1. Describe STaaS. Storage as a service (STaaS) is a managed service in which the provider supplies the

customer with access to a data storage platform. The service can be delivered on-premises

from infrastructure dedicated to a single customer, or it can be provided from the public cloud

as a shared service purchased by subscription and billed according to one or more useful

metrics. STaaS customers access individual storage services through standard system interface

protocols or application program interfaces (APIs). Typical offerings include bare-metal

storage capacity; raw storage volumes; network file systems, storage objects; and storage

applications that support file sharing and backup lifecycle management. Storage as a service was initially seen as a cost-effective way for small and mid-size

businesses that lacked the technical personnel and capital budget to implement and maintain

their storage infrastructure. Today, companies of all sizes use storage as a service. Uses of STaaS

Storage as a service can be used for data transfers and redundant storage, as well as to restore

any corrupted or lost data. CIOs may want to use STaaS for the ability to deploy resources at

an instant or to replace some existing storage space -- leaving room for on-premises storage

hardware. CIOs may also appreciate the ability to tailor storage capacity and performance per

workload. As an example, instead of maintaining a large tape library and arranging to vault (store) tapes

off site, a network administrator that uses STaaS for backups could specify what data on the

network should be backed up and how often it should be backed up. Their company would

sign a service-level agreement (SLA) whereby the STaaS provider agrees to rent storage

space on a cost-per-gigabyte-stored and cost-per-data-transfer basis, and the company's data

would then be automatically transferred at the specified time over the storage provider's

proprietary wide area network (WAN) or the internet. If the company's data were to ever

become corrupt or get lost, the network administrator could contact the STaaS provider and

request a copy of the data.

2. Mention the different storage types offered by the cloud providers and list the

example services for each type in GCP/AWS/Azure. GCP Storage Services:

Google Cloud provides three primary services for different types of storage: Persistent Disks

for block storage, Filestore for network file storage, and Cloud Storage for object storage.

These services are at the platform's core and act as building blocks for most of the Google

Cloud services and, by extension, to the systems you build on top of it. Google Cloud Persistent Disks (Block Storage:)Block storage is the traditional storage type

in the cloud and on-premises systems. A Google Cloud Persistent Disk provides block

storage and it is used by all virtual machines in Google Cloud (Google Cloud Compute

Engine). The easiest way to understand it is by imagining those Persistent Disks as mere USB

drives. They can be attached or detached from virtual machines and enable you to build, as

the name suggests, data persistence for your services whenever virtual machines are started, stopped, or terminated. Google Cloud Filestore (Network File Storage): Filestore is a fully managed Google Cloud

service that provides network file storage. Network file storage is not a new cloud concept

and very much like block storage, it also exists in your typical on-premises data centre. AWS Storage Services:

The types of AWS Storage services are:

Amazon Simple Storage EC3: Amazon Simple Storage Service (Amazon S3) is an object

storage service offering industry-leading scalability, data availability, security, and

performance. Customers of all sizes and industries can store and protect any amount of data

for virtually any use case, such as data lakes, cloud-native applications, and mobile apps. With cost-effective storage classes and easy-to-use management features, you can optimize

costs, organize data, and configure fine-tuned access controls to meet specific business, organizational, and compliance requirements. Amazon Elastic File System: Amazon Elastic File System (Amazon EFS) automatically

grows and shrinks as you add and remove files with no need for management or provisioning. Amazon FSx: Amazon FSx makes it easy and cost-effective to launch, run, and scale

feature-rich, high-performance file systems in the cloud. It supports a wide range of

workloads with its reliability, security, scalability, and broad capabilities. Amazon FSx is

built on the latest AWS compute, networking, and disk technologies to provide high

performance and lower TCO. And as a fully managed service, it handles hardware

provisioning, patching, and backups -- freeing you up to focus on your applications, end users, and your business. Amazon Elastic Block Store: Amazon Elastic Block Store (Amazon EBS) is an easy-to-use, scalable, high-performance block-storage service designed for Amazon Elastic Compute

Cloud (Amazon EC2). Azure Storage Services:

There are 4 types of storage in Azure, namely: File, Blob, Queue and Table. File Storage: Azure file storage makes it easy to move applications which depend on regular

file shares to the cloud. File storage uses the SMB 2.1 or 3.0 protocol and can be accessed by

multiple applications simultaneously. There are some SMB features which are not currently

supported. Blob Storage: Blob is an acronym for a large binary object. Blobs typically include large

files that are unstructured, such as images, video, music files, backup files etc. Blob storage can be divided into two access tiers: a hot access tier for data accessed

frequently and a cold access tier for data not accessed very often. Queue Storage is somewhat like MSMQ. It allows you to decouple your components and

have reliable asynchronous communication. In Azure Queue Storage, the number of queues is

only limited by the capacity of the storage account. Table storage is used to store semi-structured data in a key-value format in a NoSQL

datastore. Azure table storage can store petabytes of data, can scale and is inexpensive. Table

storage can be accessed using REST and some OData protocols or the Storage Explorer tool.

1.Download AWS python CLI and install

2. Go to cmd -> type -> aws configure

3 it will as for Id , Key , Region : provide the necessary details from aws console mamagent -> top right corner -> aws details (i)

-> click and open AWC CLI

-> **AWS CLI:**

Copy and paste the following into ~/.aws/credentials

[default]

aws\_access\_key\_id=ASIA27F4BZATIPGCXDII

aws\_secret\_access\_key=Qb2fWTEF53o1lRDh3XvzbhT5w7eVT0672JXjzVCp

aws\_session\_token=FwoGZXIvYXdzEPH//////////wEaDIPmnNHRtCj07wQVtiLSAdpCnBqcvN3oj1LnYx0oGHtY3JUJsVaLuAqsbhcvDzCKZpn+eV+al3OFS1z0sQpFHnCsl9pMrgi4uJygxf9TPZ1UQnyPOEf2B44V/vRklUAWP/jqIZpxxsfhOpKxBgR7iGK5hVc2tDwUdC02DxImYTWu2cznoWVFL3W5ajpBwBfKtbpXyxTL9MbLdeA9ABubGdbYgCX169BOYq6rRnwfqjNzPcmFtsXPhFKJ0TMej/esXWK+Gf4tn3Ecd6RFCTQ3OlS8cJxh2f262+/BQfhmGCJbOijjxsmbBjIt3Z6PTzr4asQJVOD5PMtNTfXl7QXZWZXHI+ex7/vDUS2z6VRkJN8t8o3d3RSB

3. Got to c:/user / .aws ->credentials file

Copy paste above into this file and save

4.. "py -m pip install boto3 in cmd to install boto3

5. Open py file and paste following to perform file upload to folder called image in aws s3 bucker ‘deccanherald’

Code :file upload to bucket

import boto3 # pip install boto3

# Let's use Amazon S3

s3 = boto3.resource("s3")

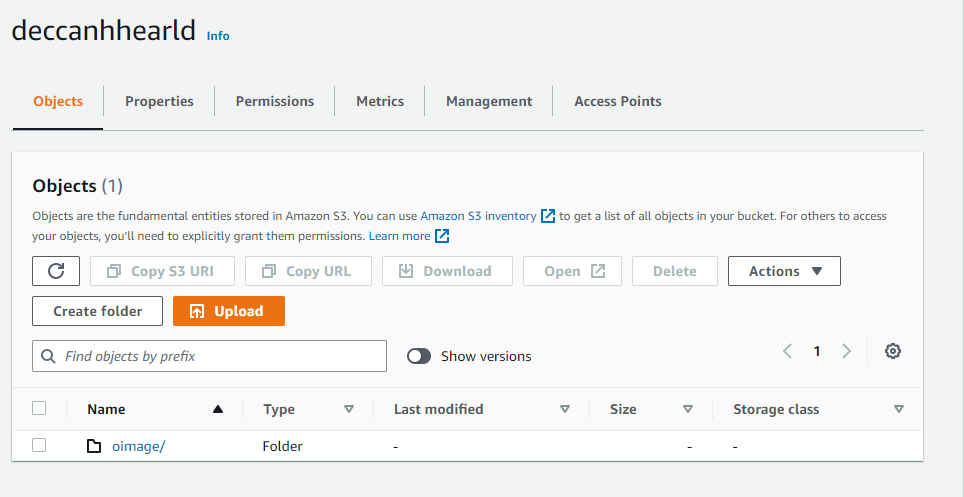
# Print out bucket names

for bucket in s3.buckets.all():

print(bucket.name)

