



CloudBerry Lab

#1 Cross Platform Cloud Backup

CloudBerry Backup with Minio.io

Installation and Configuration Guide

Contents

[Abstract](#)

[Introduction](#)

[Purpose](#)

[Architecture](#)

[Result](#)

[Requirements](#)

[Operating systems](#)

[Storage](#)

[Deployment scenario examples](#)

[Reference Physical Hardware](#)

[Distributed Minio](#)

[Installation](#)

[Docker container](#)

[GNU/Linux](#)

[Microsoft Windows](#)

[Managed Backup Service](#)

[Introduction](#)

[Storage account management](#)

[About Minio](#)

[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 white-paper we will explain how one can install and configure cloud storage software and use it for backup purposes. Document can be used by an individual or a company who wishes to become a cloud backup provider and utilize its own hardware for cloud object-based storage. We will use Minio.io object-storage software along with CloudBerry Lab products.

Introduction

Object-based 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-based storages as a part of their infrastructure, for instance for backup purposes. They choose it for its benefits and price. There is a number of public object-based cloud storages offered on the market, such as Amazon S3, Glacier, Windows Azure, Rackspace, and many others. However, there are still many companies who already have existing hardware and want to transform a part of it or transform it entirely in a cloud storage, utilizing latest technologies. It's also a huge playground for Managed Service Providers (MSP), who are willing to provide cloud storage and backup services to their final customers at the same time. New technologies, such as object-based storage require new tools, like products of CloudBerry Lab.

Purpose

This white-paper 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 MSPs' point of view. First one is to create new account on public cloud storage provider, like Amazon, for example, and resell it. However, there is a number of MSPs who run their own hardware and want to utilize its capacity to construct their own private cloud storage built on their own resources. In this white paper, we will briefly explain the second way: how this can be done by using Cloudian object-storage software. We will explain how to use CloudBerry Managed Backup to offer online backup service to the final



customer. As an add-on, we will show how this cloud storage can be accessed and used by stand-alone products of CloudBerry Lab for object-based storages.

Architecture

Minio Cloud Storage will be used in this white-paper. Minio is a distributed object storage server built for cloud applications and devops. It is written in Go lang and fully compatible with Amazon S3 APIs.

Result

In the scope of this white-paper we will configure single and distributed models in order to show seamless and scalable S3 cloud storage.

As Minio is fully compatible with AWS API we will be able to backup data by CloudBerry Backup, manage files by CloudBerry Explorer and map it as removable disk or network share to our computer using CloudBerry Drive. Thus, we proved a concept that any individual or company, MSP especially, may use this basic scenario to become a Managed Backup Provider to the cloud storage, built on their own hardware resources. We encourage everyone to go through the following sections to get the basic idea of the technical requirements, tips, and general steps of the installation.

Requirements

Operating systems

Minio is an object storage server released under Apache License v2.0. It is compatible with Amazon S3 cloud storage service. It is best suited for storing unstructured data such as photos, videos, log files, backups and container / VM images. Size of an object can range from a few KBs to a maximum of 5TB.

Minio server is light enough to be bundled with the application stack, similar to NodeJS, Redis and MySQL.

The following installation methods and OSs supported:

- Docker Container
 - Stable
 - Edge
- OS X
 - Homebrew
 - Binary (64-bit Intel)
- GNU/Linux
 - 32-bit Intel
 - 64-bit Intel
 - 32-bit Intel ARM
 - 64-bit Intel ARM
 - 32-bit Intel ARMv6
- Microsoft Windows
 - 32-bit
 - 64-bit
- FreeBSD (64-bit support)

Storage

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

Deployment scenario examples

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

Minio erasure code backend is limited by design to a minimum of 4 drives and a maximum of 16 drives. The hard limit of 16 drives comes from operational experience. Failure domain becomes too large beyond 16 drives. If you need to scale beyond 16 drives, you may run multiple instances of Minio server on different ports.



