

Ex. No: 4

Date:

Simulation of VM Scheduling Using CloudSim

AIM

To simulate VM Scheduling using CloudSim.

DESCRIPTION

CloudSim

- A Framework for modeling and simulation of Cloud Computing Infrastructures and services
- Originally built at the Cloud Computing Distributed Systems (CLOUDS) Laboratory, The University of Melbourne, Australia
- It is completely written in JAVA

Main Features of CloudSim

- Modeling and simulation
- Data center network topologies and message-passing applications
- Dynamic insertion of simulation elements
- Stop and resume of simulation
- Policies for allocation of hosts and virtual machines

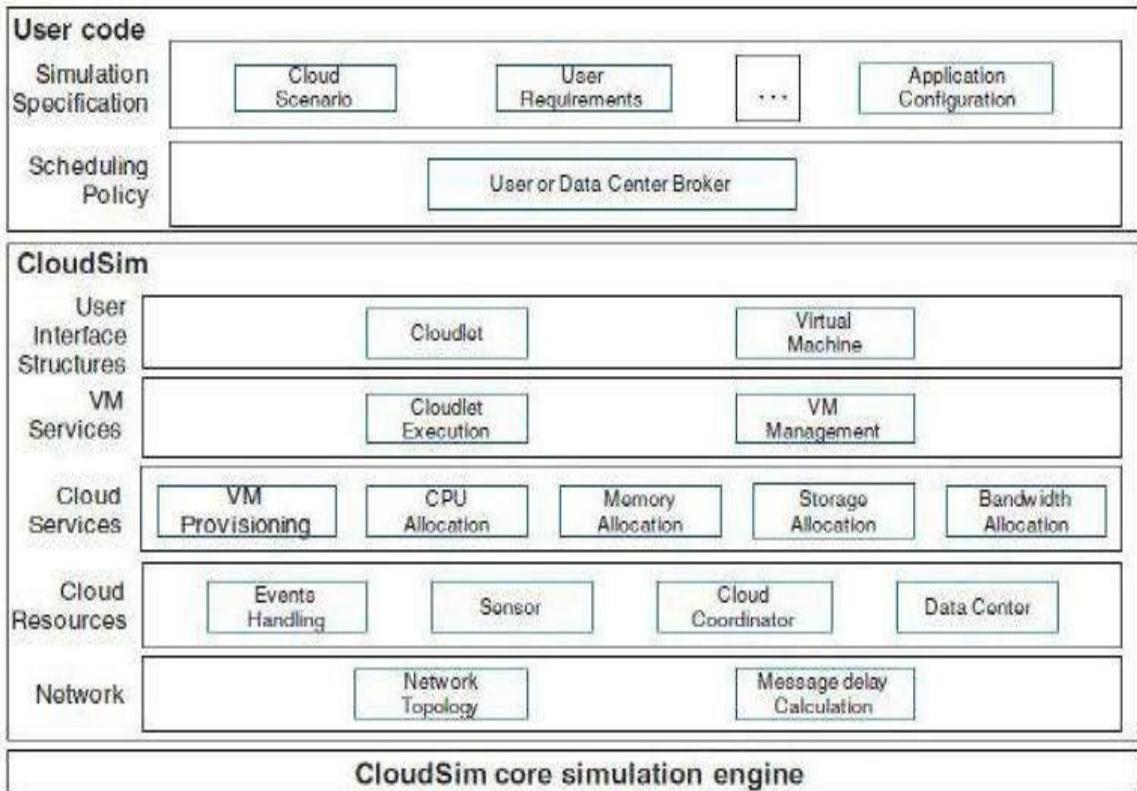
CloudSim – Essentials

- JDK 1.6 or above <http://tinyurl.com/JNU-JAVA>
- Eclipse 4.2 or above <http://tinyurl.com/JNU-Eclipse>
- Alternatively NetBeans <https://netbeans.org/downloads>
- Up & Running with cloudsim guide: <https://goo.gl/TPL7Zh>

CloudSim – Directory structure

- cloudsim/ -- top level CloudSim directory
- docs/ -- CloudSim API Documentation
- examples/ -- CloudSim examples
- jars/ -- CloudSim jar archives
- sources/ -- CloudSim source code

CloudSim - Layered Architecture



- **Core Functionalities** are queuing and processing of events, creation of Cloud system entities (services, host, data center, broker, VMs), communication between components, and management of the simulation clock.
- **CloudSim simulation layer** provides support for modeling and simulation of virtualized Cloud-based data center environments including dedicated management interfaces for VMs, memory, storage, and bandwidth.
- A Cloud host can be concurrently allocated to a set of VMs that execute applications based on SaaS provider's defined QoS levels.
- Top-most layer in the CloudSim stack is the User Code that exposes **basic entities for hosts** (number of machines, their specification, and so on), **applications** (number of tasks and their requirements), **VMs, number of users and their application types, and broker scheduling policies**.

Datacenter Entity

- The infrastructure-level services (IaaS) related to the clouds can be simulated by **extending the data center entity** of CloudSim.
- Data center entity **manages** several **host entities**.
- Hosts are assigned to one or more VMs based on a VM allocation policy that should be defined by the Cloud service provider.
- VM policy stands for the operations control policies related to VM life cycle such as provisioning of a host to a VM, VM creation, VM destruction, and VM migration.

Host Entity

- Represents a physical computing server in a Cloud.
- It assigns a pre-configured processing capability (expressed in millions of instructions per second—MIPS), memory, storage, and a provisioning policy for allocating processing cores to VMs.
- The Host component implements interfaces that support modeling and simulation of both single-core and multi-core nodes.

