

1. msiexec.exe /i <https://awscli.amazonaws.com/AWSCLIV2.msi>

```
Administrator: Windows PowerShell
PS C:\> msiexec.exe /i https://awscli.amazonaws.com/AWSCLIV2.msi
```

2. \$env:Path = [System.Environment]::GetEnvironmentVariable("Path","Machine") + ";" +
[System.Environment]::GetEnvironmentVariable("Path","User")

```
PS C:\> $env:Path = [System.Environment]::GetEnvironmentVariable("Path","Machine") + ";" + [System.Environment]::GetEnvironmentVariable("Path","User")
PS C:\> Test-Path "C:\Program Files\Amazon\AWSCLIV2\aws.exe"
```

3. Test-Path "C:\Program Files\Amazon\AWSCLIV2\aws.exe"

```
PS C:\> Test-Path "C:\Program Files\Amazon\AWSCLIV2\aws.exe"
True
PS C:\>
```

4. aws --version

```
PS C:\> aws --version
aws-cli/2.33.27 Python/3.13.11 Windows/11 exe/AMD64
PS C:\>
PS C:\> aws configure
AWS Access Key ID [None]: AKIA33MSVLE3M5B6F5ND
AWS Secret Access Key [None]: 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
Default region name [None]: ap-south-1
Default output format [None]:
PS C:\>
```

5. cd E:\

6. cd .\SQLBackups\

```
PS E:\SQLBackups> dir

Directory: E:\SQLBackups

Mode                LastWriteTime         Length Name
----                -              -          -
-a--- 22-02-2026 11.33 AM        2009 BackupToS3.ps1
-a--- 22-02-2026 11.22 AM            0 New Text Document.txt
```

Access key:- AKIA33MSVLE3M5B6F5ND

Secret access key:- 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

AKIA33MSVLE3M5B6F5ND		Actions ▾	
Description	MSSQL-DB-2019	Status	Active
Last used	None	Created	Now
Last used region	N/A	Last used service	N/A

S3 Bucket

The screenshot shows the Amazon S3 'Buckets' page. A green banner at the top indicates that a bucket named 'mssql-db-bk-2019' has been successfully created. Below the banner, there are two tabs: 'General purpose buckets' (selected) and 'Directory buckets'. Under 'General purpose buckets', there is a table with one row. The row contains a blue link 'mssql-db-bk-2019', the 'AWS Region' 'Asia Pacific (Mumbai) ap-south-1', and the 'Creation date' 'February 22, 2026, 11:05:37 (UTC+05:30)'. To the right of the table are buttons for 'Copy ARN', 'Empty', 'Delete', and 'Create bucket'.

aws configure

Enter your:

AWS Access Key ID:- AKIA33MSVLE3M5B6F5ND

AWS Secret Access Key:- vt/flsLHmovwrHB6lis8Nq5HwxYv2vuzq4ktFltb

Default region name :- ap-south-1

Default output format (press Enter to skip).

```
PS : E:\SQLBackups> & "E:\SQLBackups\BackupToS3.ps1"
& : File E:\SQLBackups\BackupToS3.ps1 cannot be loaded because running scripts is disabled on this system. For more information, see about_Execution_Policies at https://go.microsoft.com/fwlink/?LinkId=135170.
At line:1 char:3
+ & "E:\SQLBackups\BackupToS3.ps1"
+ ~~~~~
+ CategoryInfo          : SecurityError: () [], PSSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess
```

Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass

```
PS E:\SQLBackups> Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass
Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose you to the security risks described in the about_Execution_Policies help topic at https://go.microsoft.com/fwlink/?LinkId=135170. Do you want to change the execution policy?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): A
PS E:\SQLBackups>
PS E:\SQLBackups> dir
Directory: E:\SQLBackups

          Name                           Length
          Type
-----  -----
-a----  2009 BackupToS3.ps1
-a----  0 New Text Document.txt
```

& "E:\SQLBackups\BackupToS3.ps1"

```
PS E:\SQLBackups>
PS E:\SQLBackups> & "E:\SQLBackups\BackupToS3.ps1"
Starting backup for PRACTICE...
Processed 552 pages for database 'PRACTICE', file 'PRACTICE' on file 1.
Processed 2 pages for database 'PRACTICE', file 'PRACTICE_log' on file 1.
BACKUP DATABASE successfully processed 554 pages in 0.016 seconds (270.263 MB/sec).
Backup successful. Creating Password Protected Zip...

