



CloudBerry Lab

#1 Cross Platform Cloud Backup

CloudBerry Backup with the Scality S3 Server

Installation and Configuration Guide



CloudBerry Backup with the Scality S3 Server

Contents

[Abstract](#)

[Introduction](#)

[Purpose](#)

[Architecture](#)

[Result](#)

[Requirements](#)

[General](#)

[Deployment scenario examples](#)

[Reference Physical Hardware](#)

[Distributed Scality](#)

[Docker container](#)

[Building Scality S3 Server with Node.js and NPM](#)

[Environment preparation](#)

[Building Scality S3](#)

[Managed Backup Service](#)

[Introduction](#)

[Storage account management](#)

[About Scality](#)

[About CloudBerry Lab](#)

[Appendix A: CloudBerry Lab stand-alone products usage](#)

[CloudBerry Explorer](#)

[Storage account configuration](#)

[Bucket and object management](#)

[Appendix B: Useful links and information](#)



Abstract

In this guide we will explain how one can install and configure cloud storage software and use it for backup purposes. This document can be used by an individual or a company who wishes to use its own hardware for cloud storage. We will use the Scality S3 Server along with CloudBerry Backup.

Introduction

Object storage is considered a leading emerging technology and many of its features are deemed ideal for cloud storage. Benefits of object storage include massive scalability, geographic independence, multi-tenant features, and the ability to use off-the-shelf hardware. More companies are considering object storage as a part of their infrastructure for use cases like backup. They choose it for its benefits and price. There are also public cloud storage services on the market provided by such vendors as Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP), and many others. However, there are still a lot of companies who already own hardware and want to transform it partially or entirely to a cloud storage solution that uses the latest technologies. It's also a huge playground for Managed Service Providers (MSPs), who are willing to provide cloud storage and cloud backup services to their customers. New technologies, such as object-based storage require new tools, like CloudBerry Backup.

Purpose

This whitepaper may be used by Managed Service Providers who are willing to expand their portfolio by providing cloud backup services to their customers using their own private cloud storage and CloudBerry Managed Backup. There are two ways to provide cloud storage to the final customers from an MSPs point of view. The first way is to create a new account on a public cloud storage provider like Amazon and resell it. The second way is to run it on-premises to create a private cloud storage solution. This paper will focus on the latter and explain how this can be done with Scality S3 Server and how to use CloudBerry Managed Backup to offer an online backup service to the customer. As an add-on, we will show how the storage can be accessed with other CloudBerry products.

Architecture

This whitepaper will focus on the Scality S3 Server which is an open-source project that compliments S3-based application development by providing an easy to use S3 Server that can be up and running in minutes. Since S3 Server is lightweight and portable, users can build and integrate S3-based applications faster using their laptop or commodity hardware without the need to install a full object storage environment.

Result

Since Scality S3 Server is compatible with the AWS S3 API, users will be able to backup data with CloudBerry Backup, manage files with CloudBerry Explorer, and map a storage bucket from the Scality S3 Server as removable disk or network share using CloudBerry Drive. Thus, we proved the concept which any individual or company can use to provide cloud backup service using cloud storage built from their own hardware resources. We encourage everyone to go through the this paper to get the basic idea of the technical requirements, tips, and general steps of the installation.

Requirements

General

Compute

Scality S3 Server is an object-storage server released under Apache License v2.0. It is compatible with Amazon S3-based applications. It is best suited for storing unstructured data such as photos, videos, log files, backups, and container/VM images. An object is basically a file and the size of an object can range from a few KB's to a maximum of 5 TB's.

Scality S3 Server requires Node.js 4.5 and the NPM (Package Manager) 2.0 or greater. The following methods are available for installation:

- Docker Container
- Directly from source with Node.js (GitHub repository)

Storage

Scality S3 Server uses available space on the disk or network share. The storage size requirements directly depend on the source data and backup strategy (e.g. data retention policy, compression ratio, and change rate of incremental data).

Deployment scenario examples

Scality S3 server runs on a variety of hardware, operating systems and, x virtual/container environments.

Distributed Scality

Scality S3 Server provides an easy way to quickly spin up an S3 server for development and testing. For production use cases, there is Scality RING, a Software-Defined Storage (SDS) solution for petabyte-scale data storage that is designed to interoperate in the modern Software Defined Data Center SDDC.

