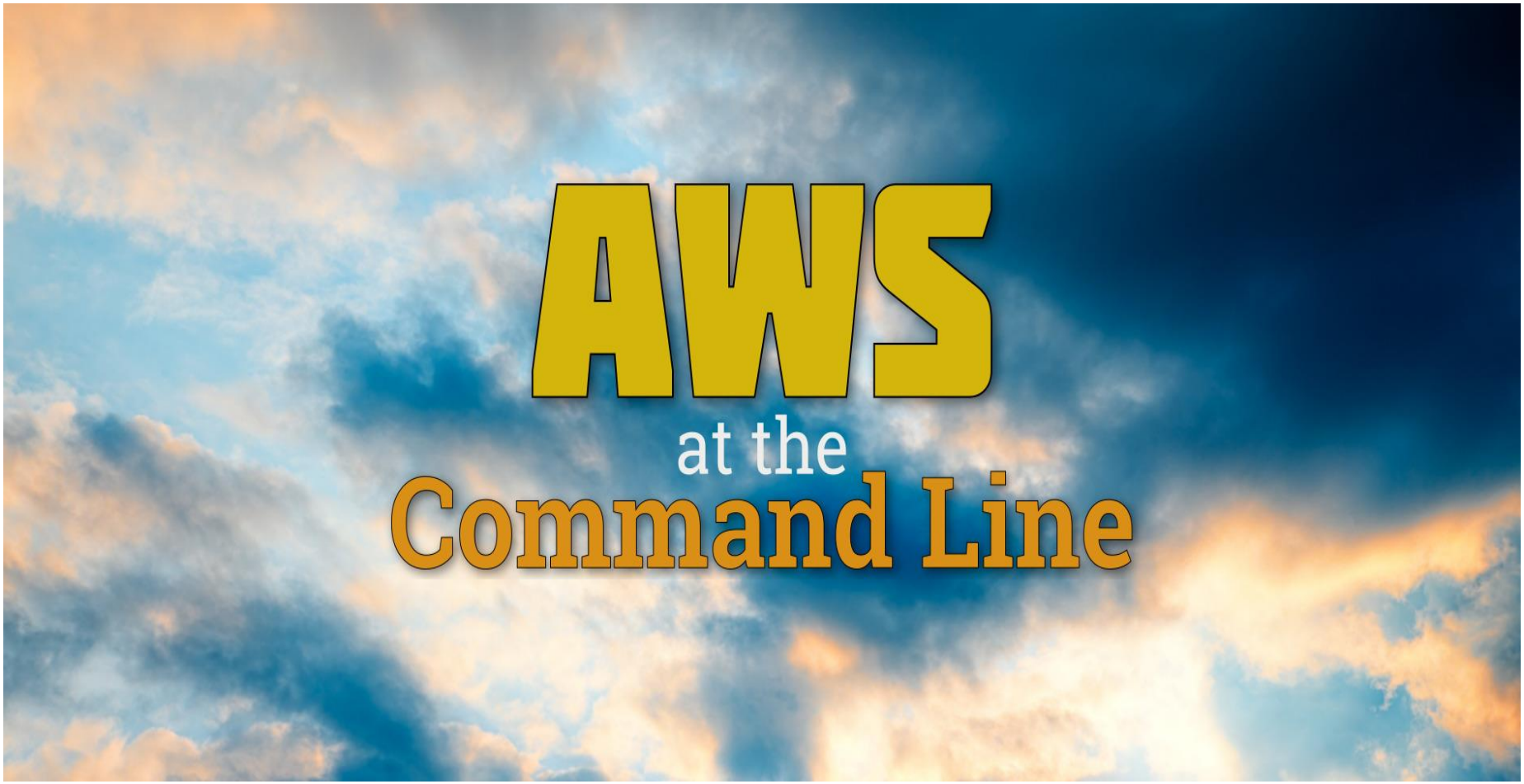

AWS
at the
Command Line

Topics to be covered—AWS CLI

1) AWS CLI Setup Introduction

2) CLI setup AWS Instance – IAM user credential or IAM role

3) CLI setup Local server – IAM user credential or IAM role

4) S3 operation with CLI

5) EC2 instance operation with CLI

6) High Availability Lab

AWS CLI

The AWS Command Line Interface (AWS CLI) is an open source tool that enables you to interact with AWS services using commands in your command-line shell. With minimal configuration, the AWS CLI enables you to start running commands that implement functionality equivalent to that provided by the browser-based AWS Management Console from the command prompt in your terminal program.