Reference Physical Hardware

- **SMC 5018A-AR12L (Intel Atom)** - SMC 1U SoC Atom C2750 platform with 12x 3.5" drive bays
- **Quanta Grid D51B-2U (OCP Compliant)** - Quanta 2U DP E5-2600v3 platform with 12x 3.5" drive bays
- **Cisco UCS C240 M4 Rack Server** - Cisco 2U DP E5-2600v3 platform with 12x 3.5" drive bays
- **Intel® Server System R2312WTTYSR** - Intel 2U DP E5-2600v3 platform with 12x 3.5" drive bays

Distributed Minio

Minio in distributed mode can help you setup a highly-available storage system with a single object storage deployment. With distributed Minio, you can optimally use storage devices, irrespective of their location in a network. In the scope of this document we configure single node.

Example 1: Start distributed Minio instance with 1 drive each on 8 nodes, by running this command on all the 8 nodes. The architecture of the configuration is shown on Image 1.

```
$ export MINIO_ACCESS_KEY=<ACCESS_KEY>  
$ export MINIO_SECRET_KEY=<SECRET_KEY>  
$ minio server http://192.168.1.11/export1 http://192.168.1.12/export2 \  
http://192.168.1.13/export3 http://192.168.1.14/export4 \  
http://192.168.1.15/export5 http://192.168.1.16/export6 \  
http://192.168.1.17/export7 http://192.168.1.18/export8
```



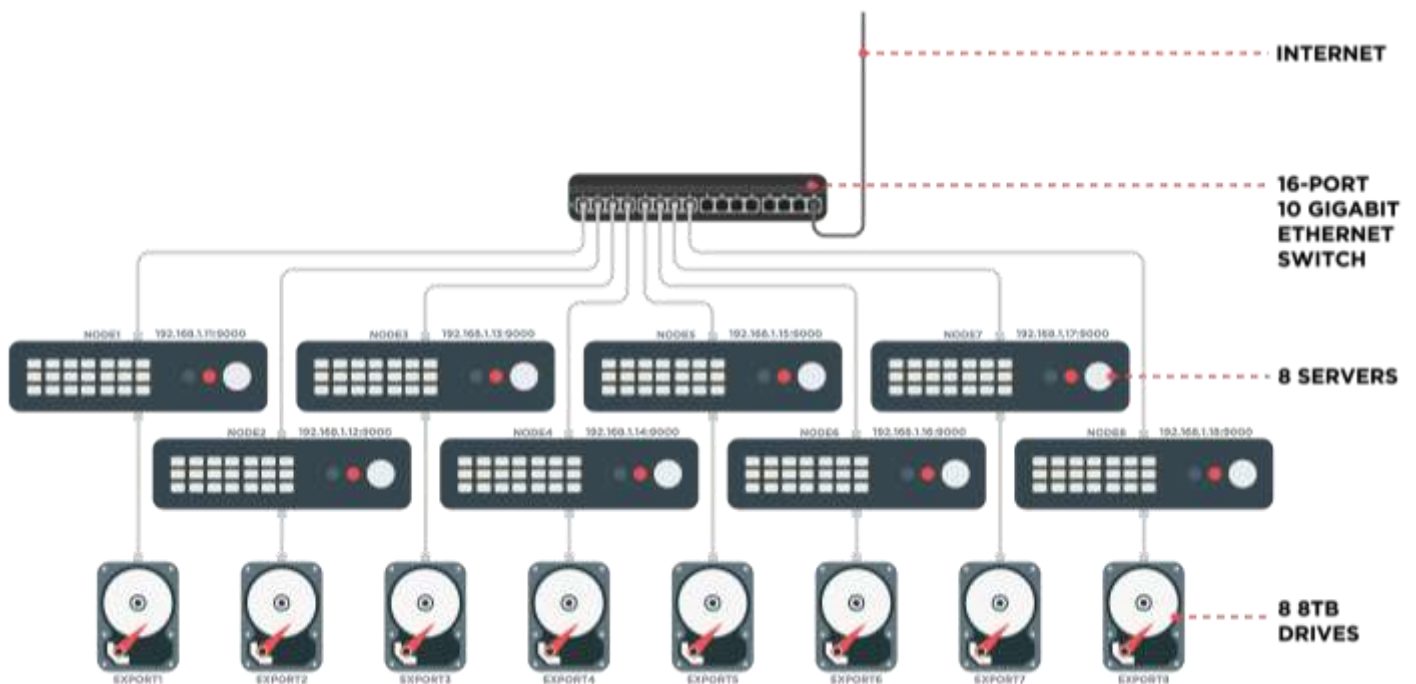


Image 1 (Distributed configuration with 8 drives, 8 nodes)

