WHIZLABS

Home / AWS / Guided Lab / AWS S3 Multipart Upload using AWS CLI

# AWS S3 Multipart Upload using AWS CLI

Level: **Advanced**

Amazon EC2 | Amazon S3 | AWS CLI | Amazon Web Services

0h 29m 18s left

15 MIN

End Lab

Open Console

Validation

## Lab Credentials

**User Name**

Whiz_User_80425.81811295

**Password**

d6292e13-2a1b-4299-a437-16557ea44134

**Access Key**

AKIAVVWNTXJ6RXJT4YSQ

**Secret Key**

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

## Lab Resources

No Lab Resources Found

## Support Documents

1. FAQs and Troubleshooting

Need help?

How to use Hands on Lab

Troubleshooting Lab

FAQs

Submit Feedback | Share

Lab Overview | Lab Steps | Lab Validation | Lab FAQs

Cloud Developer, Cloud Administrator

Storage, Developer Tools

# Lab Steps

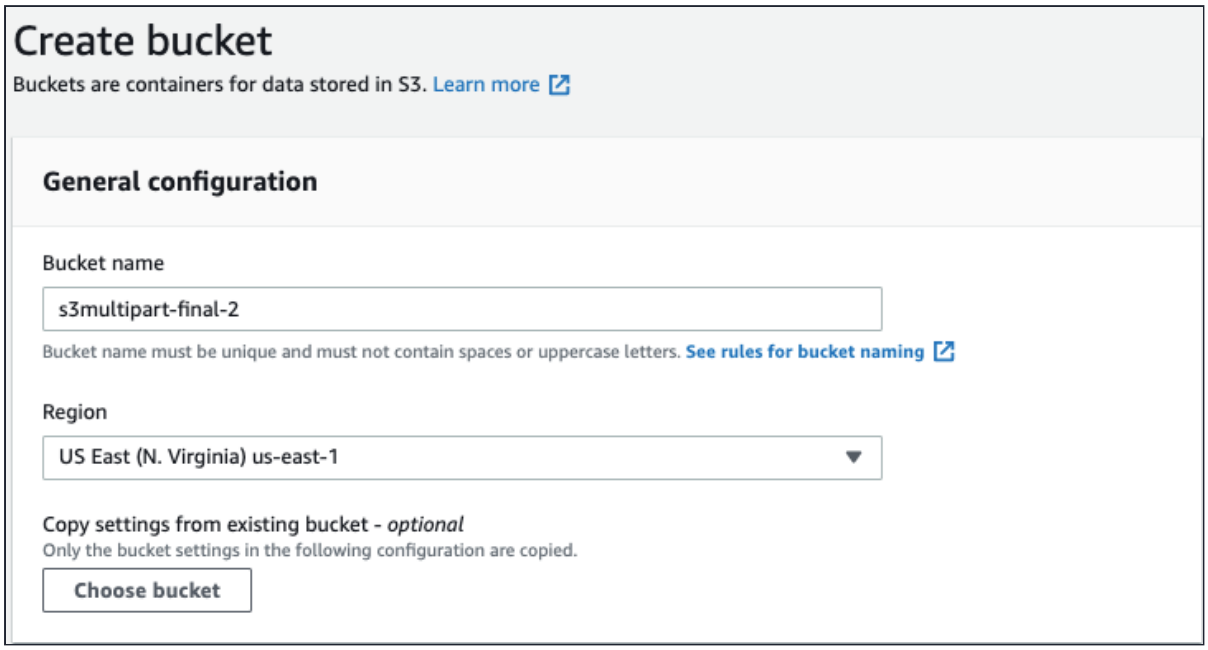
## Task 1: Sign in to AWS Management Console

1. Click on the **Open Console** button, and you will get redirected to AWS Console in a new browser tab.
2. On the AWS sign-in page,
    - Leave the Account ID as default. Never edit/remove the 12 digit Account ID present in the AWS Console. otherwise, you cannot proceed with the lab.
    - Now copy your **User Name** and **Password** in the Lab Console to the **IAM Username and Password** in AWS Console and click on the **Sign in** button.
3. Once Signed In to the AWS Management Console, Make the default AWS Region as **US East (N. Virginia) us-east-1.**

