**Task-1**

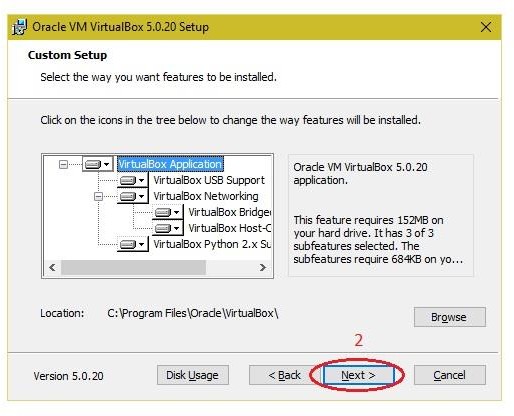
**Aim:** **Install Virtual Box/VMware Workstation on different OS.**

**Steps to install Virtual Box:**

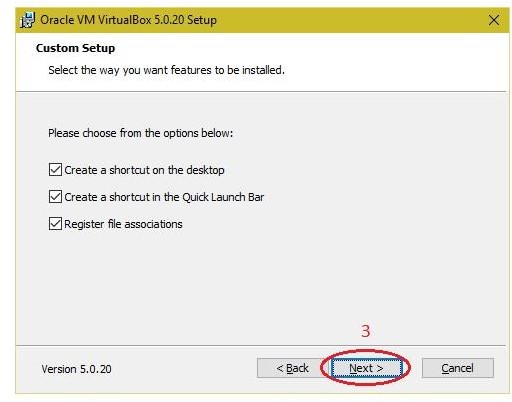
1. Download the Virtual box exe and click the exe file…and select next button.



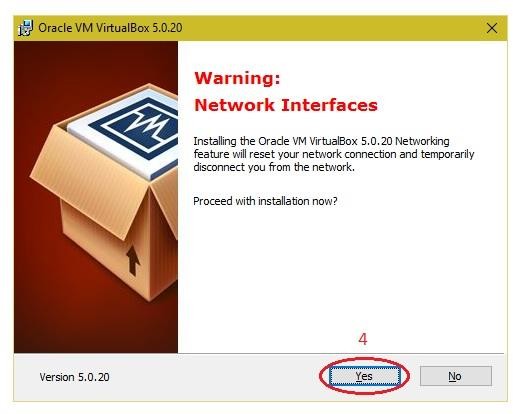
2.Click the next button.



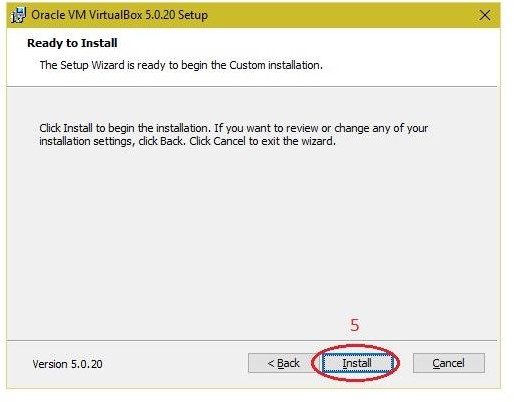
3.Click the next button



2.Click the YES button..



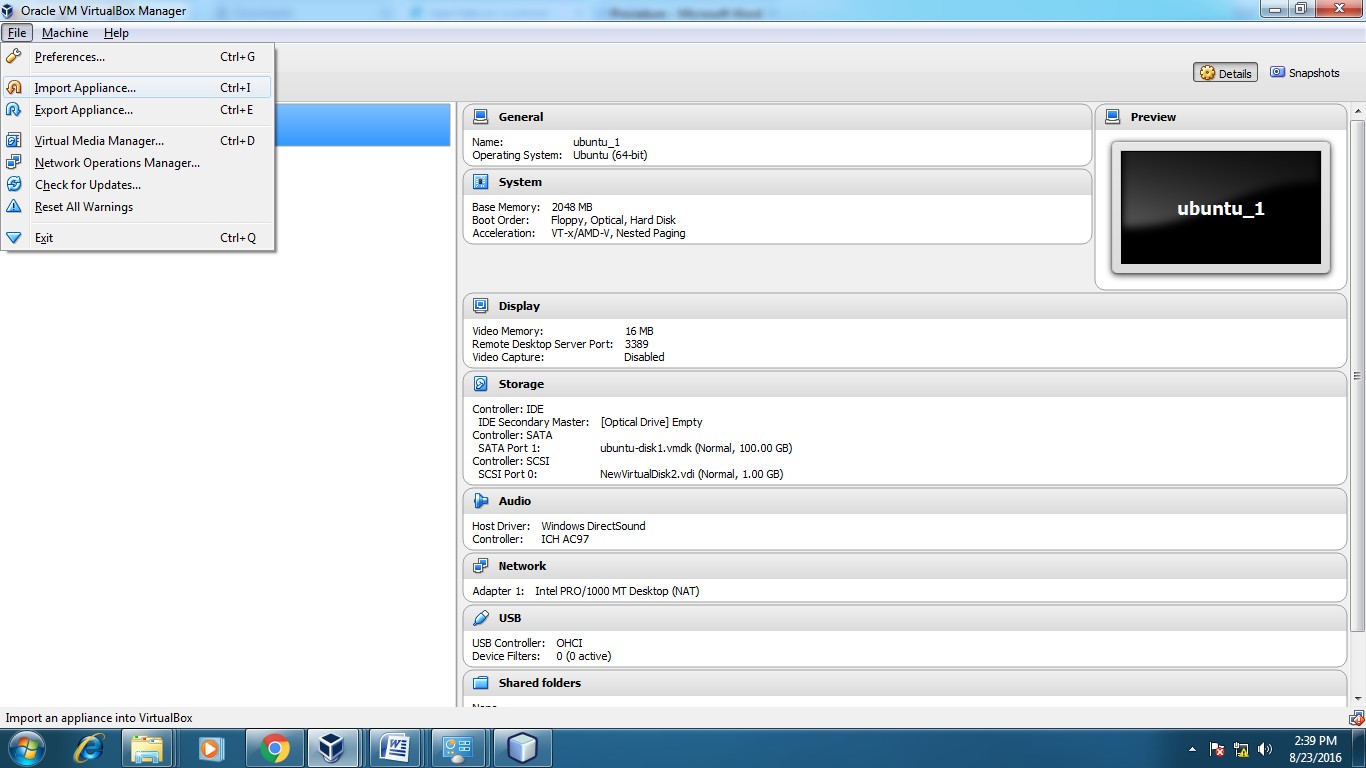
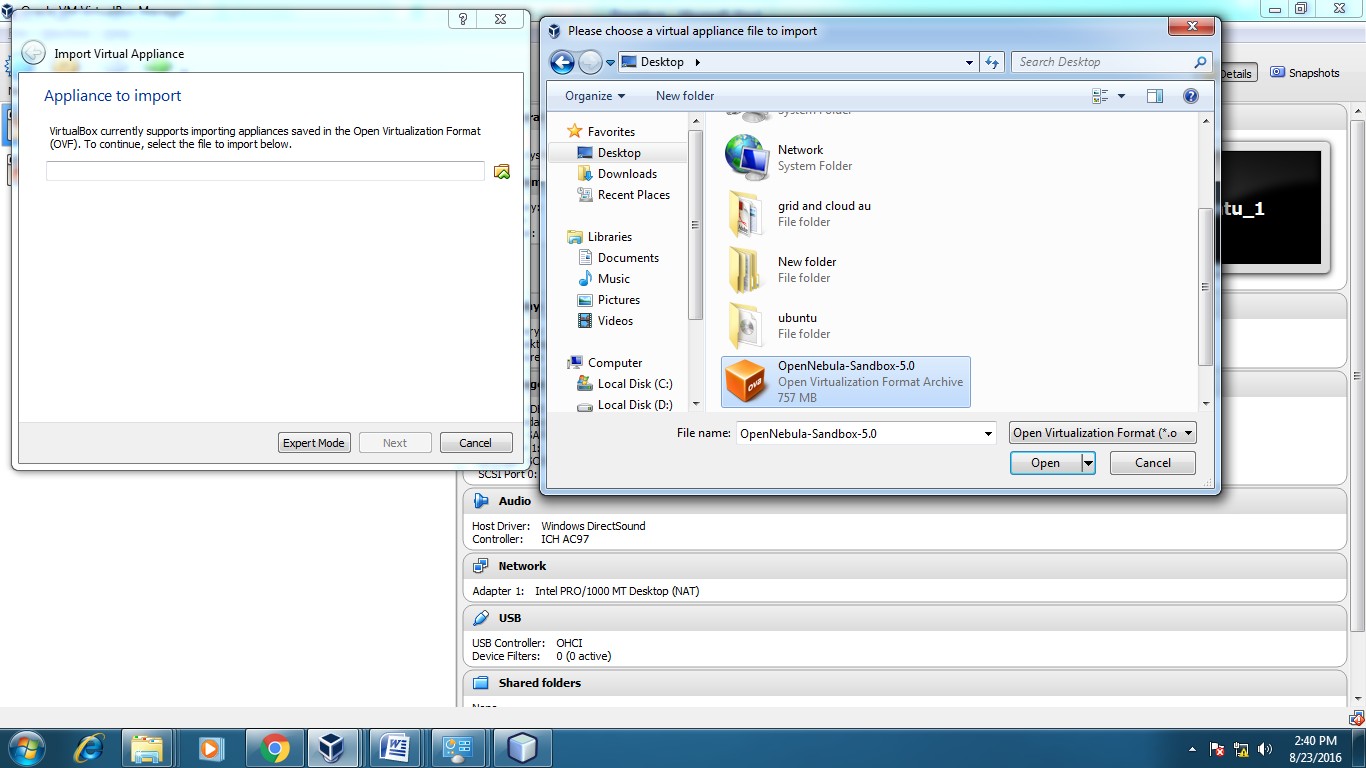
# Click the install button…



2.Then installation was completed the show virtual box icon on desktop screen….