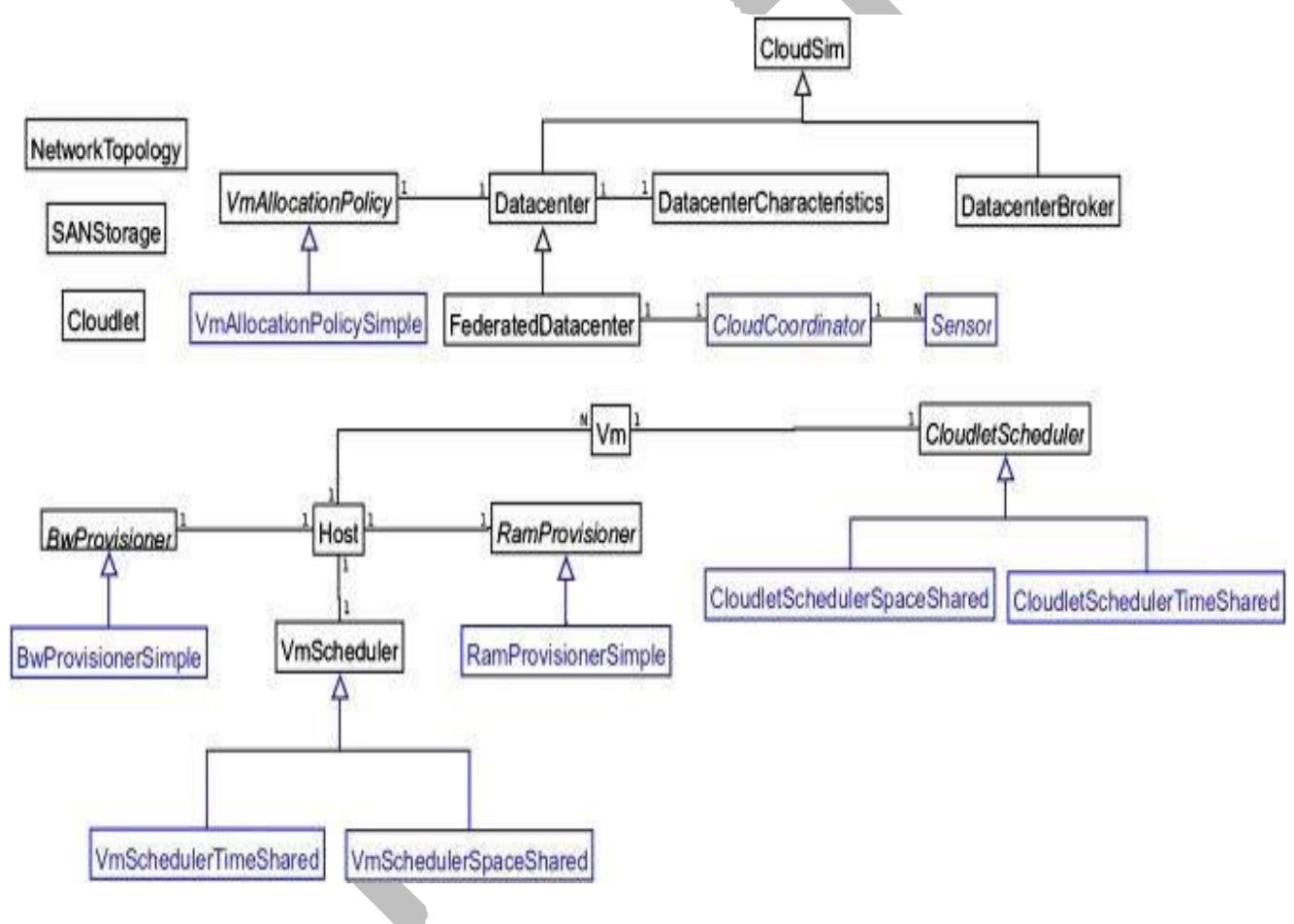
VM Allocation

- Process of creating VM instances on hosts that match the critical characteristics (storage, memory), configurations (software environment), and requirements (availability zone) of the SaaS provider.
- Once an application service (Cloudlet) is defined and modeled, it is assigned to one or more pre-instantiated VMs through a service-specific allocation policy.
- VmAllocationPolicy - VM Allocation controller component to allocate application-specific VMs to hosts in a data center. new policies can be created by the user by extending VmAllocationPolicy Entity.
- Default VmAllocationPolicy is FCFS.
- Hardware requirements, such as the number of processing cores, memory, and storage, form the basis for such provisioning.
- Other policies, including the ones likely to be expressed by Cloud providers, can also be easily simulated and modeled in CloudSim.
- Policies used by public Cloud providers (Amazon EC2, Microsoft Azure) are not publicly available

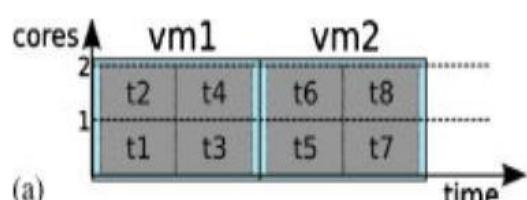
VM scheduler

- Allocation of processing cores to VMs is done by VM Scheduler using host allocation policy.
- Hardware characteristics, such as number of CPU cores, CPU share, and amount of memory (physical and secondary), that are allocated to a given VM instance
- Types:
 - Space Shared:** Assign specific CPU cores to specific VMs
 - Time Shared:** Dynamically distribute the capacity of a core among VMs

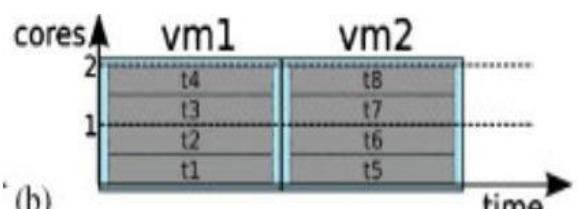
CloudSim – Component Model Classes



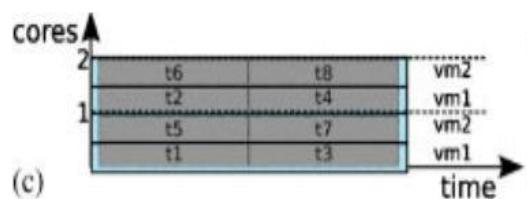
Modelling VM Allocation



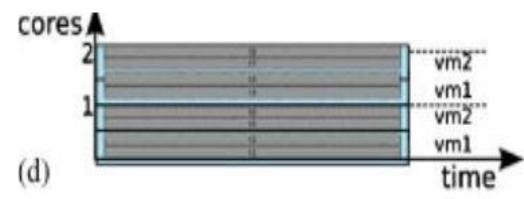
Space-shared provisioning for
VMs and tasks



