# Group 10 -Ishu Chaudhary rest… did not do the work inspite of me continuously poking them to do. Harshit and Harshavardhan did something different Title: Exploring Cloud Computing Simulations with CloudSim

## **Introduction**

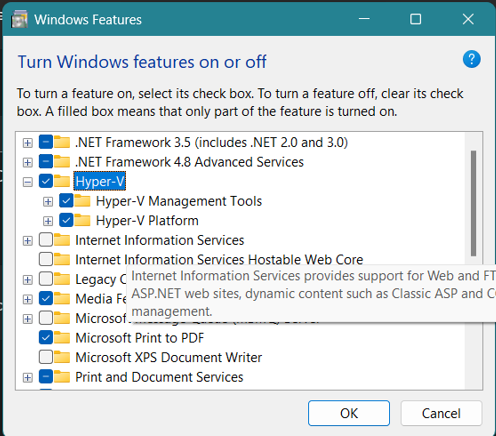
In this article, we'll delve into the world of cloud computing simulations using CloudSim. We'll walk through the process of creating a virtual environment on Windows with Hyper-V Manager, where we'll install Ubuntu 22.04. In this environment, we'll set up the Java Runtime Environment, configure environment variables, and work with CloudSim 3.0.3 to simulate cloud scenarios.

## **Creating the Virtual Environment**

**Unlocking Hyper-V Manager:**  
Enable Virtualization in BIOS: Using Hyper-V requires virtualization enabled in the motherboard's BIOS.

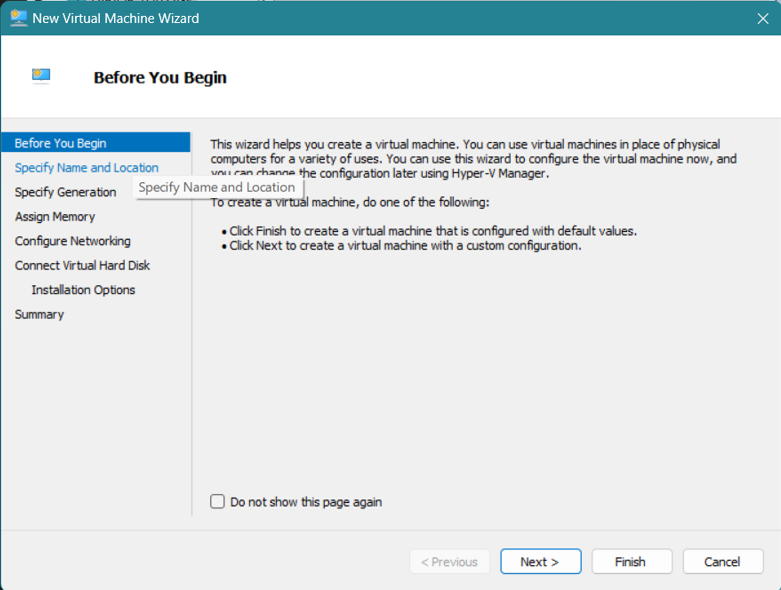
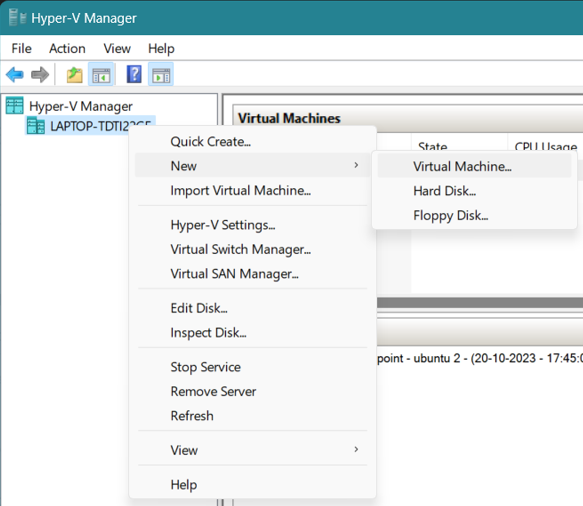
1. Open the BIOS by rebooting your computer and pressing the BIOS button (F1, F2, F10, F12, or Delete) when prompted.

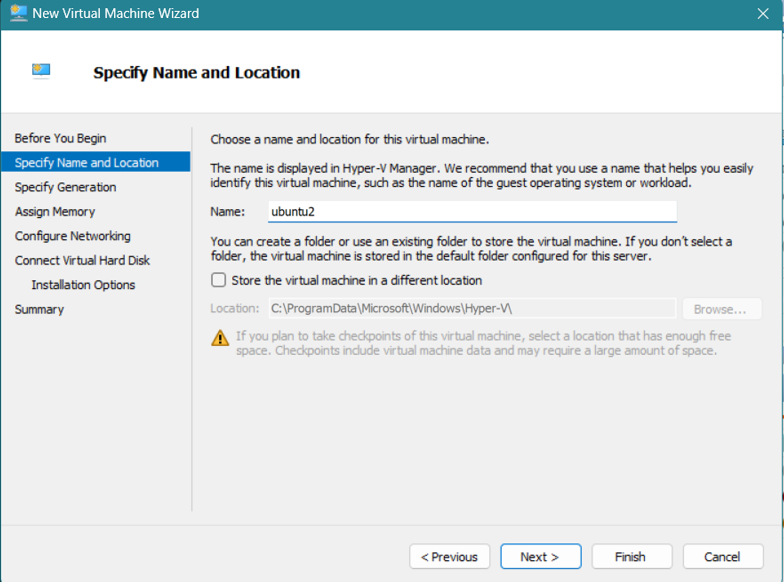
2. The virtualization settings are located under the CPU or Advanced BIOS Settings category.