**Steps to import Open nebula sandbox:**

* 1. Open Virtual box
  2. File import Appliance
  3. Browse OpenNebula-Sandbox-5.0.ova file
  4. Then go to setting, select Usb and choose USB 1.1
  5. Then Start the Open Nebula
  6. Login using username: root, password:opennebul 

**Steps to create Virtual Machine through opennebula**

Open Browser, type localhost:9869

Login using username: oneadmin, password: opennebula

Click on instances, select VMs then follow the steps to create Virtaul machine

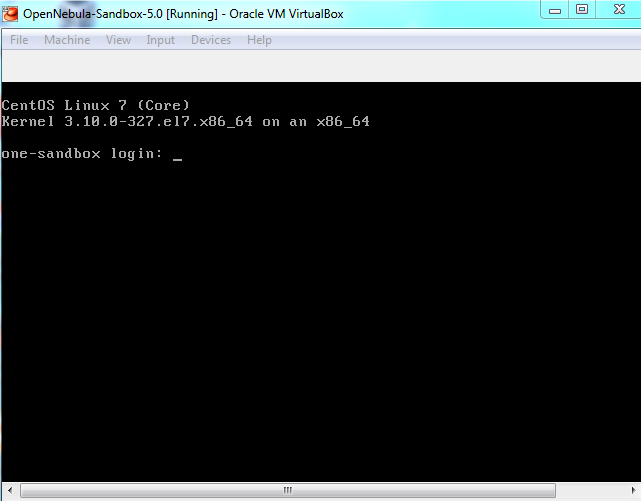
Expand the + symbol

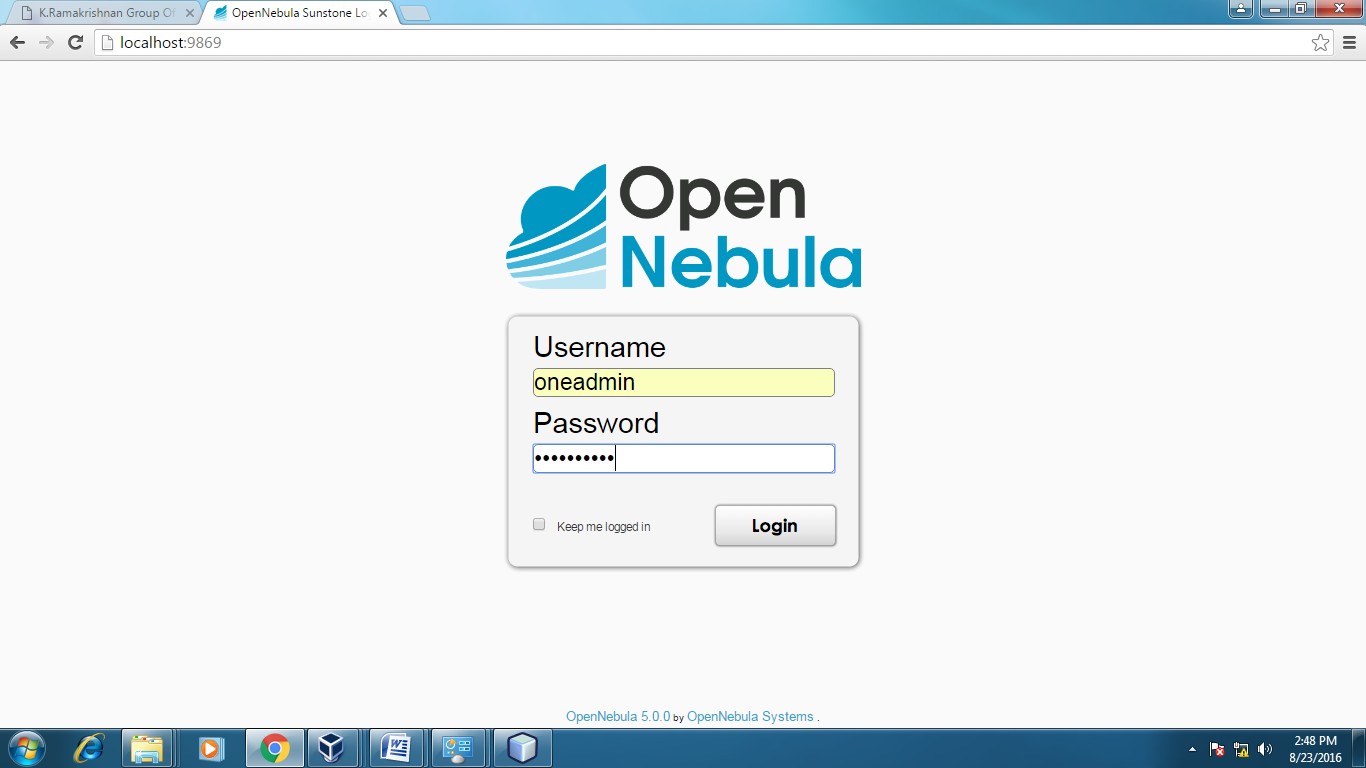
Select user one admin

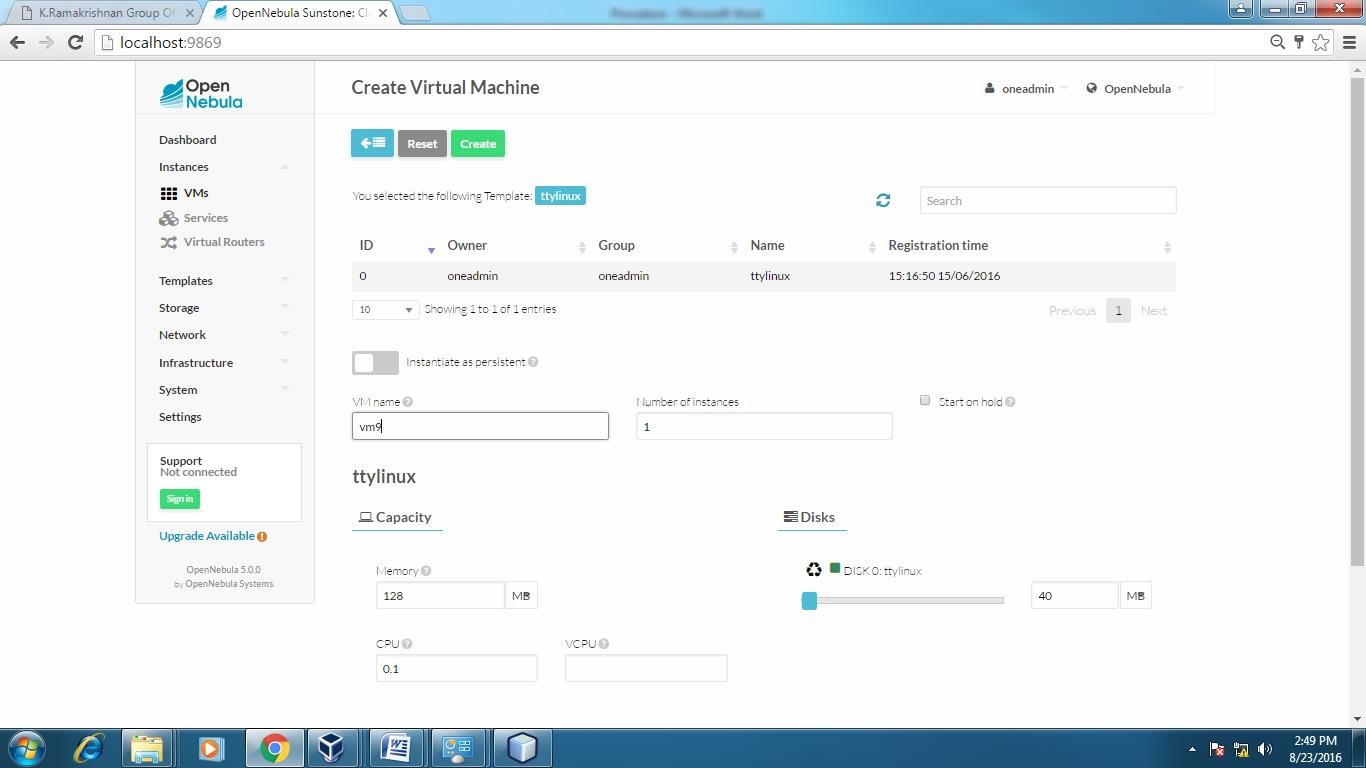
Then enter the VM name ,no. of instance, cpu.

Then click on create button.

Repeat the steps the C,D for creating more than one VMs







APPLICATIONS:

There are various applications of cloud computing in today’s network world. Many search engines and social websites are using the concept of cloud computing like www.amazon.com, hotmail.com, facebook.com, linkedln.com etc. the advantages of cloud computing in contextto scalability is like reduced risk , low cost testing ability to segment the customer base and auto-scalingbased on application load.

### **RESULT:**

Thus the procedure to run the virtual machine of different configuration.

**Task-2**

**Aim: Install different operating systems in VMware**.

VMware Workstation Pro is paid virtualization software that lets you run another operating system inside your current one. Want to use Linux for some situations but you’ve only got a Windows 10 computer? No problem — use VMware Workstation on Windows to install Ubuntu.

This program even lets you run multiple operating systems at the same time. If all you have is Windows 10 but you like to play old Windows XP games and also use a few programs that only work on a Mac, just load up the XP or Mac virtual machine, make it full screen, and use it just like you would if it were the only OS you had.

The process to install a new operating system in VMware Workstation is fairly straightforward because the setup wizard makes things super easy. Below are step-by-step screenshots showing everything you need to know, from the initial setup screen all the way to booting into the new operating system.

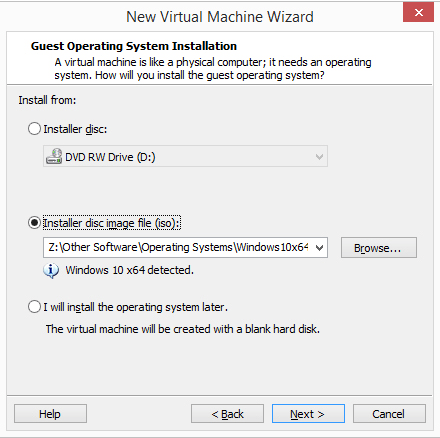
Install New OS in VMware Workstation

**Step 1**: Go to File > New Virtual Machine.

**Step 2**: Select Typical (recommended), and then press Next

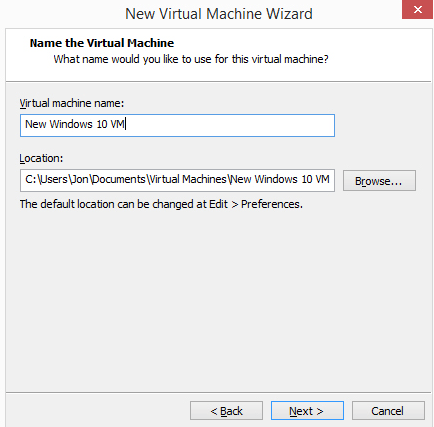
**Step 3**: Pick how to install the operating system.Select **Installer disc** if the operating system is in the disc drive. Otherwise, choose **Installer disc image file (iso)** if you have a file that contains the OS, like a Windows 10 ISO or an ISO for macOS.