Space-shared provisioning for VMs
and time-shared provisioning for tasks

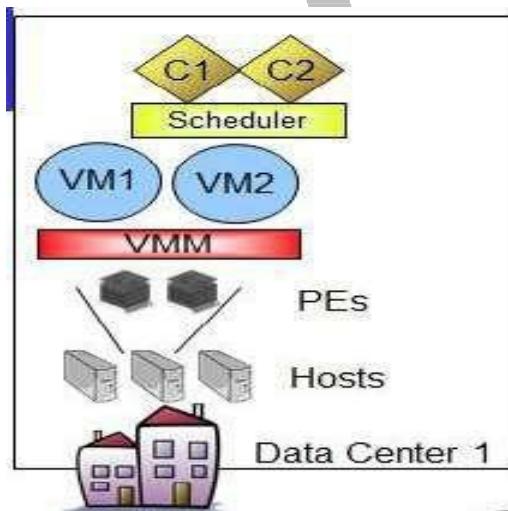


Time-shared provisioning for VMs, space-shared
provisioning for tasks

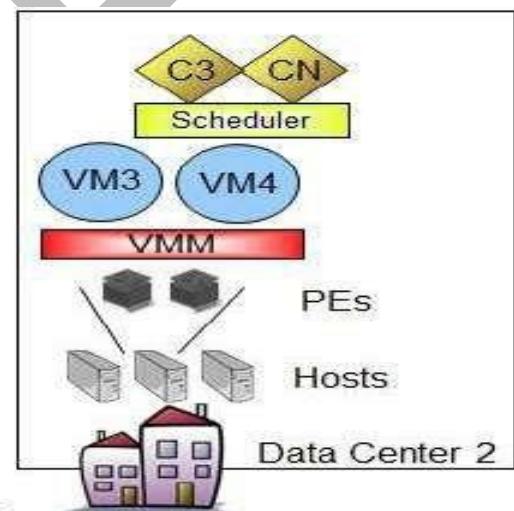


Time-shared provisioning for VMs and tasks

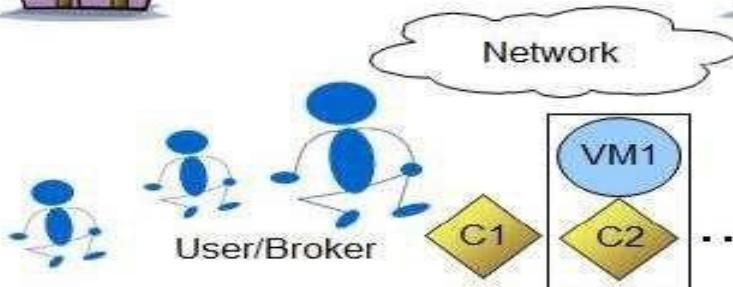
CloudSim Elements / Components



Data Center 1



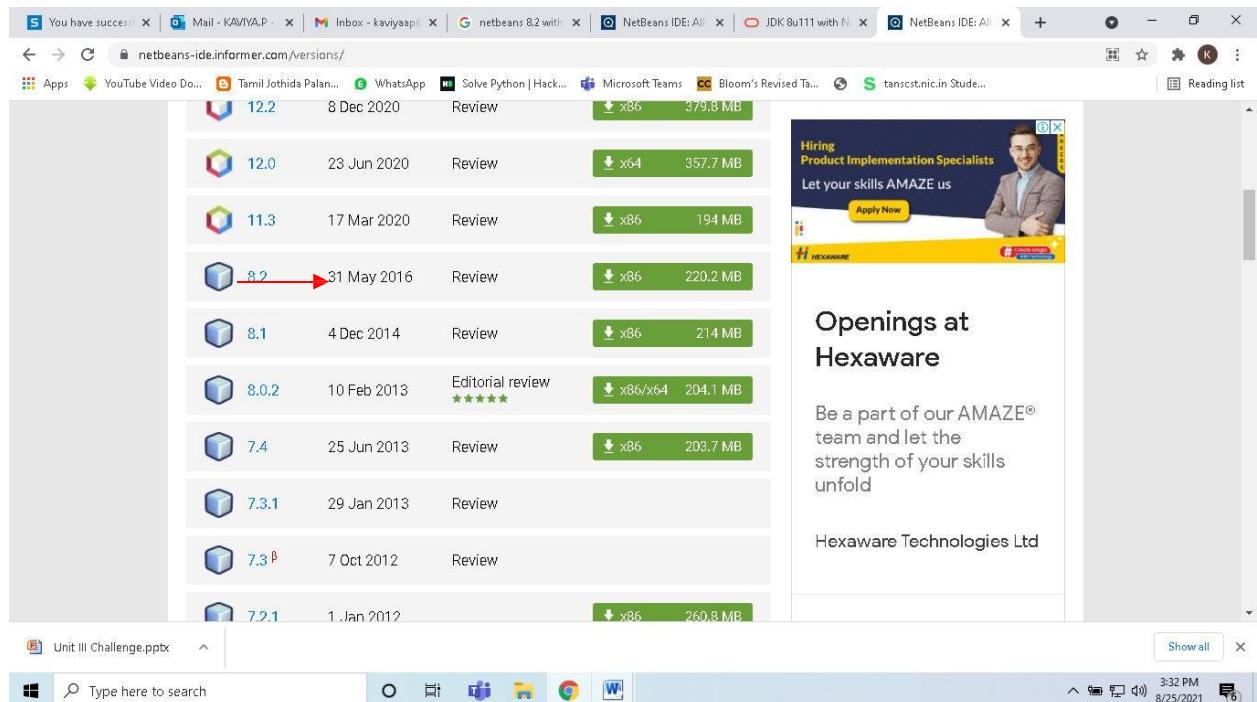
Data Center 2



PROCEDURAL STEPS

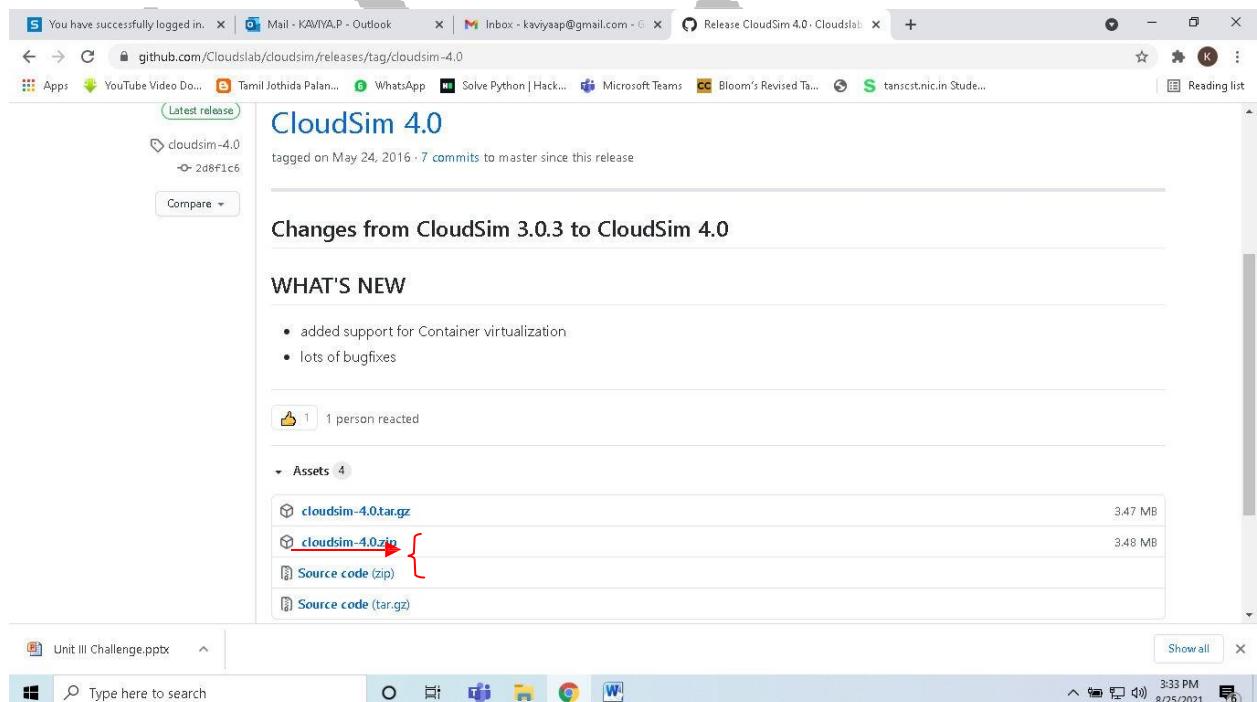
1. Download NetBeans and intall it.

