S3 storage

Introduction

What is S3?

Amazon S3 (Simple Storage Service) is a scalable object storage service used for storing and retrieving any amount of data at any time. It organizes data into containers called “buckets.” Each bucket can store an unlimited number of objects, which are the fundamental entities stored in S3.

Understanding S3 Bucket structure

- Buckets: These are the top-level containers in S3. Each bucket has a unique name and is used to store objects.
- Objects: These are the files stored in a bucket. Each object is identified by a unique key (or ID) within the bucket.
- Object Keys: While S3 does not have a traditional file system hierarchy, it uses a flat namespace. The / character in object keys is used to simulate a directory structure, making it easier to organize and manage objects. However, these are not actual directories but part of the object’s key.

S3 Endpoint Access

Accessing S3 is similar to accessing any other web service over HTTP, which most users are already familiar with. The endpoint URL follows the same structure as a typical web address, making it straightforward to understand and use.

An S3 endpoint address typically looks like this: https://dnsname.com/bucket-name/object-key

- Endpoint: https://dnsname.com
- Bucket Name: bucket-name
- Object Key: object-key

For example, if you have a bucket named my-bucket and an object with the key folder1/file.txt, the S3 URL would be: https://dnsname.com/my-bucket/folder1/file.txt

IAM Key Pairs

To access and manage your S3 resources securely, you will use IAM (Identity and Access Management) key pairs instead of a traditional login and password. An IAM key pair consists of an Access Key ID and a Secret Access Key. These keys are used to authenticate your requests to AWS services:

- Access Key ID: this is similar to a username
- Secret Access Key: this is similar to a password and should be kept secure.

Unlike a traditional login and password, different IAM key pairs can be attached to different sets of permissions defined in their policy files. These policies control what actions the keys are allowed to perform, enhancing security by ensuring that each key pair has only the necessary permissions for its intended tasks.

Request S3 bucket

To request S3 bucket you have two options. Either you want to attach it to an existing project or you want to create a new project with S3 storage.

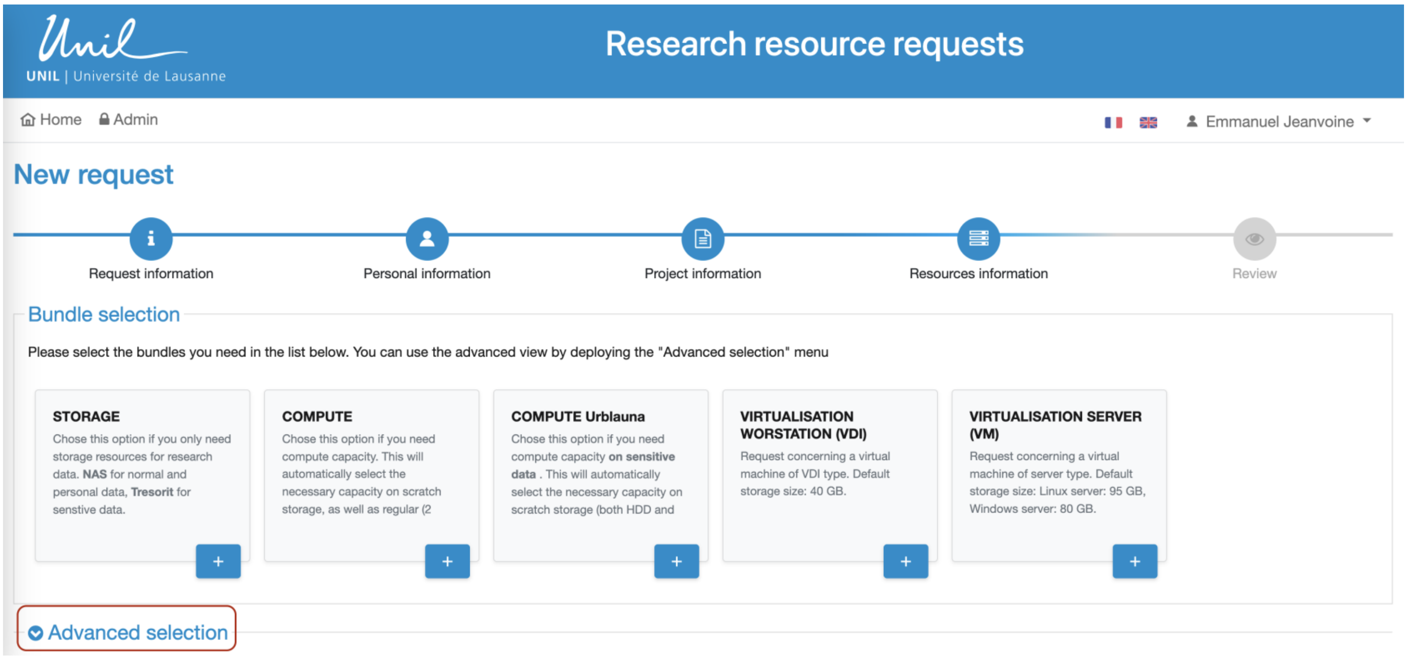
Attach S3 bucket to an existing project

In that case send an email to helpdesk@unil.ch (with subject starting with "DCSR add S3 bucket to project") and providing the following information:

- project name
- size of the bucket in GB
- access allowed from outside UNIL (in read only mode only?) or not
- if access is allowed from outside UNIL, should it be limited to a set of IP or network ranges?