Example 2: Start distributed Minio instance with 4 drives each on 4 nodes, by running this command on all the 4 nodes. The architecture of the configuration is shown on Image 2.

```
$ export MINIO_ACCESS_KEY=<ACCESS_KEY>
$ export MINIO_SECRET_KEY=<SECRET_KEY>
$ minio server http://192.168.1.11/export1 http://192.168.1.11/export2 \
  http://192.168.1.11/export3 http://192.168.1.11/export4 \
  http://192.168.1.12/export1 http://192.168.1.12/export2 \
  http://192.168.1.12/export3 http://192.168.1.12/export4 \
  http://192.168.1.13/export1 http://192.168.1.13/export2 \
  http://192.168.1.13/export3 http://192.168.1.13/export4 \
  http://192.168.1.14/export1 http://192.168.1.14/export2 \
  http://192.168.1.14/export3 http://192.168.1.14/export4
```



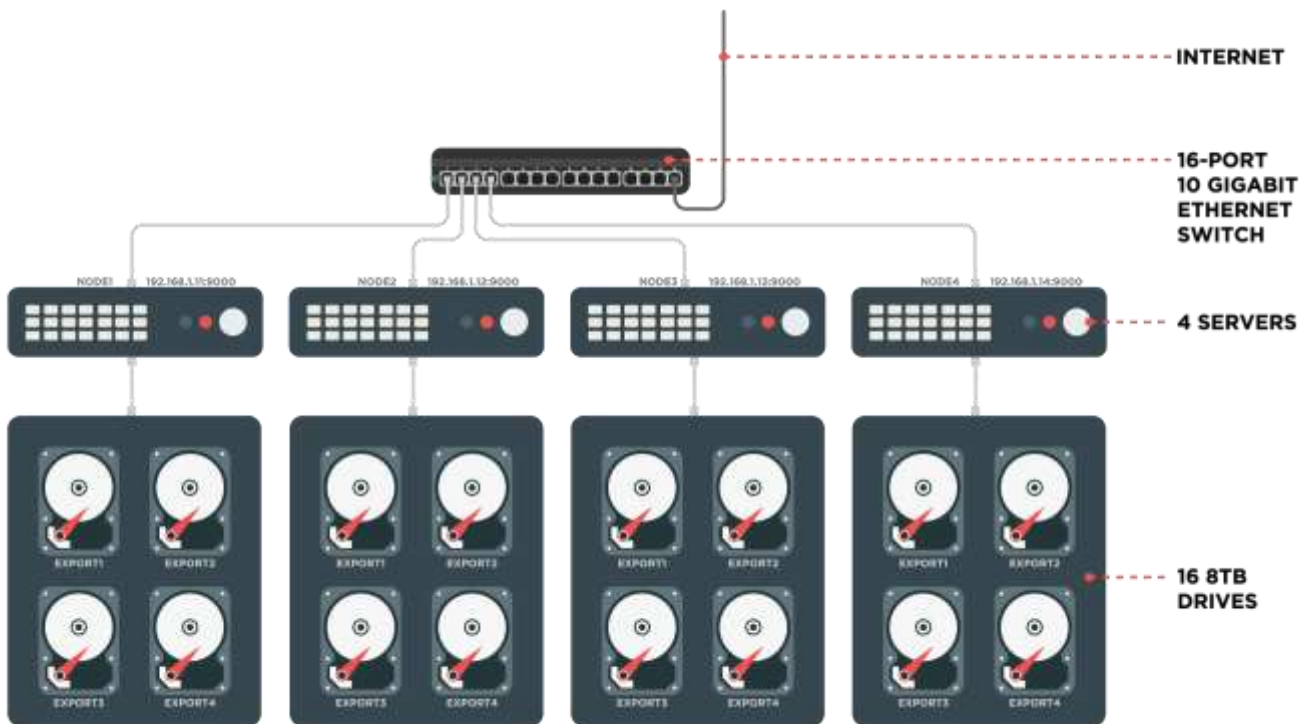


Image 2 (Distributed configuration with 16 drives, 4 nodes)

Installation

Docker container

NB!