The RING is a distributed system deployed on industry standard servers and can be seamlessly scaled-out to thousands of storage servers with 100's of petabytes of storage capacity. The RING has no single points of failure, zero downtime during upgrades, scaling, planned maintenance or unplanned system operations, and has self-healing capabilities.

To provide S3 API access to the storage, the Scality RING ships with an S3 connector that leverages the open-source S3 Server with additional features ranging from enterprise authentication to a distributed scale-out architecture. To manage users in the RING, the S3 connector supports AWS IAM, LDAP, and Active Directory authentication. The S3 connector is highly-available and can be load balanced for optimal performance.

For more information on the Scality RING, please visit <http://www.scality.com/ring/object-storage-overview>

Docker container

Note: *This document considers that you have basic experience with Docker. For a Docker technology overview, installation guides, and a general overview, please visit:*

<https://www.docker.com/products/overview>

CloudBerry Backup with the Scality S3 Server

The following command downloads the S3 Server container from the Docker Registry:

```
$ docker pull scality/s3server
```

Depending on the internet connection speed, the download may take some time. Once the download is completed, launch the S3 Server container and define the Access and Secret keys that are used to authenticate S3 applications like CloudBerry's with the S3 Server.

```
$ docker run -d --name s3server -p 8000:8000 scality/s3server
```

NB!

The arguments: -p 8000:8000 exposes the S3 Server port inside of the container to be accessible by the physical machine running Docker.

Please give the S3 Server container a few minutes to get up and running properly. The typical startup time is 1-2 minutes. You may check your running containers with the following command:

```
$ docker ps
```

You should see a similar output below that shows the S3 Server container up and running:

```
> / docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
b376626cca4e	scality/s3server	"/usr/src/app/dock..."	17 seconds ago	Up 12 seconds	0.0.0.0:8000->8000/tcp	s3server

Image 1 (docker ps output with list of containers)

CloudBerry Backup with the Scality S3 Server

The S3 Server ships with a default set of access and secret key credentials. The access key is “accessKey1” and the secret key is “verySecretKey1”.

Connecting to the S3 Server with CloudBerry Explorer

Let’s configure Explorer and start adding objects to the Scality S3 server.

Download link: <http://www.cloudberrylab.com/explorer>

Storage account configuration

Open CloudBerry Explorer and select S3 Compatible in File > New S3 Compatible Account > S3 Compatible (Image 9).

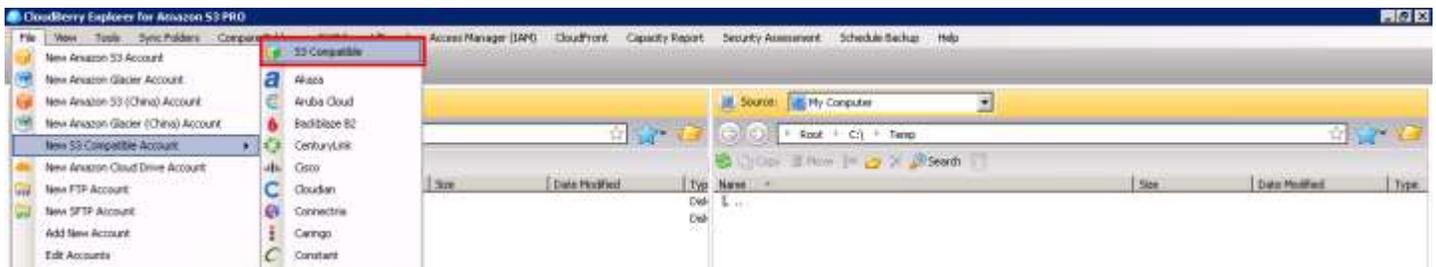


Image 7 (CloudBerry Explorer for Scality Cloud Storage)

You need to define the **Service endpoint**, **Access Key**, **Secret Key**, **Signature version**, and other settings (e.g. multipart upload support, SSL).

NB!

1. As SSL is very important (securing data in transit) HTTPS is quite hot topic as well. We did not install SSL certificates to simplify the installation.

CloudBerry Backup with the Scality S3 Server

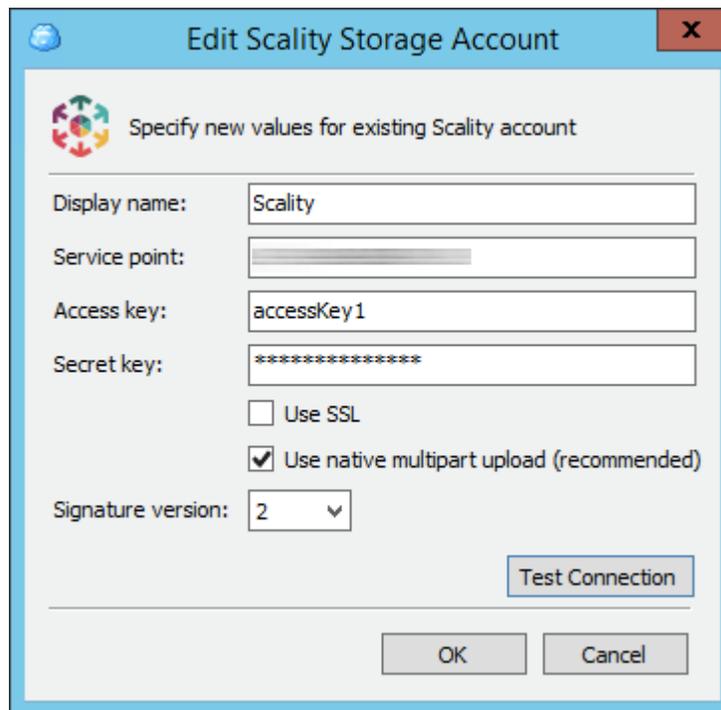


Image 9 (CloudBerry Explorer for Scality S3 Storage configuration card)

Bucket and object management

Now we are ready to manage buckets and files from the Scality S3 Storage. CloudBerry Explorer supports everything that needs to be done with our buckets and files (create, delete, update, rename, move, download, upload, etc).



Image 10 (CloudBerry Explorer for Scality Cloud Storage objects management)

Building Scality S3 Server with Node.js and NPM

NB!

This part of the white paper considers that you have basic experience and understanding of the following:

- Node.js (<https://nodejs.org/en/>)
- NPM (Node Package Manager), <https://www.npmjs.com/>
- Github (<http://github.com>)

Environment preparation

Node.js can be installed using different ways. The best way is to obtain an archive with product bits from here <https://nodejs.org/en/download/> and build it. Node.js available for various operating systems.

Let's explore how to install Node.js for Ubuntu with the build dependencies for the S3 Server:

```
$ curl -sL https://deb.nodesource.com/setup_6.x | sudo -E bash -  
$ sudo apt-get install -y nodejs build-essential python
```

In order to validate the Node.js version prerequisites for Scality, please run the following commands:

```
$ node -v  
$ npm -v
```



Both commands should display version numbers in the output. Please check the version numbers against the Scality Node.js requirements stated earlier in this paper.

Building Scality S3

NB!

This step requires basic knowledge of GitHub. The following document may help to fulfill the gap <https://help.github.com/>.

We need to do the following:

- Fetch the source from the Git repository
- Install the software dependencies using NPM
- Start the S3 Server

Let's clone the repository by executing the following:

```
$ git clone https://github.com/scality/S3.git
```

If you don't have the git client, you can install it from a Linux software repository, homebrew for Mac OS, or directly from GitHub. You can also install the GUI for using Github (optionally, if you are not comfortable with command-line interface).

The result of this command should be the following:

CloudBerry Backup with the Scality S3 Server

```
➤ ~/P/cloudberrylab on master x git clone https://github.com/scality/S3.git
Cloning into 'S3'...
remote: Counting objects: 14111, done.
remote: Compressing objects: 100% (179/179), done.
remote: Total 14111 (delta 128), reused 16 (delta 16), pack-reused 13916
Receiving objects: 100% (14111/14111), 15.24 MiB | 981.00 KiB/s, done.
Resolving deltas: 100% (9379/9379), done.
Checking connectivity... done.
```

Image 3 (git clone downloads the project from GitHub and stores in a folder named S3)

Now we are ready to install the application as defined in package.json which is the configuration file that npm uses for Node.js projects. This file contains a list of dependencies for the Scality S3 Server.

```
$ cd S3
$ npm install
```

This may take some time to finish. Once completed, let's start the Scality S3 Server:

```
$ npm start
```

Now the S3 Server has been started and you should see similar output below (compare your output with Image 4).