Link: <https://netbeans-ide.informer.com/versions/>

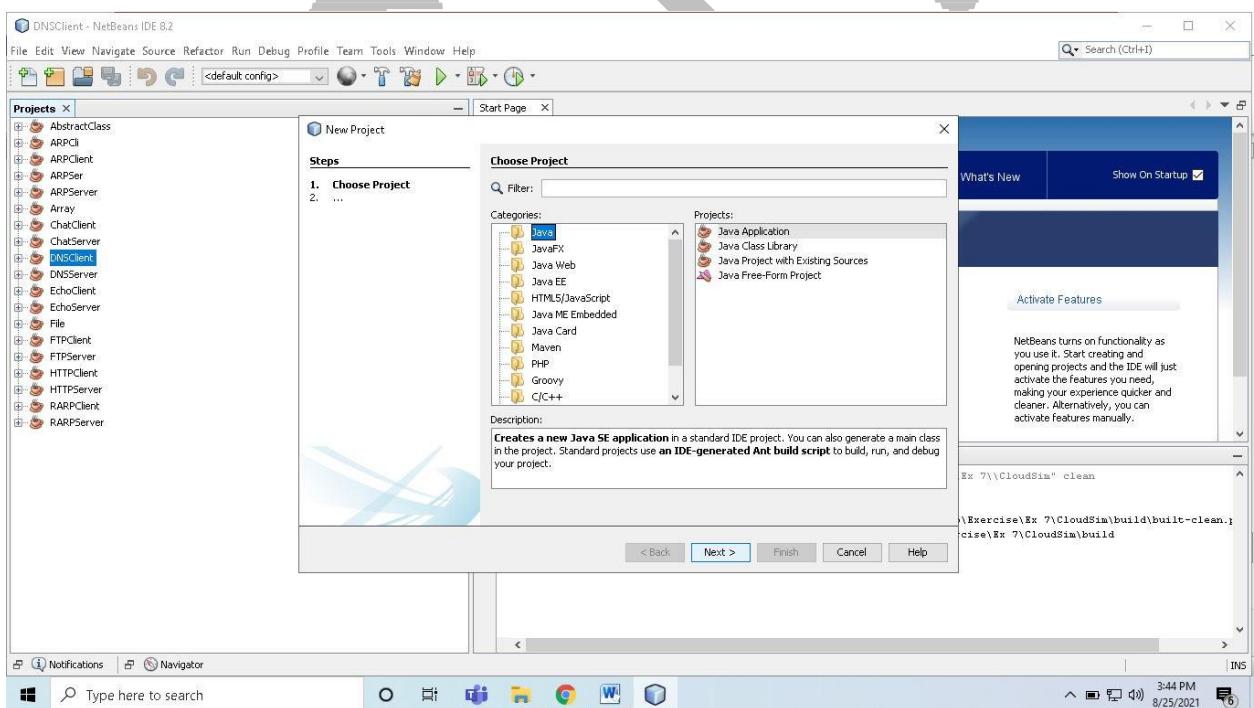
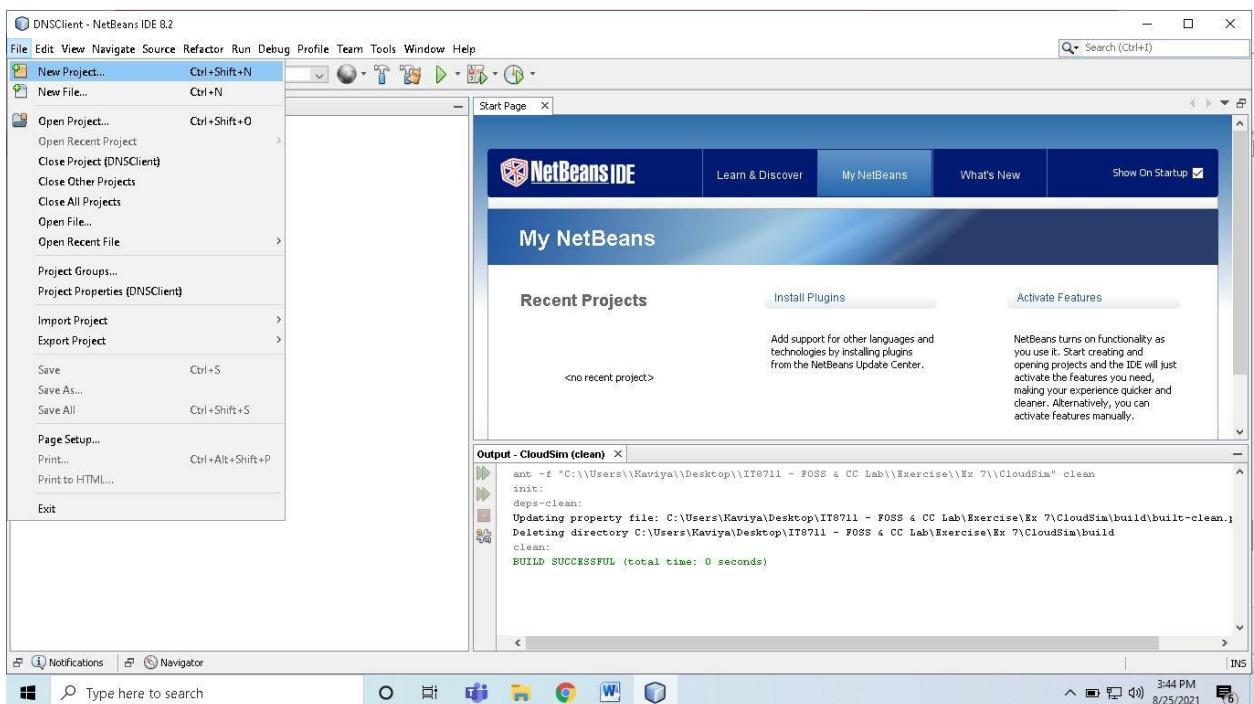


