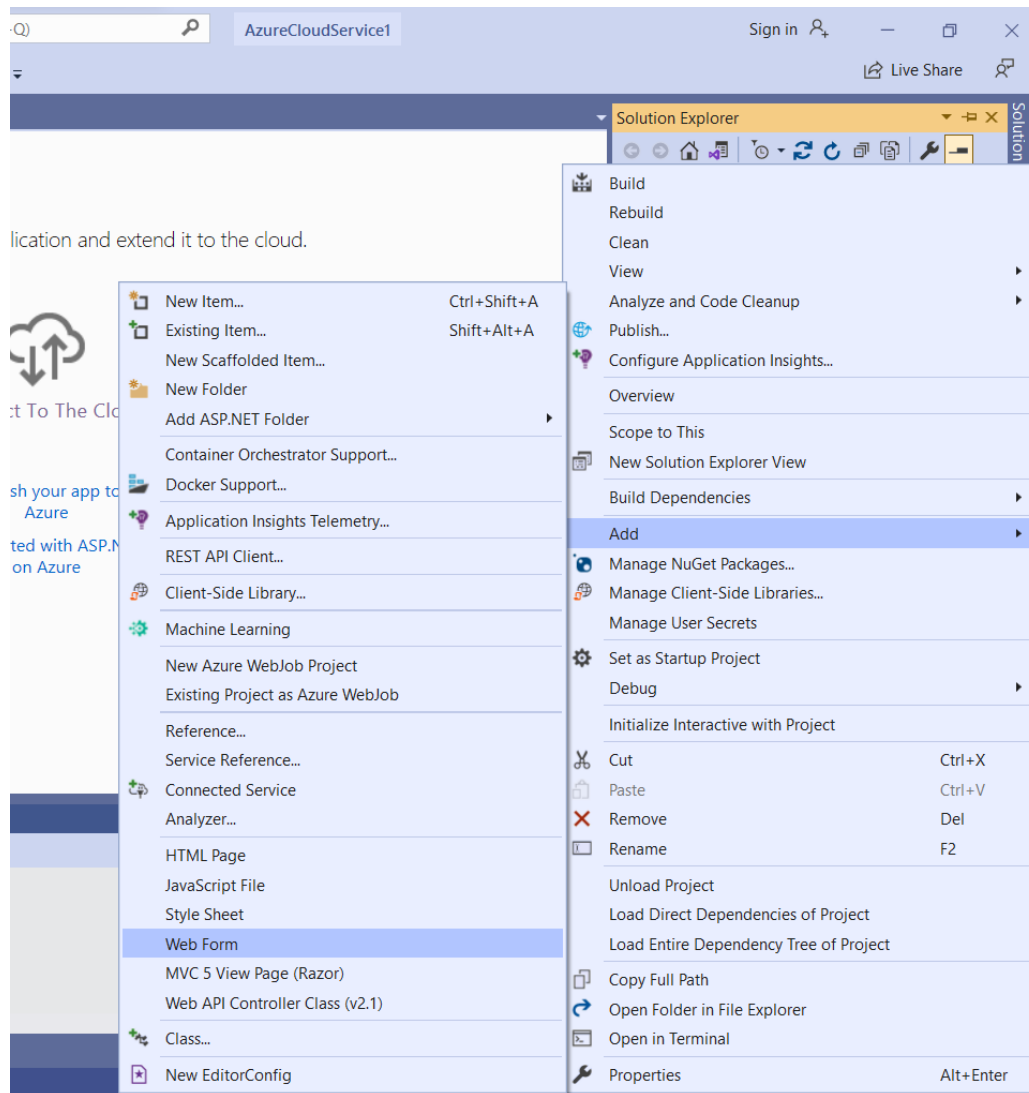
- Click on OK button
- Below window will appear
- Select Empty Option for creating empty project template and then click on Create Button. Below window will appear.



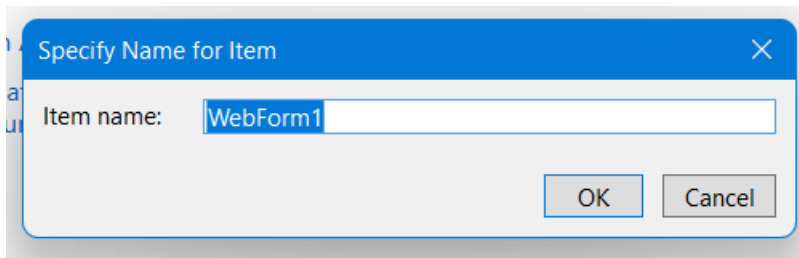
Right Click on WebRole1 in Solution Explorer Window.



