Practical - 6

# AIM: - Cloud Simulation Software Introduction: CloudSim

## What is CloudSim?

Cloud Computing is one of the hottest topics in town. It has completely transformed how modern-day applications are developed and maintained with high scalability and low latency.

**CloudSim** is an open-source framework, which is used to simulate cloud computing infrastructure and services. It is developed by the CLOUDS Lab organization and is written entirely in Java. It is used for modelling and simulating a cloud computing environment as a means for evaluating a hypothesis prior to software development in order to reproduce tests and results.

## Benefits of Simulation over the Actual Deployment:

Following are the benefits of CloudSim:

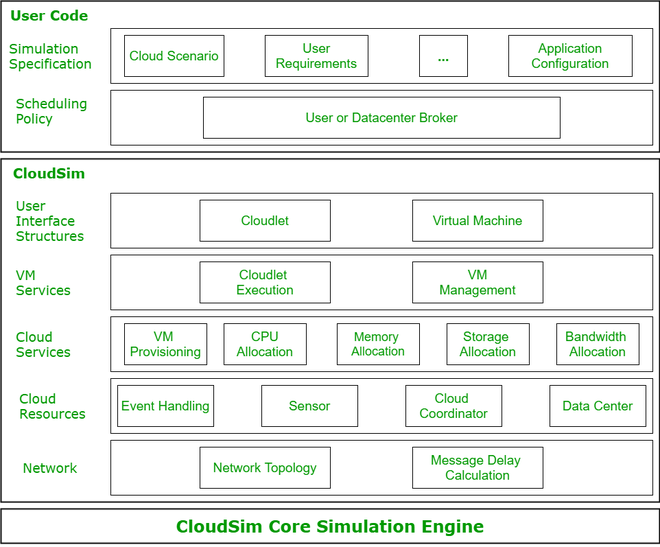
* **No capital investment involved**. With a simulation tool like CloudSim there is no installation or maintenance cost.
* **Easy to use and Scalable**. You can change the requirements such as adding or deleting resources by changing just a few lines of code.
* **Risks can be evaluated at an earlier stage**. In Cloud Computing utilization of real testbeds limits the experiments to the scale of the testbed and makes the reproduction of results an extremely difficult undertaking. With simulation, you can test your product against test cases and resolve issues before actual deployment without any limitations.
* **No need for try-and-error approaches**. Instead of relying on theoretical and imprecise evaluations which can lead to inefficient service performance and revenue generation, you can test your services in a repeatable and controlled environment free of cost with CloudSim.

## Why use CloudSim?

Below are a few reasons to opt for CloudSim:

* *Open source* and *free of cost*, so it favours researchers/developers working in the field.
* Easy to download and set-up.
* It is more *generalized* and *extensible* to support modelling and experimentation.
* Does not require any high-specs computer to work on.
* Provides *pre-defined allocation policies* and *utilization models* for managing resources, and allows implementation of user-defined algorithms as well.
* The documentation provides *pre-coded examples* for new developers to get familiar with the basic classes and functions.
* Tackle bottlenecks before deployment to reduce risk, lower costs, increase performance, and raise revenue.

## CloudSim Architecture:



*CloudSim Layered Architecture*

**CloudSim Core Simulation Engine** provides interfaces for the management of resources such as VM, memory and bandwidth of virtualized Datacenters.

**CloudSim** layer manages the creation and execution of core entities such as VMs, Cloudlets, Hosts etc. It also handles network-related execution along with the provisioning of resources and their execution and management.

**User Code** is the layer controlled by the user. The developer can write the requirements of the hardware specifications in this layer according to the scenario.

Some of the most common classes used during simulation are:

* **Datacenter**: used for modelling the foundational hardware equipment of any cloud environment, that is the Datacenter. This class provides methods to specify the functional requirements of the Datacenter as well as methods to set the allocation policies of the VMs etc.
* **Host**: this class executes actions related to management of virtual machines. It also defines policies for provisioning memory and bandwidth to the virtual machines, as well as allocating CPU cores to the virtual machines.
* **VM**: this class represents a virtual machine by providing data members defining a VM’s bandwidth, RAM, mips (million instructions per second), size while also providing setter and getter methods for these parameters.
* **Cloudlet**: a cloudlet class represents any task that is run on a VM, like a processing task, or a memory access task, or a file updating task etc. It stores parameters defining the characteristics of a task such as its length, size, mi (million instructions) and provides methods similarly to VM class while also providing methods that define a task’s execution time, status, cost and history.
* **DatacenterBroker**: is an entity acting on behalf of the user/customer. It is responsible for functioning of VMs, including VM creation, management, destruction and submission of cloudlets to the VM.
* **CloudSim**: this is the class responsible for initializing and starting the simulation environment after all the necessary cloud entities have been defined and later stopping after all the entities have been destroyed.