2. Download CloudSim 4.0 (jar and source code), extract and place it in respective folders.

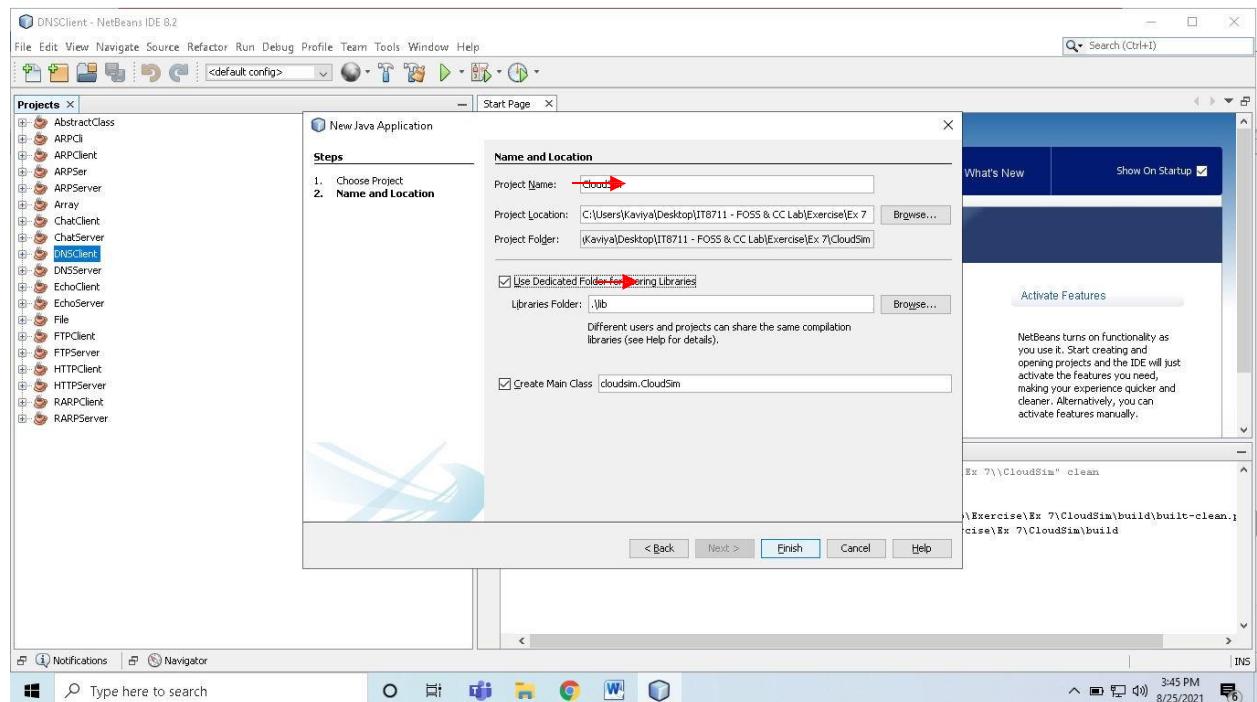
Link: <https://github.com/Cloudslab/cloudsim/releases/tag/cloudsim-4.0>



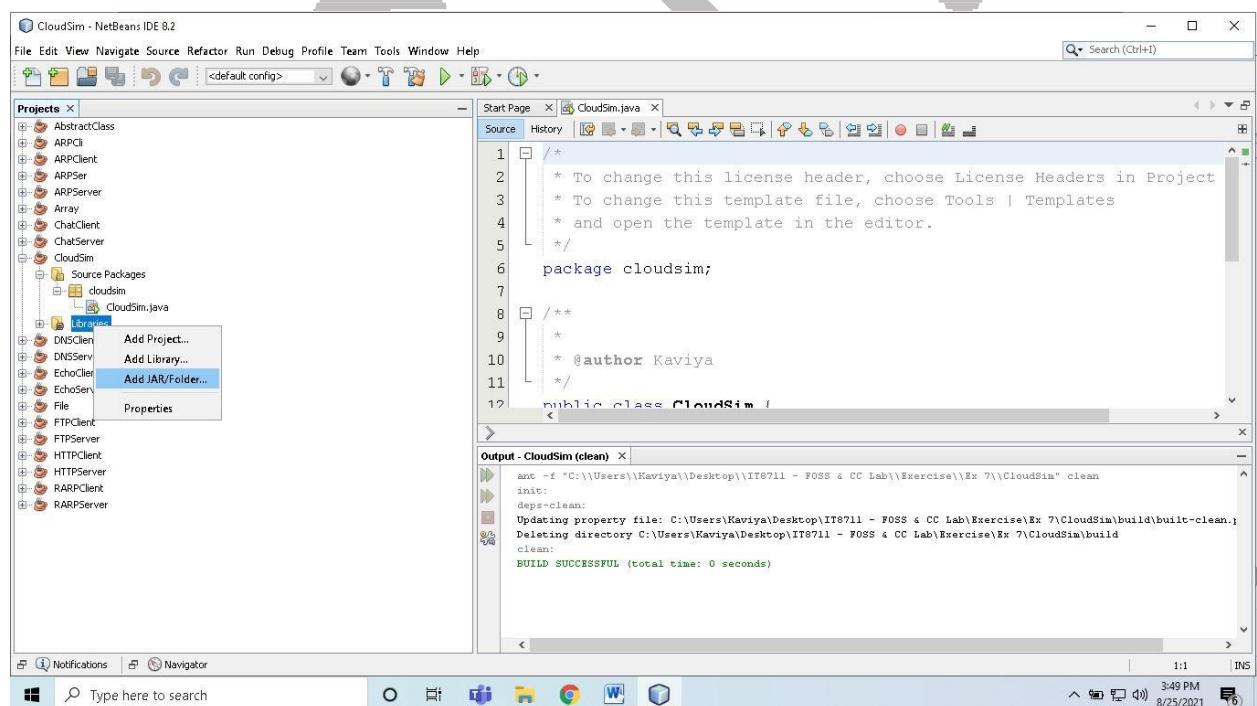
3. Open NetBeans & Create a project “CloudSim”