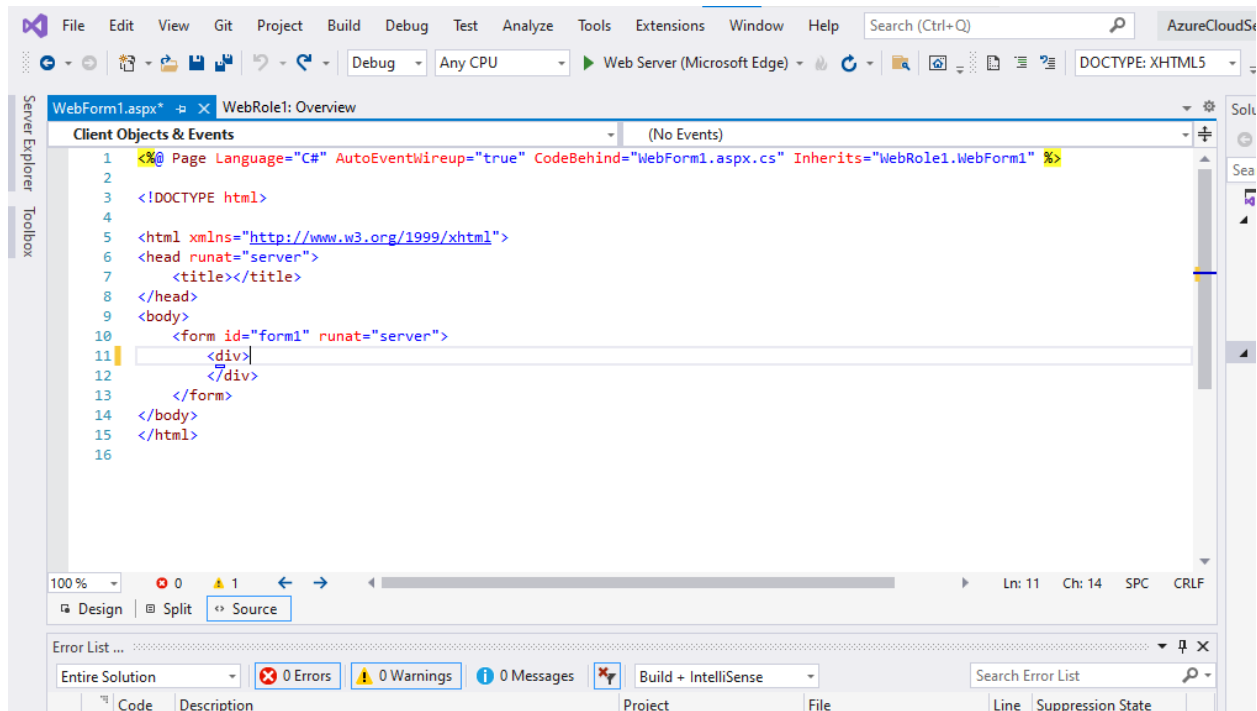
Then click on Add Button and Select Web Form



