EXP – 3

Servicer Oriented Architecture Implementation for hosting application

Aim:

To create and deploy a Calculator web service using Service-Oriented Architecture (SOA) with Eclipse and Apache Tomcat, enabling the execution of basic arithmetic operations through a web interface.

Prerequisites

1. **Eclipse JEE Oxygen 32-bit IDE**
2. **Apache Tomcat 7.0.85 (32-bit/64-bit)**: Download from Apache Tomcat
3. **Java 1.6 or Higher**

Steps

Step 1: Install Eclipse

Step 2: Install Apache Tomcat

Step 3: Configure Apache Tomcat in Eclipse

1. Open Eclipse.
2. Go to Window > Preferences > Server > Runtime Environments.
3. Click Add and select Apache Tomcat v7.0.
4. Click Next, then Browse to locate the Apache Tomcat installation path.
5. Click Finish, then Apply and Close.

Step 4: Create a Dynamic Web Project

1. In Eclipse, select File > New > Project.
2. Choose Web > Dynamic Web Project.
3. Name the project Calculator and click Finish.

Step 5: Create a Calculator Class

1. In the Project Explorer, right-click on the project Calculator.
2. Select New > Class.
3. Enter the package name com.calculator.example and class name Calculate.
4. Click Finish.

Step 6: Develop the Calculator Service

1. Implement the Calculate class with methods for add, subtract, multiply, and divide.

2. Save the class.

Step 7: Create a Web Service

1. Right-click on Java Resources in the Project Explorer.
2. Select New > Web Service.
3. Choose the service implementation class Calculate.
4. Set the configuration level to Test Service and Test Client.
5. Check the box for “Publish Web Service” and click Next.

Step 8: Deploy and Test the Service

1. Follow the wizard by clicking Next through each step until completion.
2. Start the server when prompted.
3. Finish the wizard.

Step 9: Test the Calculator Web Service

1. Use the web service client to invoke the add, subtract, multiply, and divide operations.
2. Verify the responses.

Step 10: View the WSDL File

1. In the Project Explorer, navigate to WebContent > wsdl.
2. Open Calculate.wsdl to view the service description.

EXP – 4

Create Virtual Machine for Using Cloud Applications, Instance

Aim

To set up and configure a virtual machine (VM) on a cloud platform to deploy and manage cloud applications efficiently.

Prerequisites

1. **Cloud Account:** Access to a cloud service provider (e.g., AWS, Azure, Google Cloud).
2. **Cloud Management Tools:** Familiarity with the cloud provider's management console or command-line tools.

Steps

Step 1: Log into Cloud Platform

1. **Access Cloud Console:**
    - Go to the website of your cloud provider (e.g., AWS, Azure, Google Cloud).
    - Log in using your cloud account credentials.

Step 2: Navigate to VM Services

1. **Locate VM Services:**
    - Find the section for virtual machines (e.g., EC2 for AWS, Virtual Machines for Azure).
    - Select the option to create a new VM instance.

Step 3: Configure the Virtual Machine

1. **Select OS and Size:**
    - Choose an operating system (e.g., Linux, Windows) for your VM.
    - Select the instance size (CPU, memory) based on your application needs.
2. **Set Instance Details:**
    - Provide a name for the VM.
    - Configure network settings (e.g., VPC, subnet, firewall rules).

Step 4: Launch the Virtual Machine

1. **Review and Launch:**
    - Review the configuration settings.
    - Click the button to launch or create the VM instance.

Step 5: Connect to the VM

1. **Access VM:**

- Use SSH (for Linux) or RDP (for Windows) to connect to the VM.
- Follow the instructions provided by the cloud provider to establish the connection.

Step 6: Deploy Cloud Applications

1. **Install and Configure Applications:**
    - Once connected, install the necessary cloud applications.
    - Configure the applications as per your requirements.

Step 7: Manage and Monitor the VM

1. **Monitoring Tools:**
    - Use the cloud provider's monitoring tools to track the VM's performance and health.
    - Adjust resources or configurations as needed.