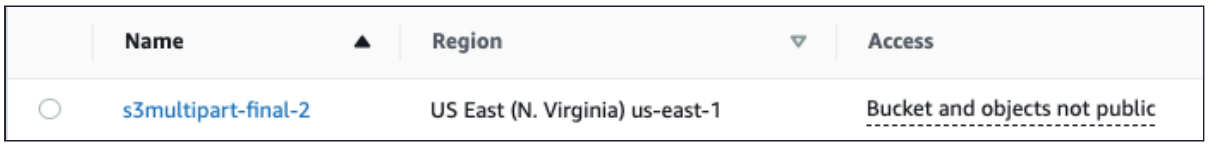
## Task 2: Create an S3 Bucket

1. Navigate to the **Services** menu at the top. Click on **S3** in the **Storage** section.
2. In the **S3** dashboard, click on the **Create bucket** button and fill in the bucket details.
3. In the General Configuration,

- Bucket name: Enter ***s3multipart-final-2***
- **Note: S3 Bucket names are globally unique, choose a name that is available. Maybe you can enter your name and create one.**
- Region: Select **US East (N. Virginia) us-east-1**




4. Object ownership: Select **ACLs disabled (recommended)** option
5. Leave all other settings as default and click on **Create bucket** button.

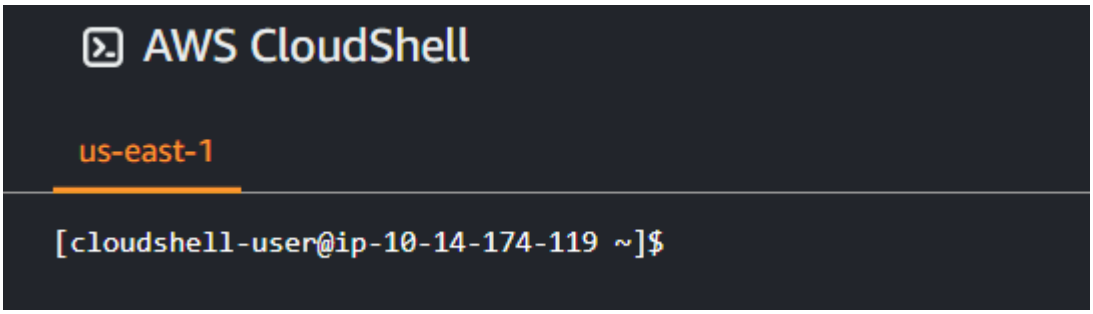



6. The S3 bucket has now been created.

## Task 3: Create an Environment in CloudShell

1. Click on Cloud Shell icon on the top right corner of AWS menu bar.


2. A new tab in your browser opens and if you see a welcome message to cloud shell, then click on the Close button in that message.
3. You will see a creating environment message on the screen.
4. Wait for a few minutes to complete the environment creation. Once the environment is created, You are ready to use the terminal.



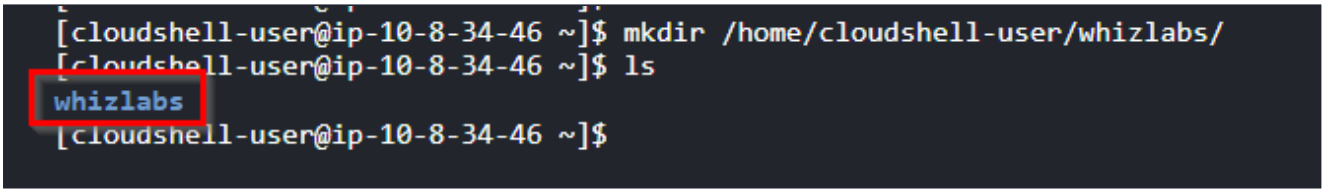

## Task 4: Copy a video file from S3 bucket

1. Under the user data section, enter the following script to copy a video file from the S3 bucket to CloudShell
2. Enter below command to make directory

```
mkdir /home/cloudshell-user/whizlabs/
```