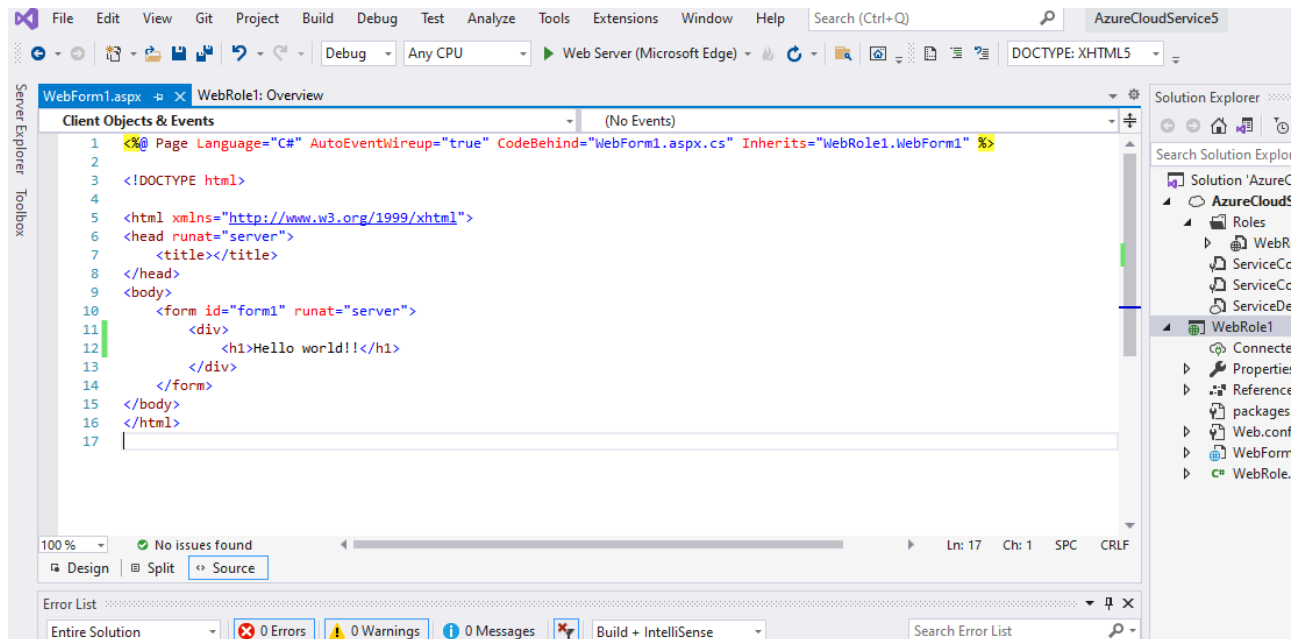
Give Name to Web Form



Click on OK Button. Below window will appear.



Add the below code in web form you have created

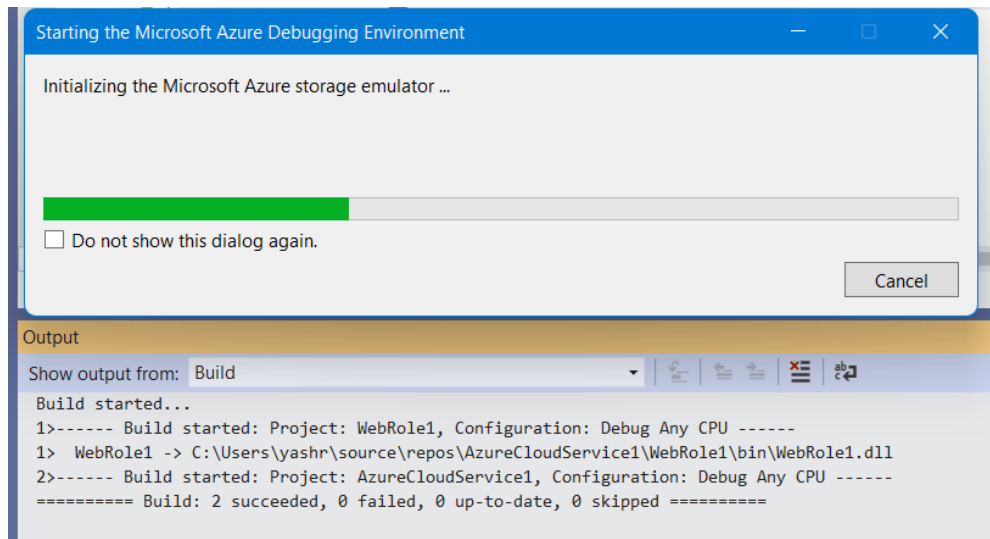


```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="WebRole1.WebForm1" %>

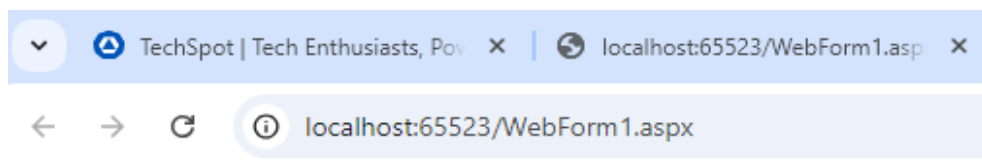
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h1>Hello world!!</h1>
        </div>
    </form>
</body>
</html>
```

Then click on Debug and Execute Project (click on green arrow)



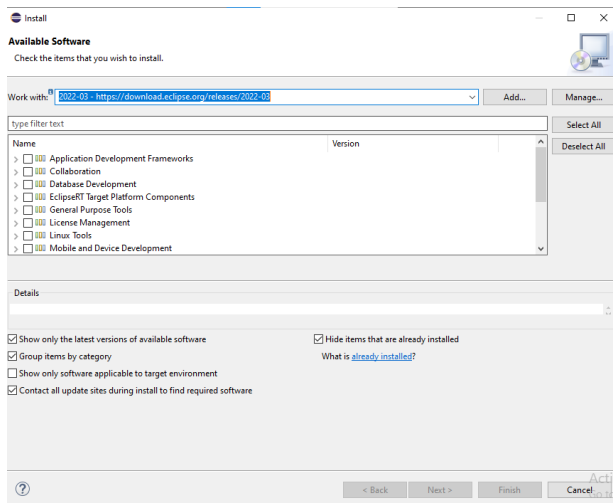
Following is the output:



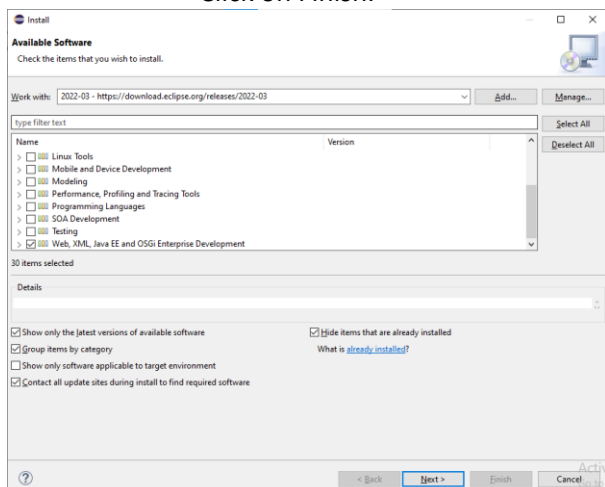
Practical 12

Develop application for Google App Engine Eclipse IDE

1. **Open Eclipse:**
 - Launch Eclipse IDE.
2. **Go to the Help Menu:**
 - Click on Help in the menu bar.
 - Select Install New Software....



3. **Add New Software:**
 - In the Work with: field, enter the appropriate update site URL for Eclipse WTP (Web Tool Is Platform).
 - Choose the appropriate software from the list (e.g., "Web XML...").
 - Click on Finish.

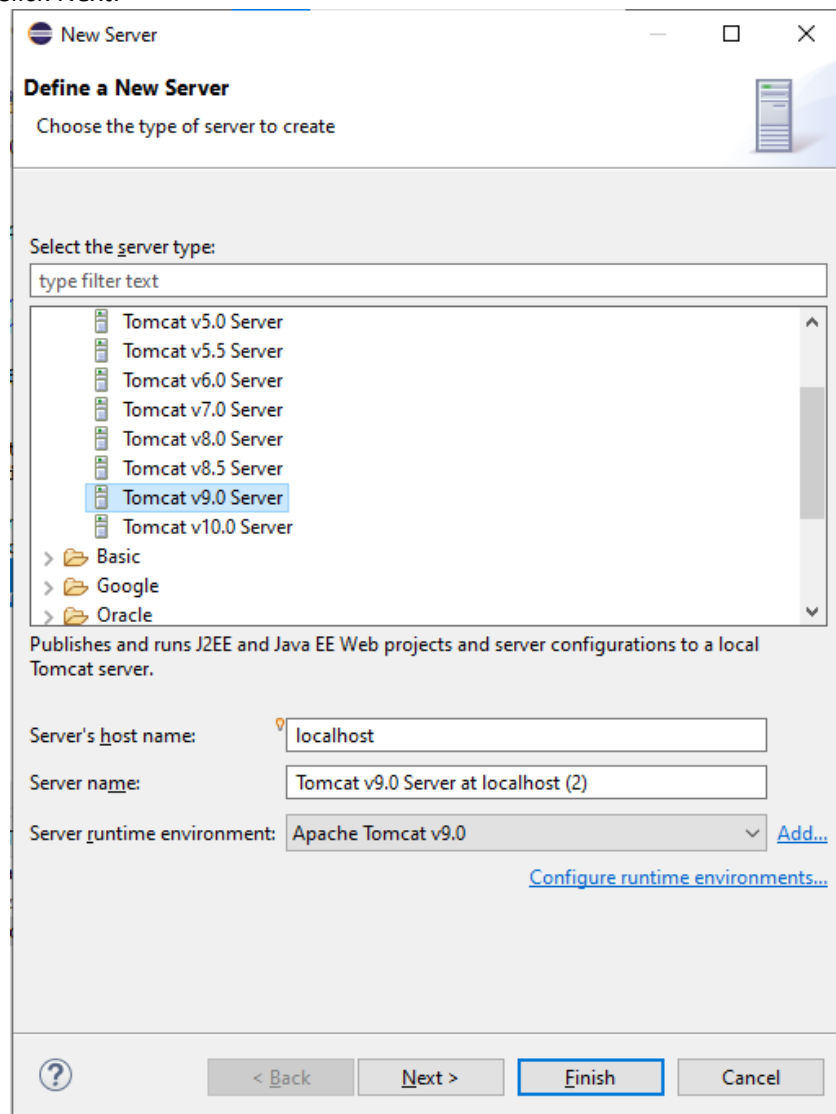


4. Create a New Server:

- Go to the Servers tab in the Eclipse workspace. If you don't see the Servers tab, you can open it by going to Window > Show View > Servers.
- Right-click in the Servers tab and select New > Server.