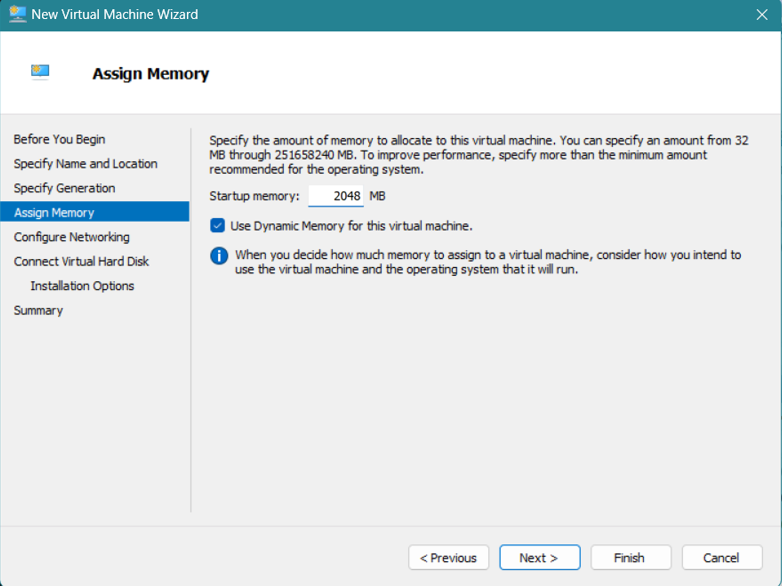
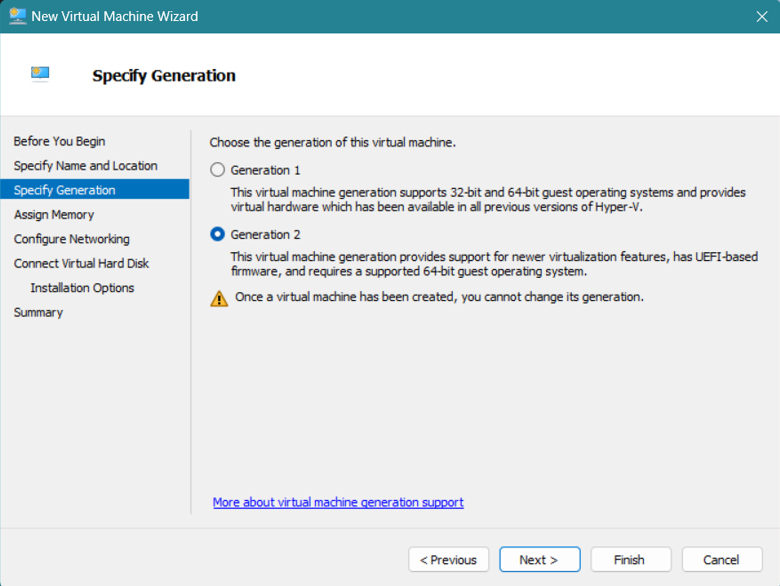
3.let's activate Hyper-V Manager from the Windows Optional Features menu.

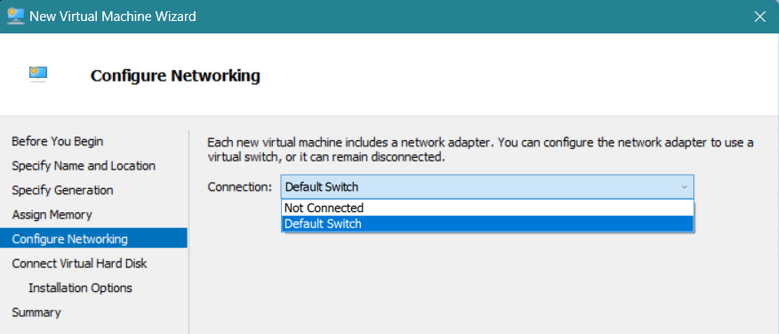
### **Crafting the Ubuntu VM:**

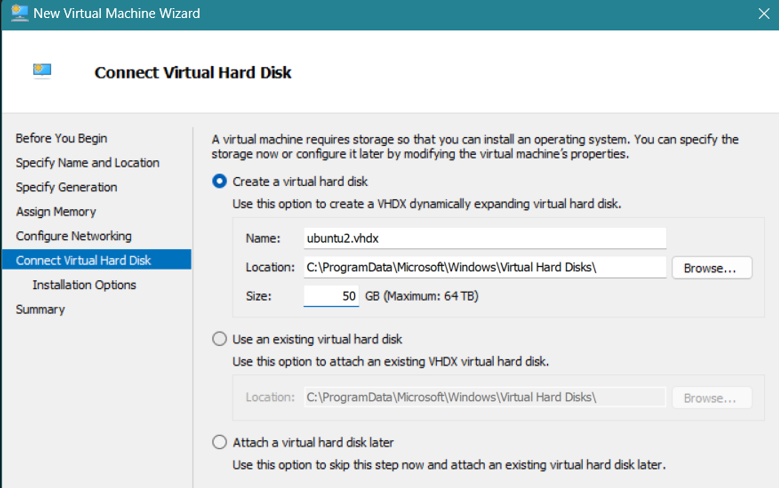
We'll craft a brand-new virtual machine (VM) within Hyper-V Manager, and we'll give it the name "ubuntu2". With that ready, we'll install Ubuntu 22.04 by loading the ISO file into the VM's DVD drive.