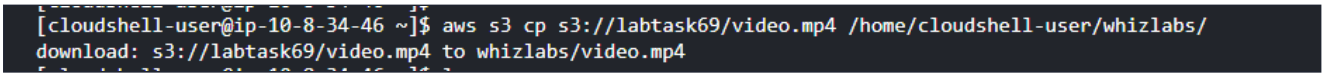
3. Enter below command to view the directory created

```
ls
```




4. Enter below command to Copy a video file from the s3 bucket

```
aws s3 cp s3://labtask69/video.mp4 /home/cloudshell-user/whizlabs/
```




## Task 5: View the Original File

Here, we are going to perform the S3 multipart upload of a video file stored in an Cloudshell. We will upload it to the S3 bucket we created in the above step.

1. Change directory to **whizlabs**

```
cd whizlabs/
```

2. View the property detail of the video file.

```
ls -l
```

```
[cloudshell-user@ip-10-8-34-46 ~]$ cd whizlabs/
[cloudshell-user@ip-10-8-34-46 whizlabs]$ ls -l
total 145800
-rw-rw-r-- 1 cloudshell-user cloudshell-user 149295511 Jun 11  2020 video.mp4
[cloudshell-user@ip-10-8-34-46 whizlabs]$
```

**Note:** This file is 143 MB in size, so we'll use the multipart feature to upload this file to s3.

## Task 6: Split the Original File

1. **Split the file into chunks**

```
split -b 40M video.mp4
```

- The split command will split a large file into many pieces (chunks) based on the option.
- **split [options] [filename]**
- Here we are dividing the 143 MB file into 40MB chunks. [ -b option means Bytes ]

2. **View the chunked files**

```
ls -lh
```

```
[cloudshell-user@ip-10-8-34-46 whizlabs]$ split -b 40M video.mp4
[cloudshell-user@ip-10-8-34-46 whizlabs]$ ls -lh
total 285M
-rw-rw-r-- 1 cloudshell-user cloudshell-user 143M Jun 11  2020 video.mp4
-rw-rw-r-- 1 cloudshell-user cloudshell-user  40M Feb  9 18:18 xaa
-rw-rw-r-- 1 cloudshell-user cloudshell-user  40M Feb  9 18:18 xab
-rw-rw-r-- 1 cloudshell-user cloudshell-user  40M Feb  9 18:18 xac
-rw-rw-r-- 1 cloudshell-user cloudshell-user  23M Feb  9 18:18 xad
[cloudshell-user@ip-10-8-34-46 whizlabs]$
```

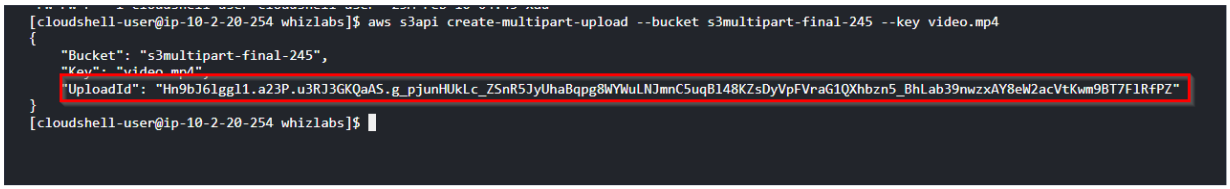
**Info:** Here "xaa" and "xad" are the chunked files that have been renamed alphabetically. Each file is 40MB in size but except the last one. The number of chunks depends on the size of your original file and the byte value used to partition the chunks.

## Task 7: Create a Multipart Upload

We are initiating the multipart upload using an **AWS CLI** command, which will generate a UploadID that will be used later.

- **Note: Replace the example bucket name below with your bucket name.**
- **Example:** aws s3api create-multipart-upload --bucket s3multipart-final --key video.mp4
- Syntax:-

```
aws s3api create-multipart-upload --bucket [Bucket name] --key
 [original file name]
```




- **Note: Please copy the UploadId and save it in the editor for later use.**

## Task 8: Uploading the File Chunks

1. Next, we need to upload each file chunk one by one, using the part number. The part number is assigned based on the alphabetic order of the file.