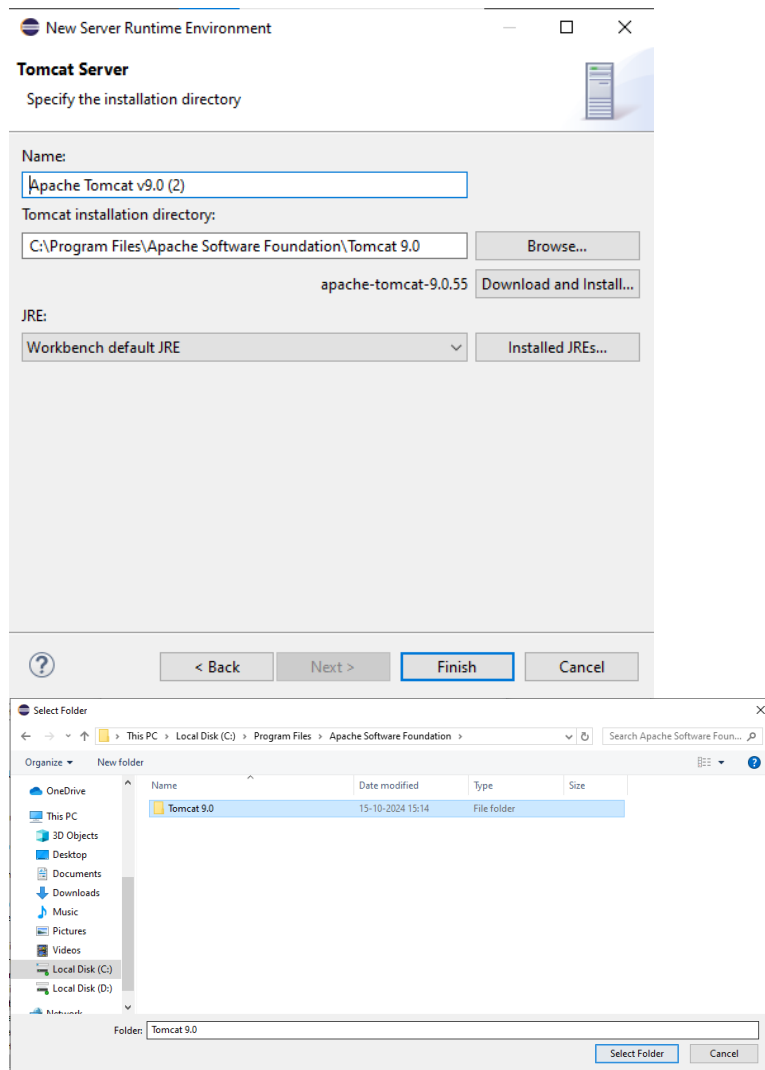
**5. Select Apache Tomcat:**

- In the New Server wizard, expand the Apache folder.
- Select Apache Tomcat v9.0 Server.
- Click Next.



6. Browse for Tomcat Installation Directory:

- In the next window, click Browse....
- Navigate to the directory where Apache Tomcat is installed.
- Select the folder and click OK.

