Department of Cyber Security and Cyber Forensics

CLOUD COMPUTING LABORATORY
(Detailed Manual)

VII SEMESTER CSCF

Compiled by

Dr. Krishna Prasad K

Professor-CSCF

1. Find procedure to Install VirtualBox/VMware Workstation with different flavours of Linux or windows OS on top of windows7 or 8 or 10.

Procedure

1. Download VirtualBox

- Visit the VirtualBox website.
- Click on 'Download VirtualBox'.
- Choose the version appropriate for your Windows (7, 8, or 10).
- Install VirtualBox

2.Run the downloaded installer.

- Follow the on-screen instructions. The default settings are usually fine.

3.Download the OS ISO File

- For Linux: Visit the website of the Linux distribution you want (e.g., Ubuntu).
- For Windows: You'll need a Windows ISO file. You can download this from the Microsoft website.

4.Create a New Virtual Machine (VM)

- Open VirtualBox.
- Click 'New' to start creating a VM.
- Name your VM and select the type and version of the OS you're installing.
- Allocate RAM (memory) to the VM (at least 2GB recommended).
- Create a virtual hard disk (recommend at least 20GB).

5.Configure the VM

- Select the VM and click 'Settings'.
- Go to 'Storage' > 'Controller: SATA' > 'Empty'.
- Click the disk icon next to 'Optical Drive' and choose 'Choose a disk file'.
- Select the ISO file you downloaded earlier.

6.Install the OS

- Select the VM and click 'Start'.
- Follow the on-screen instructions to install the OS.
- Once installed, you can start the VM from VirtualBox whenever you want.

For VMware Workstation Player

1. Download VMware Workstation Player

- Visit the VMware website.

- Click on 'Download Now'.
- Choose the version for Windows.

2.Install VMware Workstation Player

- Run the downloaded installer.
- Follow the on-screen instructions.

3.Download the OS ISO File

- Same as in VirtualBox steps.

4. Create a New Virtual Machine

- Open VMware Workstation Player.
- Click 'Create a New Virtual Machine'.
- Select 'Installer disc image file (iso)' and choose your downloaded ISO file.
- Follow the wizard to set up the VM (name, disk size, etc.).

5.Install the OS

- Once the VM setup is complete, VMware will start the OS installation.
- Follow the on-screen instructions to complete the installation.

6.Using the VM

- After installation, you can start the VM from VMware Workstation Player whenever you want.

(Note: Any one you can do not both either VMware Workstation Player or Virtual Box)

Tips for Layman

- Read Each Step Carefully: Don't rush. Read each step in the installation process carefully.
- Default Settings Are Usually Fine: If in doubt, stick with the default settings suggested by the software.
- Allocate Enough Resources: Make sure your physical machine has enough resources (RAM, CPU, storage) to run a VM smoothly.
- Internet Connection: An internet connection is required for downloading software and OS ISO files.
- Follow On-Screen Instructions: During the OS installation in the VM, just follow the on-screen instructions.

2. Install a C compiler in the virtual machine created using virtual box and execute Simple Programs.

For a Linux VM (e.g., Ubuntu)

1. Open Terminal

In your Linux VM, open the Terminal application. This is usually found in your applications menu or can be opened using a shortcut like Ctrl + Alt + T.

2. Update Package Lists

Before installing new software, it's good practice to update the package lists. Type sudo apt update and press Enter. Enter your password if prompted.

3. Install Build-Essential Package

The build-essential package includes the GNU C Compiler (GCC) and other development tools. Install it by typing sudo apt install build-essential and press Enter.

4. Verify Installation

Once the installation is complete, you can verify it by typing gcc --version in the Terminal. This should display the version of GCC that was installed.

5. Write a Simple C Program

Use a text editor (like Gedit, which comes with Ubuntu) to write your C program. For example, you can write a simple "Hello, World!" program.

6. Save the Program

Save the file with a .c extension, for example, hello.c.

7. Compile the Program

Go back to the Terminal.

