Cycle 08 AWS Homework

1. File Upload Task

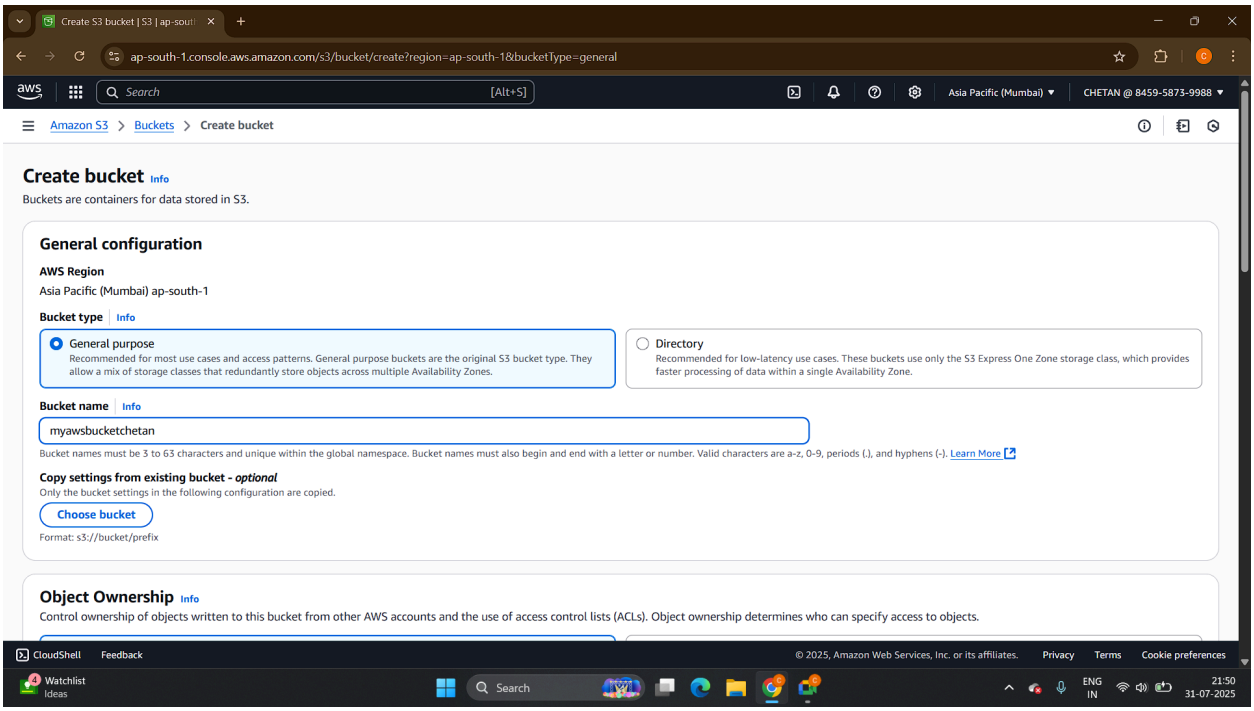
Objective: Test Amazon S3's support for diverse file formats.

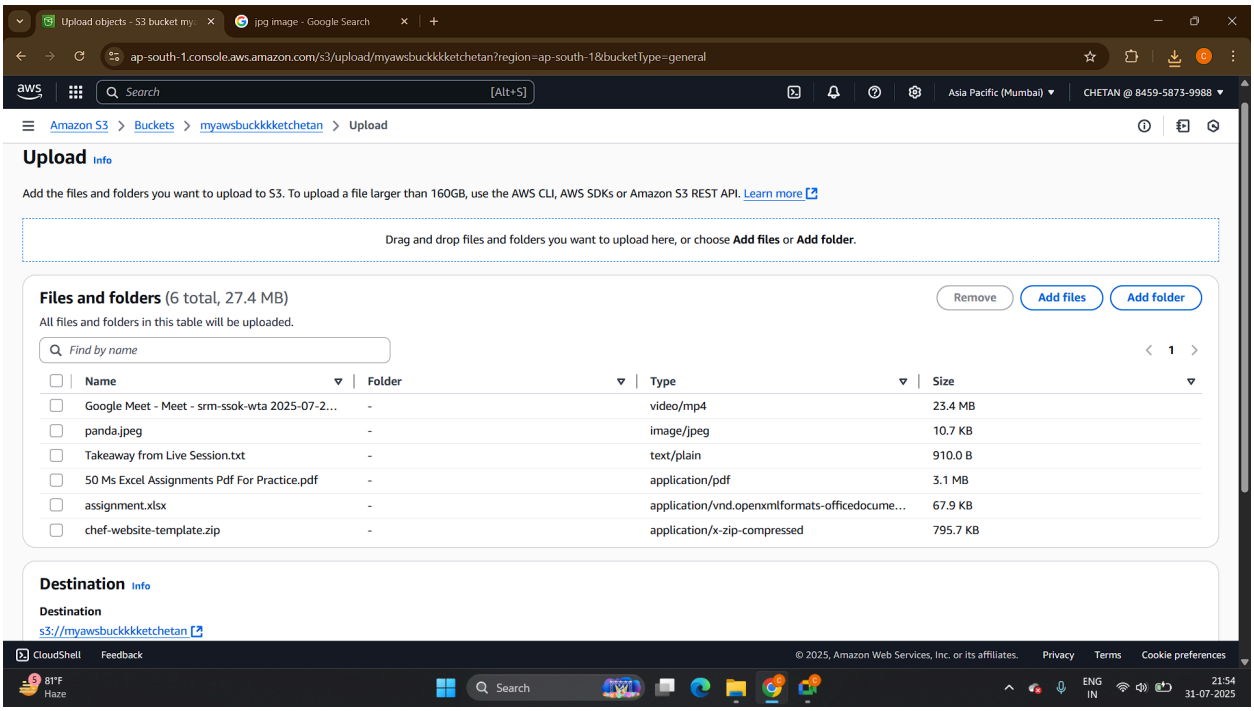
Steps:

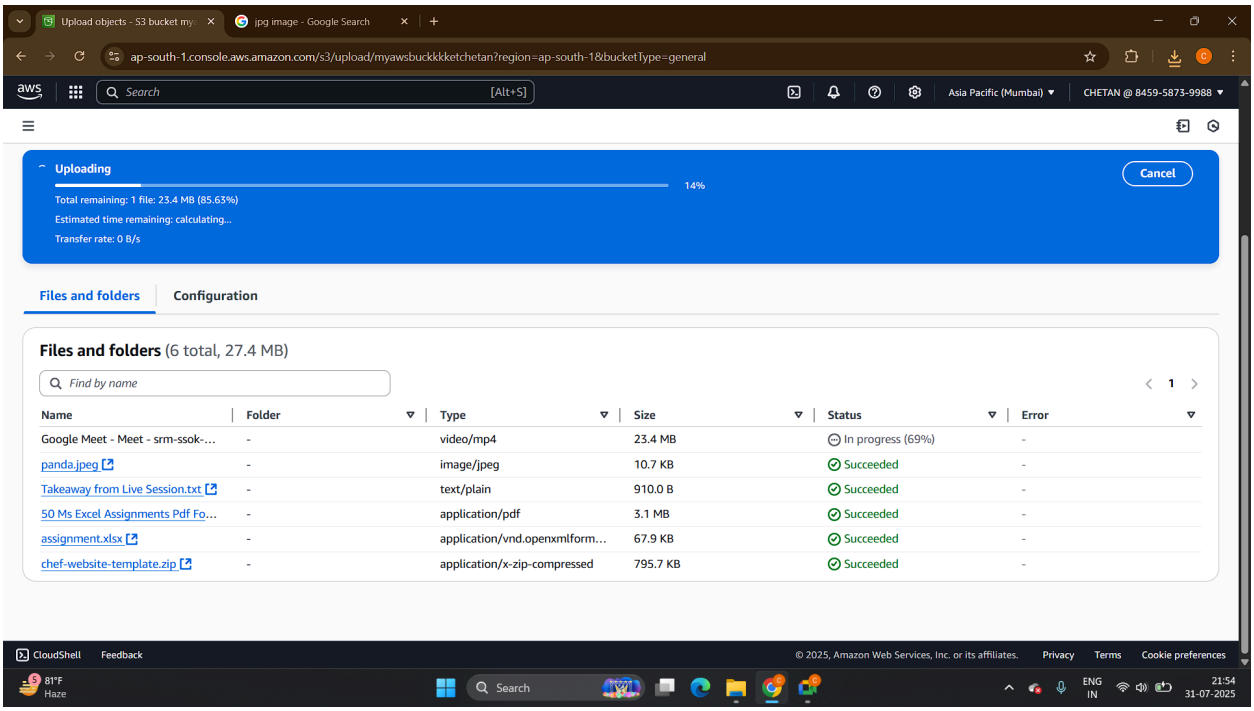
- Use AWS CLI to upload files:
  - Text file: `.txt`
  - CSV: `.csv`
  - Excel: `.xlsx`
  - Image formats: `.jpeg`, `.png`, `.gif`
  - PDF: `.pdf`
  - Small video: `.mp4` or `.mov`
- Example command:

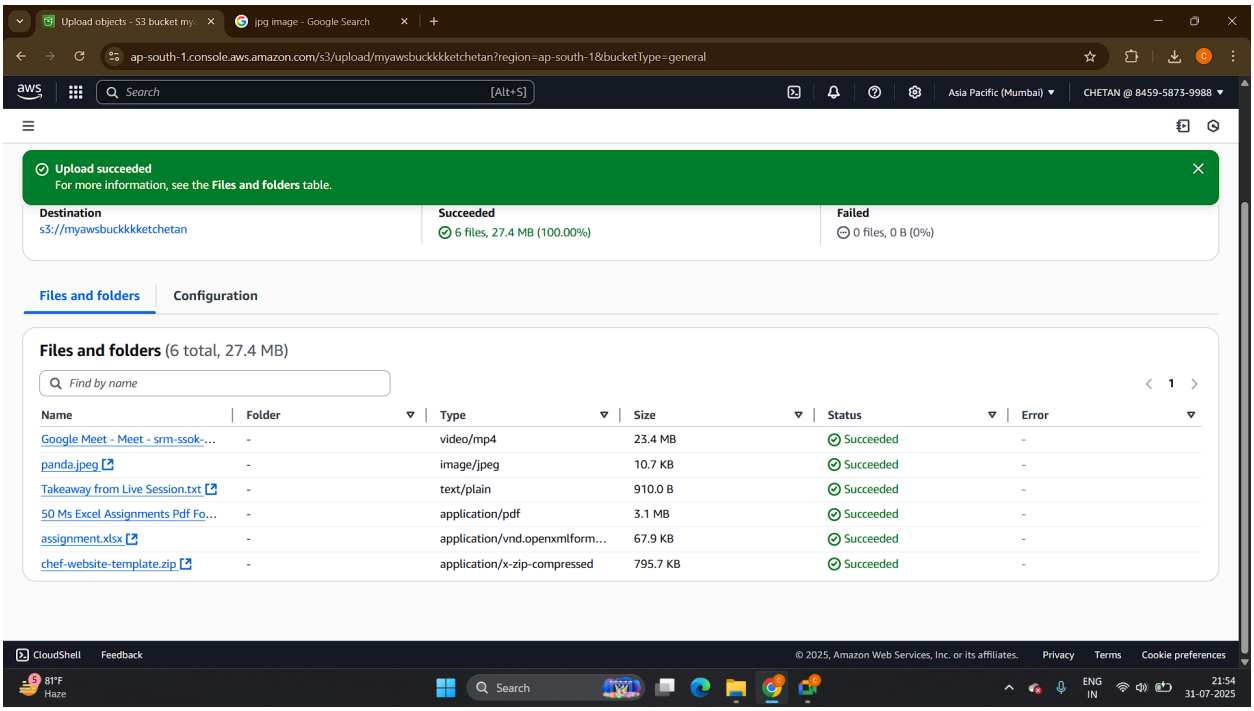
```
bash
CopyEdit
aws s3 cp filename.ext s3://your-bucket-name/
```