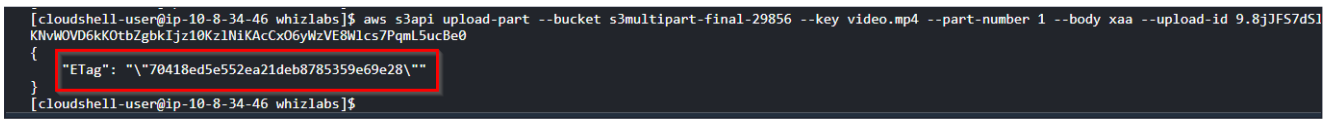
| Chunk File Name | Part Number |
|---|---|
| xaa | 1 |
| xab | 2 |
| xac | 3 |
| xad | 4 |

- **Syntax:**

```
aws s3api upload-part --bucket [bucketname] --key
[filename] --part-number [number] --body [chunk file name] -
-upload-id [id]
```

- **Example:** aws s3api upload-part --bucket s3multipart-final --key video.mp4 --part-number 1 --body xaa --upload-id 97pcMF8E31iT6spF8_AoIDVHESi0kJlj.G8oM1.jbgYWTs1KjazpK.yVt2akv3NoqfvnDc8TO9e6OikpdSEyEJblDOe.8yOx3q.suF7SlLcwjnlyfjXqVif3CAj.xgLL3jDRdB9PFTEmGr5KUog2SA-

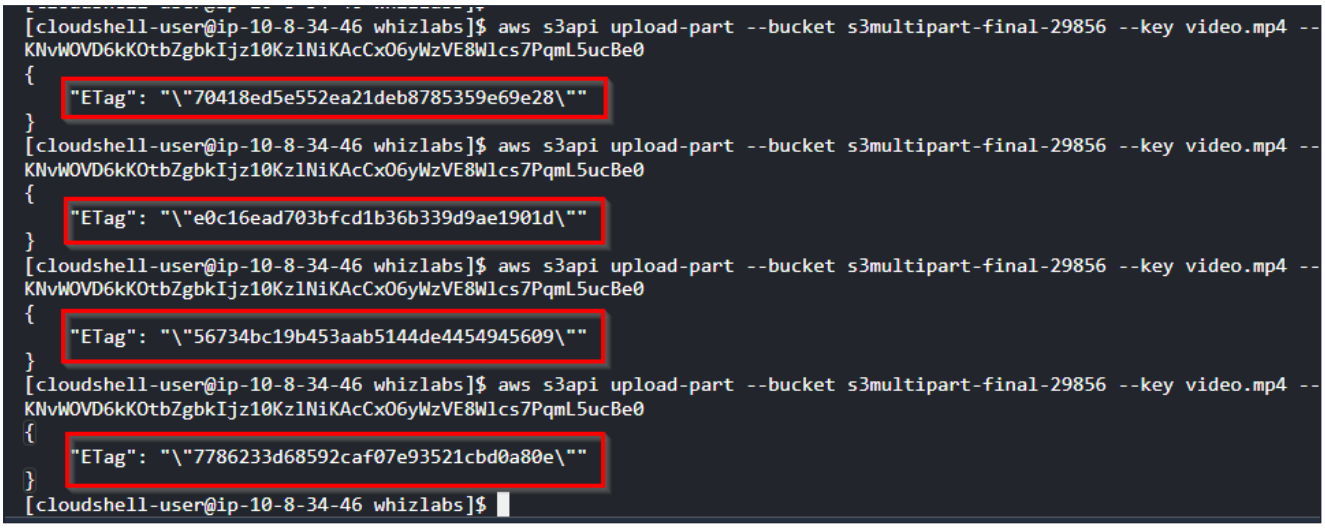
- **Note: Please replace the upload id with your upload id. Copy the ETag id and Part number for later use.**




2. **Repeat** the above CLI command for each file chunk [**Replace --part-number & --body values with the above table values**]

3. Press the **UP Arrow Key** to get back to the previous command. No need to enter the Upload ID again, just change the Part Number and Body Value.