BUCKET = "deccanhhearld"

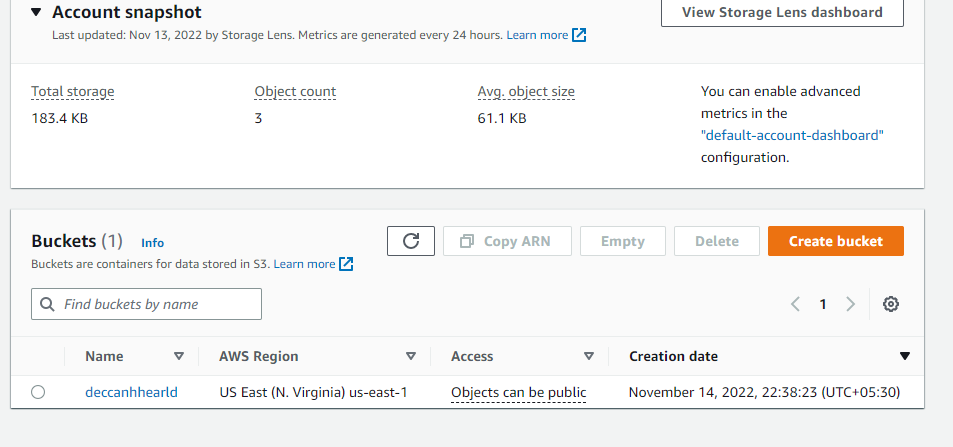
s3.Bucket(BUCKET).upload\_file('C:/Users/rakip/Downloads/1\_mFgpjcSK2p69ohyYD5DQQw.png', 'oimage/hello.png')

