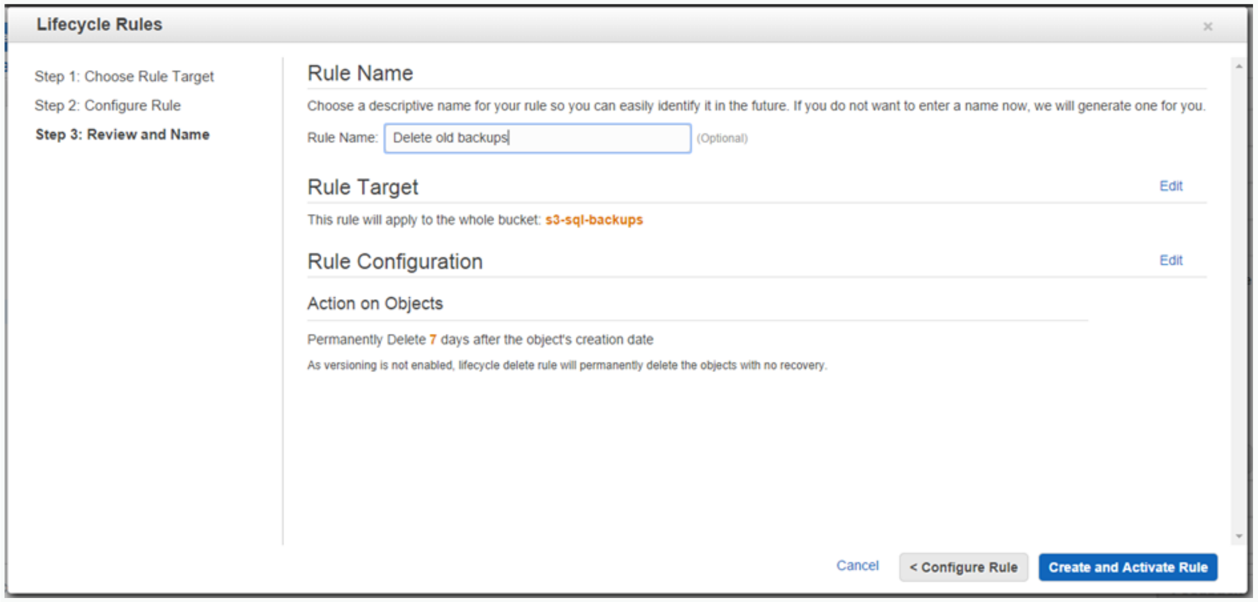
Setting up the S3 bucket

Create a new S3 bucket to store your backups, this example will use the name “*s3-sql-backups*” and assume you are using the AWS console.

Select the newly created bucket and select “Properties” from the options on the top right, then select “Lifecycle”. S3 lifecycle rules allow you to automatically delete or archive content to the Glacier service when the file is a certain age. Create a new lifecycle rule for the whole bucket, that permanently deletes the file after 7 days. This will stop you running up a large S3 bill due to storing old backups. Alter the time range to suite your needs, or consider pushing the backups to Glacier for cheaper long term storage.


Lifecycle Rules

Step 1: Choose Rule Target
Step 2: Configure Rule
Step 3: Review and Name

Rule Name

Choose a descriptive name for your rule so you can easily identify it in the future. If you do not want to enter a name now, we will generate one for you.

Rule Name: Delete old backups (Optional)

Rule Target

This rule will apply to the whole bucket: **s3-sql-backups**

Rule Configuration

Action on Objects

Permanently Delete 7 days after the object's creation date

As versioning is not enabled, lifecycle delete rule will permanently delete the objects with no recovery.

Your script will need to access this bucket, create an IAM user or role to use, and give them full control of this bucket with the following access control policy, replacing *s3-sql-backups* with the name of your bucket.

```
{
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:ListAllMyBuckets"
         ],
         "Resource":"arn:aws:s3:::*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:ListBucket",
            "s3:GetBucketLocation"
```

```
                ],
                "Resource":"arn:aws:s3:::s3-sql-backups"
            },
            {
                "Effect":"Allow",
                "Action":[
                    "s3:PutObject",
                    "s3:GetObject",
                    "s3:DeleteObject"
                ],
                "Resource":"arn:aws:s3:::s3-sql-backups/*"
            }
        ]
    }
```

The Script

The script is straightforward and uses the SQL Backup cmdlets described by #kalpeshpatelce. The backup is created, with the timestamp in the name, copied to S3 and then deleted from the local machine (along with any other old backups).