AWS CLI versions

- Version 2.x
- Version 1.x

How to install AWS CLI

1) Amazon linux 2 AMI – already installed
2) Other OS -- download from AWS documentation link

AWS CLI Setup

1) Create an IAM user and give administrator access.

2) Open it –security credential –create access key—download access key –note down access and secret key.

3) Install aws cli in windows and linux based system

4) Type command

$ aws configure

Access key:

Secret access key:

Default region: ap-south-1

Output format: JSON

AWS CLI –with S3 Bucket

To create a new bucket

```
aws  s3  mb   s3://awsbatch100
```

To upload the file to s3 bucket

```
aws   s3 cp  “C:\data1\file1.txt”  s3://awsbatch100/
```

To download file from s3 bucket

```
aws   s3  cp s3://awsbatch100/file1.txt        ./
```

To delete s3 bucket file

```
aws   s3  rm  s3://awsbatch100/file1.txt
```

AWS CLI –with S3 Bucket

To Sync the local directory with s3 bucket and vice versa

```
aws   s3 sync   c:\lab22     s3://indiabucket/lab2
aws  s3  sync    s3://indiabucket/lab2      c:\lab22
aws    s3   sync .     s3:// indiabucket/lab2    --acl public-read
```

To see s3 bucket content

```
aws  s3  ls    s3://indiabucket/lab2
```

To delete s3 bucket

```
aws s3 rb s3://bucket-name --force
```

AWS CLI –EC2

For Amazon EC2 instance

aws ec2 stop-instances --instance-ids i-1234567890abcdef0

aws ec2 start-instances --instance-ids i-1234567890abcdef0

aws ec2 terminate-instances --instance-ids i-1234567890abcdef0

aws ec2 reboot-instances --instance-ids i-1234567890abcdef0

AWS CLI—EC2

To modify the instance type

```
aws ec2 modify-instance-attribute  \
   --instance-id  i-1234567890abcdef0  \
   --instance-type  "{\"Value\":   \"m1.small\"}“
```

To modify the sourceDestCheck attribute

```
aws ec2 modify-instance-attribute --instance-id  i-1234567890abcdef0 --source-dest-check "{\"Value\":  true}"
```

AWS CLI—EC2

To modify the user data attached to an instance

Contents of original file UserData.txt:

```
#!/bin/bash
yum install vsftpd -y
systemctl start  vsftpd
systemctl enable vsftpd
```

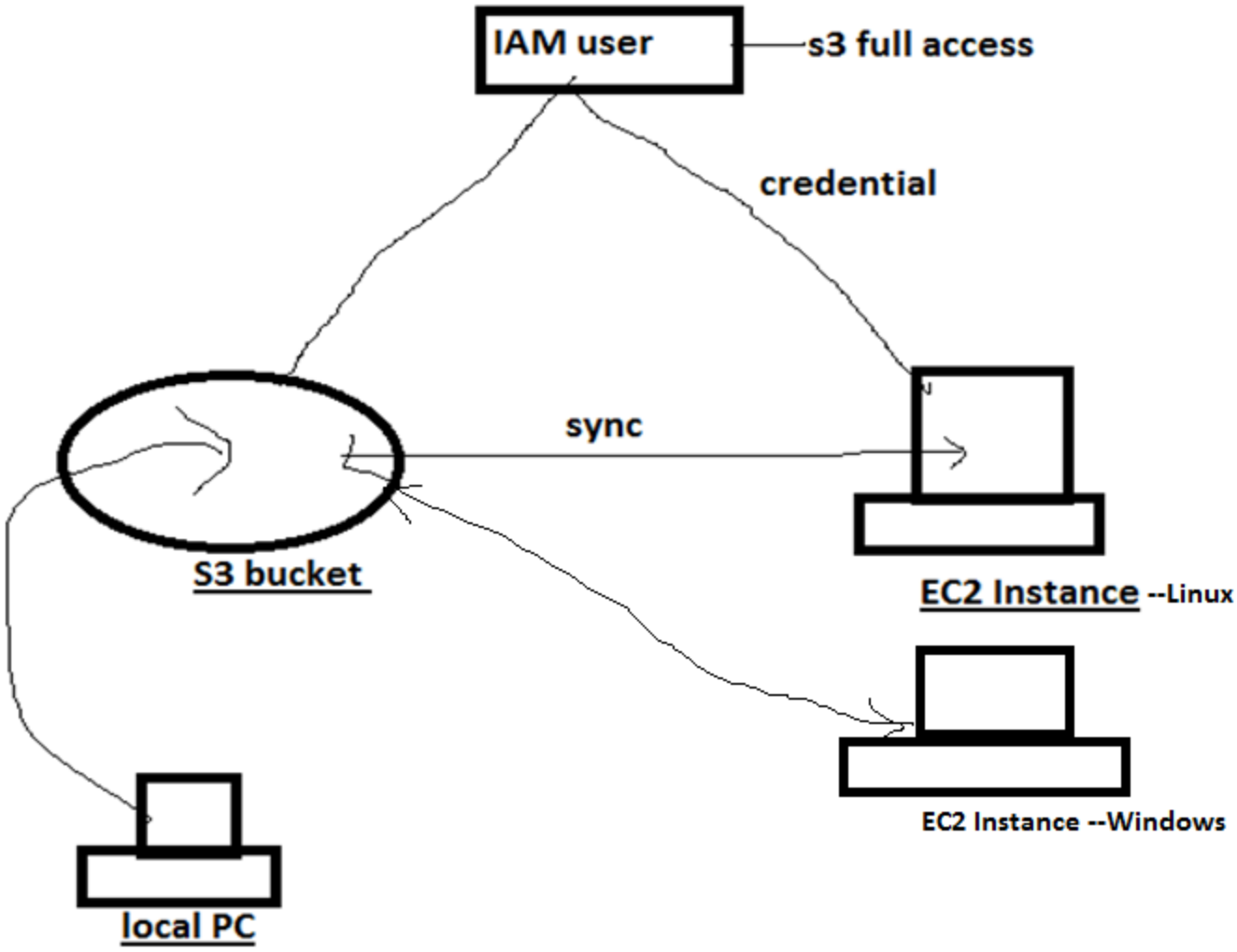
The contents of the file must be base64 encoded. The first command converts the text file to base64 and saves it as a new file.

```
base64 UserData.txt > UserData.base64.txt
```