This document considers that you have basic experience with docker technology. Otherwise technology overview, installation guides for different platforms and other information is here: <https://www.docker.com/products/overview>

The following command downloads container:

```
$ docker pull minio/minio
```

Depends on the internet connection, download of container may take some time. Once



CloudBerry Backup with Mini.io Installation and Configuration Guide.

installation completed, launch docker container with Minio and point it to the folder that will be parent (root) for all new buckets.

```
$ docker run -p 9000:9000 minio/minio server /export
```

NB!

The following parameter: -p 9000:9000 maps container's network to the docker machine's network. Change it if necessary!

This container runs on OS X with 54 GiB free space. It uses `/export` as parent path for all new S3 buckets in Minio. Let's validate if Minio available, start web browser and check <http://127.0.0.1:9000>. **AccessKey** and **SecretKey** can be founded in terminal output once Docker started (highlighted with red arrow). Use yours! Successful login guides us to the S3 storage explorer, where we can see all our buckets (currently "docker" bucket only) and basic management operations (e.g. create new bucket, upload or delete file).

```
>> / docker run -p 9000:9000 minio/minio server /export
Created minio configuration file at /root/.minio

Endpoint: http://172.17.0.2:9000 http://127.0.0.1:9000
AccessKey: GKHDKMJWFN12IXGR00TC
SecretKey: typUwJy8rV9mM0eN/yTtmsCzgzG7nDSbQ1u6pbtJ
Region: us-east-1
SQS ARNs: <none>

Browser Access:
http://172.17.0.2:9000 http://127.0.0.1:9000

Command-line Access: https://docs.minio.io/docs/minio-client-quickstart-guide
$ mc config host add myminio http://172.17.0.2:9000 GKHDKMJWFN12IXGR00TC typUwJy8rV9mM0eN/yTtmsCzgzG7nDSbQ1u6pbtJ

Object API (Amazon S3 compatible):
Go: https://docs.minio.io/docs/golang-client-quickstart-guide
Java: https://docs.minio.io/docs/java-client-quickstart-guide
Python: https://docs.minio.io/docs/python-client-quickstart-guide
JavaScript: https://docs.minio.io/docs/javascript-client-quickstart-guide

Drive Capacity: 54 GiB Free, 59 GiB Total
```

Image 3 (Console output of running Minio from Docker container on OS X)