EXP – 5

Aim

To implement para-virtualization by setting up virtual machines on VMware Workstation or Oracle's VirtualBox, and install a guest operating system for efficient resource sharing.

Prerequisites

1. **VMware Workstation or Oracle's VirtualBox:** Ensure one of these virtualization software is installed on your host machine.
2. **Guest Operating System ISO:** Download the ISO file of the guest OS you want to install (e.g., Linux, Windows).

Steps

Step 1: Install Virtualization Software

1. **Download and Install:**
    - Go to the official website of VMware or VirtualBox.
    - Download the installer and follow the instructions to install it on your host machine.

Step 2: Create a New Virtual Machine

1. **Launch Software:**
    - Open VMware Workstation or Oracle VirtualBox.
    - Click on the option to create a new virtual machine.
2. **Configure VM Settings:**
    - Select the option for installing the OS later.
    - Choose the type and version of the guest operating system.
3. **Allocate Resources:**
    - Set the amount of RAM, CPU cores, and storage for the VM.

Step 3: Install Guest Operating System

1. **Mount ISO:**
    - Attach the ISO file of the guest OS to the virtual machine's CD/DVD drive.
2. **Start VM:**
    - Power on the VM and follow the installation instructions for the guest OS.

Step 4: Configure Para-Virtualization

1. **Install Para-Virtualization Drivers:**

- After installing the guest OS, install the virtualization tools (e.g., VMware Tools or VirtualBox Guest Additions).
- These tools enable para-virtualization, allowing better interaction between the VM and the host.

2. **Adjust Settings:**
    - Fine-tune settings for network, display, and other devices for optimal performance.

Step 5: Test and Optimize

1. **Test Functionality:**
    - Run various applications on the guest OS to test performance.
    - Ensure all hardware devices are functioning correctly.
2. **Optimize Performance:**
    - Adjust resource allocation and other settings based on performance testing.

EXP – 6

Create A Web Page and Upload In Online/ Cloud Storage

Aim

To design a simple web page and host it online using cloud storage, making it accessible over the internet.

Prerequisites

1. **Web Development Tools:** A text editor (e.g., Visual Studio Code, Sublime Text) for creating the web page.
2. **Cloud Storage Account:** Access to a cloud storage service that supports web hosting (e.g., Amazon S3, Google Cloud Storage, Microsoft Azure Blob Storage).

Steps

Step 1: Create a Simple Web Page

1. **Open Text Editor:**
    - Launch your preferred text editor.
2. **Write HTML Code:**
    - Create a new file and save it as index.html.
    - Add basic HTML structure and content:

html

Copy code

```
<!DOCTYPE html>
<html>
<head>
  <title>My Web Page</title>
</head>
<body>
  <h1>Welcome to My Web Page</h1>
  <p>This is a simple web page hosted on cloud storage.</p>
</body>
</html>
```

3. **Save the File:**
    - Save the changes to the index.html file.

Step 2: Set Up Cloud Storage

1. **Access Cloud Storage Console:**
    - Log in to your cloud storage provider's management console.
2. **Create a New Bucket/Container:**
    - Create a new bucket or container to store your web page files.

Step 3: Upload Web Page to Cloud Storage

1. **Upload File:**
    - Use the cloud storage console's upload feature to upload the index.html file to your bucket or container.
2. **Configure Bucket for Web Hosting:**
    - Enable website hosting in the bucket's settings.
    - Set the index.html file as the default document.

Step 4: Set Permissions for Public Access

1. **Configure Permissions:**
    - Adjust the permissions of your bucket or the index.html file to allow public read access.

Step 5: Access Your Web Page

1. **Get Public URL:**
    - Obtain the public URL of your web page from the cloud storage console.
2. **Visit Web Page:**
    - Open a web browser and navigate to the public URL to view your web page.

EXP – 7

How Do You Create Data Base on Cloud and How to Upload on Cloud

Aim

To set up a cloud-based database and upload data, providing a scalable and accessible storage solution for applications.