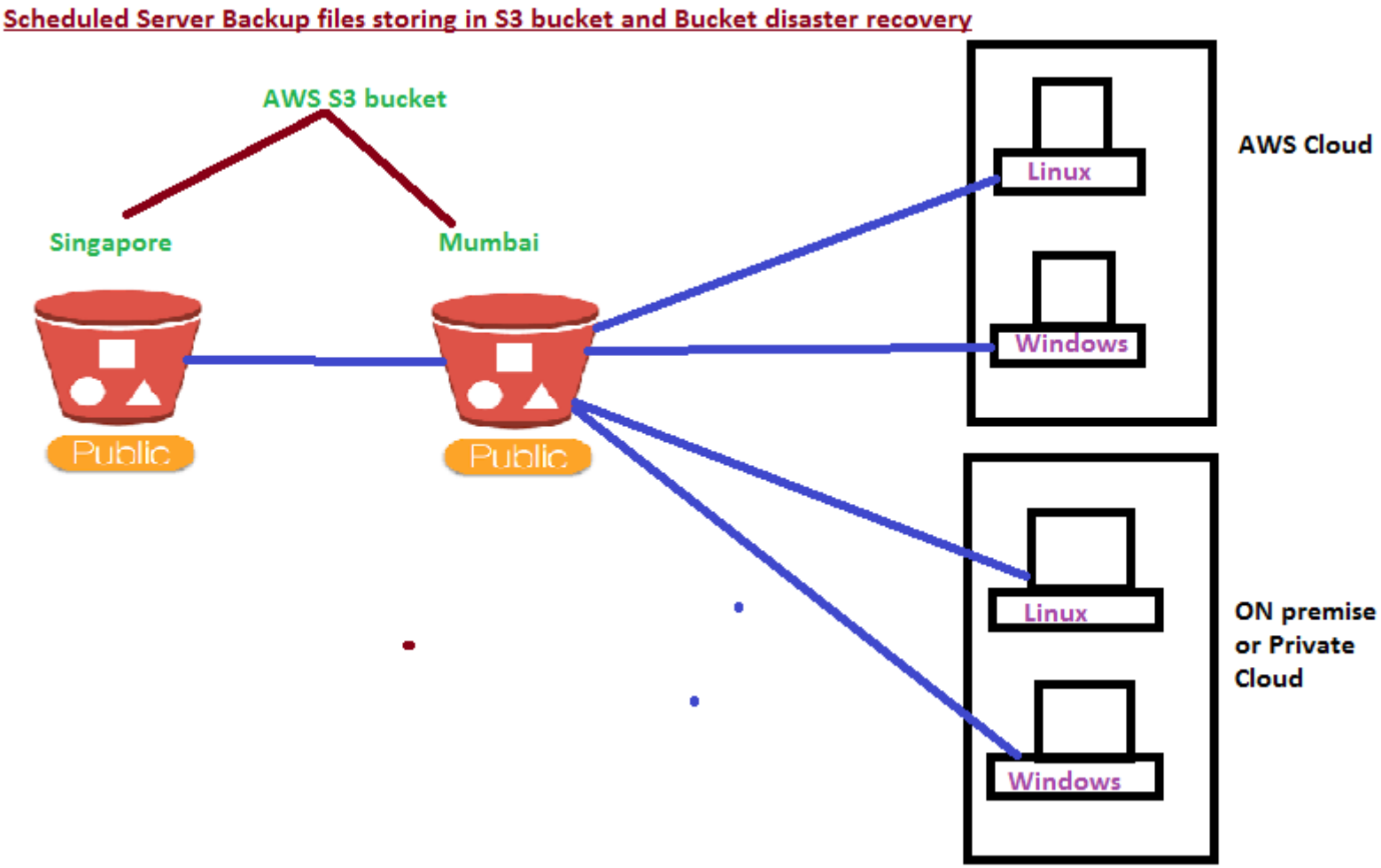
CLI command

```
aws ec2 modify-instance-attribute \
   --instance-id=i-09b5a14dbca622e76 \
   --attribute userData --value  file://UserData.base64.txt
```