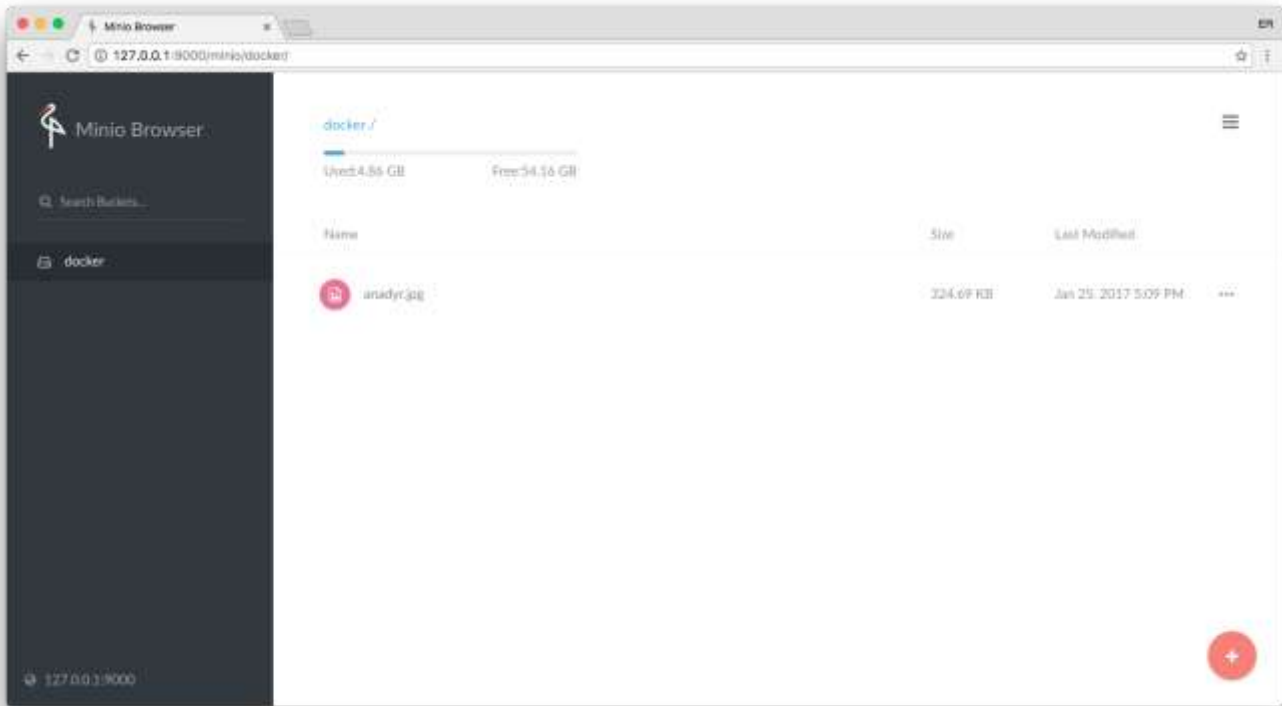


Image 4 (Minio Web interface for objects management)

GNU/Linux

NB!

This part of the document considers you know basics of Linux command line.

Select and download (depends on your architecture and platform) required binary (table 1).

Platform	Architecture	URL
GNU/Linux	64-bit Intel	https://dl.minio.io/server/minio/release/linux-amd64/minio
	32-bit Intel	https://dl.minio.io/server/minio/release/linux-386/minio
	32-bit ARM	https://dl.minio.io/server/minio/r



CloudBerry Backup with Mini.io Installation and Configuration Guide.

		elease/linux-arm/minio
	64-bit ARM	https://dl.minio.io/server/minio/release/linux-arm64/minio
	32-bit ARMv6	https://dl.minio.io/server/minio/release/linux-arm6v1/minio

Table 1 (GNU/Linux binaries)

NB!

To download binary **wget** command can be used.

```
$ cd /home/  
$ wget https://dl.minio.io/server/minio/release/linux-amd64/minio
```

Once downloaded, let's make file executable and launch the server.

```
$ chmod +x minio  
$ ./minio server /backup
```

The following occurs in terminal and let's check through web browser the availability of its endpoint (Image 5). You need **AccessKey** and **SecretKey** (marked by white arrow below).

NB!

-p <number> can be used for setting the port number (by default it is 9000, so make sure your firewall is configured accordingly).



```
root@debian-jessie:/home/username# ./minio server /home/username/file

Endpoint: http://10.132.0.2:9000 http://127.0.0.1:9000
AccessKey: DBQZQPJR5P81J7J2JJW7
SecretKey: K0014NGWqLDegfynRhSk/VGJGxmlxKl8rt40NA1L
Region: us-east-1
SQS ARNs: <none>

Browser Access:
http://10.132.0.2:9000 http://127.0.0.1:9000

Command-line Access: https://docs.minio.io/docs/minio-client-quickstart-guide
$ mc config host add myminio http://10.132.0.2:9000 DBQZQPJR5P81J7J2JJW7 K0014NGWqLDegfynRhSk/VGJGxmlxKl8rt40NA1L

Object API (Amazon S3 compatible):
Go: https://docs.minio.io/docs/golang-client-quickstart-guide
Java: https://docs.minio.io/docs/java-client-quickstart-guide
Python: https://docs.minio.io/docs/python-client-quickstart-guide
JavaScript: https://docs.minio.io/docs/javascript-client-quickstart-guide

Drive Capacity: 7.7 GiB Free, 9.7 GiB Total
```

Image 5 (Minio terminal output in GNU/Linux)

In this example Minio service runs on *Debian 8.7 "Jessie"* (<https://www.debian.org/releases/jessie/>), but it's been tested with Ubuntu, CentOS, RedHat, Suse Linux latest releases.