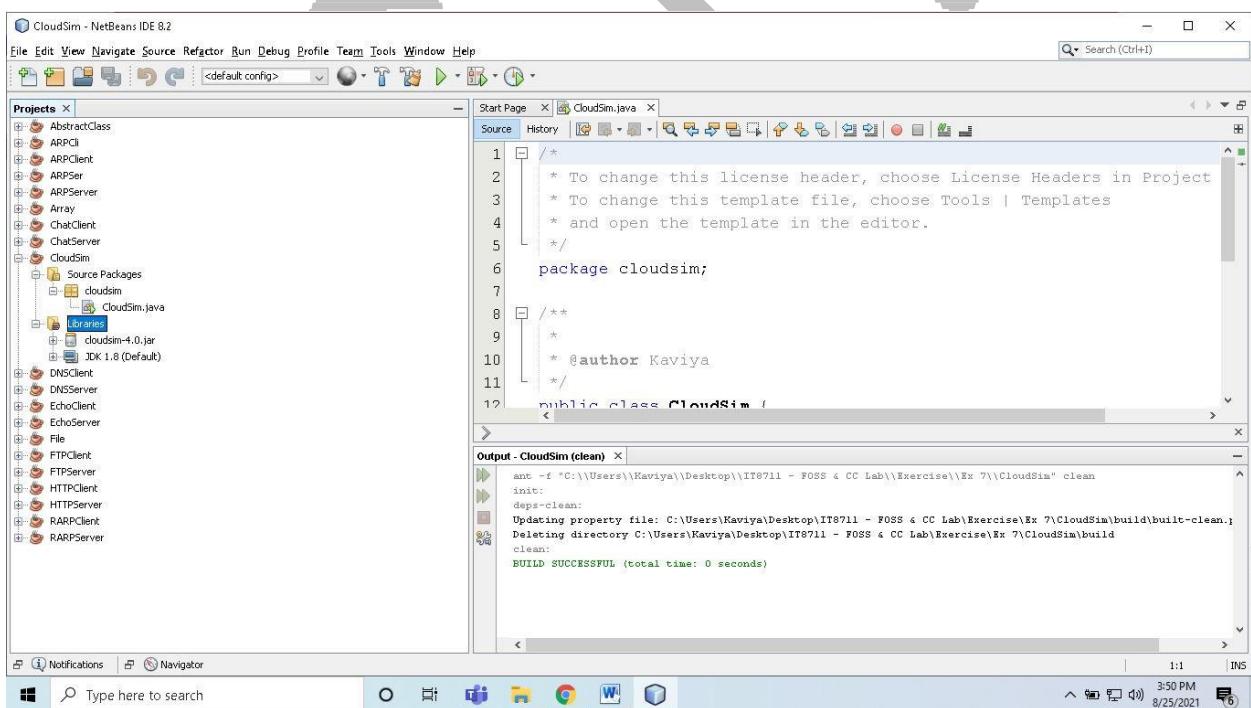
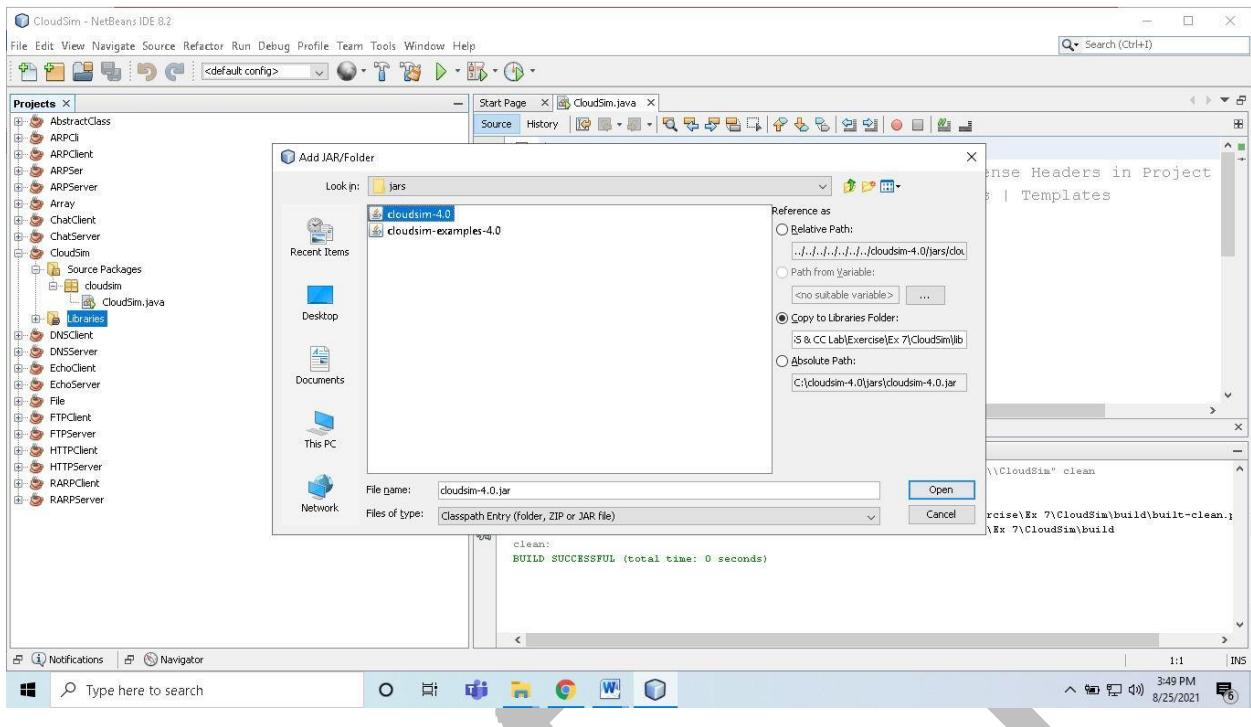


Check “Use dedicated folder for Storing Libraries” → Click “Finish”

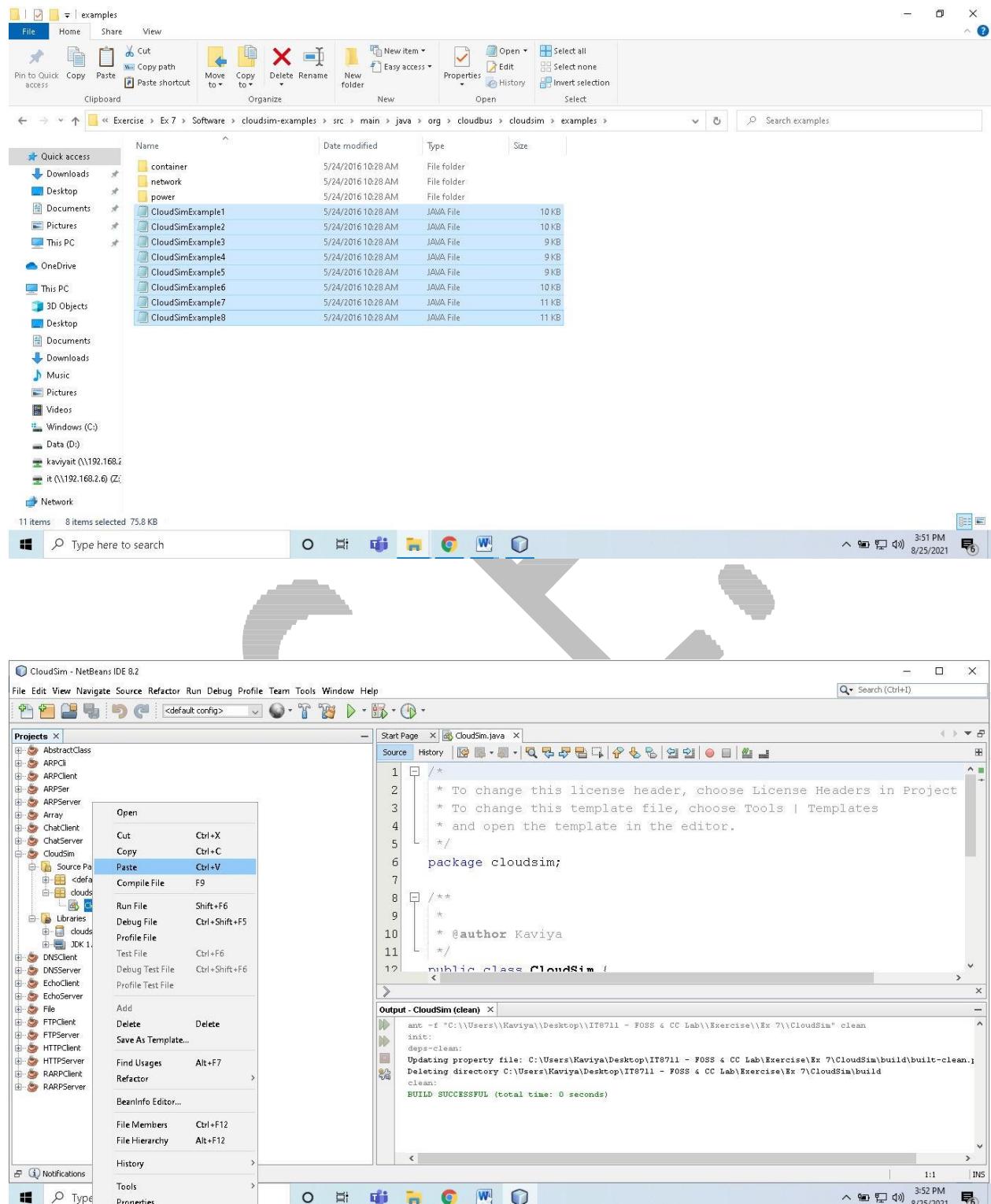


4. Add jar file “cloudsim-4.0”





5. Add “cloudsim examples” in CloudSim Project



CloudSim - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Start Page | CloudSim.java | CloudSimExample1.java | CloudSimExample2.java