Navigate to the directory where you saved your C program using the cd command.

Compile the program using GCC by typing gcc hello.c -o hello and press Enter. This will create an executable file named hello.

8. Run the Program

Run your program by typing ./hello and press Enter. You should see the output of your program, e.g., "Hello, World!".

For a Windows VM

1. Download a C Compiler

- One of the popular C compilers for Windows is MinGW. You can download it from MinGW's website.

2. Install the Compiler

- Run the downloaded installer.
- Follow the on-screen instructions. Select the architecture (32 or 64 bit) appropriate for your VM.

3. Set Environment Variable
   - After installation, you need to add the MinGW bin folder to your system's PATH environment variable.
   - Open 'Control Panel' > 'System' > 'Advanced system settings' > 'Environment Variables'.
   - Under 'System variables', find and select 'Path', then click 'Edit'.
   - Add the path to the MinGW bin folder (e.g., C:\MinGW\bin).
   - Click 'OK' to close all dialogs.
4. Verify Installation
   - Open Command Prompt and type gcc --version. This should display the GCC version, confirming the installation.
5. Write and Compile a Simple C Program
   - Use a text editor (like Notepad) to write your C program.
   - Save the file with a .c extension, for example, hello.c.
   - Open Command Prompt, navigate to the directory where you saved your C program, and compile it using GCC: gcc hello.c -o hello.exe.
6. Run the Program
   - Run your compiled program by typing hello.exe in the Command Prompt. You should see the output of your program.

Tips for Layman

Take Your Time: Don't rush through the steps. Read and understand each step before proceeding.

Be Careful with Commands: In Linux, commands are case-sensitive. Ensure you type them exactly as shown.

File Locations: Remember where you save your files, as you will need to navigate to these locations in the command line.

Internet Connection: Ensure you have an active internet connection for downloading necessary software.

3. Install Google App Engine. Create hello world app and other simple web applications usingpython/java.

Step 1: Install Python

Before installing Google App Engine, you need Python on your Windows 10 system.

1. Download Python

- Go to the Python official website.
- Click on the "Download" button for the latest version.

2. Install Python

- Run the downloaded installer.
- Ensure to check the box that says "Add Python 3.x to PATH" before clicking "Install Now".
- Follow the on-screen instructions to complete the installation.

3. Verify Python Installation

- Open Command Prompt and type python --version. This should display the installed Python version.

Step 2: Install Google Cloud SDK

Google App Engine is a part of the Google Cloud SDK.

1. Download Google Cloud SDK

- Visit the Google Cloud SDK page.
- Download the installer for Windows.

2. Install Google Cloud SDK

- Run the downloaded installer.
- Follow the on-screen instructions. It will also install the Google Cloud SDK Shell.

3. Initialize the SDK

- After installation, open Google Cloud SDK Shell (a shortcut should be on your desktop).
- Type gcloud init and press Enter.
- Follow the instructions to log in to your Google account and set up the SDK.

Step 3: Create a "Hello, World!" App in Python

- Create a Project Directory

Create a new folder on your computer where you'll store your app.

- Open Command Prompt in the Directory

Navigate to the newly created folder using the Command Prompt.

- Set Up a Virtual Environment (Optional)

Run python -m venv env to create a virtual environment named env.

- Activate it by running env\Scripts\activate.

Create the Application Files

- In the folder, create a new file named app.yaml and add the following content:

```
runtime: python39
handlers:
- url: /.*
  script: auto
```

Create another file named main.py and add:

```
from flask import Flask
app = Flask(__name__)
@app.route('/')
def hello():
    """Return a friendly HTTP greeting."""
    return 'Hello, World!'
if __name__ == '__main__':
    app.run(host='127.0.0.1', port=8080, debug=True)
```

These files create a basic web app that displays "Hello, World!"

5. Install Flask

- Flask is a simple web framework for Python. Install it by running pip install Flask in your Command Prompt.