Microsoft Windows

Download required architecture and launch with the following line:

Platform	Architecture	URL
Microsoft Windows	64-bit	https://dl.minio.io/server/minio/release/windows-amd64/minio.exe
	32-bit	https://dl.minio.io/server/minio/release/windows-386/minio.exe

Table 2 (Microsoft Windows binaries)

```
C:\Users\Administrator> cd <Drive>:\path\to\download\  
C:\Users\Administrator> minio server C:\backups
```

```
Administrator: Command Prompt - C:\Users\Administrator\Downloads\minio.exe server c:\backups
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>C:\Users\Administrator\Downloads\minio.exe server c:\backups

Endpoint: http://172.31.12.206:9000 http://127.0.0.1:9000
AccessKey: FNZKU01RZJQDFUFWX0J1
SecretKey: USnGEfSvG9EHTuDemG/AIt420qdYQhECDn3DNxM1
Region: us-east-1
SQS ARNs: <none>

Browser Access:
http://172.31.12.206:9000 http://127.0.0.1:9000

Command-line Access: https://docs.minio.io/docs/minio-client-quickstart-guide
$ mc.exe config host add myminio http://172.31.12.206:9000 FNZKU01RZJQDFUFWX0J1 USnGEfSvG9EHTuDemG/AIt420qdYQhECDn3DNxM1

Object API (Amazon S3 compatible):
Go: https://docs.minio.io/docs/golang-client-quickstart-guide
Java: https://docs.minio.io/docs/java-client-quickstart-guide
Python: https://docs.minio.io/docs/python-client-quickstart-guide
JavaScript: https://docs.minio.io/docs/javascript-client-quickstart-guide

Drive Capacity: 596 MiB Free, 40 GiB Total
```

Image 6 (Minio terminal output in Microsoft Windows)

Web browser can be used for validation. **AccessKey** and **SecretKey** required to access and explore buckets and files in S3 server.

Managed Backup Service

Introduction

[CloudBerry Managed Backup Service](#) (MBS) is built from the CloudBerry Backup technology and designed to meet the needs of managed service providers and enterprise IT departments, providing reliable backup with central management and monitoring.

CloudBerry Managed Backup 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 (MSP). With CloudBerry Managed Backup any MSP, VAR or IT service company can rebrand 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 as it allows to manage and monitor cloud backup client across

multiple machines, control billing and licenses from one web-based control panel.

Storage account management

As our Minio cloud storage is S3 compatible we can add its 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 7.