4. Each time you upload a chunk, don't forget to **save** the **Etag** value.




## Task 9: Create a Multipart JSON file

Create a file with all part numbers with their **Etag** values.

1. Creating a file named **list.json**

```
nano list.json
```

2. Copy the below JSON Script and paste it in the **list.json** file.

**Note: Replace the ETag ID according to the part number, which you received after uploading each chunk.**

```
{
    "Parts":[
        {
            "PartNumber":1,
            "ETag": "\"70418ed5e552ea21deb8785359e69e28\""
        },
        {
            "PartNumber":2,
            "ETag": "\"e0c16ead703bfcd1b36b339d9ae1901d\""
        },
        {
            "PartNumber":3,
            "ETag": "\"56734bc19b453aab5144de4454945609\""
        },
        {
            "PartNumber":4,
            "ETag": "\"7786233d68592caf07e93521cbd0a80e\""
        }
```

```
    ]
}
```

3. Save the file

    - Press **Ctrl + X** and type **Y** then press **Enter** to save the file.


## Task 10: Complete the Multipart Upload

Now we are going to join all the chunks together with the help of the JSON file we created in the above step.

1. **Syntax:**

```
aws s3api complete-multipart-upload --multipart-upload [json
file link]  --bucket [upload bucket name] --key [original
file name] --upload-id [upload id]
```

2. **Example:** aws s3api complete-multipart-upload --multipart-upload file://list.json --bucket s3multipart-final --key video.mp4 --upload-id 97pcMF8E31iT6spF8_AoIDVHESi0kJlj.G8oM1.jbgYWTs1KjazpK.yVt2akv3NoqfvnDc8TO9e6OikpdSEyEJbIDOe.8yOx3q.suF7SlLcwjnIyfjXqVif3CAj.xgLL3jDRdB9PFTEmGr5KUog2SA--

3. **Note:**

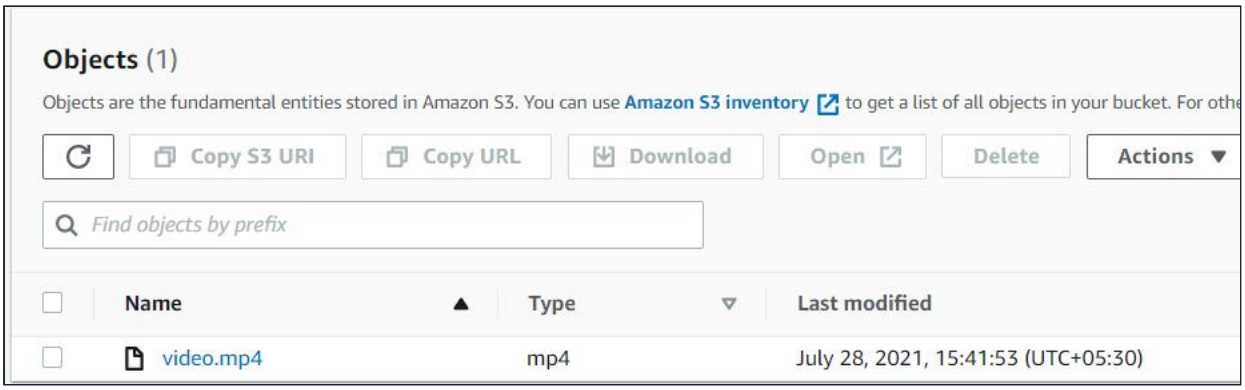
    - **Replace the example above with your bucket name.**
    - **Replace the Upload-Id value with your upload id.**

```
{
    "ETag": "\"c4bd0807a6535303d596c253cb69cd11-4\"",
    "Bucket": "s3multipart-final-3",
    "Location": "https://s3multipart-final-3.s3.amazonaws.com/video.mp4",
    "Key": "video.mp4"
}
```

### Task 11: View the File in the S3 Bucket

1. Make sure you are in the **N.Virginia** Region.
2. Navigate to the **Services** menu at the top. Click on **S3** in the **Storage** section.
3. On the S3 dashboard, click on the bucket name **s3multipart-final-2**

   **Note:** Choose the bucket name you created in the beginning if s3multipart-final was not available.
4. Inside the bucket you can see the **video.mp4** object created.