**7. Finish Configuration:**

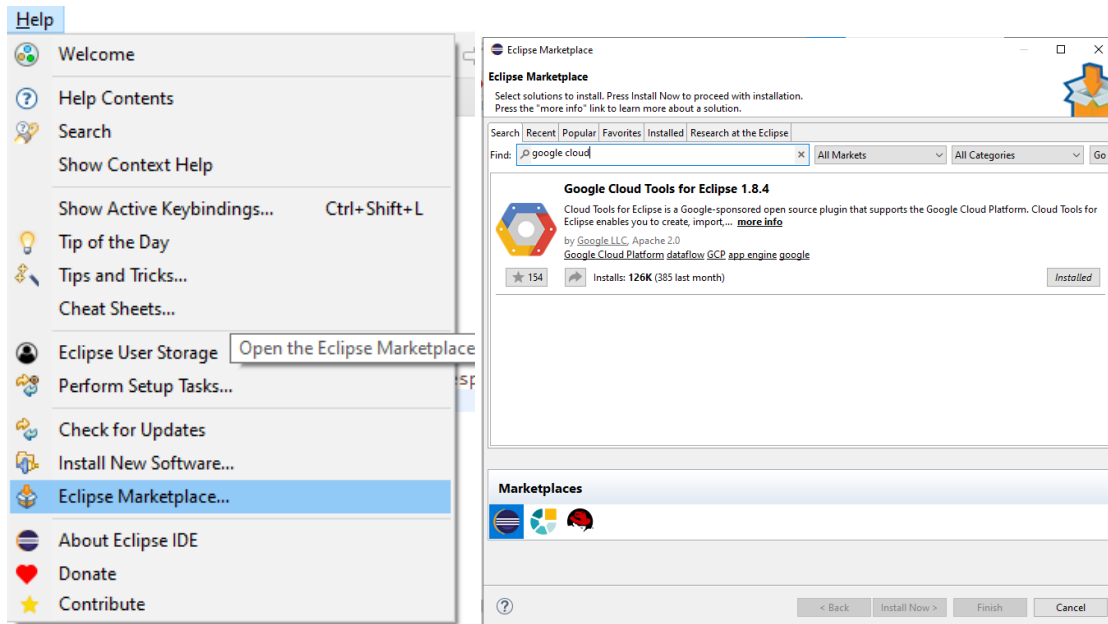
- Click Finish to complete the configuration.

1. Open Eclipse:

- Launch Eclipse IDE.

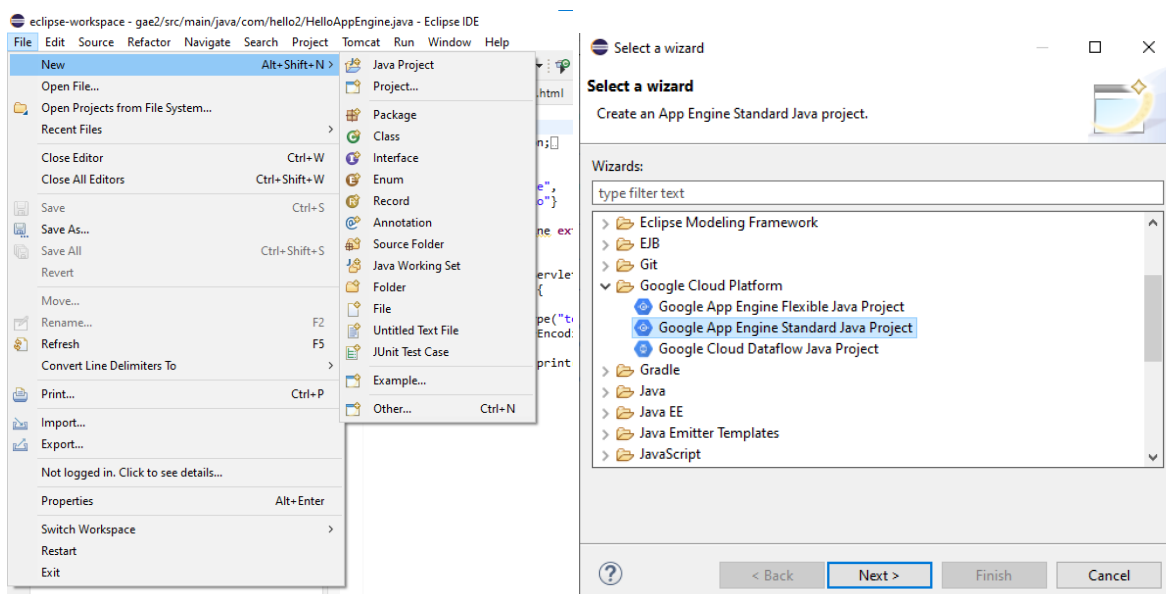
2. Install Google Cloud Tools:

- Go to Help in the menu bar.
- Select Eclipse Marketplace.
- In the search bar, type Google Cloud Tools and select the appropriate version (e.g., 1.8.4).
- Install the tools by selecting all required checkboxes.
- Restart Eclipse after the installation is complete.