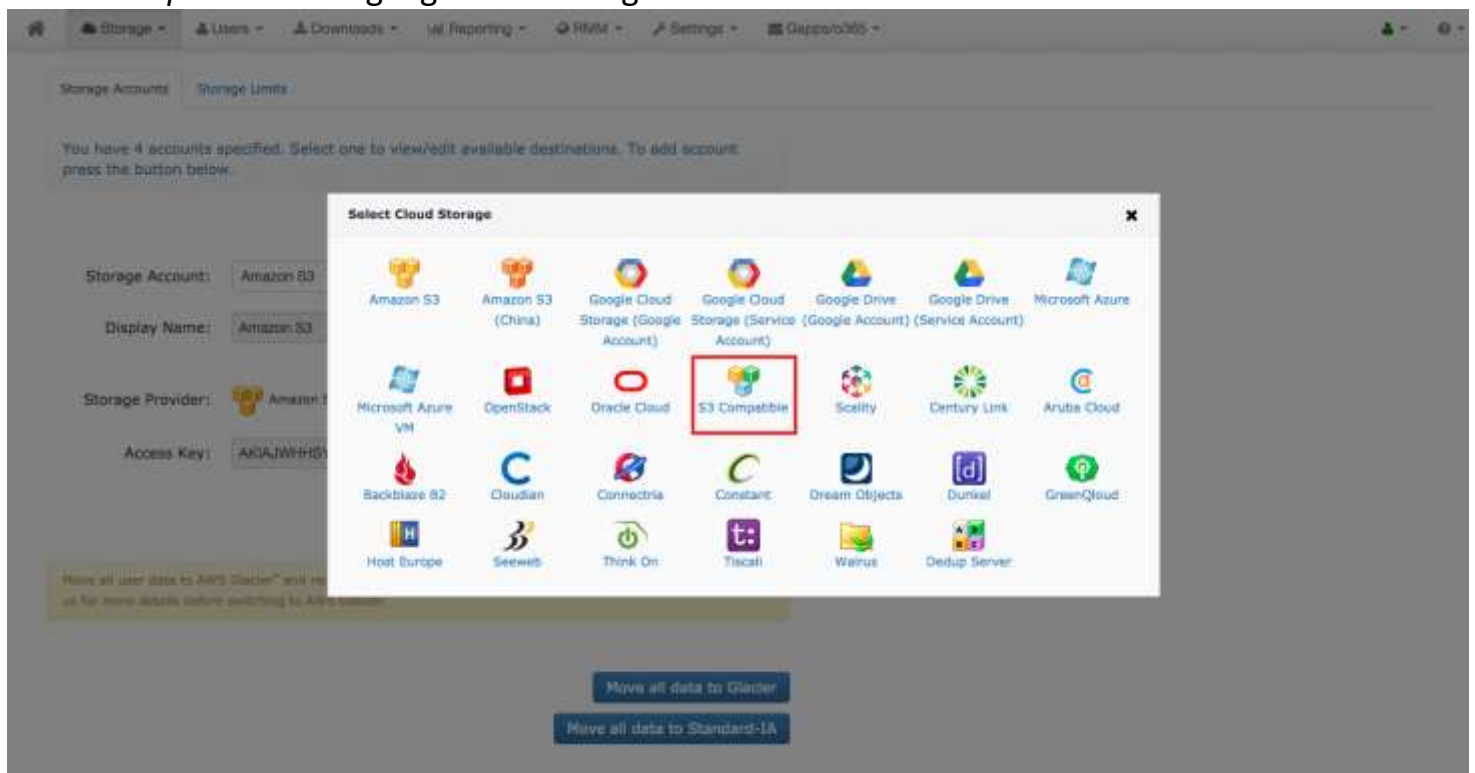


Image 7 (MBS S3 compatible storage account configuration)

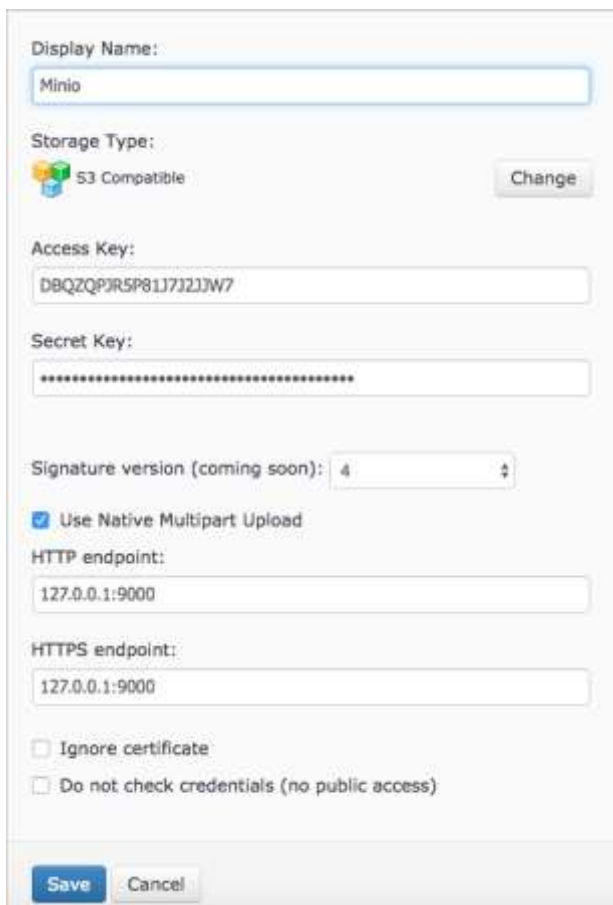
Now we need **endpoint** for HTTP and HTTPS (if you configure SSL certificate for the server), **AccessKey** and **SecretKey** to access to our storage.

NB!