Prerequisites

1. **Cloud Account:** Access to a cloud provider that offers database services (e.g., AWS, Azure, Google Cloud).
2. **Database Management Tool:** A tool to interact with the database (e.g., MySQL Workbench, pgAdmin, or cloud provider's web console).

Steps

Step 1: Choose a Cloud Database Service

1. **Log into Cloud Platform:**
    - Go to the website of your chosen cloud provider and log in.
2. **Navigate to Database Services:**
    - Find the database section in the cloud console (e.g., Amazon RDS, Azure SQL Database, Google Cloud SQL).

Step 2: Create a New Database Instance

1. **Select Database Type:**
    - Choose the type of database you want to create (e.g., MySQL, PostgreSQL, SQL Server).
2. **Configure Database Instance:**
    - Set the instance name, region, instance class, and storage capacity.
    - Configure access credentials (username and password).
3. **Launch the Database Instance:**
    - Review the settings and click "Create" or "Launch" to initiate the database instance.

Step 3: Upload Data to the Cloud Database

1. **Connect to the Database:**
    - Use a database management tool or the cloud provider's web console to connect to your database using the access credentials.
2. **Create Tables and Schemas:**
    - Define the tables and schemas in your database as needed for your data.

3. **Upload Data:**
    - Use the management tool to import data from files (e.g., CSV, SQL dump) or manually insert data using SQL commands.

Step 4: Secure and Manage the Database

1. **Set Up Security Rules:**
    - Configure security settings such as firewall rules, encryption, and access control lists.
2. **Monitor and Optimize:**
    - Use the cloud provider's monitoring tools to keep track of database performance and make optimizations as necessary.

EXP – 8

Import excel to database

Aim

To import data from an Excel file into a cloud database, enabling easy integration of spreadsheet data into a database system.

Prerequisites

1. **Cloud Database Instance:** A database set up on a cloud platform (e.g., AWS RDS, Azure SQL Database, Google Cloud SQL).
2. **Excel File:** The Excel file you want to import.
3. **Database Management Tool:** A tool to connect to and manage your cloud database (e.g., MySQL Workbench, pgAdmin, SQL Server Management Studio).

Steps

Step 1: Prepare the Excel File

1. **Open Excel:**
    - Open the Excel file you want to import.
2. **Format Data:**
    - Ensure your data is organized in a tabular format with headers in the first row.
3. **Save as CSV:**
    - Save the Excel file as a CSV (Comma-Separated Values) file:
        - Go to File > Save As.
        - Choose CSV (Comma delimited) (*.csv) as the file format.
        - Save the file.

Step 2: Prepare the Cloud Database

1. **Connect to Database:**
    - Open your database management tool and connect to your cloud database instance using your credentials.
2. **Create a Table:**
    - Define the table schema in your cloud database to match the structure of the CSV file.
    - For example, if your CSV has columns Name, Age, and Email, create a corresponding table:

sql

```
CREATE TABLE Users (
```

Name VARCHAR(255),

Age INT,

Email VARCHAR(255));

Step 3: Import CSV into the Database

1. **Upload CSV File:**
    - Depending on your database management tool, locate the option to import data. This might be under an “Import” or “Load Data” menu.
2. **Import Data:**
    - Follow the prompts to import the CSV file into the database:
        - Select the CSV file you saved earlier.
        - Map the CSV columns to the database table columns.
        - Execute the import command or process.

Example for MySQL using SQL command:

sql

LOAD DATA LOCAL INFILE 'path/to/yourfile.csv'

INTO TABLE Users

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\n'

IGNORE 1 ROWS;

Example for PostgreSQL using COPY command:

sql

COPY Users(Name, Age, Email)

FROM 'path/to/yourfile.csv'

DELIMITER ','

CSV HEADER;

Step 4: Verify the Data Import

1. **Check Data:**
    - Run a query to verify that the data has been imported correctly:

sql

SELECT * FROM Users;