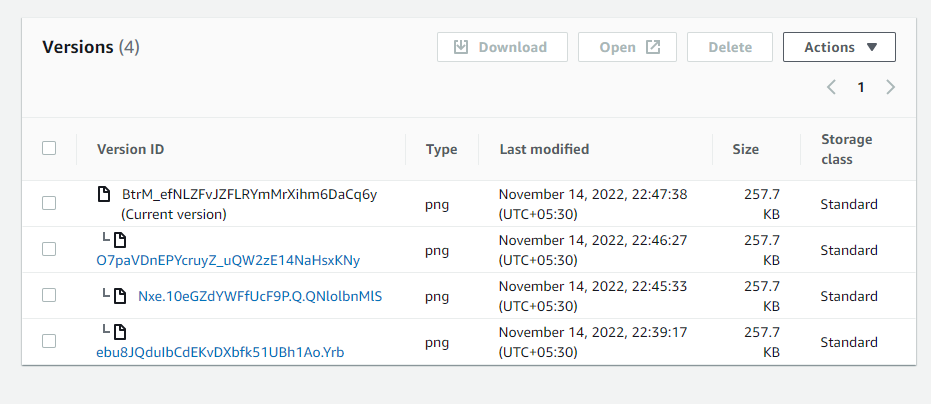
1Go to s3 service 

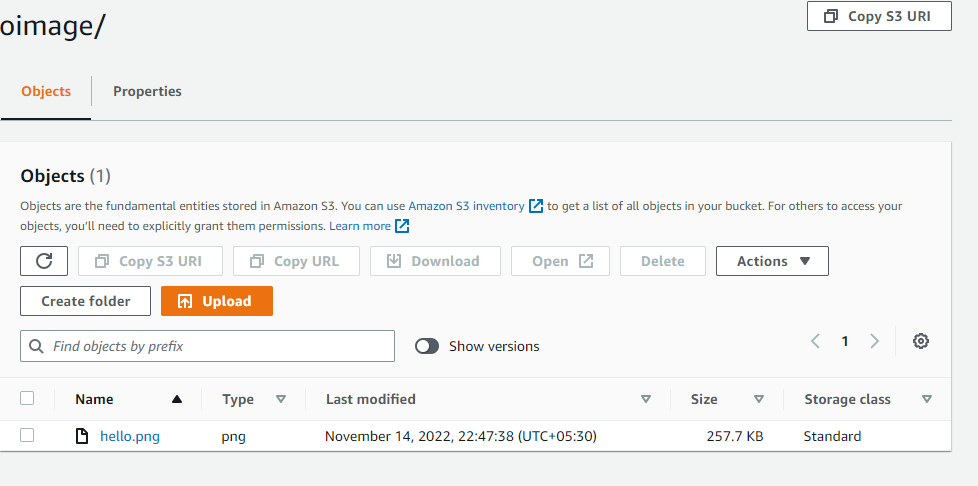
2 create bucket

3 choose name , region

4 enable public versioning and server side encryption



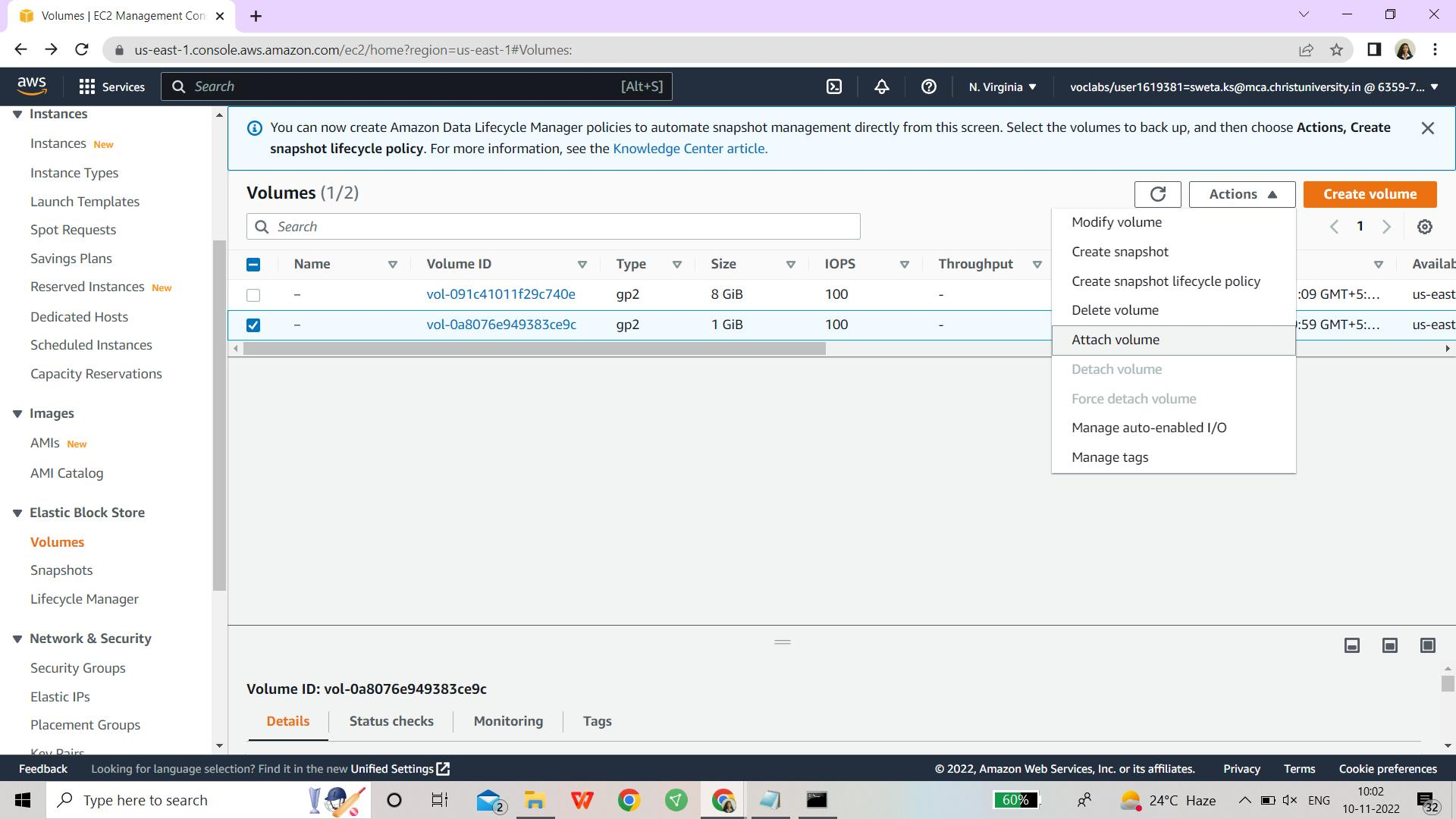
Hello.png is uploaded in image folder of deccanherald bucket