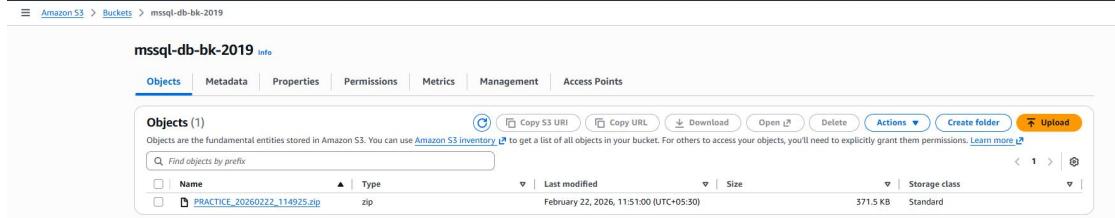
7-Zip 21.07 (x64) : Copyright (c) 1999-2021 Igor Pavlov : 2021-12-26

Scanning the drive:
1 file, 4640768 bytes (4532 KiB)

Creating archive: E:\SQLBackups\PRACTICE_20260222_114925.zip

Add new data to archive: 1 file, 4640768 bytes (4532 KiB)

Files read from disk: 1
Archive size: 380437 bytes (372 KiB)
Everything is Ok
Zip created. Uploading to S3...
upload: .\PRACTICE_20260222_114925.zip to s3://mssql-db-bk-2019/PRACTICE_20260222_114925.zip
Upload successful. Cleaning up local files...
Done.
PS E:\SQLBackups>
```



BackupToS3.ps1

```

# --- CONFIGURATION ---
$ServerInstance = "localhost\SQLEXPRESS"
$DatabaseName = "PRACTICE"
$LocalBackupPath = "E:\SQLBackups"    # Ensure this folder exists
$S3BucketName = "mssql-db-bk-2019"
$Region = "ap-south-1"

# --- ZIP CONFIGURATION ---
# If you installed 7-Zip normally, this path is correct. If not, update it.
$SevenZipPath = "C:\Program Files\7-Zip\7z.exe"
$ZipPassword = "Password@123" # <--- SET YOUR PASSWORD HERE

# --- SCRIPT LOGIC ---
$TimeStamp = Get-Date -Format "yyyyMMdd_HHmmss"
$FileName = "$DatabaseName`_$TimeStamp.bak"
$FullPath = "$LocalBackupPath\$FileName"
$ZipFileName = "$DatabaseName`_$TimeStamp.zip"
$ZipFullPath = "$LocalBackupPath\$ZipFileName"

Write-Host "Starting backup for $DatabaseName..."

# 1. Run SQL Backup
$Query = "BACKUP DATABASE [$DatabaseName] TO DISK = '$FullPath' WITH FORMAT, NAME = 'Express Backup';"
sqlcmd -S $ServerInstance -Q $Query -b

# Check if backup failed
if ($LASTEXITCODE -ne 0) {
    Write-Error "SQL Backup failed! Check permissions or path."
    exit
}

Write-Host "Backup successful. Creating Password Protected Zip..."

# 2. Create Password Protected Zip using 7-Zip
# a = add, -tzip = zip format, -p = password, -mx5 = normal compression
$Arguments = "a -tzip -p$ZipPassword -mx5 `"$ZipFullPath`" `"$FullPath`""
Start-Process -FilePath $SevenZipPath -ArgumentList $Arguments -Wait -NoNewWindow

# Check if zip was created
if (-not (Test-Path $ZipFullPath)) {
    Write-Error "Zipping failed! Check if 7-Zip is installed correctly."
    exit
}

Write-Host "Zip created. Uploading to S3..."

```

```
# 3. Upload ZIP to S3
aws s3 cp $ZipFullPath "s3://$S3BucketName/$ZipFileName" --region $Region

# 4. Cleanup
if ($?) {
    Write-Host "Upload successful. Cleaning up local files..."
    Remove-Item $FullPath # Delete .bak
    Remove-Item $ZipFullPath # Delete .zip
    Write-Host "Done."
} else {
    Write-Error "S3 Upload failed! Files retained at $LocalBackupPath"
}
```