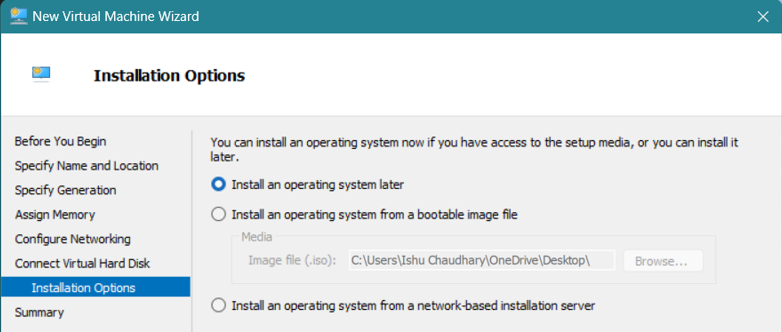


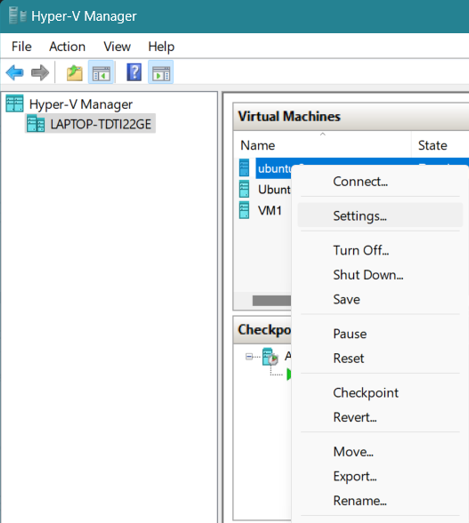


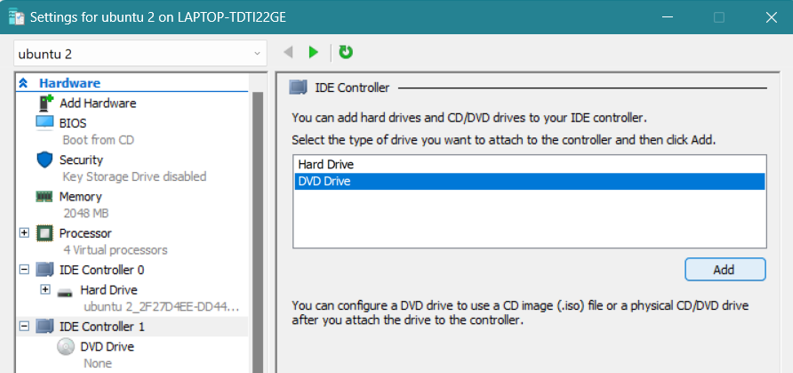


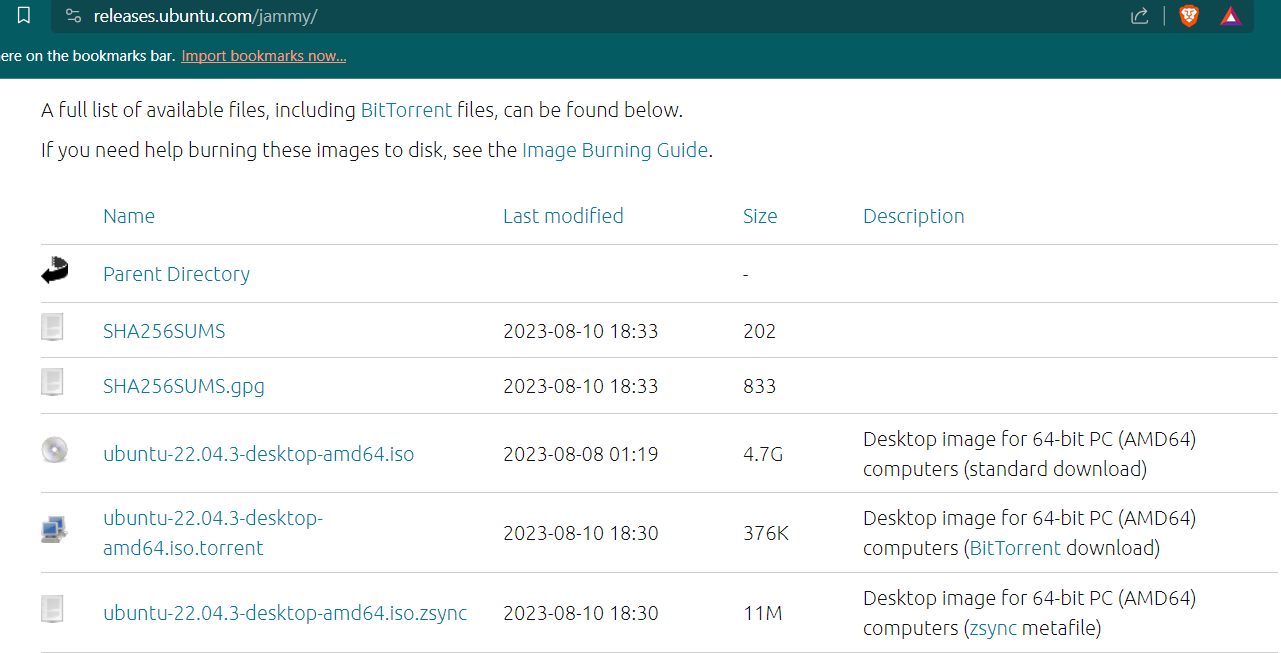


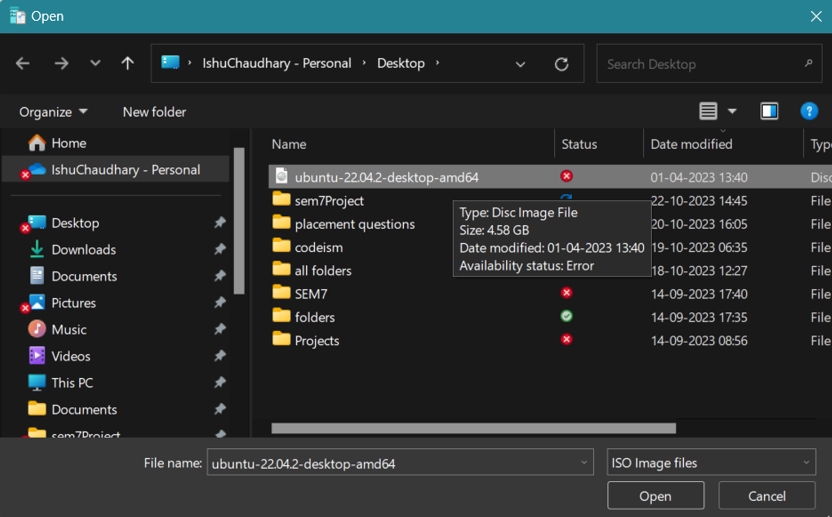