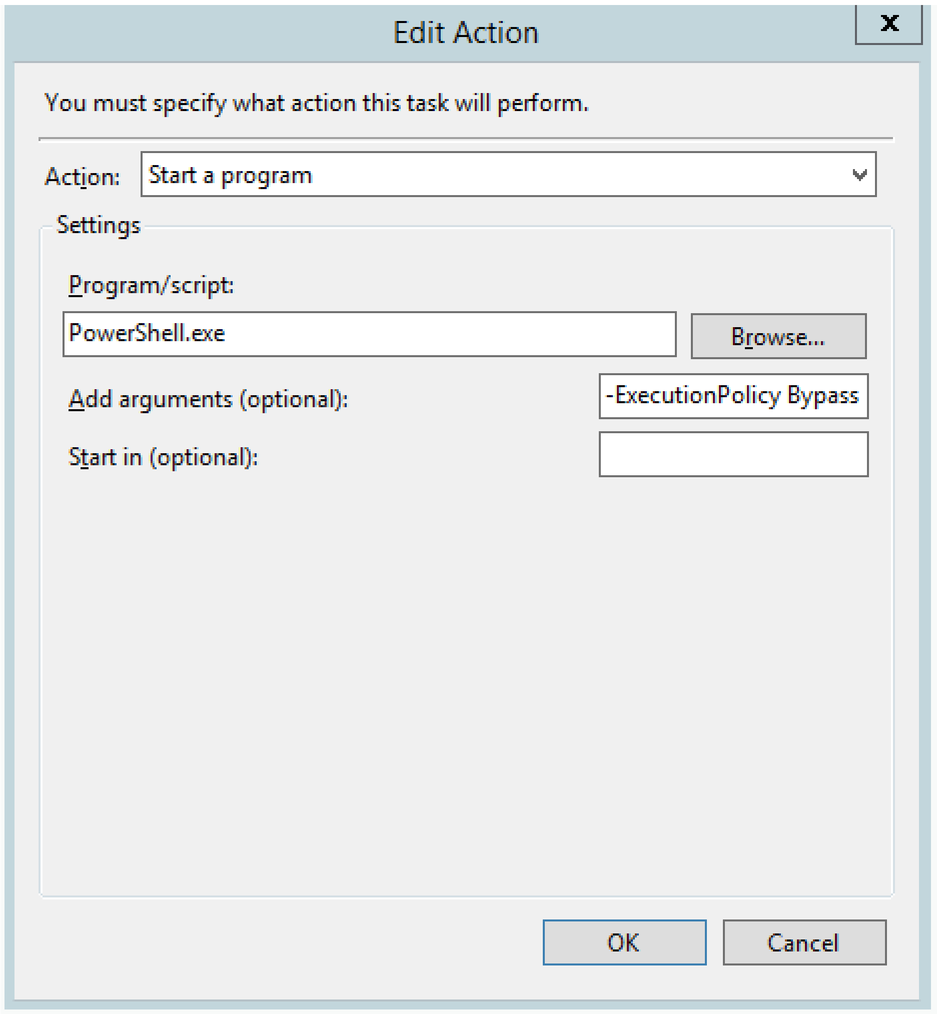
The variables at the start of the script need to be updated to point to the correct database, S3 bucket and backup location. The script assumes Windows Authentication to connect to the database. If you are running the script on an EC2 instance, and have applied the security policy to the IAM role used by the instance the AccessKey and SecretKey parameters can be removed from the Write-S3Object command.

```
$server = '.'
$database = 'MyDatabase'
$s3Bucket = 's3-sql-backups'
$backupPath = 'C:\SqlBackup\'
$region = 'eu-west-1'
# accessKey and secretKey can be removed if running on an EC2 instance and using IAM roles for security
$accessKey = 'USER-ACCESS-KEY'
$secretKey = 'USER-SECRET-KEY'

$timestamp = get-date -format yyyyMMddHHmmss
```

```
    $fileName = "$database-$timestamp.bak"
    $filePath = Join-Path $backupPath $fileName

    Backup-SqlDatabase -ServerInstance $server -Database $database -BackupFile $filePath
    # AccessKey and SecretKey can be removed if running on an EC2 instance and using IAM roles for security
    Write-S3Object -BucketName $s3Bucket -File $filePath -Key $fileName -Region $region -AccessKey $accessKey -SecretKey $secretKey
    Remove-Item $backupPath$database*.bak
```

Scheduling

The script then needs to be scheduled to run every night, I'm using scheduled tasks for this, creating a task that runs nightly and triggers the powershell script by running `Powershell.exe` with the arguments `-ExecutionPolicy Bypass C:\SqlBackup\SqlBackupToS3.ps1`.



You must specify what action this task will perform.

Action: Start a program

Settings

Program/script: PowerShell.exe

Add arguments (optional): -ExecutionPolicy Bypass

Start in (optional):