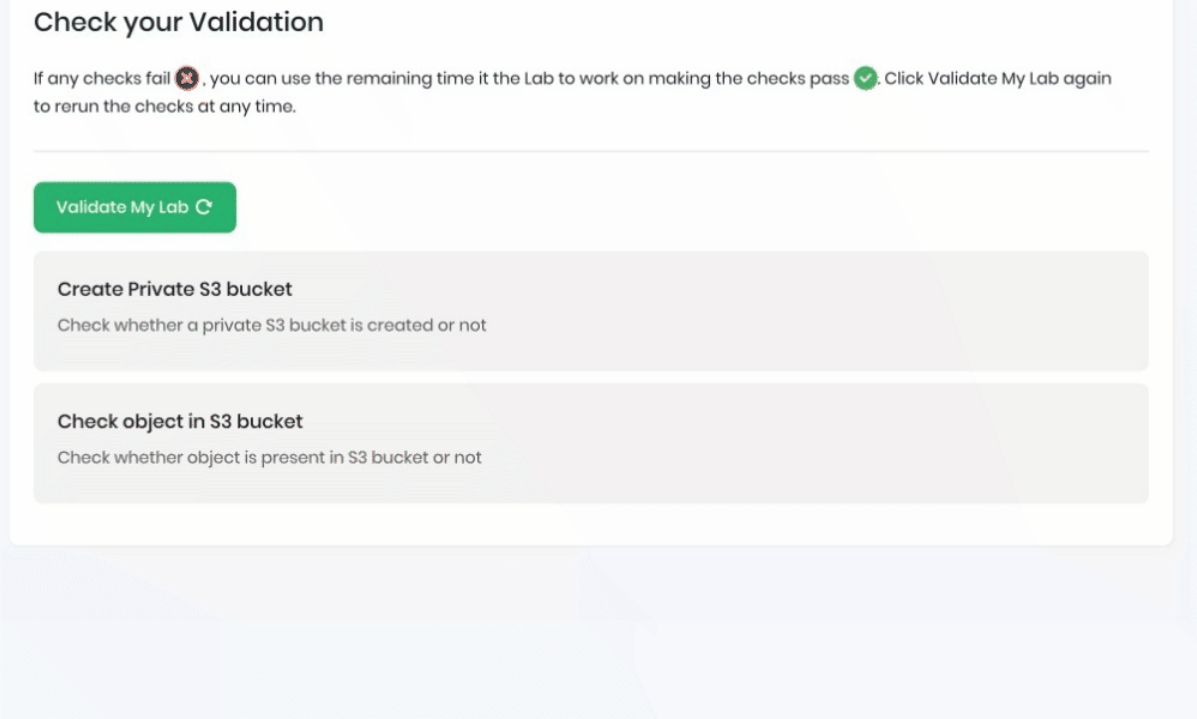



## Do you know ?

The multipart upload feature in S3 also provides event notifications, which allow you to trigger actions or workflows when specific events occur during the upload process. For example, you can configure S3 to send a notification to an Amazon Simple Notification Service (SNS) topic or an AWS Lambda function when a multipart upload is completed or fails.

### Task 12: Validation Test

1. Once the lab steps are completed, please click on the **Validation** button on the left side panel.
2. This will validate the resources in the AWS account and displays whether you have completed this lab successfully or not.
3. Sample output :




# Completion and Conclusion

1. You have successfully created an S3 bucket.
2. You have successfully created an Environment in Cloudshell and copied a file from S3 to Cloudshell.
3. You have successfully split a file into multiple parts and used the parts to perform a multipart upload via AWS CLI.
4. You have successfully validated the lab.

# End Lab

1. Sign out of AWS Account.
2. You have successfully completed the lab.
3. Once you have completed the steps click on **End Lab** from your Whizlabs dashboard.