**Step 4**: Choose precisely where the OS files are located.



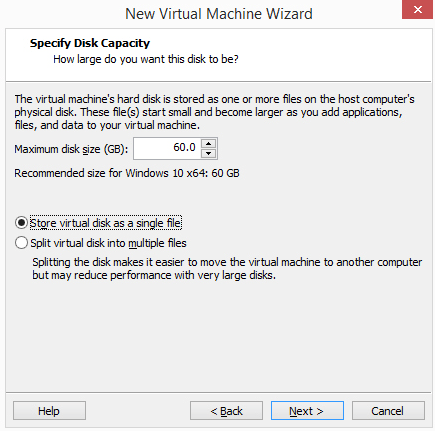
If you selected to install the operating system from a disc, choose the correct disc drive from the drop-down menu. For an ISO install, select **Browse** and locate the ISO image.

**Step 5**: Press **Next** to proceed to the screen where you name the new virtual machine and pick where its files should be stored. Fill out that information and then select **Next** again.



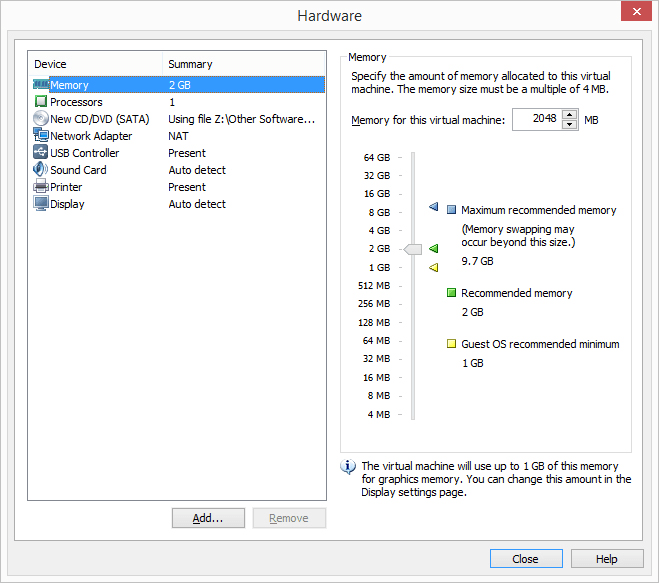
Note: For some operating system setups, you’ll be asked to enter the product key used to activate it. You should be able to skip through that step if you want to enter the key later*.*

**Step 6**: Define how storage should take place with this virtual machine, and then press **Next**.



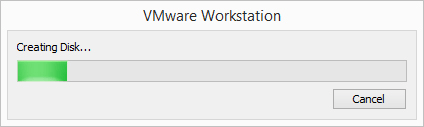
You can change the maximum size of the virtual hard drive from the small box. From the bottom of this window are two options: **Store virtual disk as a single file** and **Store virtual disk into multiple files**.

**Step 7**: Select **Customize Hardware** and make any necessary changes. You can change details about the memory, processors, disc drive, network adapter, USB controller, sound card, printer, and display.

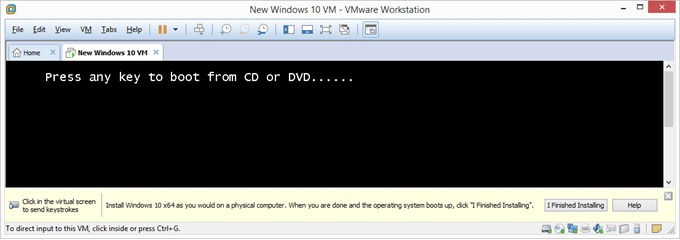


**Step 8**: Choose **Close**to exit the *Hardware* screen, and then press **Finish**.

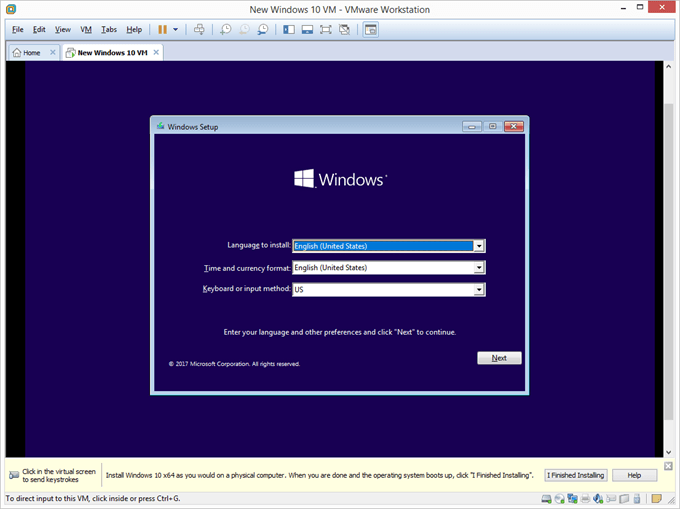
VMware Workstation will create the virtual disk you specified in Step 6 and then turn on the virtual machine automatically. This process might take a while, but you can watch the progress bar for an estimation of when it will finish.



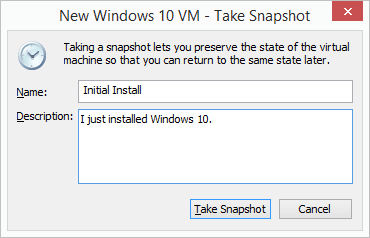
**Step 9**: Follow any on-screen prompts to begin the OS installation. For example, if you see Press any key to boot from CD or DVD, do that to start the OS setup.



**Step 10**: Follow the directions for your specific operating system install to add the OS to VMware Workstation.



The **VM**> **Snapshot**option is how you “freeze” the current state of the VM so that you can return to that same state again later. You might need to do this if the operating system fails later on, you get lots of viruses that can’t be cleaned, or you simply want to start over from a fresh install.



Once the operating system has been installed and is running, go to **VM**> **Settings** to adjust any hardware settings you changed or didn’t change from Step 7. Some options are only editable when the virtual machine is off.

At any time when the VMware virtual machine is on, you can shut it down or restart it from the **VM**> **Power** menu. This is also how you suspend the VM, which means to pause it so that you can resume at that exact same point the next time you access it. This is different from powering it on from an off state where you have to log back in again, open programs

**Task-3**

**Aim:** **Simulate a cloud scenario using simulator.**

How to use CloudSim in Eclipse

CloudSim is written in Java. The knowledge you need to use CloudSim is basic Java programming and some basics about cloud computing. Knowledge of programming IDEs such as Eclipse or NetBeans is also helpful. It is a library and, hence, CloudSim does not have to be installed. Normally, you can unpack the downloaded package in any directory, add it to the Java classpath and it is ready to be used. Please verify whether Java is available on your system.

To use CloudSim in Eclipse:

1.Download CloudSim installable files

from https://code.google.com/p/cloudsim/downloads/list and unzip

2.Open Eclipse

3.Create a new Java Project: File -> New

4.Import an unpacked CloudSim project into the new Java Project

The first step is to initialise the CloudSim package by initialising the CloudSim library, as follows

CloudSim.init(num\_user, calendar, trace\_flag)

5.Data centres are the resource providers in CloudSim; hence, creation of data centres is a second step. To create Datacenter, you need the DatacenterCharacteristics object that stores the properties of a data centre such as architecture, OS, list of machines, allocation policy that covers the time or spaceshared, the time zone and its price:

Datacenter datacenter9883 = new Datacenter(name, characteristics, new VmAllocationPolicySimple(hostList), s

6.The third step is to create a broker:

DatacenterBroker broker = createBroker();

7.The fourth step is to create one virtual machine unique ID of the VM, userId ID of the VM’s owner, mips, number Of Pes amount of CPUs, amount of RAM, amount of bandwidth, amount of storage, virtual machine monitor, and cloudletScheduler policy for cloudlets:

Vm vm = new Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, new CloudletSchedulerTimeShared())

8.Submit the VM list to the broker: broker.submitVmList(vmlist)

9.Create a cloudlet with length, file size, output size, and utilisation model:

Cloudlet cloudlet = new Cloudlet(id, length, pesNumber, fileSize, outputSize, utilizationModel, utilizationMode

10.Submit the cloudlet list to the broker: broker.submitCloudletList(cloudletList) Sample Output from the Existing Example:

Starting

CloudSimExample1... Initialising...

Starting CloudSim version 3.0 Datacenter\_0 is starting...

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>null

Broker is starting... Entities started.

: Broker: Cloud Resource List received with 1 resource(s) 0.0: Broker: Trying to Create VM #0 in Datacenter\_0

: Broker: VM #0 has been created in Datacenter #2, Host #0 0.1: Broker: Sending cloudlet 0 to VM #0

400.1 : Broker: Cloudlet 0 received

: Broker: All Cloudlets executed. Finishing 400.1: Broker: Destroying

VM #0

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.

========== OUTPUT ==========

Cloudlet ID STATUS Data center ID Finish Time 0 SUCCESS 2

0.1 400.1 VM ID Time 0 Start Time

400

\*\*\*\*\*Datacenter: Datacenter\_0\*\*\*\*\* User id

Debt

3 35.6

CloudSimExample1 finished!

**RESULT:**

The simulation was successfully executed

**Task-4**

# Aim: Implement scheduling algorithms.

Implementing scheduling algorithms involves writing code to manage the allocation of resources to different tasks or processes in a system. Here's a basic outline of how you might implement some common scheduling algorithms in a programming language like Python:

First-Come, First-Served (FCFS):

In FCFS scheduling, tasks are executed in the order they arrive.

Create a queue data structure to represent the order of arrival of tasks.

Iterate through the queue and execute each task in the order they were added.

python

Copy code

def fcfs(tasks):

for task in tasks:

execute\_task(task)

Shortest Job First (SJF):

SJF scheduling selects the task with the shortest execution time first.

Sort the tasks based on their execution time.

Execute tasks in ascending order of execution time.

python

Copy code

def sjf(tasks):

sorted\_tasks = sorted(tasks, key=lambda x: x.execution\_time)

for task in sorted\_tasks:

execute\_task(task)

Round Robin:

Round Robin scheduling allocates a fixed time slice to each task in a cyclic manner.

Implement a circular queue to store the tasks.