- Verify file upload via:
  - AWS Console
  - `aws s3 ls s3://your-bucket-name/`


Create S3 bucket | S3 | ap-south
ap-south-1.console.aws.amazon.com/s3/bucket/create?region=ap-south-1&bucketType=general
Search [Alt+S]
Asia Pacific (Mumbai)
CHETAN @ 8459-5873-9988
Amazon S3 > Buckets > Create bucket
Create bucket Info
Buckets are containers for data stored in S3.
General configuration
AWS Region
Asia Pacific (Mumbai) ap-south-1
Bucket type Info
General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.
Directory
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.
Bucket name Info
myawsbucketchetan
Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also begin and end with a letter or number. Valid characters are a-z, 0-9, periods (.), and hyphens (-). Learn More
Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.
Choose bucket
Format: s3://bucket/prefix
Object Ownership Info
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Watchlist
Ideas
Search
ENG
IN
21:50
31-07-2025


Upload objects - S3 bucket my
jpg image - Google Search
ap-south-1.console.aws.amazon.com/s3/upload/myawsbuckkkketchetan?region=ap-south-1&bucketType=general
Search [Alt+S]
Asia Pacific (Mumbai)
CHETAN @ 8459-5873-9988
Amazon S3 > Buckets > myawsbuckkkketchetan > Upload
Upload Info
Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDKs or Amazon S3 REST API. Learn more
Drag and drop files and folders you want to upload here, or choose Add files or Add folder.
Files and folders (6 total, 27.4 MB)
All files and folders in this table will be uploaded.
Remove
Add files
Add folder
Find by name
1
Name
Folder
Type
Size
Google Meet - Meet - srm-ssok-wta 2025-07-2... - video/mp4 23.4 MB
panda.jpeg - image/jpeg 10.7 KB
Takeaway from Live Session.txt - text/plain 910.0 B
50 Ms Excel Assignments Pdf For Practice.pdf - application/pdf 3.1 MB
assignment.xlsx - application/vnd.openxmlformats-officedocume... 67.9 KB
chef-website-template.zip - application/x-zip-compressed 795.7 KB
Destination Info
Destination
s3://myawsbuckkkketchetan
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
81°F
Haze
Search
ENG
IN
21:54
31-07-2025





2. Multi-Part Upload Practice

Objective: Simulate uploading large files in parts.

Steps:

- Split a large file (1GB+) using a script or tool.
- Use AWS CLI to:
  1. Initiate multipart upload:

     ```
     css
     CopyEdit
     aws s3api create-multipart-upload --bucket your-bucket-name --key l
     argefile.bin
     ```