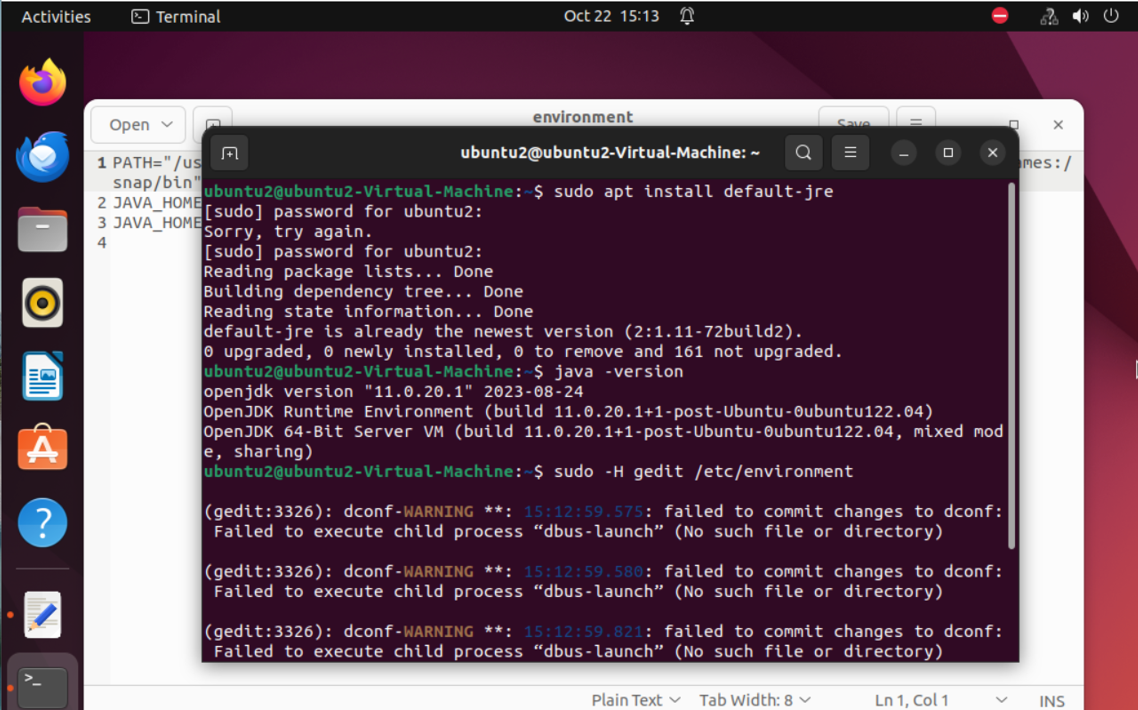
### **Installing Java:**

Inside our Ubuntu VM, we'll sprinkle in the Java Runtime Environment with the magic command

sudo apt install default-jre.