Sync S3 bucket with EC2 instance



Sync Instances with S3 bucket (Use case 3)



Sync S3 bucket with EC2 instance (Linux) --Steps

1) Open IAM –create an user and assign Amazons3fullaccess policy there.

2) Open this user –Security credential –create access key –download access key.csv file –open it with excel and note down access key and secret access key.

3) Open S3 and create one bucket (indiabucket) –open the bucket—create 3 folder there(lab1, lab2, lab3)

4)Launch Amazon linux2 AMI ---Open it – type command “aws configure” and give the data.

Access key:

Secret access key:

Default region: ap-south-1

Output format: JSON

$ mkdir lab11

$ cd lab11

$ touch file1 file2 file2

$ aws s3 sync /home/ec2-user/lab11 s3://indiabucket/lab1

Now check the data in S3 bucket

Upload some files in S3 bucket—Come to EC2 instance

$ aws s3 sync s3://indiabucket/lab1 /home/ec2-user/lab11

$ ls

Sync S3 bucket with EC2 instance (Windows) --Steps

1) Open IAM –create an user and assign Amazons3fullaccess policy there.

2) Open this user –Security credential –create access key –download access key.csv file –open it with excel and note down access key and secret access key.

3) Open S3 and create one bucket (indiabucket) –open the bucket—create 3 folder there(lab1, lab2, lab3)

4)Launch Windows instance ---Open it – Download and install “Amazon CLI “

Open cmd– type command “aws configure” and give the data.

Access key:

Secret access key:

Default region: ap-south-1

Output format: JSON

C:\ mkdir lab22

C:\ cd lab22

C:\ touch file1 file2 file2

C:\ aws s3 sync c:\lab22 s3://indiabucket/lab2

Now check the data in S3 bucket

Upload some files in S3 bucket—Come to EC2 instance

C:\ aws s3 sync s3://indiabucket/lab2 c:\lab22

C:\ ls

How to Schedule the File sync (Auto Backup)(Windows)

Windows

1) Create one batch file and write the command there

aws s3 sync c:\awsdata2\ s3://awsbatch100/windows1

Save file--- file name: s3sync.bat, save as type: all files

2) Open task Scheduler – create basic task—name:test1 –next—select daily-next-set time –next—start a program– browse and select created batch file –ok—finish.

3) Now create some files in c:\awsdata2\ folder and wait for that scheduled time –then check the output in s3 bucket.

How to Schedule the File sync (Auto Backup)(Linux)

Linux

1) Create one shell script file and write the command there

$ pwd

/home/ec2-user

$nano test1.sh

aws s3 sync /home/ec2-user/linux11 s3://awsbatch100/linux1

Save and exit

2) Chmod u+x test1.sh

3) crontab –e

- * * * * sh /home/ec2-user/test1.sh

3) Now create some files in /home/ec2-user/linux11 folder and wait for that scheduled time –then check the output in s3 bucket.

Note: the given time is to run the script in every one minue