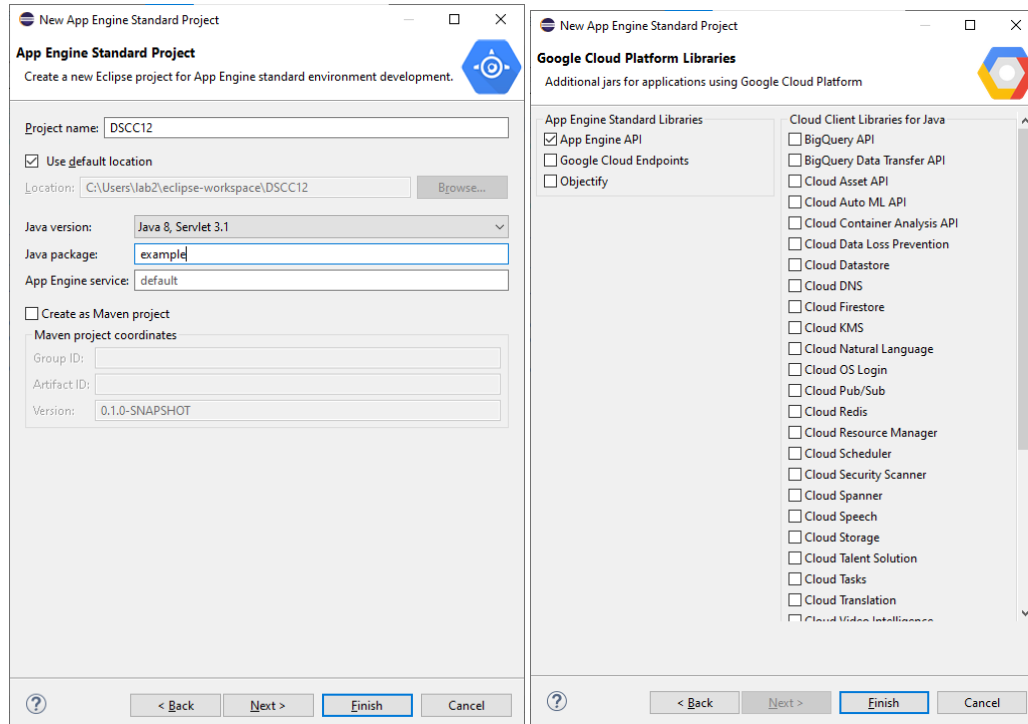
3. Create a Google Cloud Platform Project:

- Go to File > New > Other....
- Select Google Cloud Platform > Google Cloud Platform Standard Project.
- Click Next.



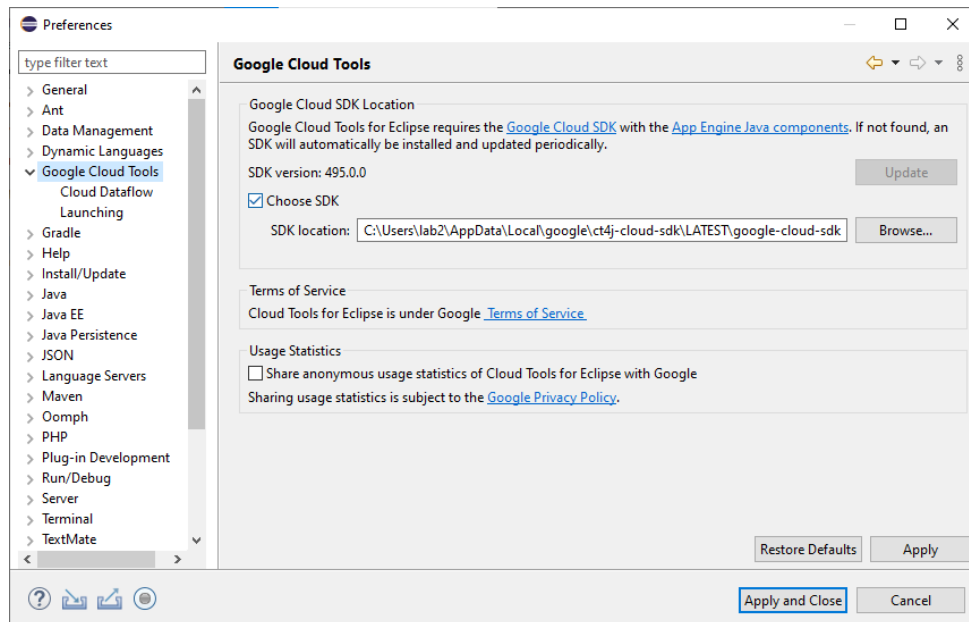
4. Configure the Project:

- Give your project a name and specify the Java package.
- Select the App Engine API.
- Click Finish to create the project.