Iterate through the queue, executing each task for a fixed time slice.

python

Copy code

def round\_robin(tasks, time\_slice):

queue = deque(tasks)

while queue:

task = queue.popleft()

execute\_task(task, time\_slice)

if task.remaining\_time > 0:

queue.append(task)

Priority Scheduling:

Priority scheduling selects tasks based on their priority.

Assign a priority value to each task and sort them based on priority.

Execute tasks in descending order of priority.

python

Copy code

def priority\_scheduling(tasks):

sorted\_tasks = sorted(tasks, key=lambda x: x.priority, reverse=True)

for task in sorted\_tasks:

execute\_task(task)

Shortest Remaining Time First (SRTF):

SRTF scheduling selects the task with the shortest remaining execution time.

Maintain a priority queue (min-heap) of tasks based on their remaining execution time.

Execute the task with the shortest remaining time.

python

Copy code

import heapq

def srtf(tasks):

current\_time = 0

remaining\_tasks = [(task.execution\_time, task) for task in tasks]

heapq.heapify(remaining\_tasks)

while remaining\_tasks:

\_, task = heapq.heappop(remaining\_tasks)

execute\_time = min(task.execution\_time, task.remaining\_time)

execute\_task(task, execute\_time)

current\_time += execute\_time

task.remaining\_time -= execute\_time

if task.remaining\_time > 0:

heapq.heappush(remaining\_tasks, (task.remaining\_time, task))

**Example:**

**def sjf(tasks):**

**sorted\_tasks = sorted(tasks, key=lambda x: x.execution\_time)**

**for task in sorted\_tasks:**

**execute\_task(task)**

**class Task:**

**def \_\_init\_\_(self, task\_id, execution\_time, priority=None):**

**self.task\_id = task\_id**

**self.execution\_time = execution\_time**

**self.priority = priority**

**self.remaining\_time = execution\_time**

**def execute\_task(task, time\_slice=None):**

**if time\_slice is None or time\_slice >= task.remaining\_time:**

**print(f"Executing Task {task.task\_id} for {task.remaining\_time} units")**

**task.remaining\_time = 0**

**else:**

**print(f"Executing Task {task.task\_id} for {time\_slice} units")**

**task.remaining\_time -= time\_slice**

**# Define tasks**

**tasks = [**

**Task(1, 10, 3),**

**Task(2, 5, 1),**

**Task(3, 8, 2),**

**Task(4, 3, 4),**

**]**

**# Execute tasks using different scheduling algorithms**

**print("FCFS (First-Come, First-Served):")**

**fcfs(tasks)**

**print()**

**print("SJF (Shortest Job First):")**

**sjf(tasks)**

**print()**

**print("Round Robin:")**

**round\_robin(tasks, time\_slice=3)**

**print()**

**print("Priority Scheduling:")**

**priority\_scheduling(tasks)**

**print()**

**print("SRTF (Shortest Remaining Time First):")**

**srtf(tasks)**

**Task-5**

**Aim:** **To study cloud security management**.

**Objectives:** From this experiment, the student will be able,

* To understand the security features of Cloud.
* To learn the technique of application security management and its complexity
* To understand the importance of cloud security management from application point of view

**Outcomes:** The learner will be able to

* + Student can study and implement single-sign-on.
  + To use current techniques, skills, and tools necessary for computing practice.
  + To match the industry requirements in the domains of Database management, Programming and Networking with the required management skills.

1. **Hardware / Software Required:** Ubuntu operating system, Virtual machine, WAMP/ZAMP server, Any tool or technology can be used for implementation of web application e.g., JAVA, PHP, etc.

1. **Theory:**

Cloud computing security is the set of control-based technologies and policies designed to adhere to regulatory compliance rules and protect information, data applications and infrastructure associated with cloud computing use. Because of the cloud's very nature as a shared resource, identity management, privacy and access control are of particular concern. With more organizations using cloud computing and associated cloud providers for data operations, proper security in these and other potentially vulnerable areas have become a priority for organizations contracting with a cloud computing provider.

Cloud computing security processes should address the security controls the cloud provider will incorporate to maintain the customer's data security, privacy and compliance with necessary regulations. The processes will also likely include a business continuity and databackup plan in the case of a cloud security breach.

**Physical security.**

Cloud service providers physically secure the IT hardware (servers, routers, cables etc.) against unauthorized access, interference, theft, fires, floods etc. and ensure that essential supplies (such as electricity) are sufficiently robust to minimize the possibility of disruption. This is normally achieved by serving cloud applications from 'world-class' (i.e. professionally specified, designed, constructed, managed, monitored and maintained) data centers.

**Personnel security**

Various information security concerns relating to the IT and other professionals associated with cloud services are typically handled through pre-, para- and post-employment activities such as security screening potential recruits, security awareness and training programs, proactive security monitoring and supervision, disciplinary procedures and contractual obligations embedded in employment contracts, service level agreements, codes of conduct, policies etc.

**Application security**

Cloud providers ensure that applications available as a service via the cloud (SaaS) are secure by specifying, designing, implementing, testing and maintaining appropriate application security measures in the production environment. Note that - as with any commercial software - the controls they implement may not necessarily fully mitigate all the risks they have identified, and that they may not necessarily have identified all the risks that are of concern to customers. Consequently, customers may also need to assure themselves that cloud applications are adequately secured for their specific purposes, including their compliance obligations.

**Procedure:**

Security using MFA(Multi Factor Authentication) device code:

1) goto aws.amazon.com

2) click on "My Account"

3) select "AWS management console" and click on it

4) Give Email id in the required field

if you are registering first time then select "I am a new user" radio button

5) click on "sign in using our secure server" button

6) follow the instruction and complete the formalities

(Note: do not provide any credit card details or bank details)

sign out from

7) Again go to "My Account"

select "AWS management console" and click on it

Sign in again by entering the user name and valid password ( check "I am returning user and my password is" radio button)

Now you are logged in as a Root User

All AWS project can be viewed by you, but you cant make any changes in it or you cant create new thing as you are not paying any charges to amazon (for reason refer step:6)

**To create the user in a root user follow the steps mentioned below:**

1) click on "Identity and Access Management" in security and identity project

2) click in "Users" from dashboard

It will take you to "Create New Users"

click on create new user button

enter the "User Name"

(select "Generate and access key for each user" checkbox, it will create a user with a specific key)

click on "Create" button at right bottom

3) once the user is created click on it

4) go to security credentials tab

5) click on "Create Access Key", it will create an access key for user.

6) click on "Manage MFA device" it will give you one QR code displayed on the screen

you need to scan that QR code on your mobile phone using barcode scanner (install it in mobile phone)you also need to install "Google Authenticator" in your mobile phone to generate the MFA code

7) Google authenticator will keep on generating a new MFA code after every 60 seconds

that code you will have to enter while logging as a user.

Hence, the security is maintained by MFA device code...

one can not use your AWS account even if it may have your user name and password, because MFA code is on your MFA device (mobiel phone in this case) and it is getting changed after every 60 seconds.

**Permissions in user account:**

After creating the user by following above mentioned steps; you can give certain permissions to specific user

1) click on created user

2) goto "Permissions" tab

3) click on "Attach Policy" button

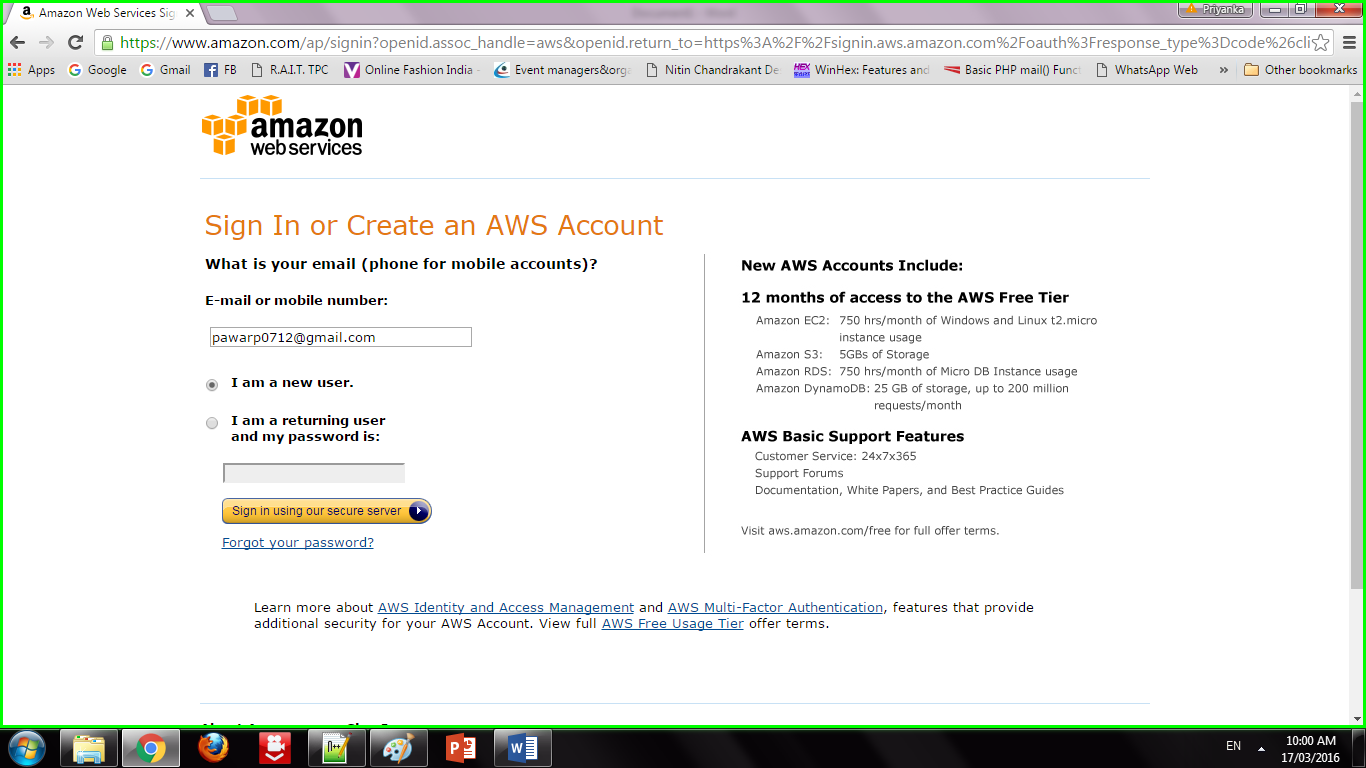
4) select the needed policy from given list and click on apply.