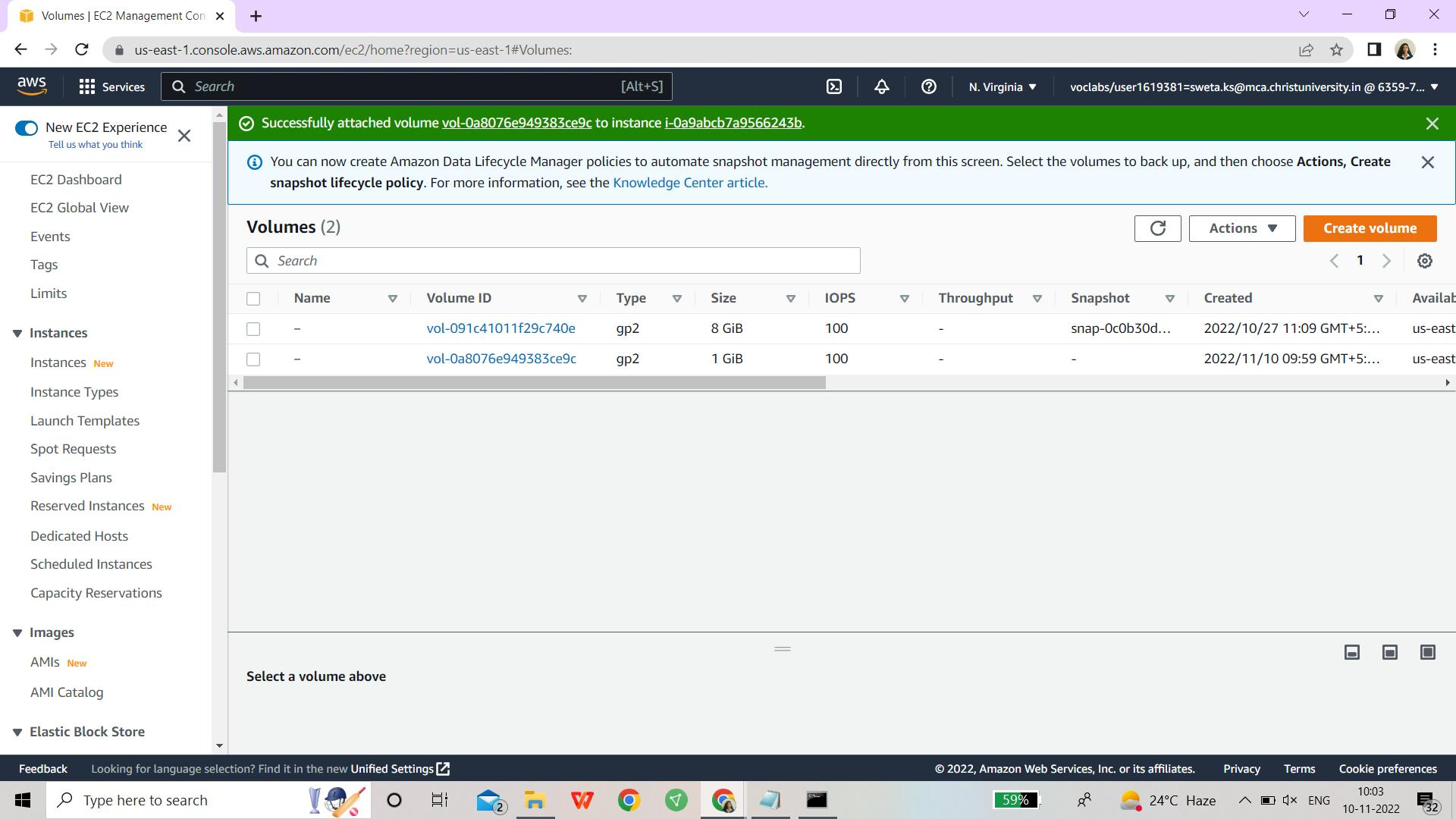
Multiple versions of same object png

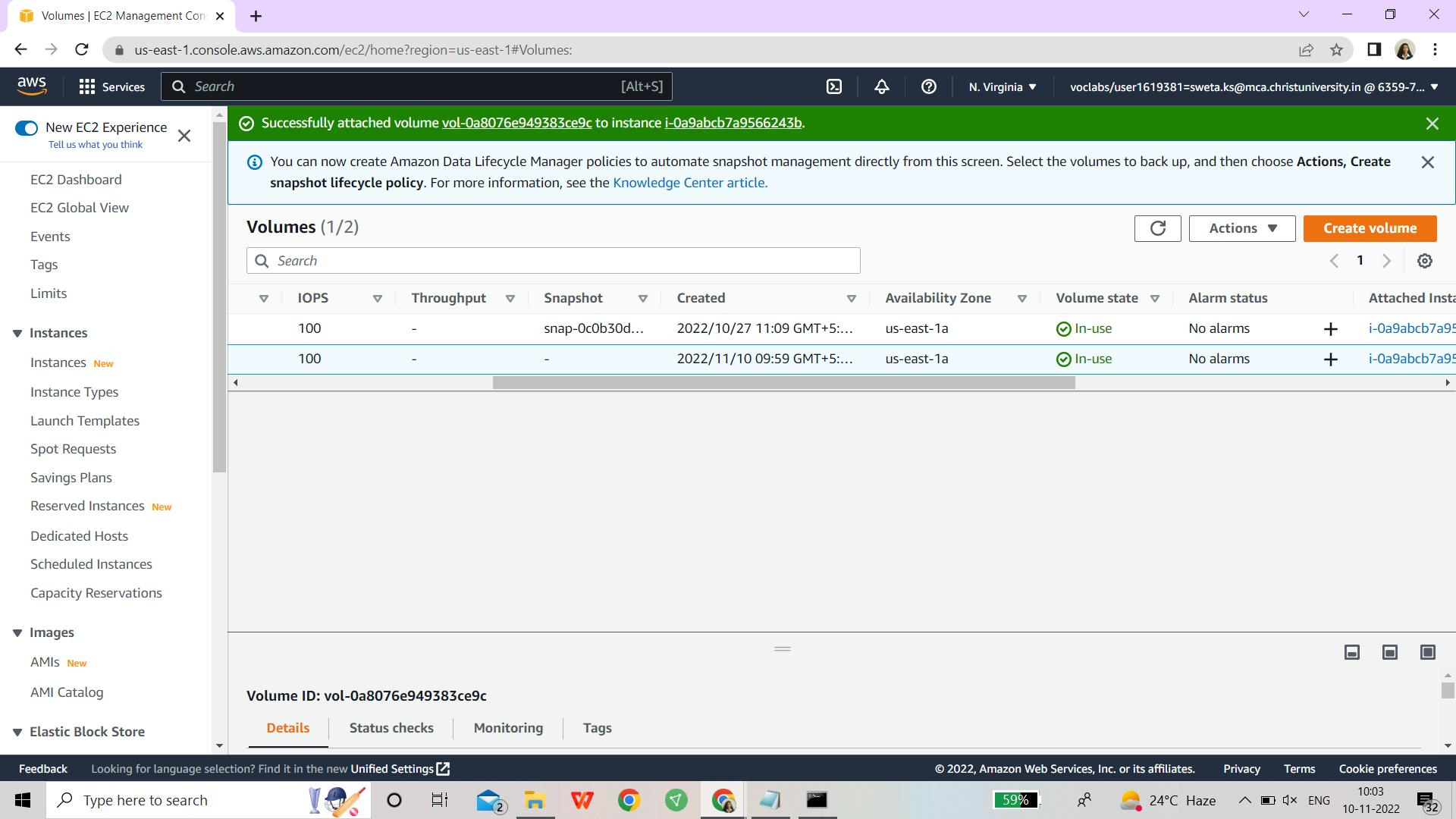
storage can be accessed using REST and some OData protocols or the Storage Explorer tool. 3. Demonstrate the following:

i). Create a New Block Store and attach it to a VM instance. create an EBS block volume:

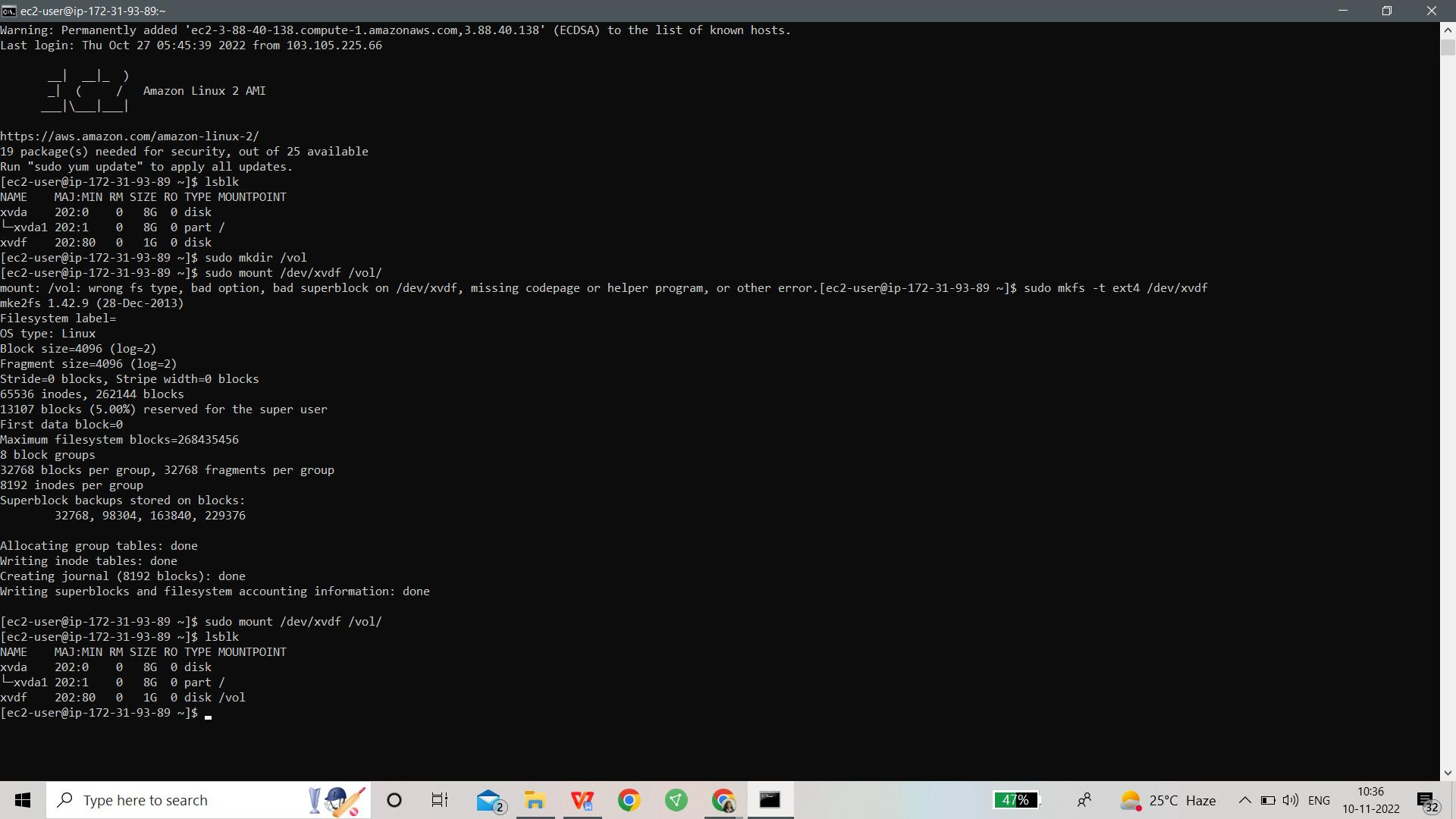


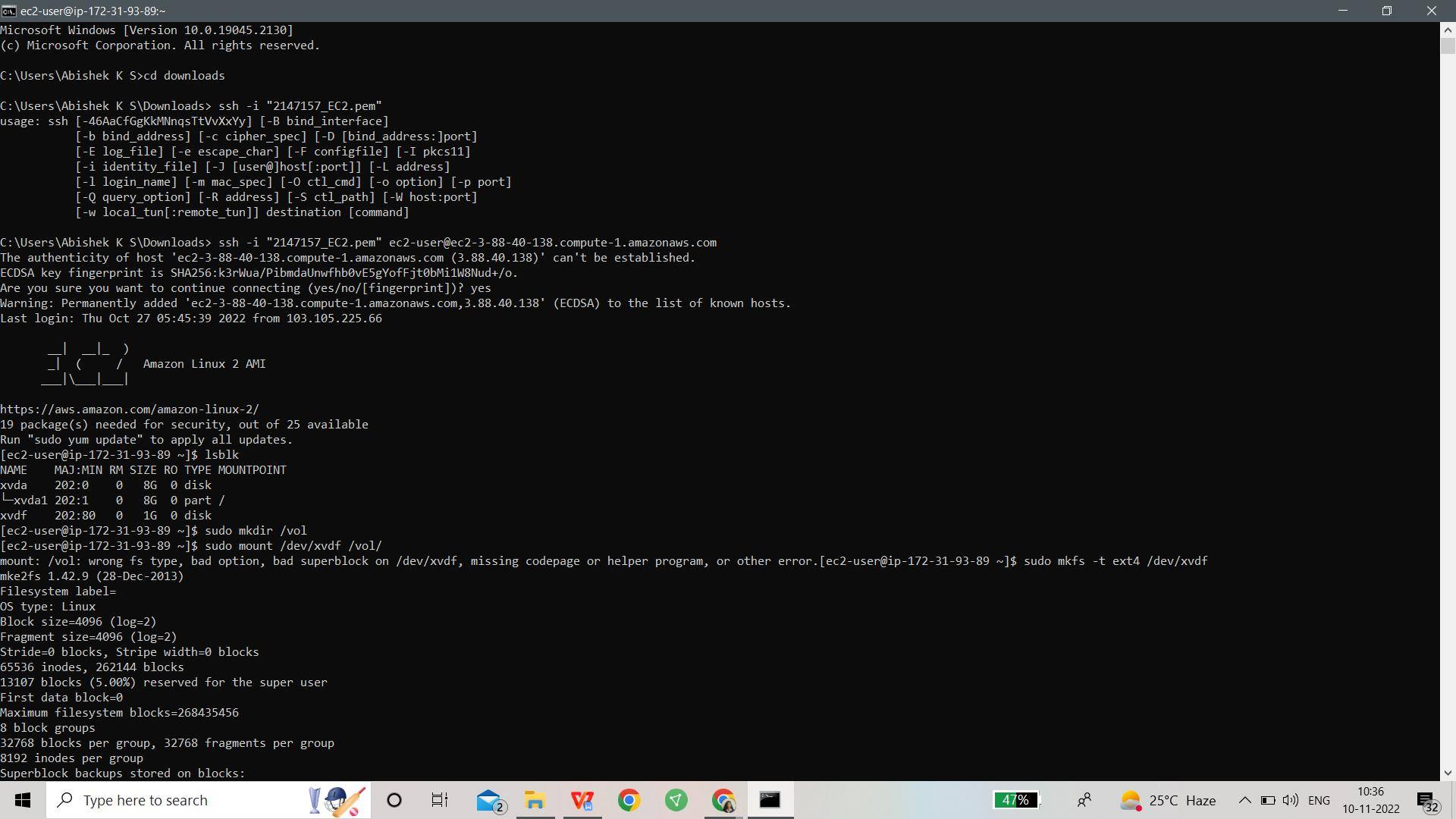
attach volume





mount the volume:





**4.Assume** **that** **An** **online** **news** **Agency** **wants** **to** **support** **its** **agents** **and** **editors** **for** **fast**

**news** **publication** **using** **cloud** **technologies.** **They** **expect** **a** **small** **software** **application** **for**

**the** **newsagents** **and** **editors** **for** **the** **following** **requirements.**

**i).** **Provision** **to** **upload** **the** **images** **ofthe** **events** **from** **the** **venue** **to** **the** **folder** **named**

**“oimage”** **present** **in** **the** **cloud** **storage.**

Creating bucket is explained at beginning