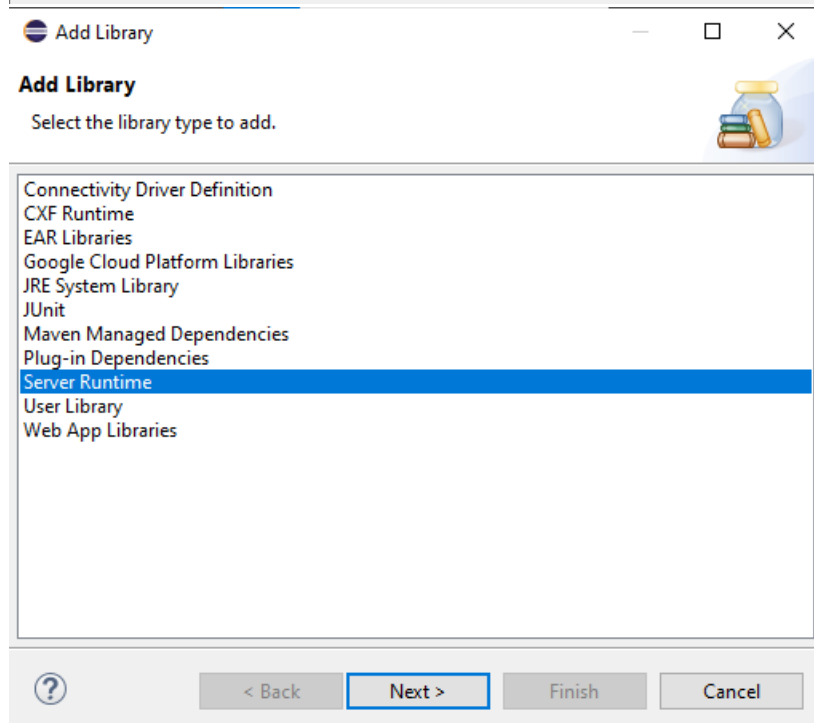
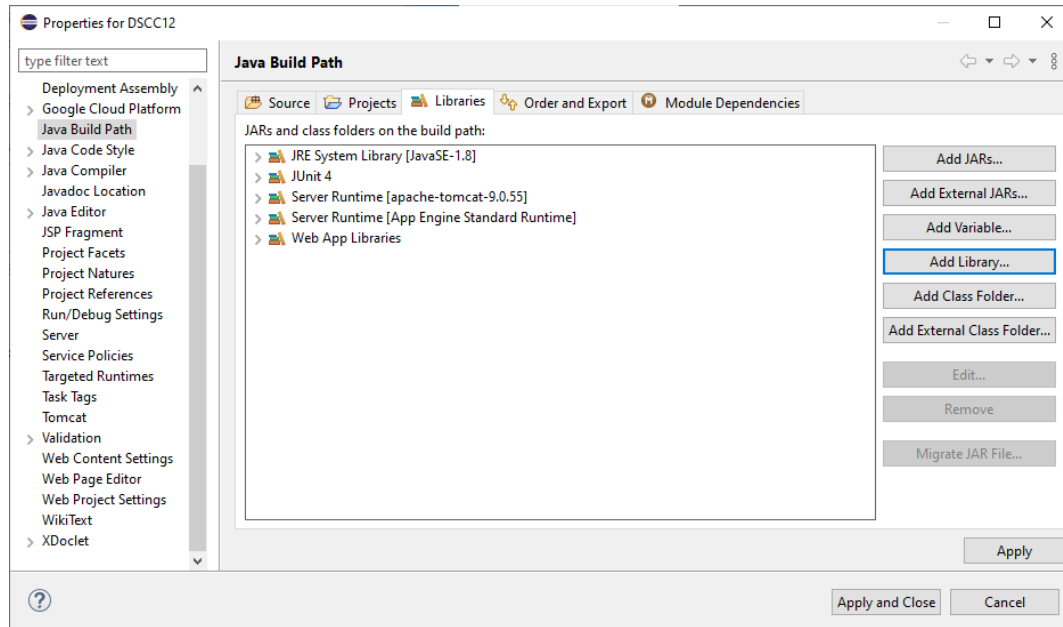
5. Set Up Google Cloud SDK:

- Go to Window > Preferences.
- Select Google Cloud Tools.
- Click on Choose SDK... and follow the prompts to set up the SDK.



6. Configure Project Build Path:

- **Right-click on your project in the Project Explorer.**
- **Select Properties.**
- **Go to Java Build Path.**
- **Click on Add Library....**
- **Select Server Runtime and choose Apache Tomcat.**
- **Click Apply and Close.**



7. Run on Server:

- Right-click on your project.
- Select Run As > Run on Server.
- Choose Apache Tomcat v9.0 Server and follow the prompts to deploy your project.

