How to use Amazon S3 as a datastore backend

This article describes how to setup Magnolia/Jackrabbit to use Amazon S3 as a datastore backend.

This can be helpful for Docker-like deployments. In this type of deployment, it's common to move as much of the content and configuration out of your file system as possible (by default the datastore stores the binaries on the FS). Performance when handling big files should not be affected by using S3.

Add Maven dependency to your project

It will add `jackrabbit-aws-ext` library to your project. This library contains necessary classes Jackrabbit uses to store the binaries in the S3.

```
<dependency>
  <groupId>org.apache.jackrabbit</groupId>
  <artifactId>jackrabbit-aws-ext</artifactId>
  <version>2.18.1</version>
</dependency>
```

Create `aws.properties` file

Add information about your AWS S3 account so Jackrabbit can connect to it.

```
# AWS account ID
accessKey=
# AWS secret key
secretKey=
# AWS bucket name
s3Bucket=
# AWS bucket region
s3Region=
# S3 endpoint to be used. This parameter is optional
# and has a higher precedence over endpoint derived
# via S3 region.
# Recommended to set as the URL constructed from the info above might not be correct. The format is bucket.s3.amazonaws.com.
s3EndPoint=
connectionTimeout=120000
socketTimeout=120000
maxConnections=20
maxErrorRetry=10
# maximum concurrent threads to write to S3.
writeThreads=10
# proxy configurations (optional)
proxyHost=
proxyPort=
```

Configure the Datastore in the Jackrabbit config file

```
<DataStore class="org.apache.jackrabbit.aws.ext.ds.S3DataStore">
  <param name="config" value="/path/to/your/aws.properties"/>
</DataStore>
```

`S3DataStore` class extends CachingDataStore. This means by default the disk space will be still used to store the cached items. The location can be specified by the `path` parameter. Otherwise the repository folder will be used.

```
<param name="path" value="/path/where/you/want/the/s3/cache/stored"/>
```

The cache can be disabled so the items will always be read from the S3 and no FS is needed. To do so, set `cacheSize` parameter to 0.

```
<param name="cacheSize" value="0"/>
```

Useful links

- aws.properties example
- S3DataStore class
- AWS console