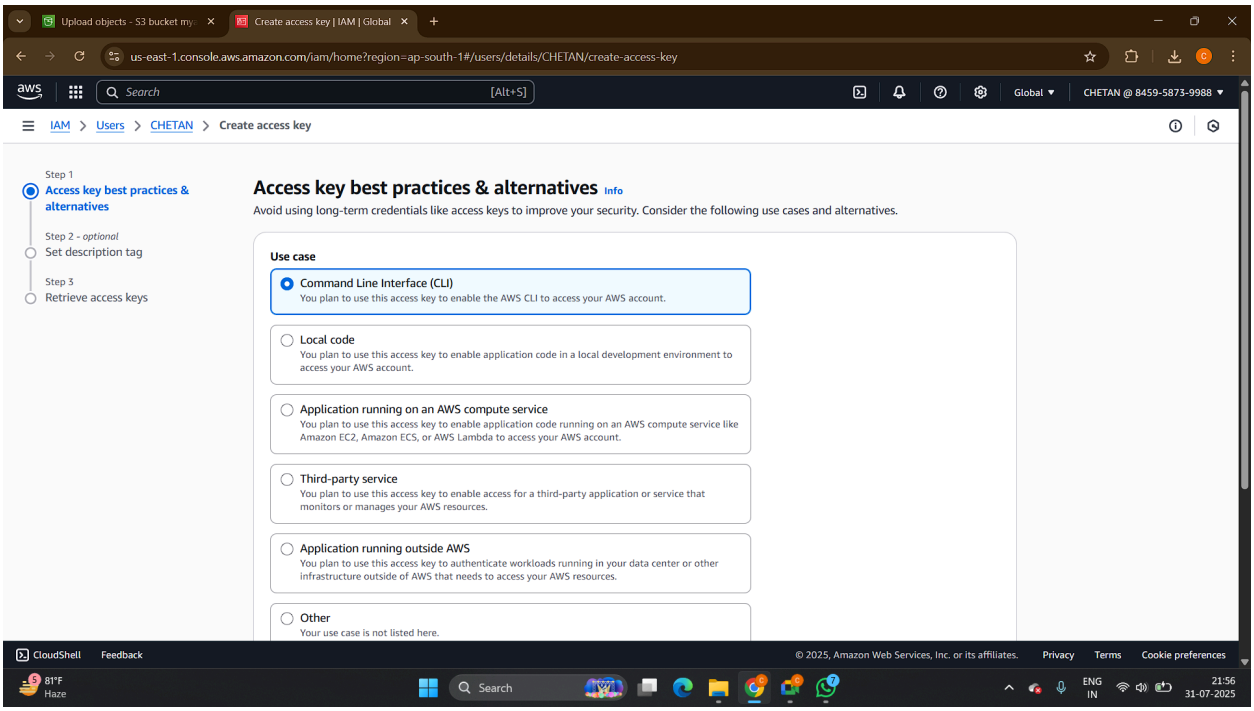
  2. Upload each part with `upload-part`:

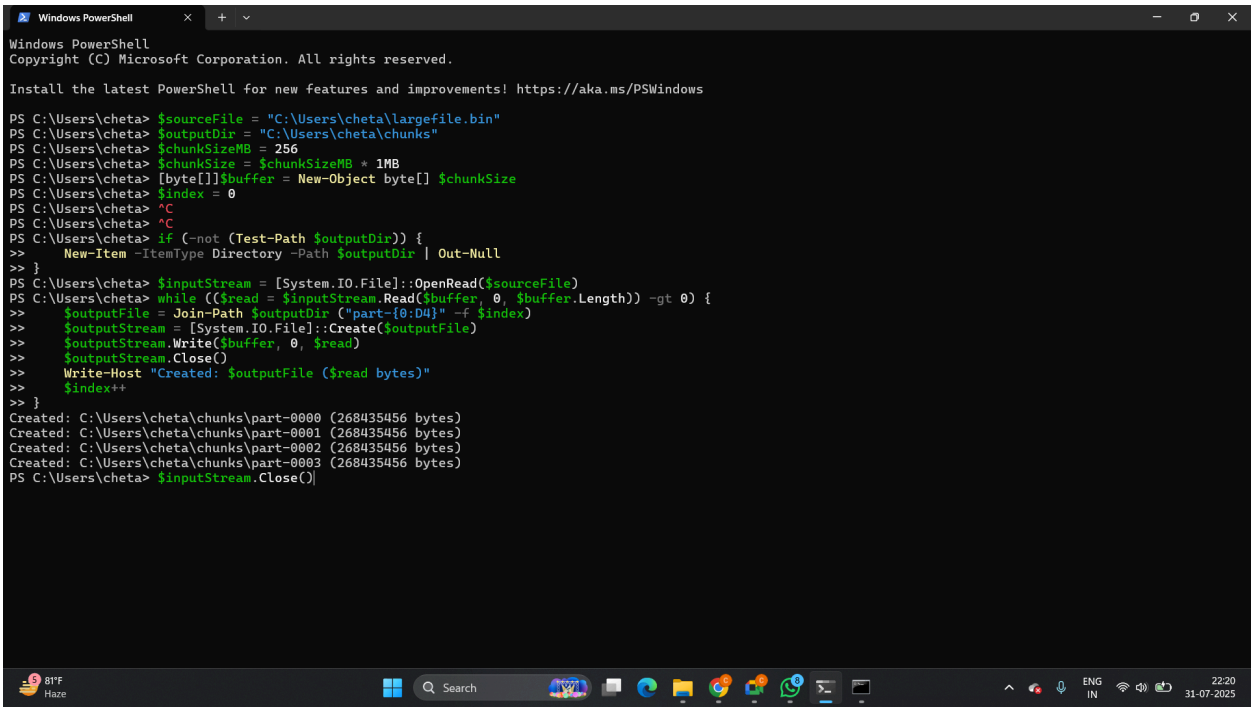
     ```
     css
     CopyEdit
     aws s3api upload-part --bucket your-bucket-name --key largefile.bin
     --part-number <n> --body path/to/part --upload-id <UploadId>
     ```