```
➤ ~/P/c/S3 on master o npm start
09
> s3@1.1.0 start /Users/evgeny/Projects/cloudberrylab/S3
> node init.js && node index.js
```

Image 4 (Scality S3 Server is ready)

Managed Backup Service

Introduction

[CloudBerry Managed Backup Service](#) (MBS) is built from CloudBerry Backup technology and is designed to meet the needs of managed service providers and enterprise IT departments. MBS provides reliable backup with central management and monitoring and is integrated with Amazon Web Services, Microsoft Azure, Google Cloud Platform and almost every S3-compatible or OpenStack-based cloud storage services.

For service providers using CloudBerry Managed Backup, any MSP, VAR or IT service company can rebrand it and provide robust and reliable cloud backup service with remote management and monitoring.

For corporate use, CloudBerry Managed Backup is also a good fit for internal use by businesses of all sizes because it has features to manage and monitor cloud backup clients across multiple machines and control billing and licenses from a single web-based control panel.

Storage account management

Since the Scality S3 server is AWS S3 compatible, we can add its service endpoint to our MBS account. In order to do this we need to:

- Be either MBS Administrator or Sub-Administrator with certain permissions (learn more https://mbs.cloudberrylab.com/Admin/Help.aspx?c=Contents/multiple_admins.html)
- Navigate to Storage > Storage Accounts and click “Add Account” and select “S3 compatible” as highlighted on Image 5.

CloudBerry Backup with the Scality S3 Server

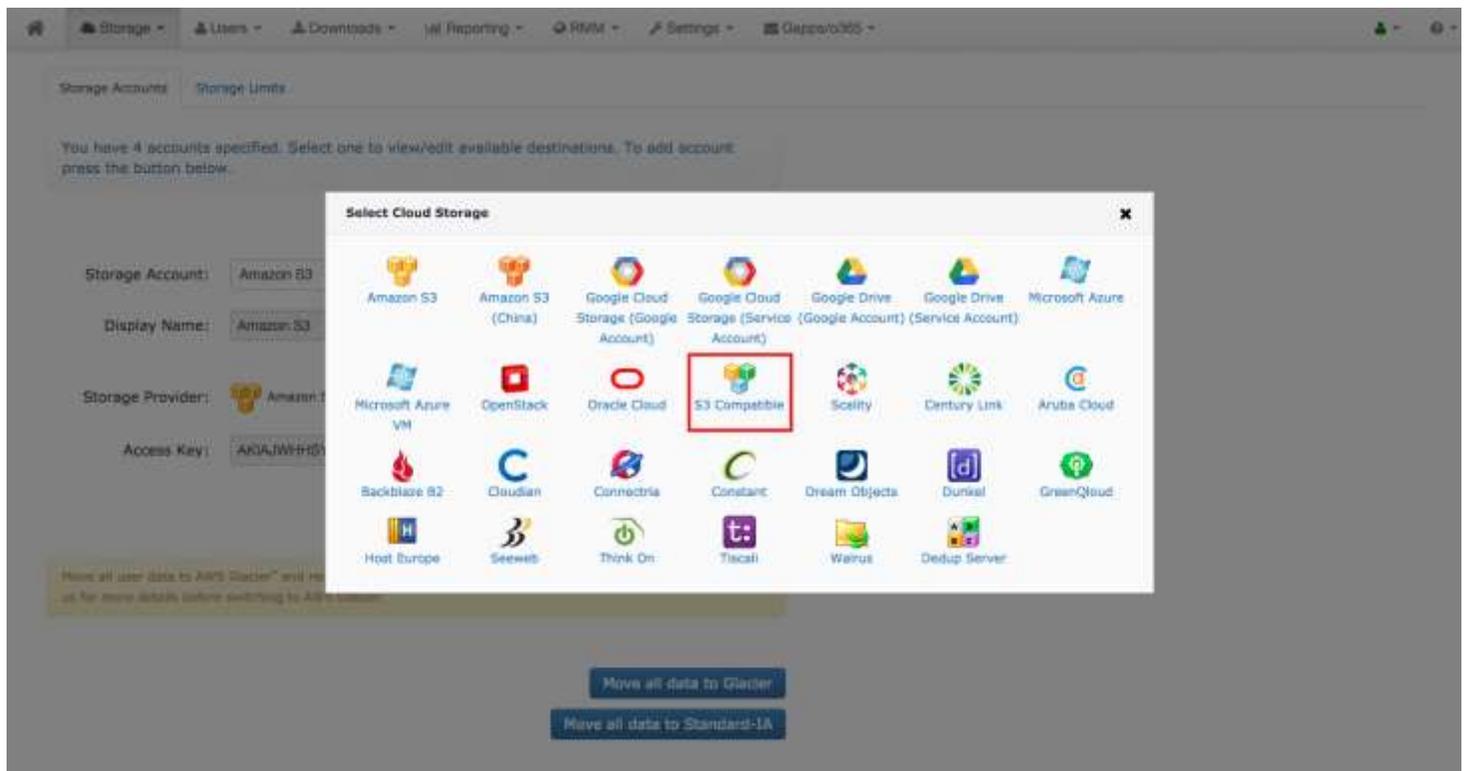


Image 5 (MBS S3 compatible storage account configuration)

Now we need to enter the the **endpoint** for HTTP and HTTPS (if you configure SSL certificate for the server), **AccessKey** and **SecretKey** to access to the Scality S3 Server storage.

Display Name:
Scality

Storage Type:
S3 Compatible Change

Access Key:
accessKey1

Secret Key:

Signature version (coming soon): 2

Use Native Multipart Upload

HTTP endpoint:
127.0.0.1:8000

HTTPS endpoint:
127.0.0.1:8000

Ignore certificate
 Do not check credentials (no public access)

Save Cancel

NB!

2. As SSL is very important (securing data in transit) HTTPS is quite a hot topic as well. We did not install SSL certificates on the Scality S3 server to simplify the installation process.

3. The Scality S3 Server and CloudBerry Backup both support v2 and v4 of the AWS S3 signatures.

More details can be found here:

<http://docs.aws.amazon.com/AmazonS3/latest/API/bucket-policy-s3-sigv4-conditions.html>

This popup requires:

- **Access Key** and **Secret Key**