## Features of CloudSim:

CloudSim provides support for simulation and modelling of:

1. Large scale virtualized Datacenters, servers and hosts.
2. Customizable policies for provisioning host to virtual machines.
3. Energy-aware computational resources.
4. Application containers and federated clouds (joining and management of multiple public clouds).
5. Datacenter network topologies and message-passing applications.
6. Dynamic insertion of simulation entities with stop and resume of simulation.
7. User-defined allocation and provisioning policies.

Installation:

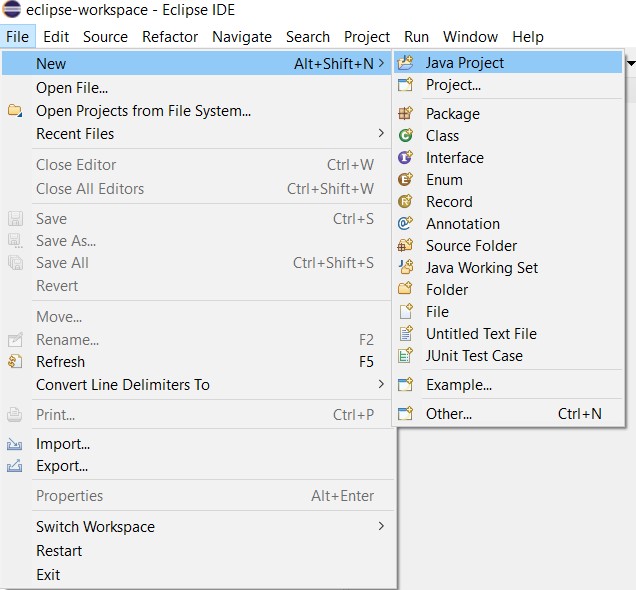
***Prerequisites:***

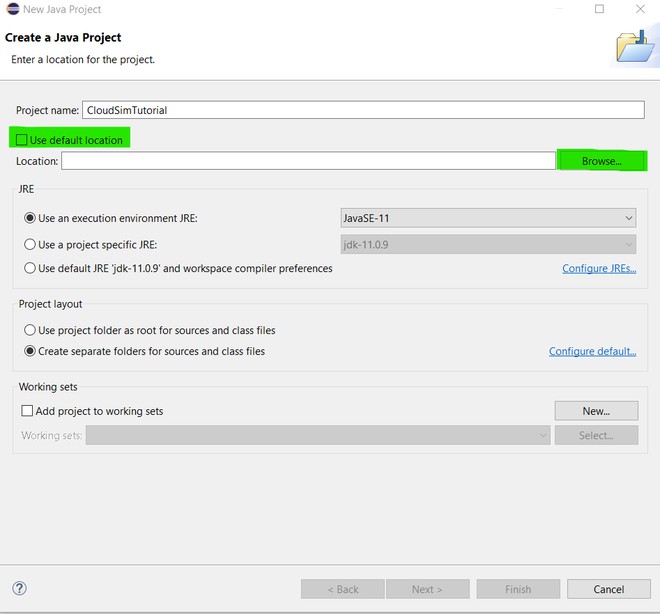
* Knowledge of Core Java language features such as [OOP](https://www.geeksforgeeks.org/object-oriented-programming-oops-concept-in-java/) and [Collections](https://www.geeksforgeeks.org/collections-in-java-2/).
* Basics of [Cloud Computing](https://www.geeksforgeeks.org/cloud-computing/).
* CloudSim is available for download from website.
* For this, we have downloaded zip file of CloudSim 3.0.3.

***Note:*** *CloudSim also uses some utilities of Apache’s commons-math3 library. Download its Binaries zip file*

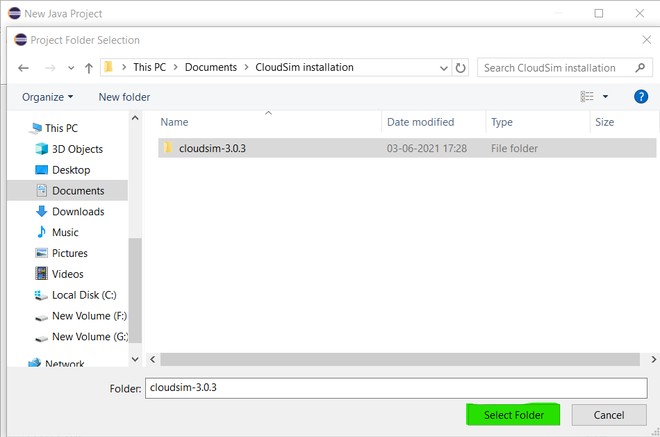
**Step 1:** From the zip folder extracts *cloudsim-3.0.3* into a folder. Also, extract the *commons-math3-3.6.1 jar* into the same folder.