1. As SSL is very important (securing data in transit) HTTPS is quite hot topic as well. We did not install SSL certificate on our Minio server as there are many certificates,

methods and configuration specifics. Example of Cert deployment can be found here: <http://docs.minio.io/docs/generate-let-s-encrypt-certificate-using-concert-for-minio>

2. Minio Cloud storage provides support of both S3 Signatures: v2 and v4. CloudBerry Backup and MBS portal provides support of both signature versions. 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 8 (MBS S3 compatible storage account configuration card)

About Minio

Minio is a lightweight object storage server released under Apache License v2.0. It is compatible with Amazon S3 cloud storage service. It is best suited for storing unstructured data such as photos, videos, log files, backups and container / VM images. Size of an object can range from a few KBs to a maximum of 5TB. Minio has advanced features like erasure

code, bit rot protection and lambda functions. Application developers often deploy Minio in a dockerized container and orchestrate it with Kubernetes.

Minio server is light enough to be bundled with the application stack, similar to Node JS, Redis and MySQL.

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: CloudBerry Lab stand-alone products usage

CloudBerry Explorer

Explorer supports many cloud providers including S3 compatible. Let's configure and start adding objects to our Minio S3 server.

Direct 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 9 (CloudBerry Explorer for Minio Cloud Storage)

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

- NB!**
3. As SSL is very important (securing data in transit) HTTPS is quite hot topic as well. We did not install SSL certificate on our Minio server as there are many certificates, methods and configuration specifics. Example of Cert deployment can be found here: <http://docs.minio.io/docs/generate-let-s-encrypt-certificate-using-concert-for-minio>

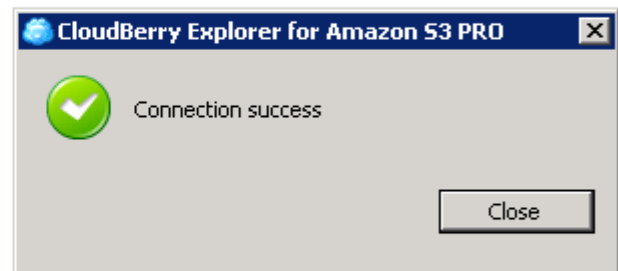
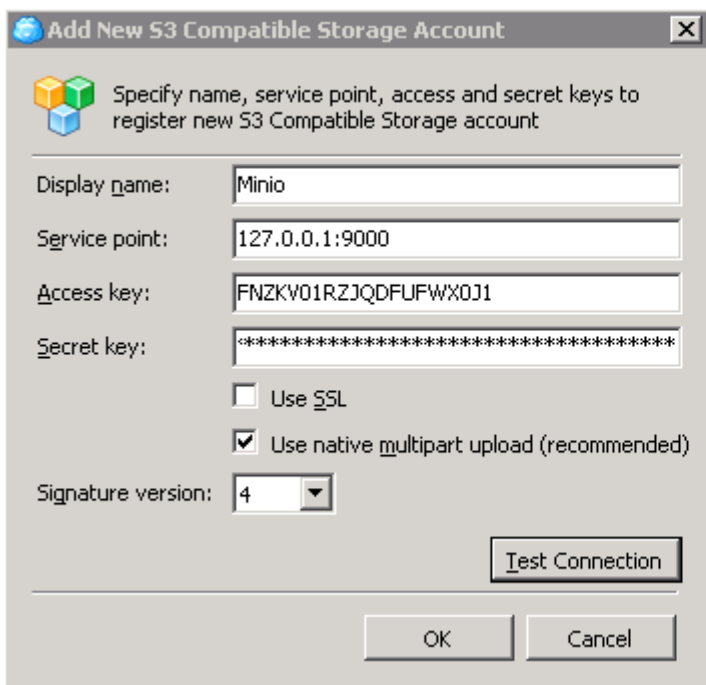


Image 10 (CloudBerry Explorer for Minio Cloud Storage configuration card)

Bucket and object management

Now we are ready to manage buckets and files in our Minio Cloud Storage. Explorer supports everything we may need to do with our buckets and files (create, delete, update, rename, move, download, upload etc).



Image 11 (CloudBerry Explorer for Minio Cloud Storage objects management)

Appendix B: 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>
- *SSL certificate for Minio Cloud Server (Certbot)* — <http://docs.minio.io/docs/generate-let-s-encrypt-certificate-using-concert-for-minio>