- **HTTP** and **HTTPS service endpoints**
- **Signature version**
- Other additional parameters

We also recommend to enable “Use Native Multipart Upload” in order to use chunks and multiple parallel streams of data upload. This makes transfer faster, supports retransmit (even after long network outage). Multipart upload first was introduced by AWS.

Image 6 (MBS S3 compatible storage account configuration card)

About Scality

Scality, world leader in object and cloud storage, develops cost-effective Software Defined Storage (SDS): the RING, which serves over 500 million end-users worldwide with over 800 billion objects in production; and the open-source S3 Server. Scality RING software deploys on any industry-standard x86 server, uniquely delivering performance, 100% availability and data durability, while integrating easily in the datacenter thanks to its native support for directory integration, traditional file applications and over 45 certified applications. Scality’s complete solutions excel at serving the specific storage needs of Global 2000 Enterprise, Media and Entertainment, Government and Cloud Provider customers while delivering up to 90% reduction in TCO versus legacy storage. A global company, Scality is headquartered in San Francisco

About CloudBerry Lab

CloudBerry's offerings include powerful, easy-to-use backup management capabilities and military-grade encryption using customer-controlled keys. Customers can choose to store their backup data with more than 20 online storage providers, including Amazon S3, Microsoft Azure, Google Cloud, HP Cloud, Rackspace, IBM Softlayer and others. CloudBerry also partners with thousands of VARs and MSPs to provide them with turnkey, white-label data protection services. CloudBerry Lab is an Amazon Web Services Advanced Technology Partner.

Appendix A: Useful links and information

Standalone products (MBS is not required):

- *CloudBerry Backup for Windows* — <https://www.cloudberrylab.com/backup>
- *CloudBerry Backup for Mac* — <https://www.cloudberrylab.com/mac>
- *CloudBerry Backup for Linux* — <https://www.cloudberrylab.com/linux>
- *CloudBerry Explorer* — <http://www.cloudberrylab.com/explorer>
- *CloudBerry Drive* — <https://www.cloudberrylab.com/drive>

"S3 compatible" interface technologies overview:

- *Multipart upload* — <http://docs.aws.amazon.com/AmazonS3/latest/dev/mpuoverview.html>
- *Scality GitHub repository* — <https://github.com/scality/S3>
- *Scality Enterprise S3 storage details* — <https://s3.scality.com/v1.0/page/s3-enterprise-edition>
- *Scality RING* — <http://www.scality.com/ring/object-storage-overview/>