**Step 2:** Open Eclipse IDE and go to *File -> New -> Java Project.*

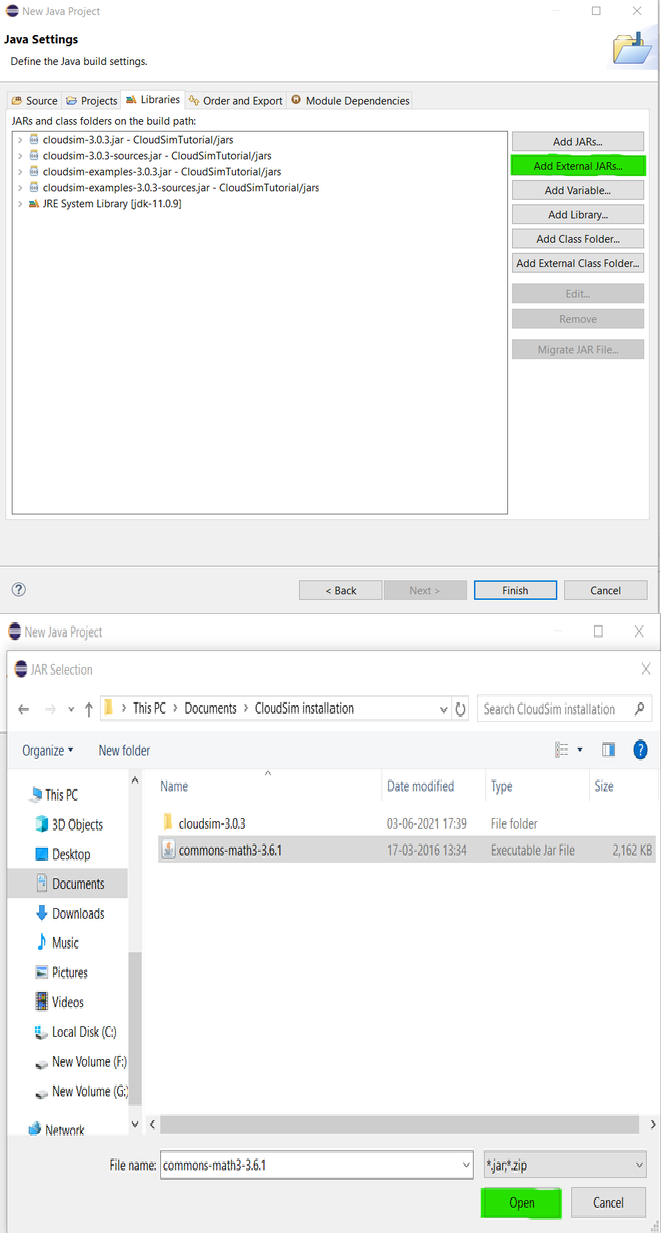
**Step 3:** Enter any name for your project and then uncheck the *Use default location* box just under it and click on *Browse*.

Browse to the folder where you extracted your files and select the *cloudsim-3.0.3* folder.



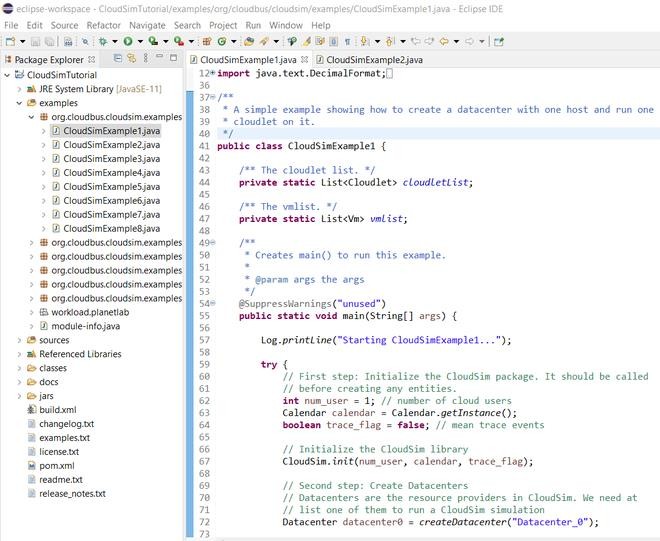
Don’t click on *Finish* yet, because we need to add a jar file to our project.

**Step 4** Click *Next* and go to ***Libraries -> Add External JARs****.* Now browse to the same folder where you extracted your *commons-math3* jar file and *Open* it.



**Step 5** Finally click on *Finish* and wait for the project to build. After the project has been built, from the *Project Explorer* you can click on your project and from the dropdown

go-to *examples -> org.cloudbus.cloudsim.examples* where you can find pre-written sample codes and try to run them.



## Scope

With the flexibility and generalizability of the CloudSim framework, it is easy to model heavy cloud environments which would otherwise require experimentation on paid computing infrastructures. Extensible capabilities of scaling the infrastructure and resources to fit any scenario helps in fast and efficient research of several topics in cloud computing.

CloudSim has been used in several areas of research such as:

* Task Scheduling
* Green Computing
* Resource Provisioning
* Secure Log Forensics.