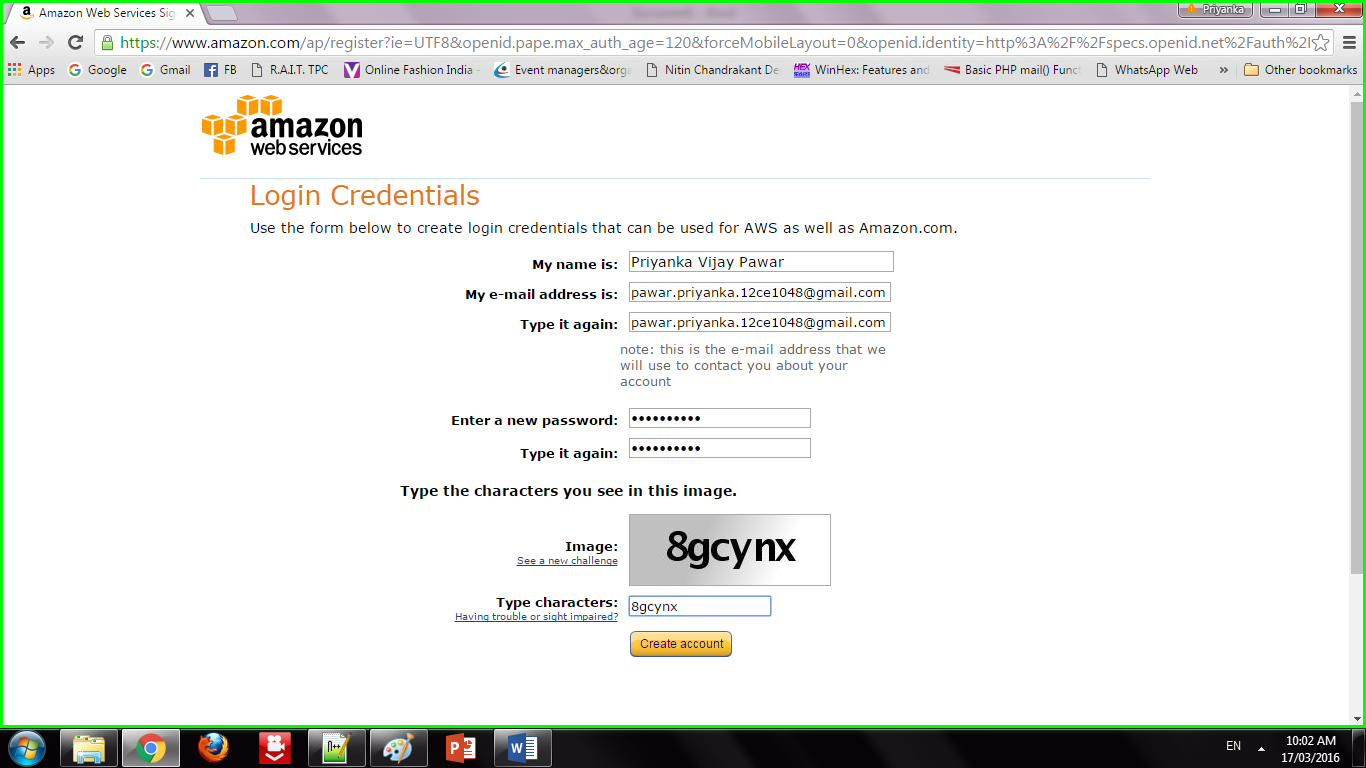
**Result:**

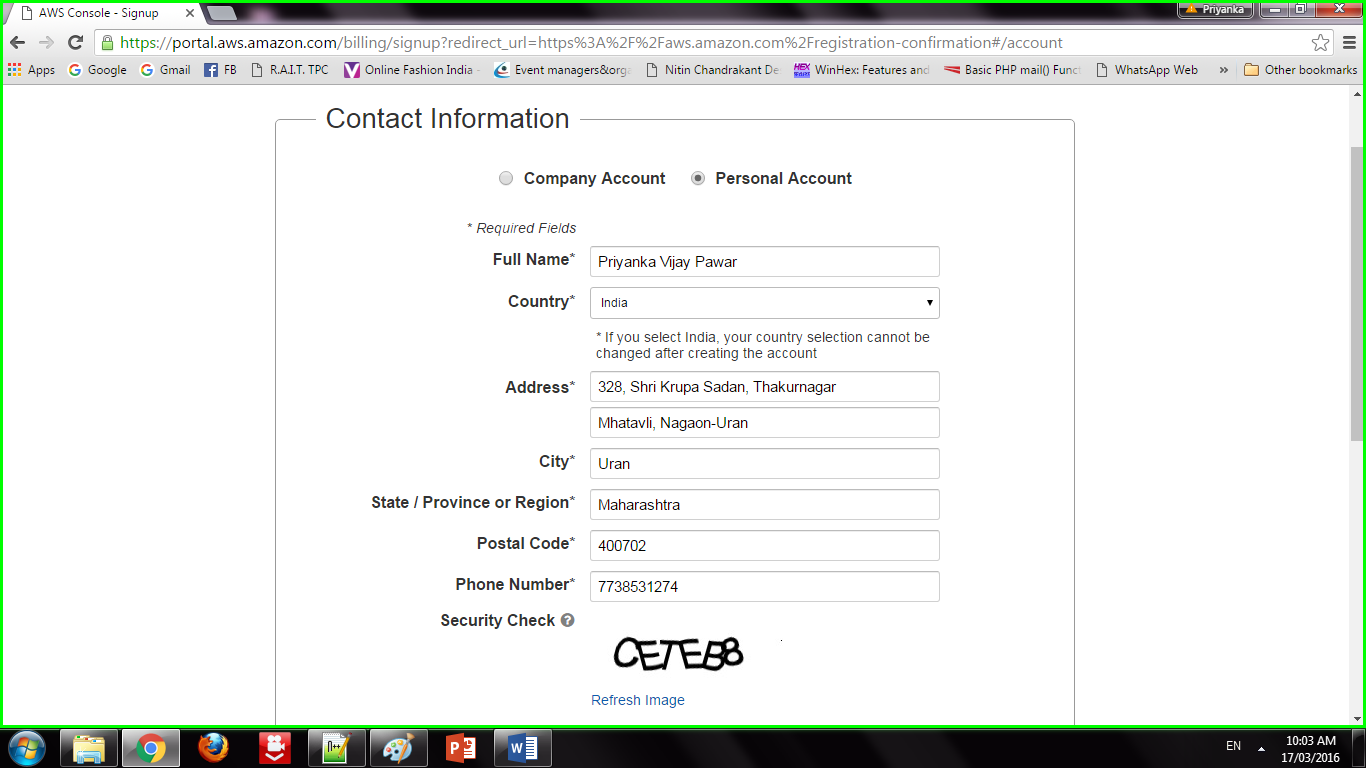
Step 1 :goto aws.amazon.com



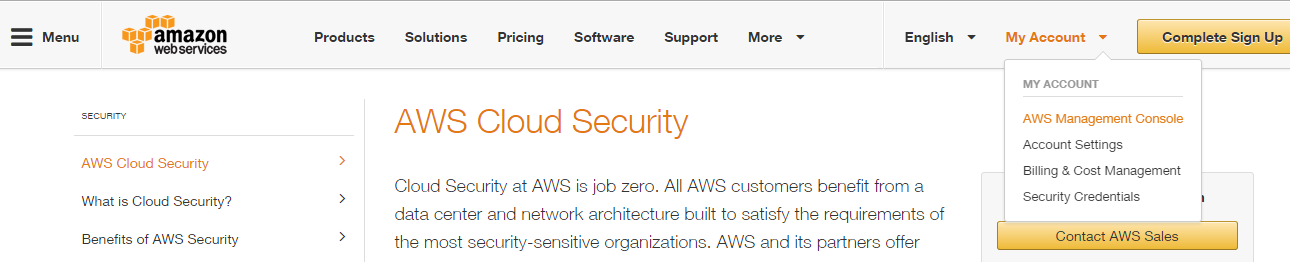
**Step 2** :Click on "My Account". Select "AWS management console" and click on it. Give Email id in the required field