Add S3 bucket to new project

When requesting a new project using the dedicated [web application](), on the resource information selection screen, click on advanced selection as follows:



Bundle selection

Please select the bundles you need in the list below. You can use the advanced view by deploying the "Advanced selection" menu

STORAGE
Chose this option if you only need storage resources for research data. **NAS** for normal and personal data, **Tresorit** for senstive data.
+

COMPUTE
Chose this option if you need compute capacity. This will automatically select the necessary capacity on scratch storage, as well as regular (2
+

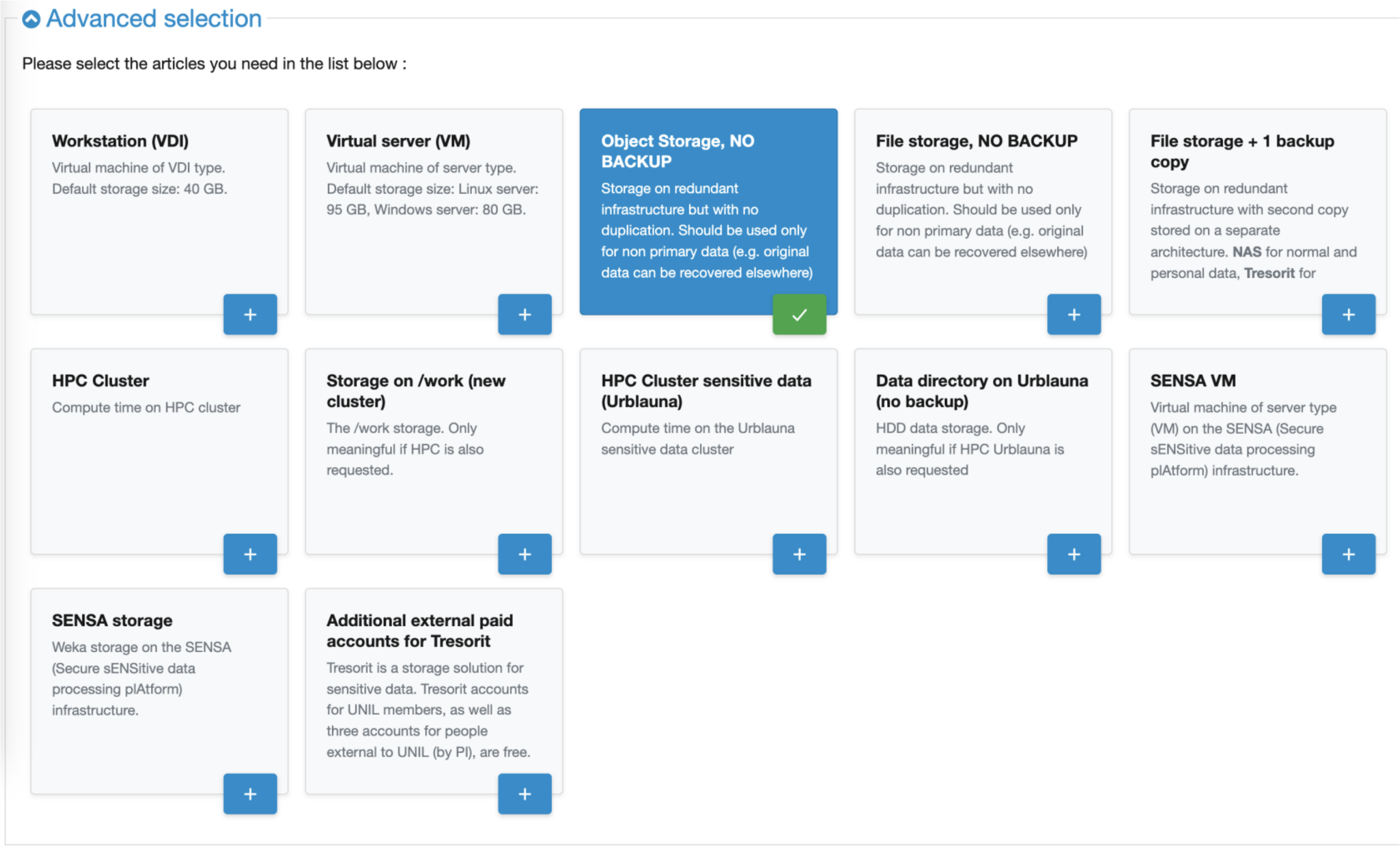
COMPUTE Urblauna
Chose this option if you need compute capacity **on sensitive data** . This will automatically select the necessary capacity on scratch storage (both HDD and
+