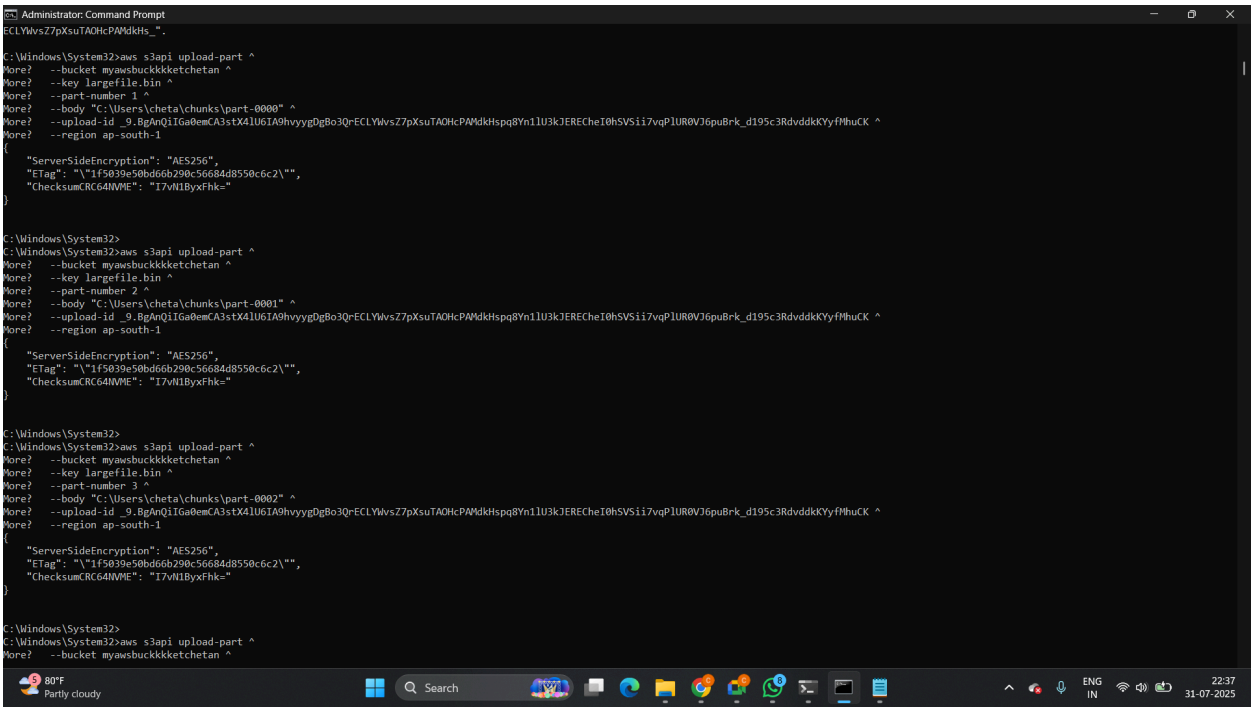
  3. Complete upload with a JSON file of `ETags`:

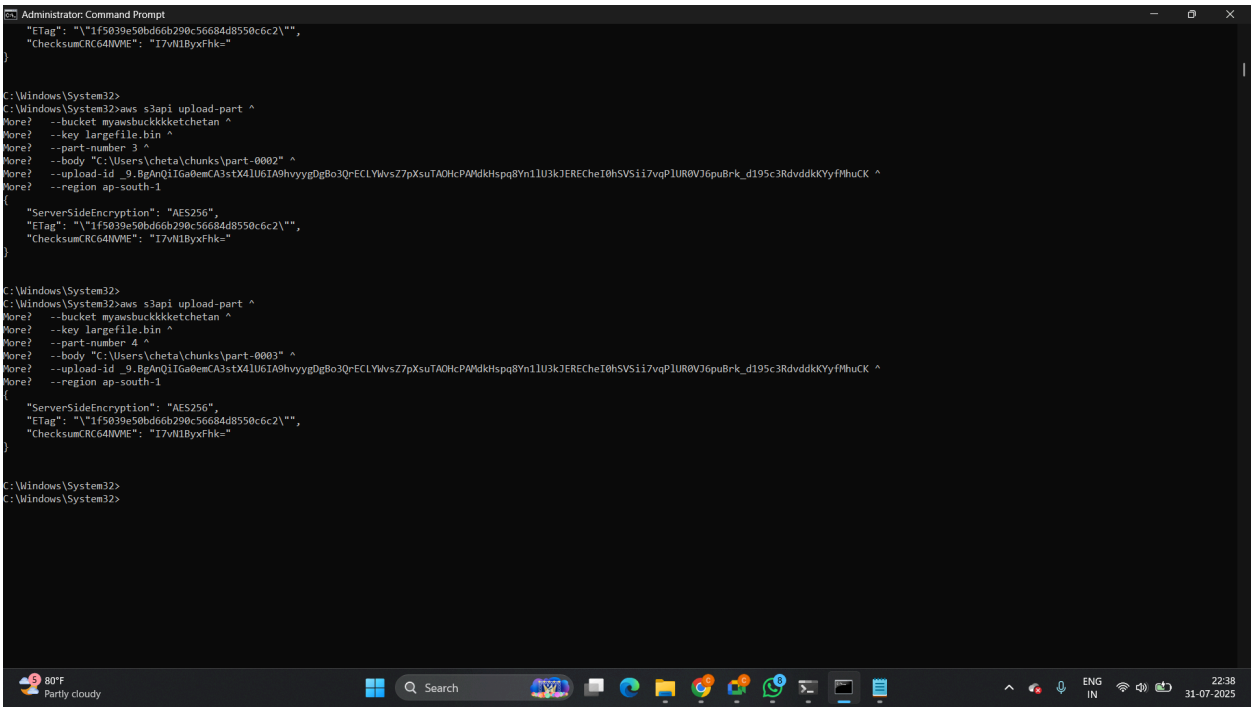
     ```
     css
     CopyEdit
     aws s3api complete-multipart-upload --bucket your-bucket-name --k
     ey largefile.bin --upload-id <UploadId> --multipart-upload file://parts.j
     son
     ```