Search (Ctrl+F)

Projects

- AbstractClass
- ARPCli
- ARPClient
- ARPSer
- ARPServer
- Array
- ChatClient
- ChatServer
- CloudSim
- Source Packages
 - cloudsim
 - CloudSim.java
 - CloudSimExample1.java
 - CloudSimExample2.java
 - CloudSimExample3.java
 - CloudSimExample4.java
 - CloudSimExample5.java
 - CloudSimExample6.java
 - CloudSimExample7.java
 - CloudSimExample8.java
- Libraries
 - cloudsim-4.0.jar
 - JDK 1.8 (Default)
- DNSClient
- DNSServer
- EchoClient
- EchoServer
- File
- FTPClient
- FTPServer
- HTTPClient
- HTTPServer
- Random

Output - CloudSim (clean) | 10:1 IN5

```
ant -f "C:\\Users\\Kaviya\\Desktop\\IT8711 - FOSS & CC Lab\\Exercise\\Ex 7\\CloudSim" clean
init:
deps=clean:
Created dir: C:\\Users\\Kaviya\\Desktop\\IT8711 - FOSS & CC Lab\\Exercise\\Ex 7\\CloudSim\\build
Updating property file: C:\\Users\\Kaviya\\Desktop\\IT8711 - FOSS & CC Lab\\Exercise\\Ex 7\\CloudSim\\build\\built-clean.l
Deleting directory C:\\Users\\Kaviya\\Desktop\\IT8711 - FOSS & CC Lab\\Exercise\\Ex 7\\CloudSim\\build
clean:
BUILD SUCCESSFUL (total time: 0 seconds)
```

Type here to search

CloudSim - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Start Page | CloudSim.java | CloudSimExample1.java | CloudSimExample2.java

Search (Ctrl+F)

Projects

- AbstractClass
- ARPCli
- ARPClient
- ARPSer
- ARPServer
- Array
- ChatClient
- ChatServer
- CloudSim
- Source Packages
 - cloudsim
 - CloudSim.java
 - CloudSimExample1.java
 - CloudSimExample2.java
 - CloudSimExample3.java
 - CloudSimExample4.java
 - CloudSimExample5.java
 - CloudSimExample6.java
 - CloudSimExample7.java
 - CloudSimExample8.java
- Libraries
 - cloudsim-4.0.jar
 - JDK 1.8 (Default)
- DNSClient
- DNSServer
- EchoClient
- EchoServer
- File
- FTPClient
- FTPServer
- HTTPClient
- HTTPServer
- Random

Output - CloudSim (run) | 10:1 IN5

```
52   : vmlist. */
53   : static List<Vm> vmlist;
54
55   states main() to run this example
56
57   static void main(String[] args) {
58
59     Log.printLine("Starting CloudSimExample2...");
60
61     try {
62       // First step: Initialize the CloudSim package. It should be called
63       // before creating any entities.
64       int num_user = 1; // number of cloud users
65       Calendar calendar = Calendar.getInstance();
66       boolean trace_flag = false; // mean trace events
67
68       // Initialize the CloudSim library
69       CloudSim.init(num_user, calendar, trace_flag);
70
71       // Second step: Create Datacenters
72       //Datacenters are the resource providers in CloudSim. We need at least one of them to
73       @SuppressWarnings("unused")
74     }
```

Type here to search

CloudSim - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

<default config>

Projects x

- AbstractClass
- ARPCli
- ARPClient
- ARPSer
- ARPServer
- Array
- ChatClient
- ChatServer
- CloudSim
- Source Packages cloudsim
 - CloudSim.java
 - CloudSimExample1.java
 - CloudSimExample2.java
 - CloudSimExample3.java
 - CloudSimExample4.java
 - CloudSimExample5.java
 - CloudSimExample6.java
 - CloudSimExample7.java
 - CloudSimExample8.java
- Libraries cloudsim-4.0.jar JDK 1.8 (Default)
- DNSClient
- DNSServer
- EchoClient
- EchoServer
- File
- FTPClient
- FTPServer
- HTTPClient
- HTTPServer
- Random

Start Page x CloudSim.java x CloudSimExample1.java x CloudSimExample2.java x

```
    // VM DESCRIPTION
    int vmid = 0;
    int mips = 250;
    long size = 10000; //image size (MB)
    int ram = 512; //vm memory (MB)
    long bw = 1000;
    int pesNumber = 1; //number of cpus
    String vmm = "Xen"; //VMM name

    //create two VMs
    Vm vm1 = new Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, new CloudletSchedulerRoundRobin());
    vmid++;

    Vm vm2 = new Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, new CloudletSchedulerRoundRobin());

    //add the VMs to the vmList
    vmList.add(vm1);
    vmList.add(vm2);

    //submit vm list to the broker
    broker.submitVmList(vmList);

    //Fifth step: Create two Cloudlets

```

Output - CloudSim (run) x

```
>>> run:
```

10:1 4:02 PM 8/25/2021 IN5

CloudSim - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

<default config>

Projects x

- AbstractClass
- ARPCli
- ARPClient
- ARPSer
- ARPServer
- Array
- ChatClient
- ChatServer
- CloudSim
- Source Packages cloudsim
 - CloudSim.java
 - CloudSimExample1.java
 - CloudSimExample2.java
 - CloudSimExample3.java
 - CloudSimExample4.java
 - CloudSimExample5.java
 - CloudSimExample6.java
 - CloudSimExample7.java
 - CloudSimExample8.java
- Libraries cloudsim-4.0.jar JDK 1.8 (Default)
- DNSClient
- DNSServer
- EchoClient
- EchoServer
- File
- FTPClient
- FTPServer
- HTTPClient
- HTTPServer
- Random

Start Page x CloudSim.java x CloudSimExample1.java x CloudSimExample2.java x

```
    cloudletList = new ArrayList<Cloudlet>();

    //Cloudlet properties
    int id = 0;
    pesNumber=1;
    long length = 250000;
    long fileSize = 300;
    long outputSize = 300;
    UtilizationModel utilizationModel = new UtilizationModelFull();

    Cloudlet cloudlet1 = new Cloudlet(id, length, pesNumber, fileSize, outputSize, utilizationModel);
    cloudlet1.setUserId(brokerId);

    id++;
    Cloudlet cloudlet2 = new Cloudlet(id, length, pesNumber, fileSize, outputSize, utilizationModel);
    cloudlet2.setUserId(brokerId);

    //add the cloudlets to the list
    cloudletList.add(cloudlet1);
    cloudletList.add(cloudlet2);

    //submit cloudlet list to the broker
    broker.submitCloudletList(cloudletList);