### **Fine-Tuning Environment Variables:**

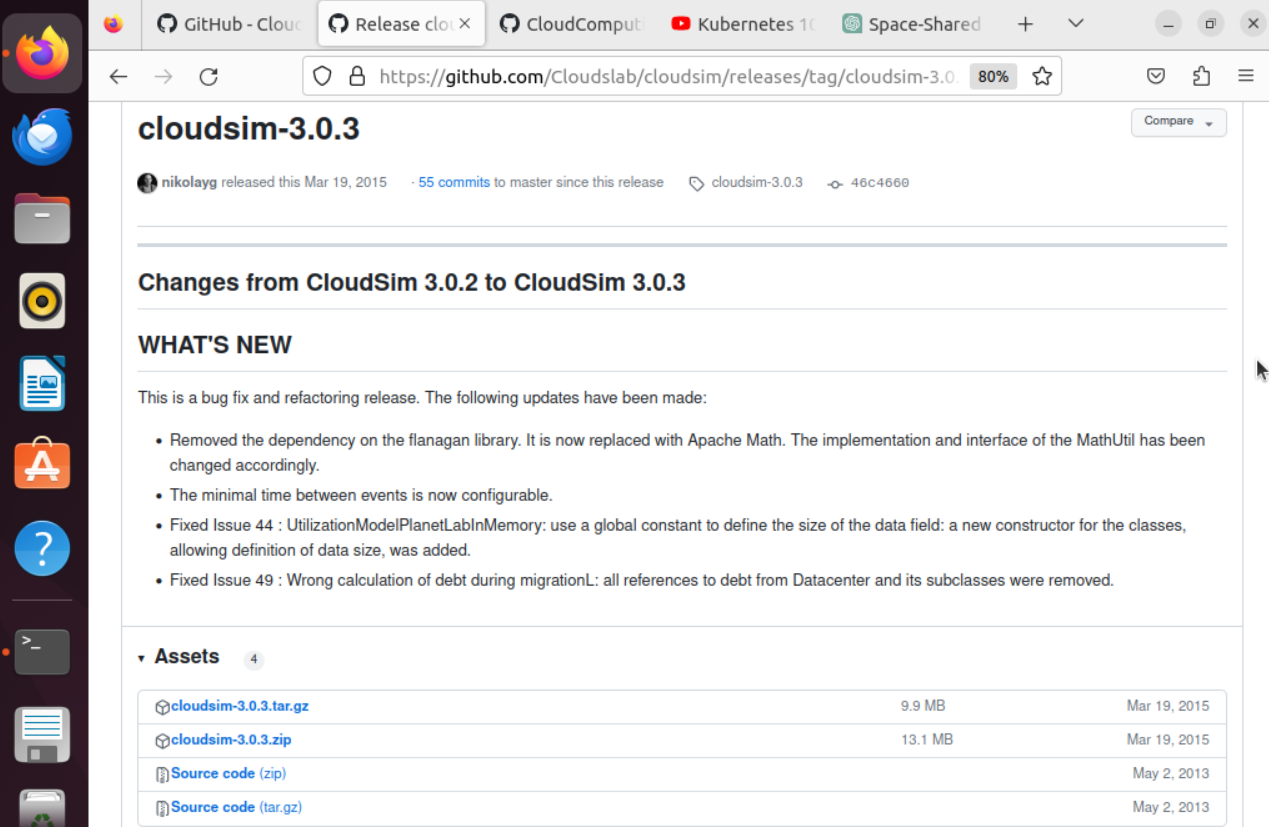
Now, let's fine-tune our environment by setting up Java environment variables. This involves using the command   
sudo -H gedit /etc/environment  
to add -> JAVA\_HOME="/usr/lib/jvm/java-11-openjdk-amd64".





## **Working with CloudSim**

### **Snagging CloudSim:**

We'll snag CloudSim 3.0.3 from a handy GitHub link, typically found on the CloudSim website.  
github.com/Cloudslab/cloudsim  


### **Customizing the Datacenter Broker:**

The Datacenter Broker is our secret sauce in CloudSim. It acts as a mediator between cloud servers and the cloud infrastructure, orchestrating the provisioning of VMs and the scheduling of cloudlets.

You can give it a personalized twist by modifying the DataCenterBroker Java file to implement diverse cloud scheduling algorithms.

### **Crafting the Main Simulation File:**

We'll whip up a Java file for our cloud computing simulation, naming it "Simulation.java."

The main course includes   
  
1.importing CloudSim packages and classes  
2.static variables that stores list od VMs and Cloudlets  
3.crafting VMs and cloudlets  
4.Datacenter creation (with properties (RAM, CPU, Cores, OS, Architecture etc)  
5.Broker Creation

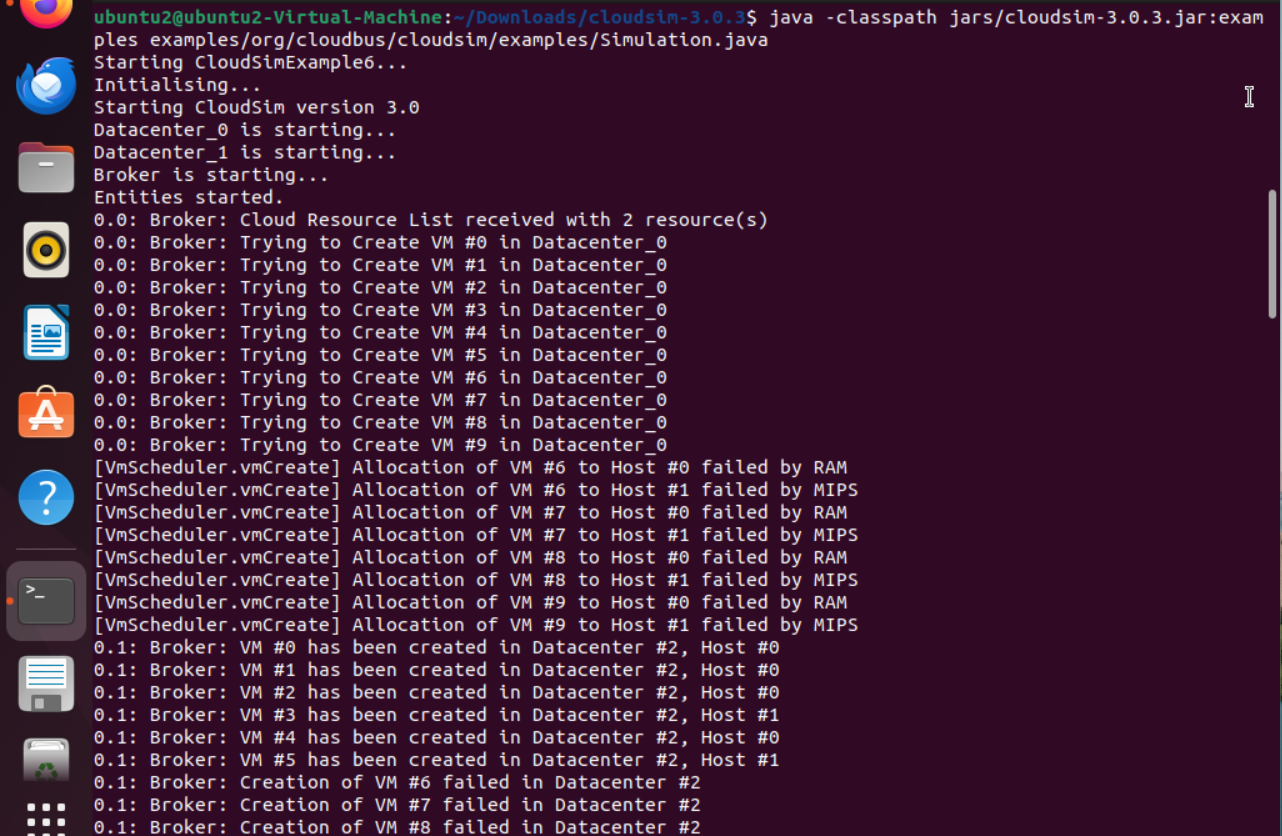
4. submission of cloudlets and VMs to broker  
5.simulation execution  
6.simulation results  
7.simulation termination  
8.PrintCloudletList (prints cloudlet details like IDs, status, DataCenter IDs, VM IDs, execution time, start times, finish times and UserIDs)