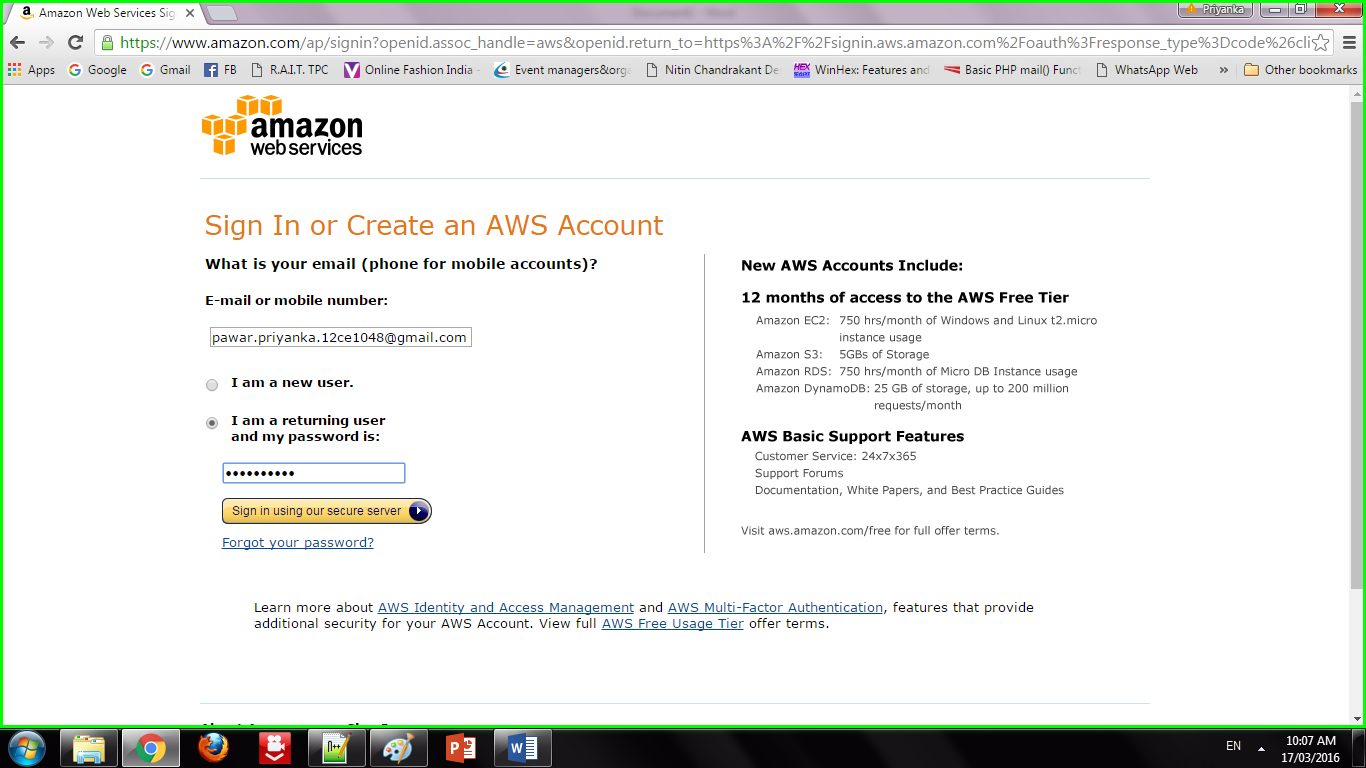


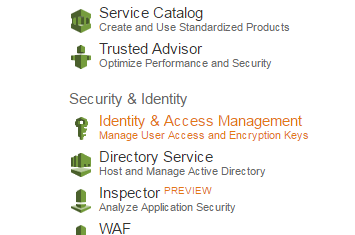


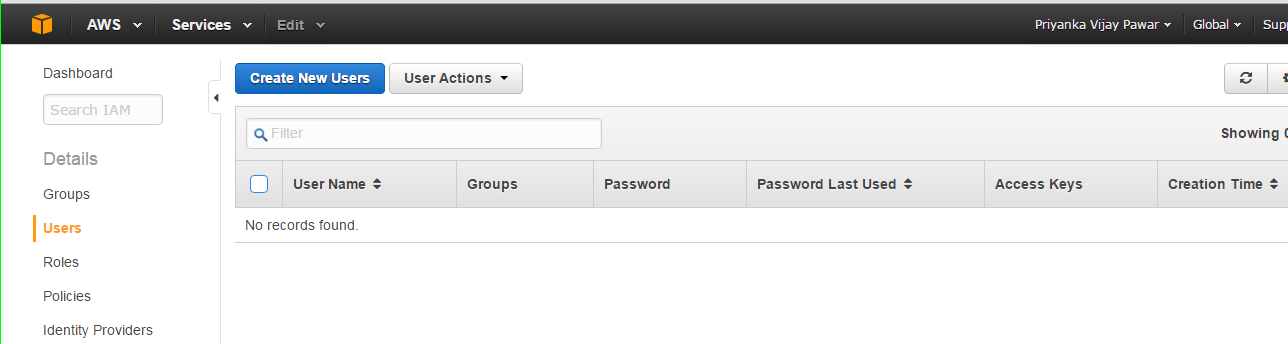
Step 3: Addition of security features



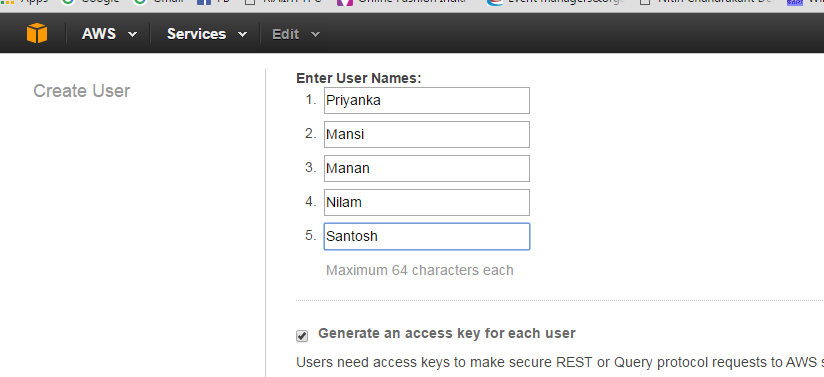
Step 4: Sign in to an AWS account

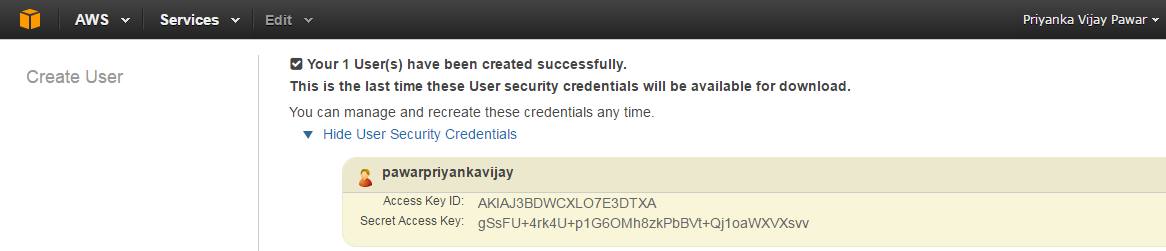


****

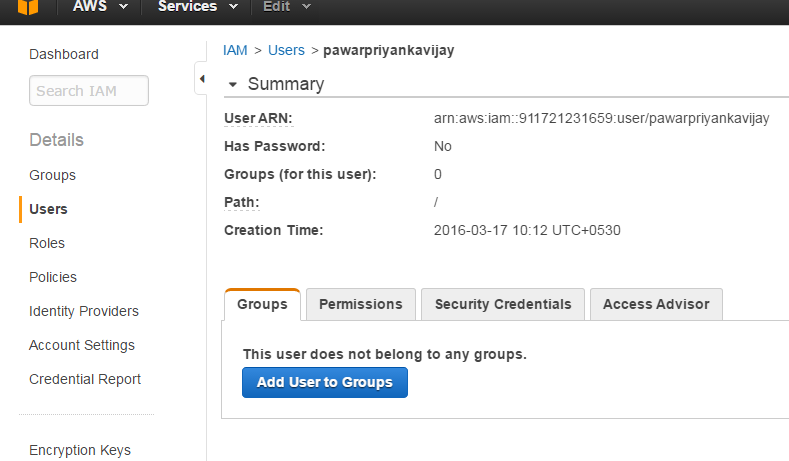
****

**Step 5:** Creation of users

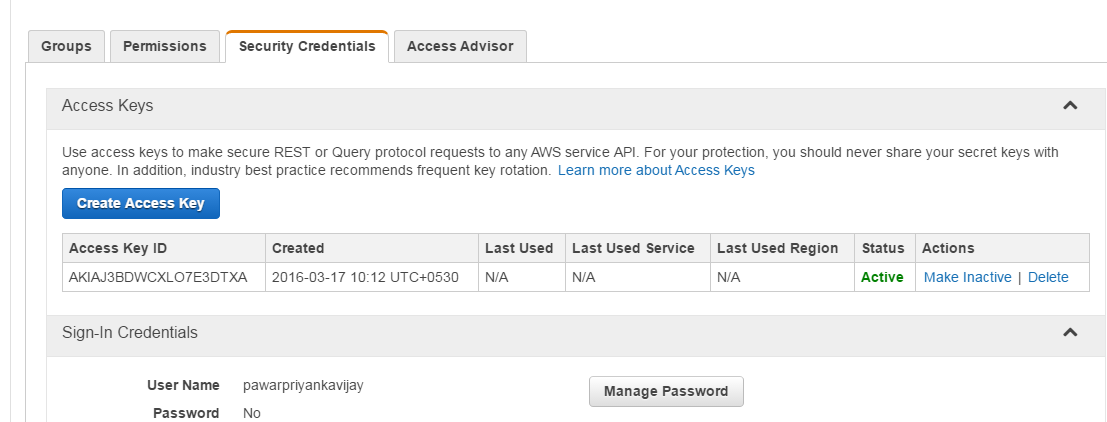
****

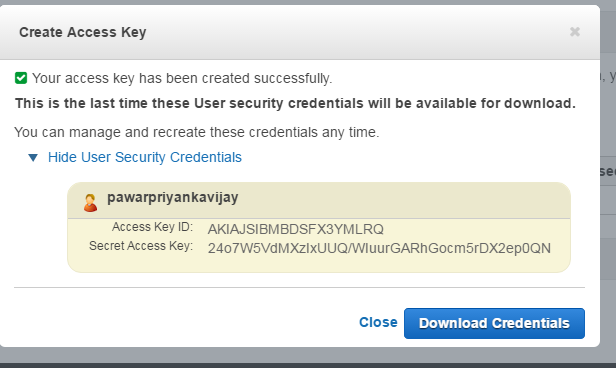
****

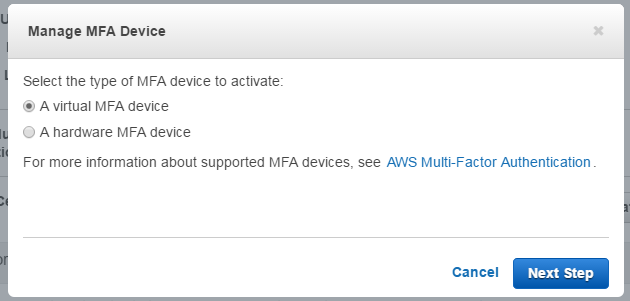
Step 6: Adding users to group

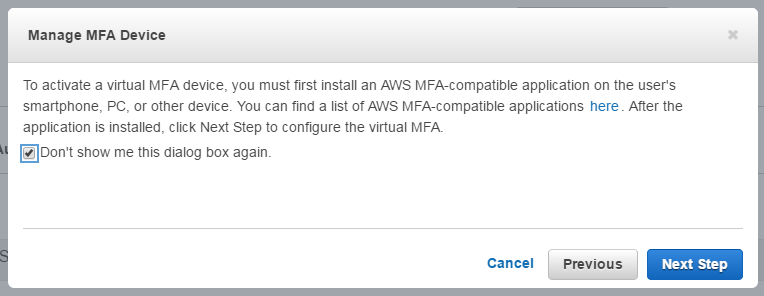
****

Step 7: Creating Access key

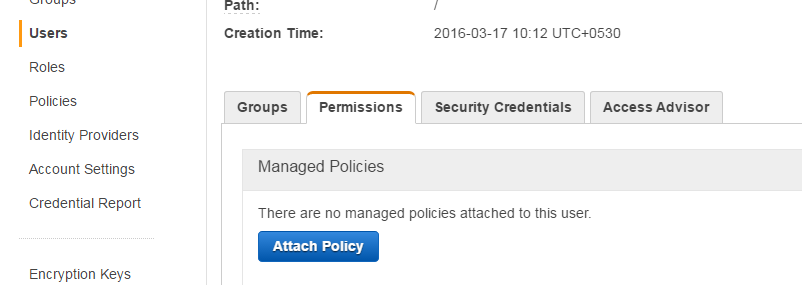
****

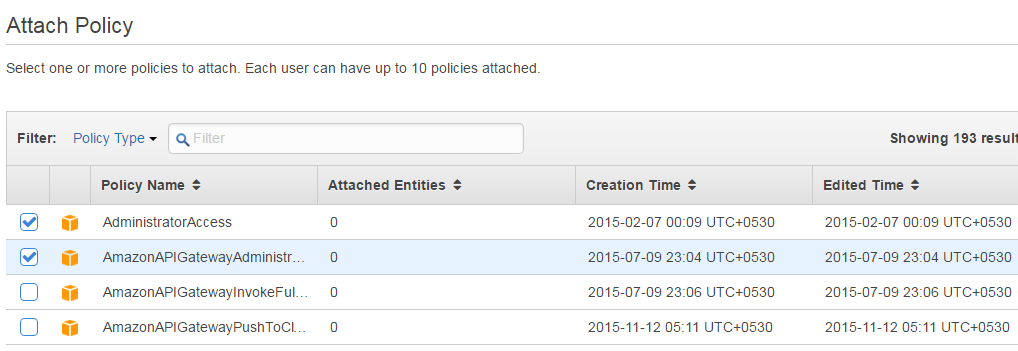
****

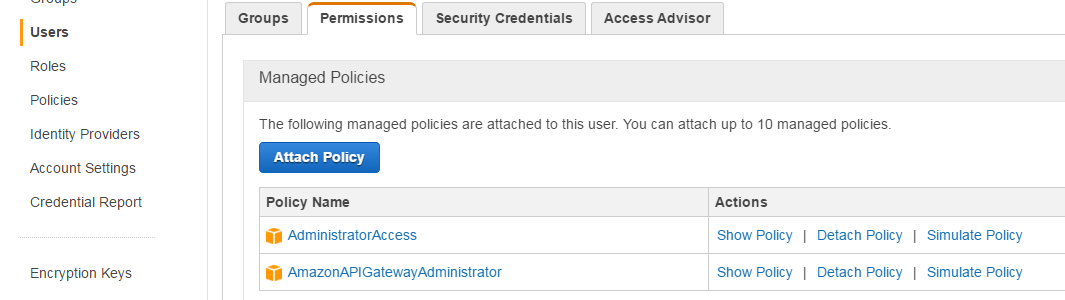
****

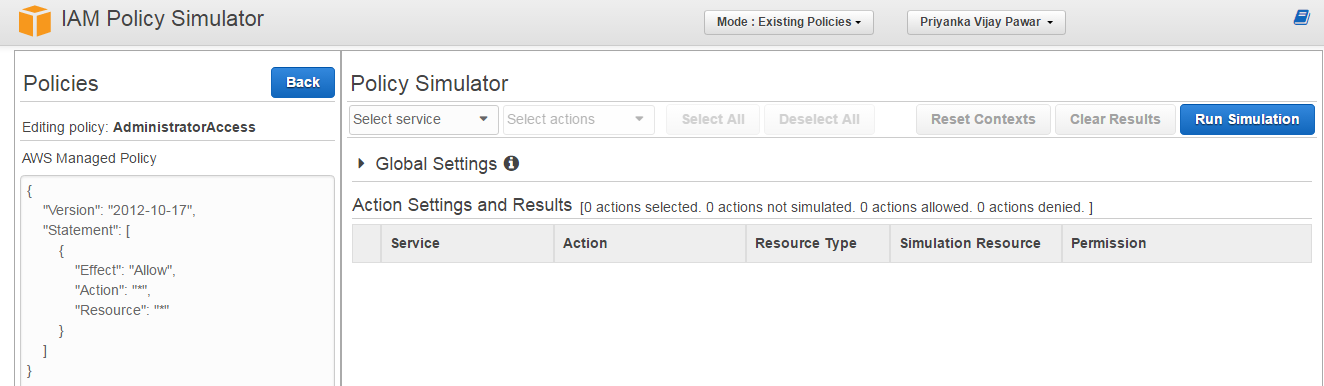
****

Step 8 : Setting permissions to users

****

****

****

****

1. **Conclusion:**

We have studied how to secure the cloud and its data. Amazon EWS provides the best security with its extended facilities and services like MFA device. It also gives you the ability to add your own permissions and policies for securing data more encrypted.