6. Run the App Locally

- Inside the project directory, run python main.py.
- Open a web browser and go to http://localhost:8080. You should see your "Hello, World!" message.

Step 4: Deploy to Google App Engine

1. Navigate to Your Project Directory in Google Cloud SDK Shell

- Use the cd command to navigate to your project directory.

2. Deploy Your Application

- Run gcloud app deploy.
- Follow the prompts to deploy your app.

3. Open Your Application

- After deployment, you can access your app via the URL provided by Google App Engine.

Tips for Layman

- Follow Instructions Carefully: Each step is crucial, so read and follow them carefully.
- Use Correct File Names and Extensions: Make sure to name your files correctly with the right extensions (app.yaml, main.py).
- Check Internet Connection: An active internet connection is required for downloading software and deploying the app.
- Google Account Needed: You need a Google account to use Google Cloud services.
- Billing Account: While Google App Engine has a free tier, Google may require you to enable billing. Check their pricing details for more information.

4. Use GAE launcher to launch the web applications.

1. Using Google Cloud SDK to Launch Web Applications
    - Install Google Cloud SDK
    - If you haven't already installed the Google Cloud SDK, please refer to the instructions in my previous message about installing and initializing the Google Cloud SDK.
2. Open the Google Cloud SDK Shell
    - On Windows, you can find the Google Cloud SDK Shell in your Start menu, or you might have a shortcut on your desktop.
3. Navigate to Your Project Directory
    - Use the cd command in the Google Cloud SDK Shell to navigate to your project directory. For example, cd path\to\your\project.
4. Run Your Application Locally
    - To run your application locally, use the gcloud app deploy command. This command deploys your app to the App Engine server.
    - It will prompt you for any necessary information during the deployment process.
5. View Your Running Application
    - Once deployed, Google Cloud SDK will provide you with a URL where you can view your running application.
6. Stop the Running Application
    - If you need to stop your application, you can do so through the Google Cloud Platform Console.

Tips for Layman

- Follow Each Step: Make sure to follow each step carefully and enter the commands as they are written.
- Google Account Required: You will need a Google account to use Google Cloud services.
- Internet Connection Needed: Ensure you have an active internet connection for deploying applications.
- Be Patient: Deployment can take a few minutes, so be patient while the process completes.
- Keep Your Project Organized: Keep all files related to your project in a single folder to make navigation and deployment easier.
- Billing Considerations: While Google App Engine has a free tier, be aware of possible charges if you exceed the free usage limits. It's a good idea to review Google App Engine's pricing details.

5. Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.

Step 1: Set Up Your Development Environment

1. Install Java Development Kit (JDK)
   - Download and install the latest JDK from the Oracle website.
   - Follow the installation instructions, and make sure Java is properly set up by typing java -version in your command prompt or terminal.
2. Install an Integrated Development Environment (IDE)
   - Download and install an IDE that supports Java, like Eclipse or IntelliJ IDEA.
3. Download CloudSim
   - Download the latest version of CloudSim from the CloudSim website or its GitHub repository.
   - Extract the downloaded ZIP file to a folder.

Step 2: Import CloudSim into Your IDE

1. Open Your IDE and Create a New Project
   - Open Eclipse or IntelliJ IDEA and create a new Java project.
2. Import CloudSim Libraries

   - In your project, include the CloudSim jar files from the extracted CloudSim folder.
   - For Eclipse: Right-click on the project → Build Path → Configure Build Path → Libraries → Add External JARs.
   - For IntelliJ IDEA: File → Project Structure → Libraries → + → Java → select the CloudSim

jar files.

Step 3: Implement Your Scheduling Algorithm

1. Understand CloudSim Architecture

- Familiarize yourself with CloudSim's architecture and existing scheduling algorithms. This understanding is crucial to implement your own.

2. Create a New Class for Your Algorithm

- In your project, create a new Java class where you will write your scheduling algorithm.
- You may need to extend or implement specific CloudSim classes or interfaces depending on your algorithm's requirements.

