**Assignment No :- 2**

**Title**: Installation and configuration of CloudSim

**Aim**: 1. Explore cloud simulator

2. Workflow of cloud simulator

3. Four Scenarios

1. Set the Number of users for the current simulation. This user count is directly proportional to a number of brokers in the current simulation.
2. Initialize the simulation, provided with the current time, number of users and trace flag.
3. Create a Datacenter.
4. Create a Datacenter broker.
5. Create a Virtual Machine(s).
6. Submit Virtual Machine to Datacenter broker.
7. Create Cloudlet(s) by specifying their characteristics.
8. Submit Cloudlets to a Datacenter broker.
9. Send a call to Start Simulation.
10. Once no more events are executed, send the call to Stop Simulation.
11. Finally, print the final status of the Simulation.

|  | **Example1.java** | **Example2.java** |
| --- | --- | --- |
| **Title** | Create the datacenter with One VM | Create the datacenter with Two VMs |
| **dataCenter** | #DC2  @SuppressWarnings("unused")  Datacenter datacenter0 = *createDatacenter*("Datacenter\_0"); | #DC2  @SuppressWarnings("unused")  Datacenter datacenter0 = *createDatacenter*("Datacenter\_0"); |
| **Host Description** | int hostId = 0;  int ram = 2048; // host memory (MB)  long storage = 1000000; // host storage  int bw = 10000;  hostList.add(  new Host(  hostId,  new RamProvisionerSimple(ram),  new BwProvisionerSimple(bw),  storage,  peList,  new VmSchedulerTimeShared(peList)  )  ); | |
| **Cloudlet** | CLoudlet=1  cloudletList = new ArrayList<Cloudlet>();  // Cloudlet properties  int id = 0;  long length = 400000;  long fileSize = 300;  long outputSize = 300;  UtilizationModel utilizationModel = new UtilizationModelFull();  Cloudlet cloudlet = new Cloudlet(id, length, pesNumber, fileSize, outputSize, utilizationModel, utilizationModel, utilizationModel);  cloudlet.setUserId(brokerId);  cloudlet.setVmId(vmid);  // add the cloudlet to the list  cloudletList.add(cloudlet);  // submit cloudlet list to the broker  broker.submitCloudletList(cloudletList);  // Sixth step: Starts the simulation  CloudSim.startSimulation();  CloudSim.stopSimulation();  //Final step: Print results when simulation is over  List<Cloudlet> newList = broker.getCloudletReceivedList();  printCloudletList(newList); | CLoudlet=2  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, utilizationModel, utilizationModel);  cloudlet1.setUserId(brokerId);  Id++;  Cloudlet cloudlet2 = new Cloudlet(id, length, pesNumber, fileSize, outputSize, utilizationModel, utilizationModel, 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); |
| **VM list** | Vm vm1 = **new** Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, **new** CloudletSchedulerTimeShared()); | Vm vm1 = **new** Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, **new** CloudletSchedulerTimeShared());    Vmid++;  Vm vm2 = **new** Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, **new** CloudletSchedulerTimeShared());  //add the VMs to the vmList  *vmlist*.add(vm1);  *vmlist*.add(vm2); |

|  | **Example3.java** | **Example4.java** |
| --- | --- | --- |
| **Title** | Create the DataCenter with  Two Cloudlet and Two VMs | Create the Two datacenters with Two VMs and Two Cloudlets |
| **dataCenter** | #DC2  @SuppressWarnings("unused")  Datacenter datacenter0 = *createDatacenter*("Datacenter\_0"); | #DC2  @SuppressWarnings("unused")  Datacenter datacenter0 = *createDatacenter*("Datacenter\_0");  @SuppressWarnings("unused")  #DC3  Datacenter datacenter1 = *createDatacenter*("Datacenter\_1"); |
| **Host Description** | **int** hostId=0;**int** ram = 2048; //host memory (MB)  **long** storage = 1000000; //host storage  **int** bw = 10000;  hostList.add(  **new** Host(  hostId,  **new** RamProvisionerSimple(ram),  **new** BwProvisionerSimple(bw),  storage,  peList,  **new** VmSchedulerTimeShared(peList)  )); // This is our first machine | |
| **Cloudlet** | Cloudlet no=2 //the creation of the cloudlet  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, utilizationModel, utilizationModel);  cloudlet1.setUserId(brokerId);  Id++;  Cloudlet cloudlet2 = new Cloudlet(id, length, pesNumber, fileSize, outputSize, utilizationModel, utilizationModel, 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); | |
| **VM list** | #VM0  Vm vm1 = **new** Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, **new** CloudletSchedulerTimeShared());  Vmid++;  #VM1  Vm vm2 = **new** Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, **new** CloudletSchedulerTimeShared());  //add the VMs to the vmList  *vmlist*.add(vm1);  *vmlist*.add(vm2); | #VM0  Vm vm1 = **new** Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, **new** CloudletSchedulerTimeShared());  Vmid++;  #VM1  Vm vm2 = **new** Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, **new** CloudletSchedulerTimeShared());  //add the VMs to the vmList  *vmlist*.add(vm1);  *vmlist*.add(vm2); |