Uploading a video to an Amazon S3 bucket – how to:

**Using the AWS Management Console**

1. **Log in to AWS**:
   * Go to the [AWS Management Console](https://aws.amazon.com/console/).
2. **Navigate to S3**:
   * Open the S3 service from the console.
3. **Open Your Bucket**:
   * Select the bucket where you want to upload the video.
4. **Upload the Video**:
   * Click the **Upload** button.
   * Drag and drop the video file or use the file picker to select it from your system.
5. **Set Permissions** (Optional):
   * You can set access permissions for the file (e.g., public or private) during the upload process.
6. **Complete the Upload**:
   * Click **Upload** to start the process.

**Using Boto3 (Python SDK)**

Here’s how to upload a video programmatically using Boto3 in Python:

import boto3

from botocore.exceptions import NoCredentialsError

# Initialize the S3 client

s3 = boto3.client(

's3',

aws\_access\_key\_id='YOUR\_AWS\_ACCESS\_KEY',

aws\_secret\_access\_key='YOUR\_AWS\_SECRET\_KEY',

region\_name='YOUR\_AWS\_REGION'

)

def upload\_to\_s3(file\_name, bucket\_name, object\_name=None):

if object\_name is None:

object\_name = file\_name # Default to file name

try:

s3.upload\_file(file\_name, bucket\_name, object\_name)

print(f"File {file\_name} uploaded to {bucket\_name}/{object\_name}")

except FileNotFoundError:

print(f"The file {file\_name} was not found.")

except NoCredentialsError:

print("Credentials not available.")

# Example usage

upload\_to\_s3("path/to/video.mp4", "your-s3-bucket-name")

**Using AWS CLI**

1. **Install and Configure AWS CLI**:
   * Install the AWS CLI: [Installation Guide](https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2.html).
   * Configure it using aws configure and provide your AWS credentials and default region.
2. **Upload the Video**:
3. aws s3 cp /path/to/video.mp4 s3://your-s3-bucket-name/

**Important Considerations**

1. **Bucket Permissions**:
   * Ensure your bucket has the correct permissions to allow uploads. Update the bucket policy or use IAM roles for access control.
2. **Large Files**:
   * For large video files (greater than 5 GB), use **multipart upload** in the AWS Management Console, CLI, or SDK to handle the upload efficiently.
3. **Public Access**:
   * If you want the video to be accessible publicly, set its object ACL to public-read during upload or update its permissions later.
4. **Cost Implications**:
   * Be aware of S3 costs, including storage and data transfer fees. Check the [AWS S3 Pricing](https://aws.amazon.com/s3/pricing/) page.

**Example: Multipart Upload with Boto3 (For Large Files)**

import boto3

# Initialize S3 client

s3 = boto3.client('s3')

bucket\_name = "your-s3-bucket-name"

key\_name = "large-video.mp4"

file\_path = "path/to/large-video.mp4"

# Initiate multipart upload

response = s3.create\_multipart\_upload(Bucket=bucket\_name, Key=key\_name)

upload\_id = response['UploadId']

# Read and upload parts

parts = []

chunk\_size = 5 \* 1024 \* 1024 # 5 MB chunks

with open(file\_path, 'rb') as file:

for i, chunk in enumerate(iter(lambda: file.read(chunk\_size), b"")):

part = s3.upload\_part(

Bucket=bucket\_name,

Key=key\_name,

PartNumber=i + 1,

UploadId=upload\_id,

Body=chunk

)

parts.append({"PartNumber": i + 1, "ETag": part["ETag"]})

# Complete the multipart upload

s3.complete\_multipart\_upload(

Bucket=bucket\_name,

Key=key\_name,

UploadId=upload\_id,

MultipartUpload={"Parts": parts}

)

print(f"File {key\_name} uploaded successfully!")

This ensures efficient and robust uploading of videos to S3.