

# Object Lifecycle Management

This tool basically will help you manage your capacity consumption by putting an expiration date on your old file's configurations after a set amount of days. This tool is best paired up with S3 Versioning to put an expiration date on older versions of a file to prevent large capacity consumption on files that will not be used.

---

## Notes:

To simplify the process, setting up aliases will help with management and organization of code. In our demo, our alias is shown as:

```
alias myaws='/usr/local/bin/aws --endpoint-url http://90.0.0.11'
alias mys3='myaws s3api'
```

## Instructions:

1. Using AWSCLI, set up a Configuration Input File in a JSON format with the following rules to dictate how many days the version has before it expires. Modify this file to be appropriate for your environment.

```
{
  "Rules": [
    {
      "ID": "90day",
      "Prefix": "",
      "Status": "Enabled",
      "NoncurrentVersionExpiration": {
        "NoncurrentDays": 90
      }
    }
  ]
}
```

2. Use the PUT command to place to configure a bucket with this file.

```
> mys3 put-bucket-lifecycle --bucket versioning-enabled --lifecycle-configuration file://90-day-expiration.json
```

3. Displaying the information about the bucket to make sure that our bucket has been configured correctly, the bucket information will show the rules that are sent in the Configuration Input File.

```
> mys3 get-bucket-lifecycle --bucket versioning-enabled
{
  "Rules": [
    {
      "Status": "Enabled",
      "Prefix": "",
      "NoncurrentVersionExpiration": {
        "NoncurrentDays": 90
      },
      "ID": "90day"
    }
  ]
}
```