

# Sandbox Tutorial: Simulating Hybrid Storage and Data Migration to Amazon S3

To migrate files from an on-premises Linux system (UTM VM) to Amazon S3 in AWS Sandbox, demonstrating cloud-based file storage, versioning, and data transfer workflows. This lab simulates a hybrid storage scenario without using AWS Storage Gateway, and focuses on uploading and managing files in S3.

## 1. Create the S3 Buckets

One bucket receives files (source), the other receives replicated files (destination). Both must have **Versioning enabled**.

### US-East-1

The screenshot displays the 'Create bucket' page in the AWS Management Console for the US-East-1 region. The breadcrumb navigation at the top shows 'Amazon S3 > Buckets > Create bucket'. The main heading is 'Create bucket' with an 'info' link. Below this, a note states 'Buckets are containers for data stored in S3.' The 'General configuration' section includes the 'AWS Region' set to 'US East (N. Virginia) us-east-1'. Under 'Bucket type', 'General purpose' is selected with a radio button, accompanied by a description: '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.' The 'Directory' option is also visible but unselected. The 'Bucket name' field contains 'sandbox-source-karim', with a note below stating that names must be 3 to 63 characters and unique. A 'Copy settings from existing bucket - optional' section includes a 'Choose bucket' button and a format example 's3://bucket/prefix'. At the bottom, the 'Bucket Versioning' section shows 'Enable' selected with a radio button, with a description of versioning and a 'Learn more' link.

## 2. Prepare Your Linux VM (UTM)

Connect to your **UTM Linux instance**.

### Update packages:

```
sudo apt update && sudo apt -y upgrade
```

### Install AWS CLI (if not installed):

Since im running aarch64

Downloading the aarch64 AWS cli and installing it

A terminal window with a title bar showing three colored circles (red, yellow, green) and a folder icon, followed by the text 'karim — karim@karim-vm: ~'. The terminal content shows a green prompt 'karim@karim-vm:~\$' followed by the command 'uname -m', the output 'aarch64', and another green prompt 'karim@karim-vm:~\$' with a grey cursor block.

```
karim — karim@karim-vm: ~  
[karim@karim-vm:~$ uname -m  
aarch64  
karim@karim-vm:~$ █
```

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-aarch64.zip" -o "awscliv2.zip"
```

```
unzip awscliv2.zip
```

```
sudo ./aws/install
```

```
aws --version
```

```
karim@karim-vm:~$ aws --version
aws-cli/2.28.21 Python/3.13.7 Linux/6.8.0-79-generic exe/aarch64.ubuntu.24
karim@karim-vm:~$ █
```

## Configuring the AWS

```
[karim@karim-vm:~$ aws configure
[AWS Access Key ID [None]: AKIA2TTGXG5HLNXWR5EM
[AWS Secret Access Key [None]: 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
[Default region name [None]: us-east-1
[Default output format [None]: json]█
```

## Verify Connection

```
[karim@karim-vm:~$ aws s3 ls
2025-08-31 09:48:37 sandbox-source-karim
karim@karim-vm:~$ █
```

## Create a Local Folder with Test Files

```
mkdir ~/s3test
cd ~/s3test
echo "Hello AWS Sandbox" > file1.txt
echo "Another test file" > file2.txt
```

```

karim@karim-vm:~$ mkdir ~/s3test
karim@karim-vm:~$ ls
academy-regular.ovpn  backupDir      host.pcap      nmap_res.txt   sample.txt
aws                  cryptography   http.cap       output.txt     sqlProject
awsliv2.zip          firstDir       myfile.txt     project        testDir
backup.log           gitprojects   myscript.sh    s3test
karim@karim-vm:~$ aws s3 ls
2025-08-31 09:48:37 sandbox-source-karim
karim@karim-vm:~$ cd ~/s3test
karim@karim-vm:~/s3test$ echo "Hello AWS Sandbox" > file1.txt
karim@karim-vm:~/s3test$ echo "Another test file" > file2.txt
karim@karim-vm:~/s3test$
karim@karim-vm:~/s3test$ cat file1.txt
Hello AWS Sandbox
karim@karim-vm:~/s3test$ cat file2.txt
Another test file
karim@karim-vm:~/s3test$ █

```

## Upload Files to S3

- `cp` → copy
- `-recursive` → uploads everything inside the folder

```
aws s3 cp ~/s3test s3://<bucket-name>/ --recursive
```

```

[karim@karim-vm:~/s3test$ aws s3 cp ~/s3test s3://sandbox-source-karim/ --recursive]
ve
upload: ./file1.txt to s3://sandbox-source-karim/file1.txt
upload: ./file2.txt to s3://sandbox-source-karim/file2.txt
[karim@karim-vm:~/s3test$ █

```

## Verify Upload

```
aws s3 ls s3://<bucket-name>/
```

```

[karim@karim-vm:~/s3test$ aws s3 ls s3://sandbox-source-karim/
2025-08-31 09:58:31      18 file1.txt
2025-08-31 09:58:31      18 file2.txt
[karim@karim-vm:~/s3test$

```

Now our UTM Linux VM is connected to AWS Sandbox and successfully sending files to S3 🚀