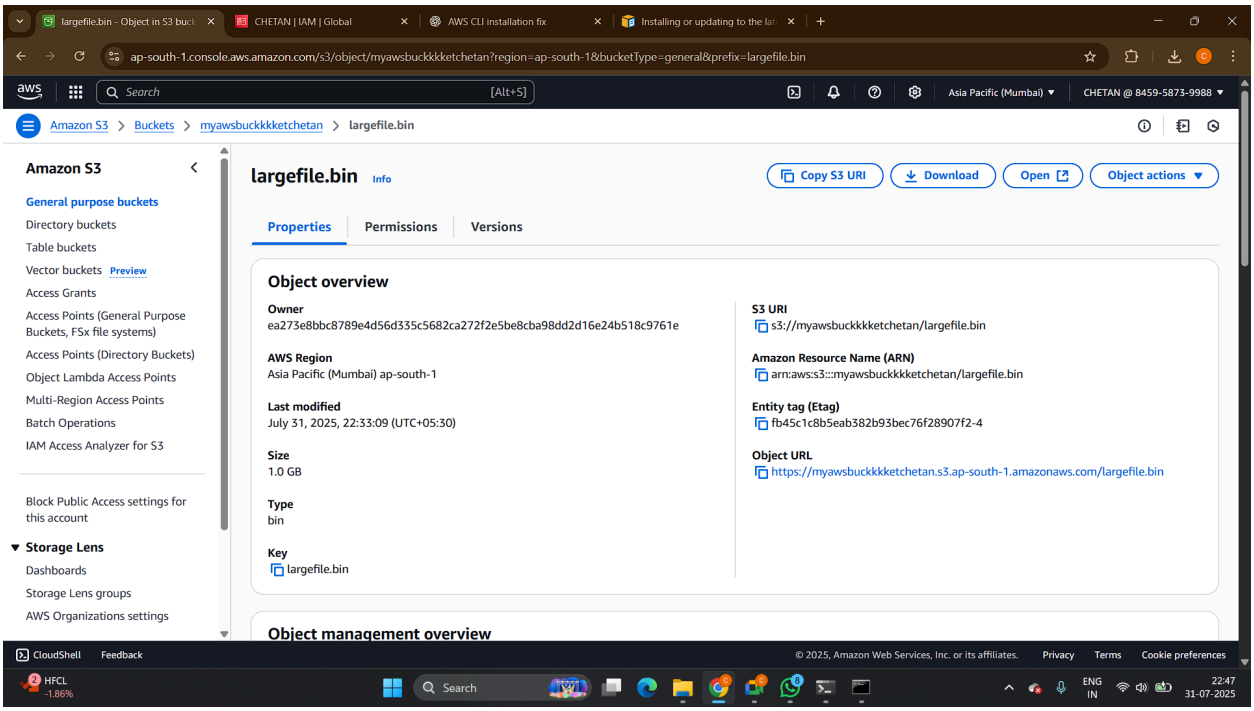
- You can simulate interruption by pausing/stopping midway, then resume using the same UploadId.


Upload objects - S3 bucket my
Create access key | IAM | Global
us-east-1.console.aws.amazon.com/iam/home?region=ap-south-1#/users/details/CHETAN/create-access-key
Search
[Alt+S]
Global
CHETAN @ 8459-5873-9988
IAM > Users > CHETAN > Create access key
Step 1
Access key best practices & alternatives
Step 2 - optional
Set description tag
Step 3
Retrieve access keys
Access key best practices & alternatives Info
Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.
Use case
Command Line Interface (CLI)
You plan to use this access key to enable the AWS CLI to access your AWS account.
Local code
You plan to use this access key to enable application code in a local development environment to access your AWS account.
Application running on an AWS compute service
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.
Third-party service
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.
Application running outside AWS
You plan to use this access key to authenticate workloads running in your data center or other infrastructure outside of AWS that needs to access your AWS resources.
Other
Your use case is not listed here.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
81°F
Haze
Search
ENG
IN
21:56
31-07-2025


Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\Users\cheta> $sourceFile = "C:\Users\cheta\largefile.bin"
PS C:\Users\cheta> $outputDir = "C:\Users\cheta\chunks"
PS C:\Users\cheta> $chunkSizeMB = 256
PS C:\Users\cheta> $chunkSize = $chunkSizeMB * 1MB
PS C:\Users\cheta> [byte[]]$buffer = New-Object byte[] $chunkSize
PS C:\Users\cheta> $index = 0
PS C:\Users\cheta> ^C
PS C:\Users\cheta> ^C
PS C:\Users\cheta> if (-not (Test-Path $outputDir)) {
>> New-Item -ItemType Directory -Path $outputDir | Out-Null
>> }
PS C:\Users\cheta> $inputStream = [System.IO.File]::OpenRead($sourceFile)
PS C:\Users\cheta> while (($read = $inputStream.Read($buffer, 0, $buffer.Length)) -gt 0) {
>> $outputFile = Join-Path $outputDir ("part-{0:D4}" -f $index)
>> $outputStream = [System.IO.File]::Create($outputFile)
>> $outputStream.Write($buffer, 0, $read)
>> $outputStream.Close()
>> Write-Host "Created: $outputFile ($read bytes)"
>> $index++
>> }
Created: C:\Users\cheta\chunks\part-0000 (268435456 bytes)
Created: C:\Users\cheta\chunks\part-0001 (268435456 bytes)
Created: C:\Users\cheta\chunks\part-0002 (268435456 bytes)
Created: C:\Users\cheta\chunks\part-0003 (268435456 bytes)
PS C:\Users\cheta> $inputStream.Close()
81°F
Haze
Search
ENG
IN
22:20
31-07-2025