```

Output - CloudSim (run) x

```
>>> run:
```

10:1 4:02 PM 8/25/2021 IN5

CloudSim - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Start Page | CloudSim.java | CloudSimExample1.java | CloudSimExample2.java

Projects | Source History | Search (Ctrl+F)

```

133 //bind the cloudlets to the vms. This way, the broker
134 // will submit the bound cloudlets only to the specific VM
135 broker.bindCloudletToVm(cloudlet1.getId(),vm1.getId());
136 broker.bindCloudletToVm(cloudlet2.getId(),vm2.getId());
137
138 // Sixth step: Starts the simulation
139 CloudSim.startSimulation();
140
141
142 // Final step: Print results when simulation is over
143 List<Cloudlet> newList = broker.getCloudletReceivedList();
144
145 CloudSim.stopSimulation();
146
147 printCloudletList(newList);
148
149 Log.printLine("CloudSimExample2 finished!");
150
151 catch (Exception e) {
152     e.printStackTrace();
153     Log.printLine("The simulation has been terminated due to an unexpected error");
154 }
155

```

Output - CloudSim (run) | run:

4:02 PM 8/25/2021 IN5

6. Run the CloudSim examples

CloudSim - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Start Page | CloudSim.java | CloudSimExample1.java | CloudSimExample2.java

Projects | Source History | Search (Ctrl+F)

Open

- Cut Ctrl+X
- Copy Ctrl+C
- Paste Ctrl+V
- Compile File F9
- Run File Shift+F6**
- Debug File Ctrl+Shift+F5
- Profile File
- Test File Ctrl+F6
- Debug Test File Ctrl+Shift+F6
- Profile Test File

Add

Delete

Save As Template...

Find Usages Alt+F7

Refactor >

BeanInfo Editor...

File Members Ctrl+F12

File Hierarchy Alt+F12

History >

Tools

Properties

```

1 /*
2  * Title:      CloudSim Toolkit
3  * Description: CloudSim (Cloud Simulation) Toolkit for Modeling a
4  *              of Clouds
5  * Licence:    GPL - http://www.gnu.org/copyleft/gpl.html
6  *
7  * Copyright (c) 2009, The University of Melbourne, Australia
8  */
9
10 package cloudsim;
11
12

```

Output - CloudSim (clean) | clean:

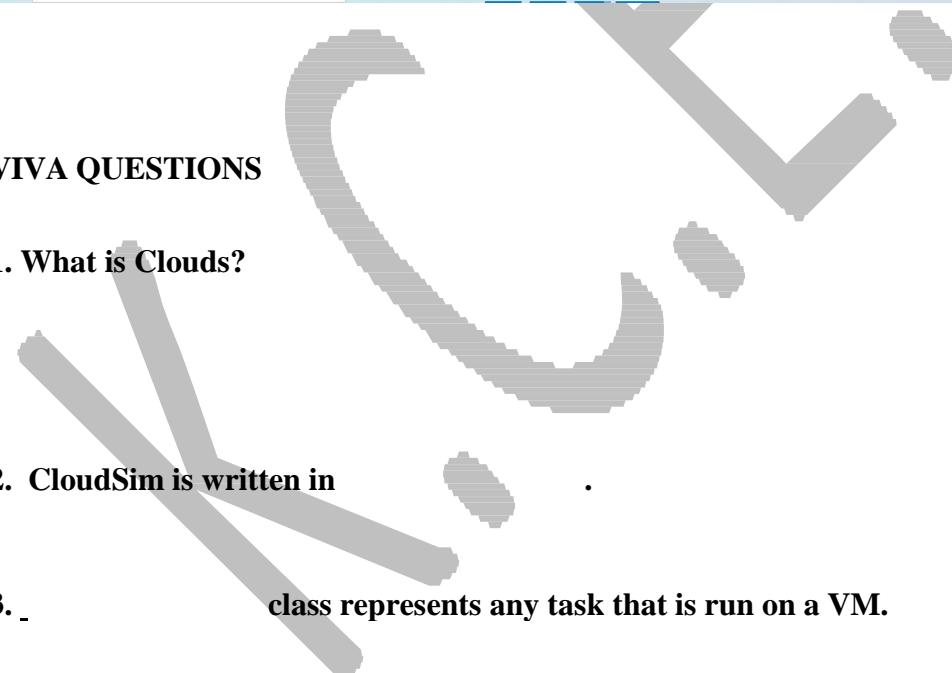
```

ant -f "C:\\Users\\Kaviya\\Desktop\\IT8711 - FOSS & CC Lab\\Exercise\\Ex 7\\CloudSim" clean
init:
deps=clean:
Created dir: C:\\Users\\Kaviya\\Desktop\\IT8711 - FOSS & CC Lab\\Exercise\\Ex 7\\CloudSim\\build
Updating property file: C:\\Users\\Kaviya\\Desktop\\IT8711 - FOSS & CC Lab\\Exercise\\Ex 7\\CloudSim\\build\\built-clean.ln
Deleting directory C:\\Users\\Kaviya\\Desktop\\IT8711 - FOSS & CC Lab\\Exercise\\Ex 7\\CloudSim\\build
clean:
BUILD SUCCESSFUL (total time: 0 seconds)

```

3:59 PM 8/25/2021 IN5

7. Output for CloudSimExample2.java



CloudSim - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Start Page X CloudSim.java X CloudSimExample1.java X CloudSimExample2.java X

Source History |

Output - CloudSim (run) X

```
run:
Starting CloudSimExample2...
Initializing...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.0: Broker: Trying to Create VM #1 in Datacenter_0
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: VM #1 has been created in Datacenter #2, Host #0
0.1: Broker: Sending cloudlet 0 to VM #0
0.1: Broker: Sending cloudlet 1 to VM #1
1000.1: Broker: Cloudlet 0 received
1000.1: Broker: Cloudlet 1 received
1000.1: Broker: All Cloudlets executed. Finishing...
1000.1: Broker: Destroying VM #0
1000.1: Broker: Destroying VM #1
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.

=====
Cloudlet ID STATUS Data center ID VM ID Time Start Time Finish Time
0 SUCCESS 2 0 1000 0.1 1000.1
1 SUCCESS 2 1 1000 0.1 1000.1
CloudSimExample2 finished!
```

Notifications Navigator

Type here to search

10:1 IN5

4:00 PM 8/25/2021

VIVA QUESTIONS

1. What is Clouds?

2. CloudSim is written in _____.

3. _____ class represents any task that is run on a VM.

4. _____ is responsible for functioning of VMs, including VM creation, management, destruction, and submission of cloudlets to the VM.

5. _____ is the default VM allocation policy.