**Task-6**

**Aim: To study and implementation of identity management.**

**Objectives:** From this experiment, the student will be able to,

* Understand concepts of virtualization and to use cloud as Infrastructure as a services.
* Learn the technique and its complexity
* Understand the importance of this technique from application point of view

**Outcomes:**

**Hardware / Software Required:**

**Theory:**

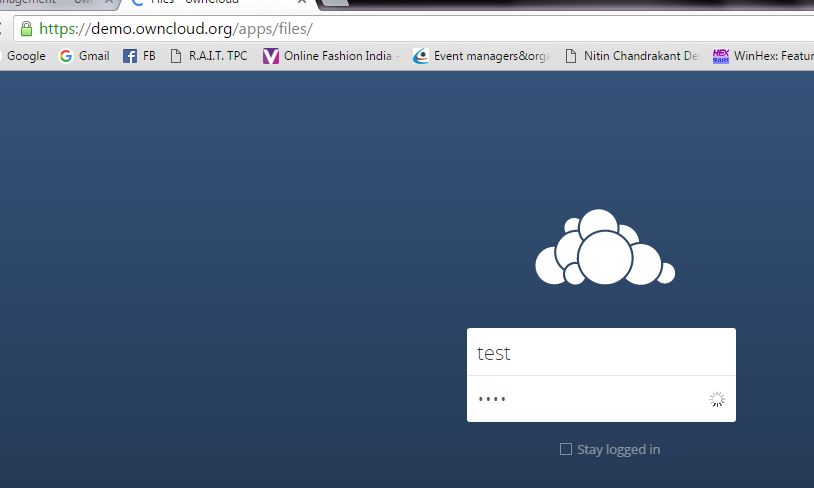
Identity Management

**Procedure:**

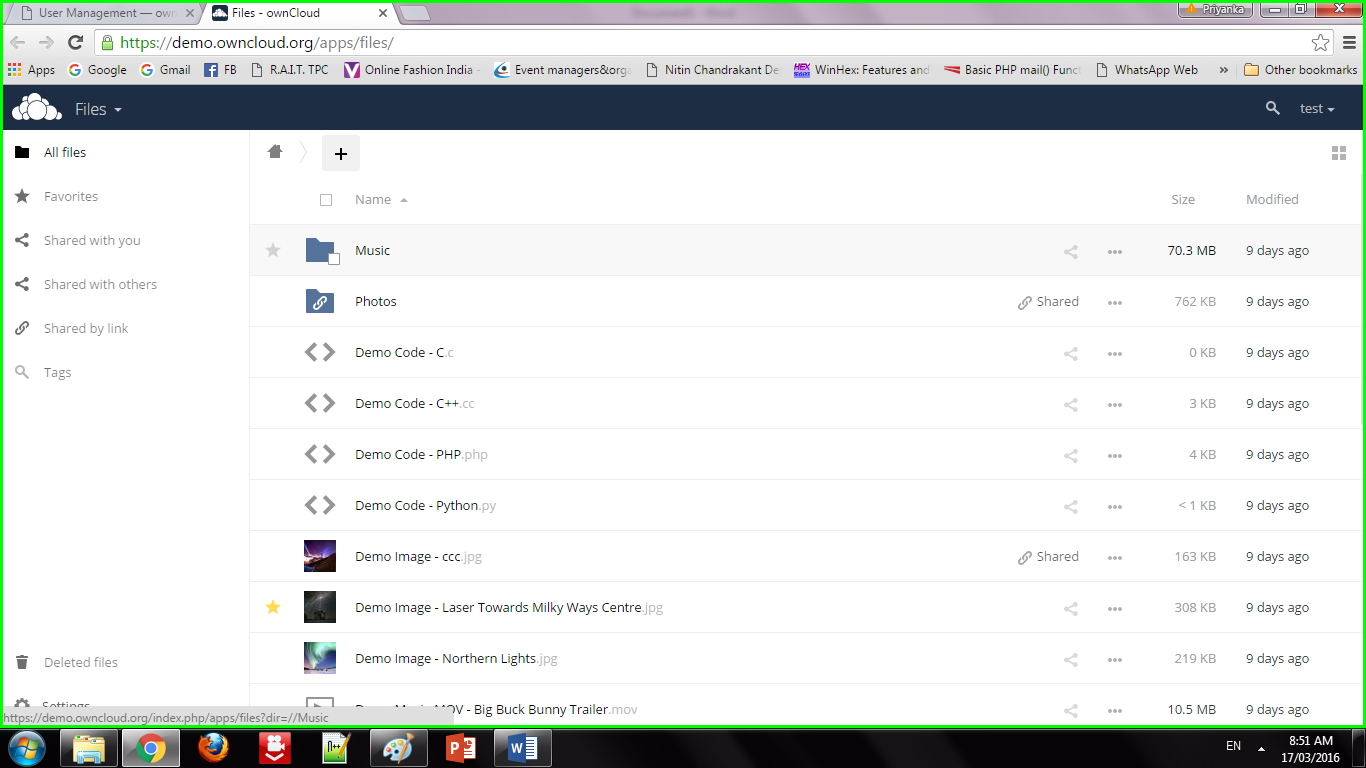
**Result:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_SNAPSHOTS\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

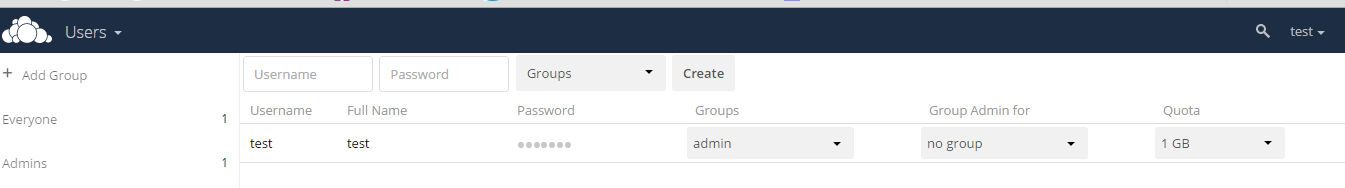
OwnCloud is open source file sync and share software for everyone from individuals operating the free ownCloud Server edition, to large enterprises and service providers operating the ownCloud Enterprise Subscription. ownCloud provides a safe, secure, and compliant file synchronization and sharing solution on servers that you control. You can share one or more files and folders on your computer, and synchronize them with your ownCloud server.