3. Write Your Algorithm

- Implement your scheduling algorithm in the newly created class. This will require programming skills in Java and an understanding of the algorithm you wish to implement.

Step 4: Set Up a Simulation Environment

1. Create a Simulation Class

- Write a Java class to create a cloud environment using CloudSim classes.
- Define cloud resources like hosts, data centers, virtual machines (VMs), and cloudlets (tasks).

2. Integrate Your Scheduling Algorithm

- In the simulation class, configure the data center or VMs to use your custom scheduling algorithm.

Step 5: Run the Simulation and Analyze Results

1. Run Your Simulation

- Execute your simulation class.
- The IDE should compile and run the simulation, using your custom scheduling algorithm.

2. Analyze the Output

- The output will be displayed in the IDE's console.
- Analyze this output to understand how your algorithm performs.

Tips for Layman

- Start Simple: If you're new to CloudSim or Java, start with simple modifications to existing algorithms before attempting to implement a new one.
- Refer to Documentation: Make extensive use of CloudSim documentation and Java resources.
- Debugging: Use the debugging features of your IDE to troubleshoot any issues in your code.
- Seek Help: If you're stuck, seek help from online communities or forums focused on CloudSim or Java programming.

6. Find a procedure to transfer the files from one virtual machine to another virtual machine.

Lab Manual: Transferring Files Between Virtual Machines

Objective:

To safely and efficiently transfer files from one virtual machine (VM) to another.

Prerequisites:

- Two operational virtual machines (VM1 and VM2).
- Basic familiarity with operating systems on the VMs.
- Network connectivity between both VMs.

Tools Needed:

- File transfer software (e.g., FTP client, SCP client) installed on both VMs.
- Optional: Shared network drive or cloud storage access.

Steps:

1. Establish Network Connectivity:
   - Ensure both VMs are on the same network or can communicate over the internet.
   - Check IP addresses of both VMs for reference.
2. Choose a File Transfer Method:
   - Options include FTP, SCP, or shared drives.
   - We'll use SCP (Secure Copy Protocol) for this guide.
3. Install SCP Client:
   - On Windows VMs, an application like WinSCP can be used.
   - On Linux VMs, SCP is usually pre-installed.
4. Preparing Files for Transfer:
   - On VM1, locate the files you want to transfer.
   - Compress them into a zip file for easier transfer (optional).

5. Using SCP for Transfer:
    - Open SCP client on VM1.
    - Enter the IP address of VM2.
    - Provide necessary authentication (username/password).
    - Browse and select the file(s) on VM1.
    - Choose the target directory on VM2.
    - Initiate the transfer.
6. Verifying Transfer:
    - Once transfer is complete, log into VM2.
    - Check the target directory to ensure files are present.
    - Verify file integrity (size, no corruption).
7. Troubleshooting:
    - If transfer fails, check network connectivity and credentials.
    - Ensure there's enough storage space on VM2.
    - Check for any firewall or security settings that might block the transfer.
8. Alternative Methods:
    - For non-technical users, consider using a shared network drive or cloud storage.
    - Upload files from VM1 to the shared location, then download to VM2.

7. Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version)

1. Introduction to TryStack
    - Explain what TryStack is: a free, online platform for experimenting with OpenStack, which is a cloud computing software.
    - Briefly mention the benefits of using virtual machines in the cloud.
2. Account Creation and Login
    - Direct the user to the TryStack website.
    - Guide them through the account creation process, including any verification steps.
3. Explain how to log in to their new TryStack account.
    - Navigating the TryStack Dashboard
    - Provide an overview of the TryStack dashboard.
    - Highlight key areas such as the project overview, compute resources, and network settings.
4. Setting up a Project
    - Instruct how to create a new project (if needed).
    - Discuss the importance of organizing resources within a project.