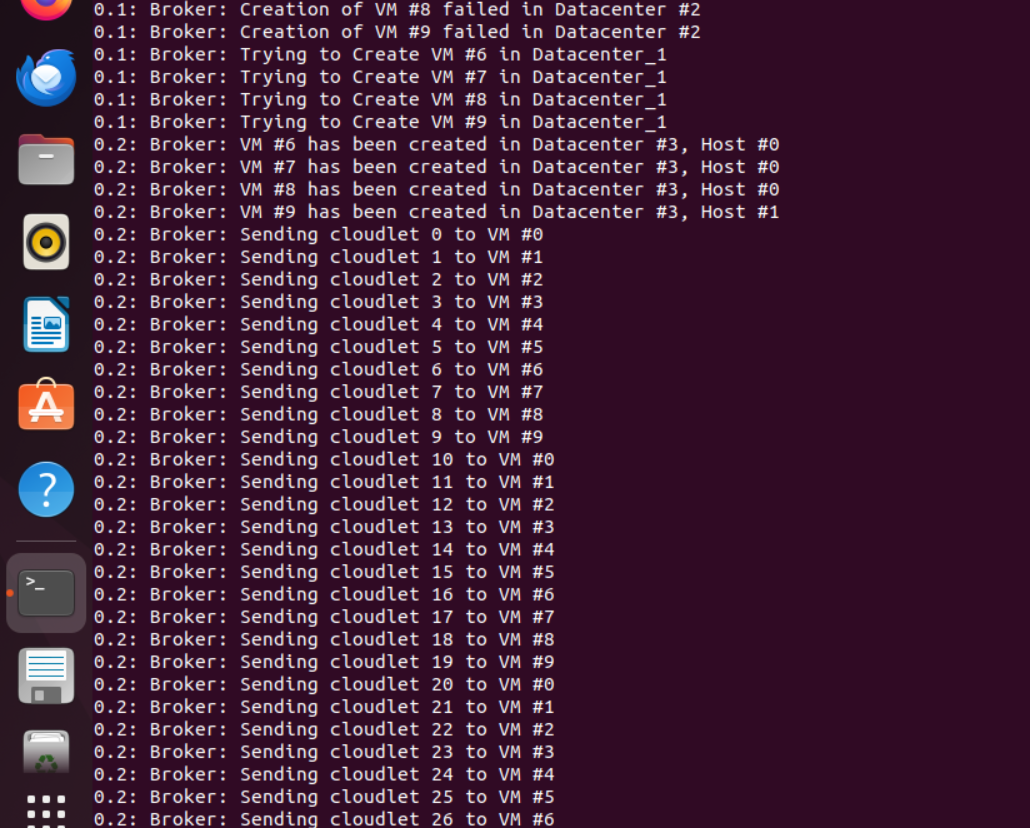
## **Running the Simulation**

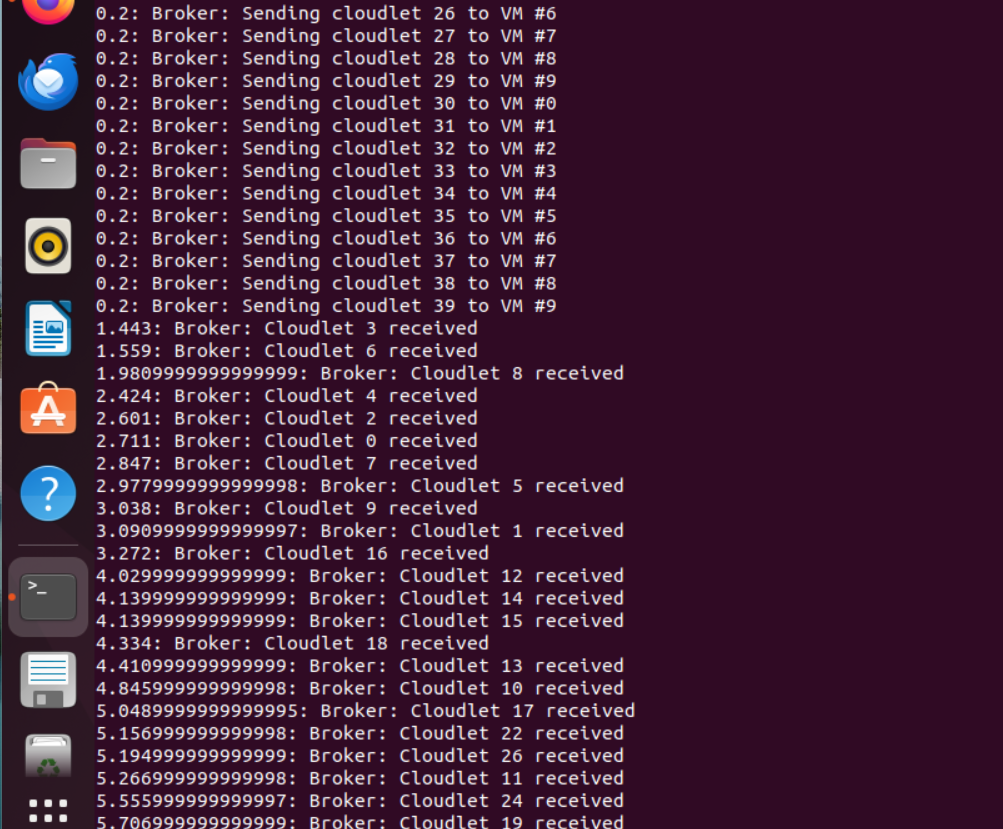
To run our simulation file, we used the command:

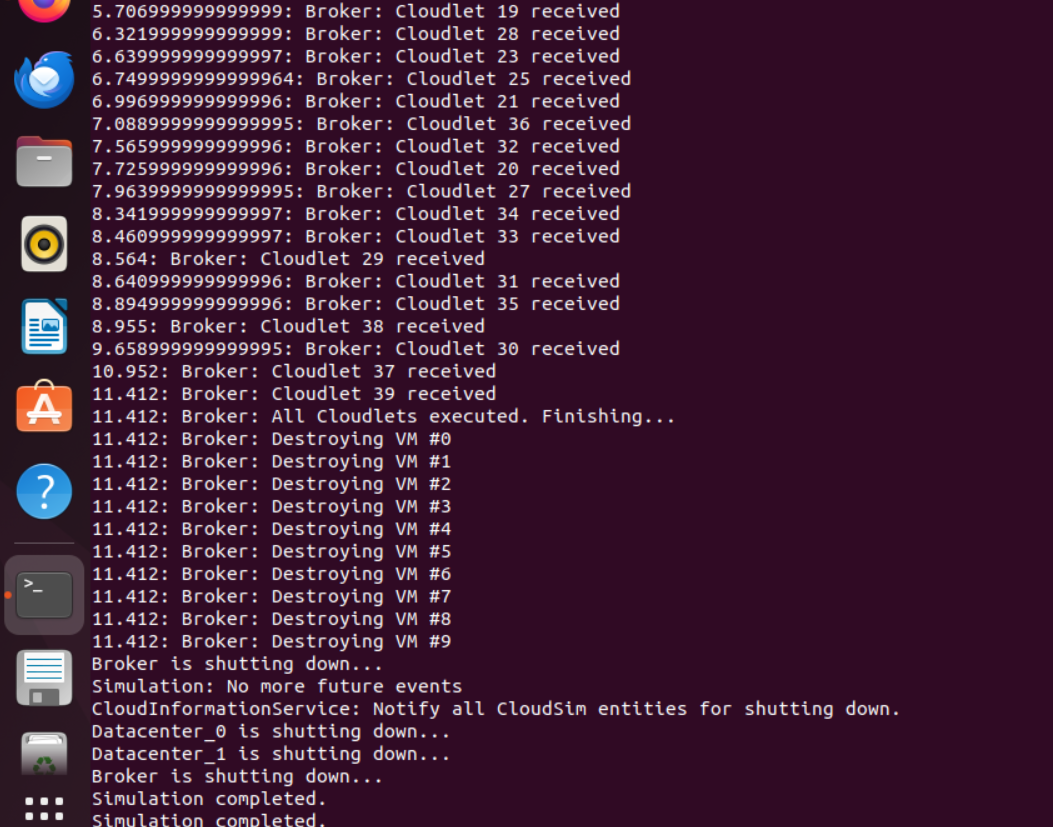
java -classpath java/cloudsim-3.0.3.jar:examples example/org /cloudbus/cloudsim/ examples/ simulation.java