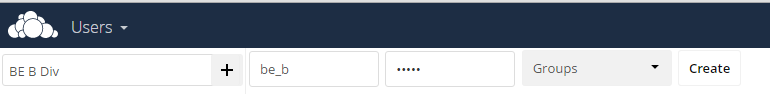


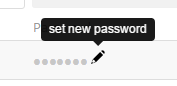
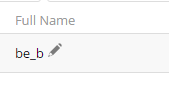
**Step 2:** By default, the ownCloud Web interface opens to your Files page. You can add, remove, and share files, and make changes based on the access privileges set by you (if you are administering the server) or by your server administrator. You can access your ownCloud files with the ownCloud web interface and create, preview, edit, delete, share, and re-share files. Your ownCloud administrator has the option to disable these features, so if any of them are missing on your system ask your server administrator.



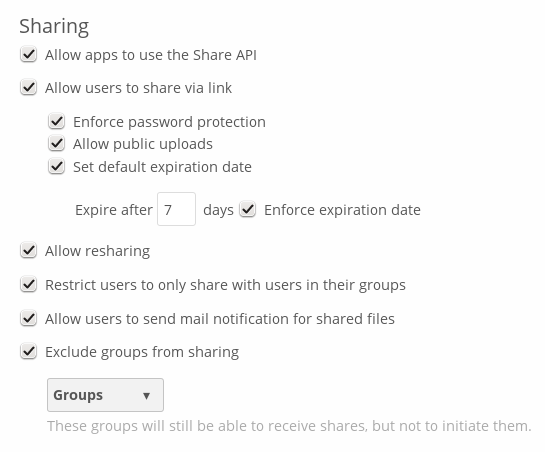
**Step 3:** **Apps Selection Menu:** Located in the upper left corner, click the arrow to open a dropdown menu to navigate to your various available apps. **Apps Information field:** Located in the left sidebar, this provides filters and tasks associated with your selected app. **Application View:** The main central field in the ownCloud user interface. This field displays the contents or user features of your selected app.

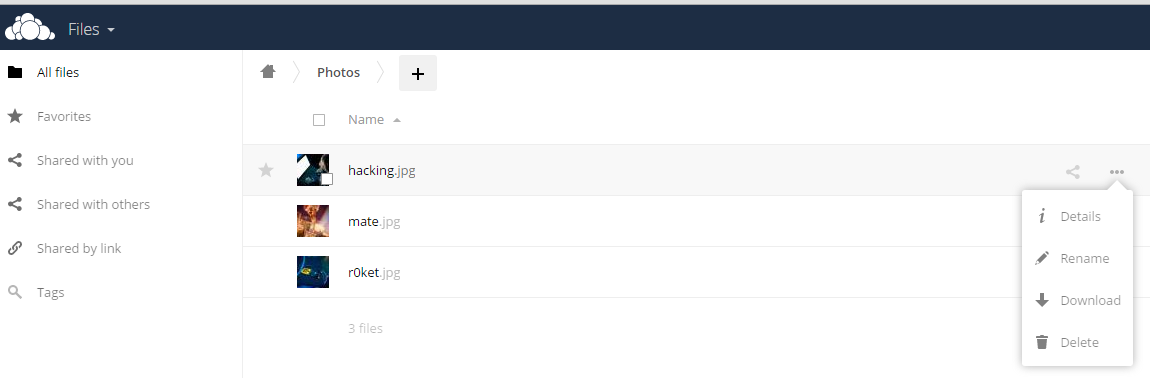


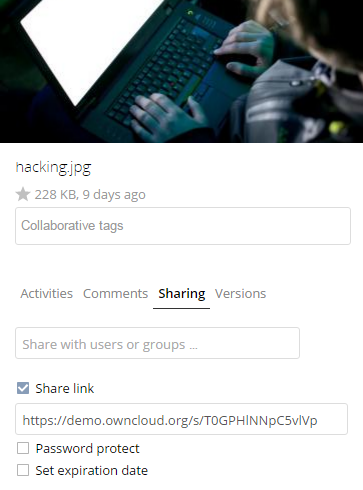


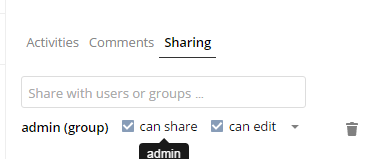
 

**Step 4:** Share the file or folder with a group or other users, and create public shares with hyperlinks. You can also see who you have shared with already, and revoke shares by clicking the trash can icon. If username auto-completion is enabled, when you start typing the user or group name ownCloud will automatically complete it for you. If your administrator has enabled email notifications, you can send an email notification of the new share from the sharing screen.









**Step 5: Five Share permissions are**

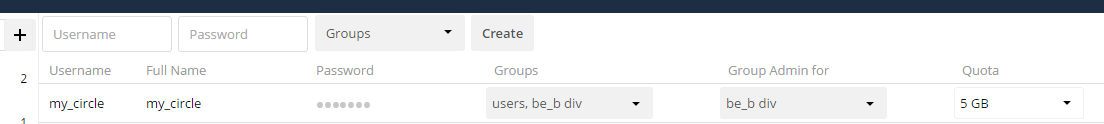
Can share; allows the users you share with to re-share.

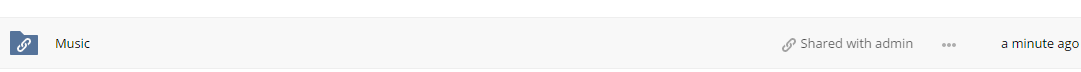
Can edit; allows the users you share with to edit your shared files, and to collaborate using the Documents app.

Create; allows the users you share with to create new files and add them to the share.

Change; allows uploading a new version of a shared file and replacing it.

Delete; allows the users you share with to delete shared files.





**Conclusion:**

We have studied how to use ownCloud for ensuring identity management of the users. We can create multiple groups and provide privileges to view or modify data as per defined permissions. It also enables simplified look and feel to be used by anyone.

**Task-7**

**Aim:** **Case Study - Amazon Web Services/Microsoft Azure/Google cloud services.**

Amazon Web Services (AWS) is a subsidiary of Amazon providing on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered pay-as-you-go basis. AWS offers a wide range of services including computing power, storage options, networking, databases, machine learning, analytics, and more.

Strengths:

**1.Market Leader**: AWS is the largest and most widely adopted cloud platform in the world, with a significant market share.

**2.Extensive Service Portfolio:** AWS offers an extensive range of services catering to diverse needs, from startups to enterprise-level applications.

**3.Global Infrastructure:** AWS has a vast global network of data centers, enabling low-latency access and high availability across regions.

**4.Ecosystem and Community:** AWS has a large and active community of users, developers, and partners, providing extensive support, resources, and integration options.

**5.Innovation:** AWS continually innovates and introduces new services and features, staying ahead of competitors and addressing emerging needs.

Weaknesses:

**1.Complex Pricing Structure:** AWS pricing can be complex, with various factors affecting costs such as usage, region, and service selection.

**2.Vendor Lock-in:** Utilizing AWS services extensively may lead to vendor lock-in, making it challenging to migrate to other platforms.

**3.Security Concerns:** While AWS has robust security measures, incidents like data breaches or outages can raise concerns about the security and reliability of cloud services.

**Microsoft Azure:**

Microsoft Azure is a cloud computing service created by Microsoft for building, testing, deploying, and managing applications and services through Microsoft-managed data centers. It offers a wide range of services including computing, analytics, storage, and networking.

Strengths:

**1.Integration with Microsoft Products:** Azure seamlessly integrates with other Microsoft products and services, providing a cohesive ecosystem for organizations already using Microsoft technologies.

**2.Hybrid Capabilities:** Azure offers strong support for hybrid cloud deployments, allowing businesses to integrate on-premises infrastructure with cloud services.

**3.Enterprise Focus:** Microsoft's strong presence in the enterprise market gives Azure an advantage in catering to the needs of large organizations, including compliance and regulatory requirements.

**4.Developer Tools and Support:** Azure provides robust developer tools, documentation, and support, making it easier for developers to build and deploy applications on the platform.

**5.Global Presence:** Azure has an extensive global network of data centers, ensuring low-latency access and high availability across regions.

**Weaknesses:**

**1.Learning Curve:** Azure's breadth of services and features can lead to a steep learning curve for users unfamiliar with the platform.

**2.Less Market Share:** While Azure is a major player in the cloud market, it still lags behind AWS in terms of market share, which may impact perceptions of reliability and adoption.

3**.Complexity:** Similar to AWS, Azure's pricing and service options can be complex, requiring careful planning and management to optimize costs and resources.

**Google Cloud Platform (GCP):**

**Google Cloud Platform (GCP**) is a suite of cloud computing services offered by Google, providing infrastructure as a service, platform as a service, and serverless computing environments.

**Strengths:**

**1.Global Network Infrastructure:** GCP leverages Google's extensive global network infrastructure, offering high performance, low-latency access, and scalability.

**2.Data Analytics and Machine Learning:** Google has deep expertise in data analytics and machine learning, reflected in GCP's powerful analytics and AI services, including BigQuery and TensorFlow.

**3.Containerization and Kubernetes:** GCP has strong support for containerization technologies, particularly Kubernetes, making it a popular choice for container-based deployments and microservices architectures.

**4.Innovation:** Google is known for innovation, and GCP frequently introduces new services and features, often leveraging cutting-edge technologies and advancements.

**5.Pricing and Discounts:** GCP's pricing structure is relatively straightforward, and it offers various discounts and pricing options, including sustained use discounts and committed use discounts.

**Weaknesses**:

**1.Market Perception:** Despite its technical prowess, GCP still lags behind AWS and Azure in terms of market share and enterprise adoption, which may impact perceptions of reliability and maturity.

**2.Less Diverse Service Portfolio**: While GCP offers a wide range of services, it may have fewer options compared to AWS and Azure in certain areas, limiting flexibility for some use cases.

**3.Customer Support:** Some users have reported challenges with GCP's customer support, particularly in comparison to AWS and Azure, which offer more extensive support options.

Each of these cloud service providers has its own strengths and weaknesses, and the choice between them often depends on factors such as specific business requirements, existing technology stack, budget considerations, and organizational preferences.