5. Accessing the Compute Service
    - Explain how to navigate to the Compute service in the dashboard.
    - Describe what the Compute service is used for.
6. Creating a Virtual Machine

   Detailed steps to create a new virtual machine:
    - Click on the “Launch Instance” button.
    - Fill in the details like Instance Name, Flavor (size of the VM), and Source (image or snapshot to use).
    - Select an appropriate image (OS like Ubuntu, CentOS, etc.).
    - Configure network settings, such as assigning a floating IP.
    - Set up security groups (firewall settings) for the VM.
    - Add a key pair for SSH access.

   Explain each step in simple terms, avoiding jargon.
7. Launching the Virtual Machine
    - Guide on how to launch the virtual machine after setting up all the configurations.
    - Explain what to expect during the launching process (status changes, notifications).
8. Accessing and Using the Virtual Machine
    - Describe how to access the VM once it's running, typically via SSH.
    - Provide a basic overview of common tasks that can be performed on the VM.
9. Monitoring and Managing the Virtual Machine
    - Teach how to monitor the status of the VM.
    - Explain how to resize, suspend, or terminate the VM.
10. Troubleshooting Common Issues
    - Include a section on troubleshooting common problems, like connectivity issues or errors during VM creation.

8. Install Hadoop single node cluster and run simple applications like wordcount.

Objective:

- To install and configure a Hadoop single-node cluster on a system.
- To run a simple WordCount application to demonstrate the functioning of the cluster.

Requirements:

- A computer with a minimum of 4GB RAM and a dual-core processor (or better).
- Operating System: Linux (Ubuntu recommended) or a virtual machine running Linux.
- Basic knowledge of Linux command line.
- Internet connection for downloading software.

Part 1: Installing Hadoop

1. **Install Java Development Kit (JDK):**
   - Open a terminal and update the package index:

```sql
sudo apt-get update
```

   - Install the default JDK package:

```arduino
sudo apt-get install default-jdk
```

2. **Verify Java Installation:**
   - Check the Java version to confirm installation:

```
java -version
```

3. **Download Hadoop:**
   - Go to the Apache Hadoop website (http://hadoop.apache.org/) and download the latest stable release of Hadoop.
4. **Extract Hadoop:**
   - Extract the downloaded Hadoop tar file to a directory (e.g., `/usr/local/hadoop`):

```bash
sudo tar zxvf hadoop-x.y.z.tar.gz -C /usr/local/hadoop
```

5. **Configure Hadoop Environment:**
   - Open `.bashrc` file in your home directory:

```bash
nano ~/.bashrc
```

   - Add the following lines at the end of the file:

```bash
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
```

   - Source the `.bashrc` to apply changes:

```bash
source ~/.bashrc
```

6. **Configure Hadoop Settings:**
   - Navigate to the Hadoop configuration directory:

```bash
cd $HADOOP_HOME/etc/hadoop
```

   - Edit the `hadoop-env.sh` file to set the Java Home:

```bash
export JAVA_HOME=$(readlink -f /usr/bin/java | sed "s:bin/java::")
```

   - Configure core-site.xml, hdfs-site.xml, mapred-site.xml, and yarn-site.xml as per your environment needs.
7. **Format the Hadoop Filesystem:**
   - Initialize the Hadoop filesystem:

```lua
hdfs namenode -format
```

8. **Start Hadoop Services:**
   - Start NameNode and DataNode daemons:

```sql
start-dfs.sh
```

   - Start ResourceManager and NodeManager daemons:

```sql
start-yarn.sh
```

Part 2: Running the WordCount Application

1. **Create Input Directory in HDFS:**
   - Create a directory in HDFS to store input files:

```bash
hdfs dfs -mkdir /wordcount_input
```

2. **Add Input Files:**
   - Put a text file into the input directory:

```arduino
hdfs dfs -put localfile.txt /wordcount_input
```

3. **Run WordCount:**
   - Navigate to the Hadoop directory:

```bash
cd $HADOOP_HOME
```

   - Run the WordCount example:

```bash
hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-x.y.z.jar wordcount /
```

4. **Check the Output:**
   - View the output of the WordCount:

```bash
hdfs dfs -cat /wordcount_output/part-r-00000
```