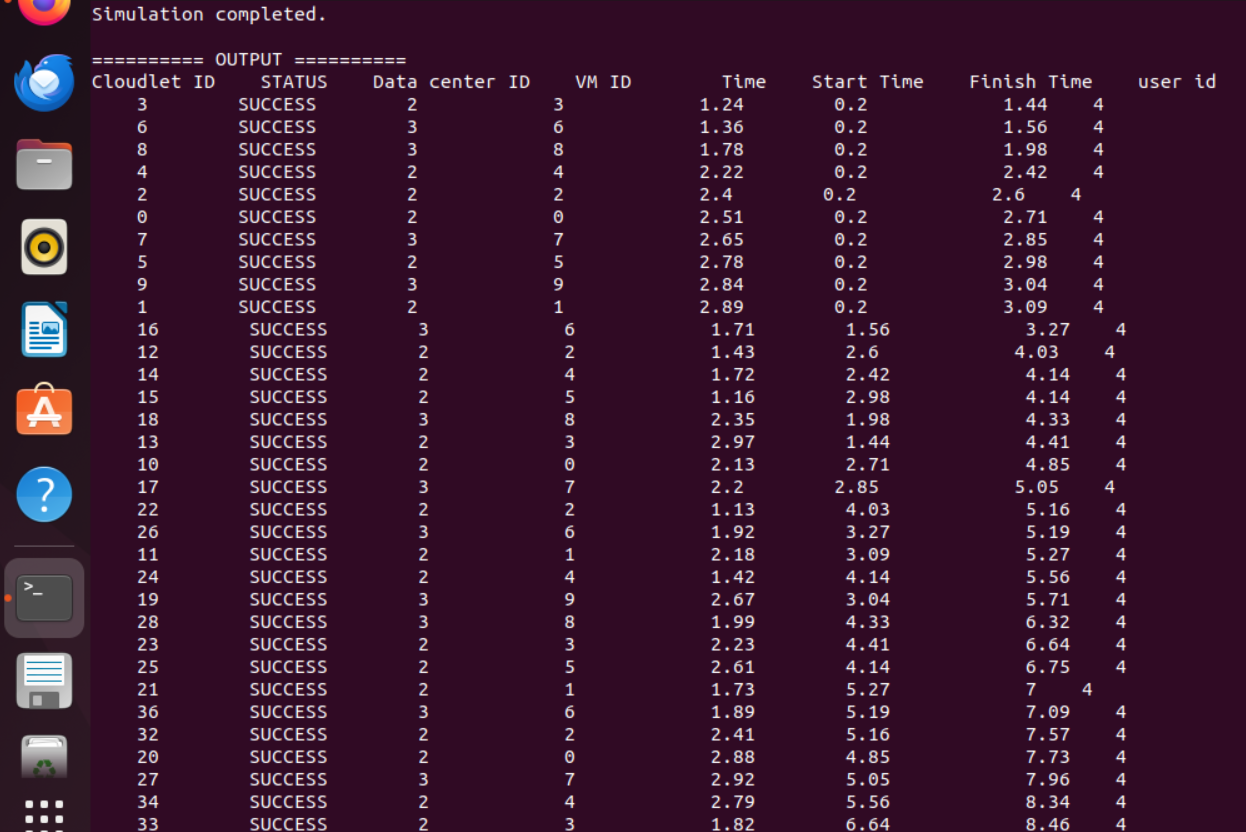
I discovered this command in the process outlined in the readme.txt and examples.txt files located in the CloudSim 3.0.3 folder we extracted after downloading. We also delved into various tutorials on simulating cloud environments.

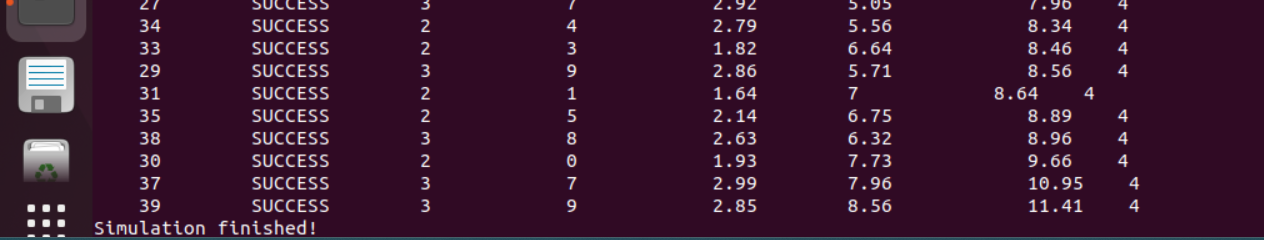












## **Demystifying Simulation Details**

In our mystical cloud realm, we've brewed up a simulation featuring not one but two Datacenters, a broker, 10 VMs, and 40 cloudlets.

Our scheduling wizardry is powered by "CloudletSchedulerSpaceShared," which is perfect for space-shared scheduling.

## **Code https://github.com/Ishu-dev/cloudComputingProjectSem7**

## **Glossary**

**Processing Elements (PEs):**

These are the brainiac cores inside a VM, responsible for churning through instructions and performing computations.

**Apache ANT:**

Think of this as your magical wand for building and deploying Java projects.

Datacenter Broker: This is the maestro within the CloudSim realm. It plays matchmaker between the cloud servers and the cloud infrastructure, orchestrating VM provisioning and cloudlet scheduling on behalf of the user.

**Cloudlet:**

Picture a cloudlet as a little spark of computation that leaps from a device to a nearby cloud server. It's a part of a larger application, and it dances its routine on the cloud server rather than the device, making our lives more efficient.

**Broker:**

In the world of cloud simulation, a broker is your trusty assistant. It represents you, the user, in the cloud computing world. It handles the nitty-gritty of resource allocation, including VMs and cloudlets, ensuring your needs and app demands are met.

## **Conclusion**

With CloudSim, you unlock the power to explore and experiment with cloud computing scenarios, scheduling algorithms, and resource allocation strategies. By setting up a virtual environment, installing the necessary tools, and crafting your simulations, you embark on a journey of discovery in the world of cloud computing