Administrator: Command Prompt
ECLYWvsZ7pXsuTAOHcPAMdkHs_".
C:\Windows\System32>aws s3api upload-part ^
More? --bucket myawsbuckkkketchetan ^
More? --key largefile.bin ^
More? --part-number 1 ^
More? --body "C:\Users\cheta\chunks\part-0000" ^
More? --upload-id _9.BgAnQiIGa0emCA3stX4lU6IA9hvyygDgBo3QrECLYWvsZ7pXsuTAOHcPAMdkHspq8Yn1lU3kJERECheI0hSVSii7vqPlUR0VJ6puBrk_d195c3RdvddkKYyfMhuCK ^
More? --region ap-south-1
{
"ServerSideEncryption": "AES256",
"ETag": "\"1f5039e50bd66b290c56684d8550c6c2\"",
"ChecksumCRC64NVME": "I7vN1ByxFhk="
}
C:\Windows\System32>
C:\Windows\System32>aws s3api upload-part ^
More? --bucket myawsbuckkkketchetan ^
More? --key largefile.bin ^
More? --part-number 2 ^
More? --body "C:\Users\cheta\chunks\part-0001" ^
More? --upload-id _9.BgAnQiIGa0emCA3stX4lU6IA9hvyygDgBo3QrECLYWvsZ7pXsuTAOHcPAMdkHspq8Yn1lU3kJERECheI0hSVSii7vqPlUR0VJ6puBrk_d195c3RdvddkKYyfMhuCK ^
More? --region ap-south-1
{
"ServerSideEncryption": "AES256",
"ETag": "\"1f5039e50bd66b290c56684d8550c6c2\"",
"ChecksumCRC64NVME": "I7vN1ByxFhk="
}
C:\Windows\System32>
C:\Windows\System32>aws s3api upload-part ^
More? --bucket myawsbuckkkketchetan ^
More? --key largefile.bin ^
More? --part-number 3 ^
More? --body "C:\Users\cheta\chunks\part-0002" ^
More? --upload-id _9.BgAnQiIGa0emCA3stX4lU6IA9hvyygDgBo3QrECLYWvsZ7pXsuTAOHcPAMdkHspq8Yn1lU3kJERECheI0hSVSii7vqPlUR0VJ6puBrk_d195c3RdvddkKYyfMhuCK ^
More? --region ap-south-1
{
"ServerSideEncryption": "AES256",
"ETag": "\"1f5039e50bd66b290c56684d8550c6c2\"",
"ChecksumCRC64NVME": "I7vN1ByxFhk="
}
C:\Windows\System32>
C:\Windows\System32>aws s3api upload-part ^
More? --bucket myawsbuckkkketchetan ^
80°F
Partly cloudy
Search
ENG
IN
22:37
31-07-2025


Administrator: Command Prompt
"ETag": "\"1f5039e50bd66b290c56684d8550c6c2\"",
"ChecksumCRC64NVME": "I7vN1ByxFhk="
}
C:\Windows\System32>
C:\Windows\System32>aws s3api upload-part ^
More? --bucket myawsbuckkkketchetan ^
More? --key largefile.bin ^
More? --part-number 3 ^
More? --body "C:\Users\cheta\chunks\part-0002" ^
More? --upload-id _9.BgAnQiIGa0emCA3stX4lU6IA9hvyygDgBo3QrECLYWvsZ7pXsuTAOHcPAMdkHspq8Yn1lU3kJERECheI0hSVSii7vqPlUR0VJ6puBrk_d195c3RdvddkKYyfMhuCK ^
More? --region ap-south-1
{
"ServerSideEncryption": "AES256",
"ETag": "\"1f5039e50bd66b290c56684d8550c6c2\"",
"ChecksumCRC64NVME": "I7vN1ByxFhk="
}
C:\Windows\System32>
C:\Windows\System32>aws s3api upload-part ^
More? --bucket myawsbuckkkketchetan ^
More? --key largefile.bin ^
More? --part-number 4 ^
More? --body "C:\Users\cheta\chunks\part-0003" ^
More? --upload-id _9.BgAnQiIGa0emCA3stX4lU6IA9hvyygDgBo3QrECLYWvsZ7pXsuTAOHcPAMdkHspq8Yn1lU3kJERECheI0hSVSii7vqPlUR0VJ6puBrk_d195c3RdvddkKYyfMhuCK ^
More? --region ap-south-1
{
"ServerSideEncryption": "AES256",
"ETag": "\"1f5039e50bd66b290c56684d8550c6c2\"",
"ChecksumCRC64NVME": "I7vN1ByxFhk="
}
C:\Windows\System32>
C:\Windows\System32>
80°F
Partly cloudy
Search
ENG
IN
22:38
31-07-2025




```
C:\Windows\System32>aws configure
AWS Access Key ID [****************PE6O]: AKIA4J5YR7AKB3UAPE6O
AWS Secret Access Key [****************hpmO]: 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
Default region name [ap-south-1]: ap-south-1
Default output format [None]:

C:\Windows\System32>fsutil file createnew C:\Users\cheta\largefile.bin 1073741824
File C:\Users\cheta\largefile.bin is created

C:\Windows\System32>aws s3api create-multipart-upload --bucket myawsbuckkkketchetan --key largefile.bin
{
    "ServerSideEncryption": "AES256",
    "Bucket": "myawsbuckkkketchetan",
    "Key": "largefile.bin",
    "UploadId": ".HK.2KNQDbiyQudA1qE99uy9vtmOZSBzrxo6afGV3NRUgeBkcZky_obv3lptstgz6fFKYwxC2F8BXIRyLtJtlivqvVStxLUVdj.AF7TtgKTkkPDi8PkWKibXaXjhcoxC"
}
```

```
:\Windows\System32>aws s3api create-multipart-upload ^
ore?   --bucket myawsbuckkkketchetan ^
ore?   --key largefile.bin ^
ore?   --region ap-south-1
{
    "ServerSideEncryption": "AES256",
    "Bucket": "myawsbuckkkketchetan",
    "Key": "largefile.bin",
    "UploadId": "_9.BgAnQiIGa0emCA3stX4lU6IA9hvyygDgBo3QrECLYWvsZ7pXsuTAOHcPAMdkHspq8Yn1lU3kJERECheI0hSVSii7vqPlUR0VJ6puBrk_d195c3RdvddkKYyfMhuCK"
}
```

Parts.json

C: > Users > cheta > {} Parts.json > ...

```json
{
  "Parts": [
    {
      "ETag": "\"1f5039e50bd66b290c56684d8550c6c2\"",
      "PartNumber": 1
    },
    {
      "ETag": "\"etag-part-2-here\"",
      "PartNumber": 2
    },
    {
      "ETag": "\"etag-part-3-here\"",
      "PartNumber": 3
    },
    {
      "ETag": "\"etag-part-4-here\"",
      "PartNumber": 4
    }
  ]
}
```

```
C:\Windows\System32>aws s3api complete-multipart-upload ^
More?   --bucket myawsbuckkkketchetan ^
More?   --key largefile.bin ^
More?   --upload-id _9.BgAnQiIGa0emCA3stX4lU6IA9hvyygDgBo3QrECLYWvsZ7pXsuTAOHcPAMdkHspq8Yn1lU3kJEREChe
hSVSii7vqPlUR0VJ6puBrk_d195c3RdvddkKYyfMhuCK ^
More?   --multipart-upload file://C:\Users\cheta\Parts.json ^   --region ap-south-1
{
    "ServerSideEncryption": "AES256",
    "Location": "https://myawsbuckkkketchetan.s3.ap-south-1.amazonaws.com/largefile.bin",
    "Bucket": "myawsbuckkkketchetan",
    "Key": "largefile.bin",
    "ETag": "\"fb45c1c8b5eab382b93bec76f28907f2-4\"",
    "ChecksumCRC64NVME": "LboFOsM6Fuk=",
    "ChecksumType": "FULL_OBJECT"
}
```

3. Bucket Naming Rules Validation

Objective: Learn valid/invalid S3 bucket naming.

Valid Bucket Names:

- `my-bucket123`
- `data-store-01`

Invalid Examples (and reasons):

- `My.Bucket` → Uppercase letters and dots are discouraged
- `192.168.1.1` → Looks like an IP address
- `my_bucket` → Underscores are not allowed

AWS CLI will return error messages like:

- `InvalidBucketName`
- `Bucket name should not be formatted as IP address`

Optional – Weekend Exploration Tasks

1. S3 Versioning and Override Behavior

Steps:

- Enable versioning:

```
sql
CopyEdit
aws s3api put-bucket-versioning --bucket your-bucket --versioning-confi
guration Status=Enabled
```

- Upload a file multiple times with the same key.
- List versions:

```
css
CopyEdit
```

```
aws s3api list-object-versions --bucket your-bucket
```

- Restore a previous version using `copy-object` with `-version-id`.

---

2. Static Website Hosting

Steps:

- Create and configure bucket:

```
arduino
CopyEdit
aws s3api put-bucket-website --bucket your-bucket --website-configurati
on file://config.json
```

Sample `config.json`:

```
json
CopyEdit
{
  "IndexDocument": { "Suffix": "index.html" }
}
```

- Upload `index.html`
- Make bucket public (IAM/ACL/policy)

---

3. Cost Estimation Exercise

Tools:

- AWS Pricing Calculator

Inputs:

- S3 Standard Storage: 1TB

- Operations: 10,000 PUT, 10,000 GET

Estimate monthly costs and document it

---

4. Cross-Region Access Testing

Steps:

- Create two buckets: one in `us-east-1`, one in `ap-south-1`
- Upload object to one bucket
- Test access from other region using:

```
bash
CopyEdit
aws s3 cp s3://bucket-name/object .
```

---

5. AWS CLI/SDK Integration

Using AWS CLI:

- Create bucket:

```
css
CopyEdit
aws s3api create-bucket --bucket my-cli-bucket --region ap-south-1
```

- Upload/download files:

```
bash
CopyEdit
aws s3 cp file.txt s3://my-cli-bucket/
aws s3 cp s3://my-cli-bucket/file.txt .
```

Using Boto3 (Python SDK):

- Automate operations in a Python script:

```
python
CopyEdit
import boto3
s3 = boto3.client('s3')
s3.create_bucket(Bucket='my-boto3-bucket')
s3.upload_file('file.txt', 'my-boto3-bucket', 'file.txt')
```