VIRTUALISATION WORSTATION (VDI)
Request concerning a virtual machine of VDI type. Default storage size: 40 GB.
+

VIRTUALISATION SERVER (VM)
Request concerning a virtual machine of server type. Default storage size: Linux server: 95 GB, Windows server: 80 GB.
+

Advanced selection

Then choose "Object Storage, NO BACKUP" article :



Advanced selection

Please select the articles you need in the list below :

Workstation (VDI)
Virtual machine of VDI type. Default storage size: 40 GB. +

Virtual server (VM)
Virtual machine of server type. Default storage size: Linux server: 95 GB, Windows server: 80 GB. +

Object Storage, NO BACKUP
Storage on redundant infrastructure but with no duplication. Should be used only for non primary data (e.g. original data can be recovered elsewhere) ✓

File storage, NO BACKUP
Storage on redundant infrastructure but with no duplication. Should be used only for non primary data (e.g. original data can be recovered elsewhere) +

File storage + 1 backup copy
Storage on redundant infrastructure with second copy stored on a separate architecture. **NAS** for normal and personal data, **Tresorit** for +

HPC Cluster
Compute time on HPC cluster +

Storage on /work (new cluster)
The /work storage. Only meaningful if HPC is also requested. +

HPC Cluster sensitive data (Urblauna)
Compute time on the Urblauna sensitive data cluster +

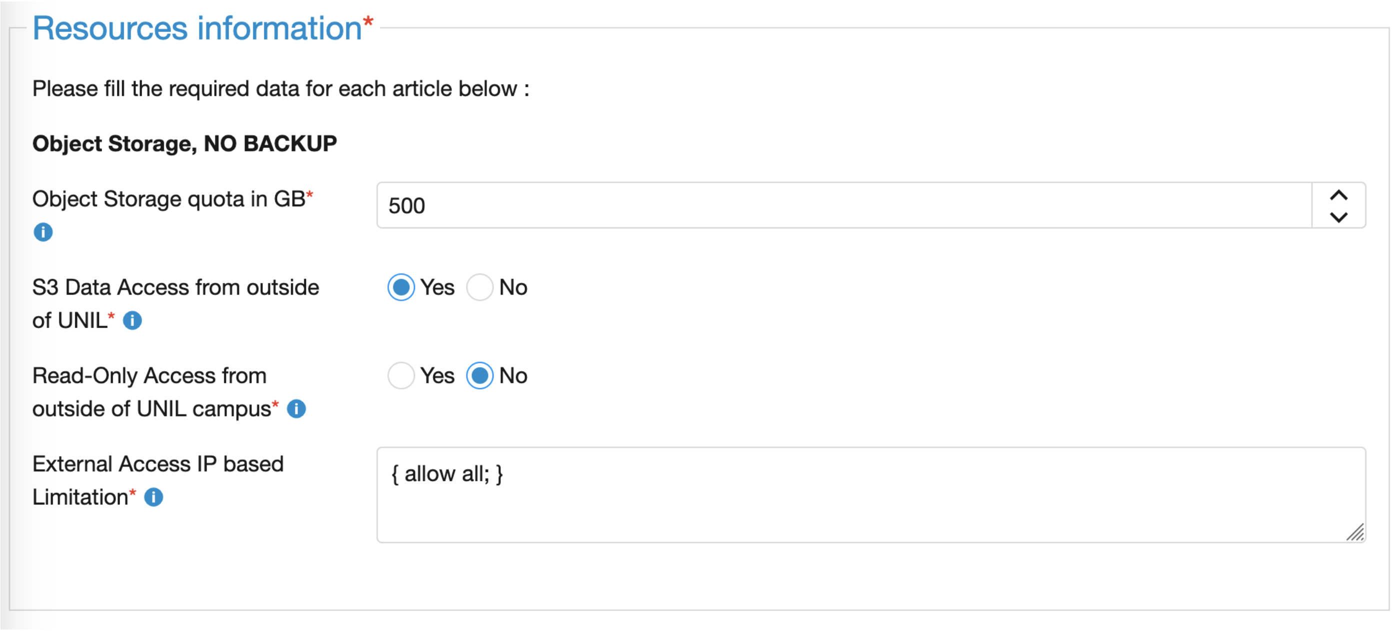
Data directory on Urblauna (no backup)
HDD data storage. Only meaningful if HPC Urblauna is also requested +

SENSA VM
Virtual machine of server type (VM) on the SENSA (Secure sENSitive data processing plAtform) infrastructure. +

SENSA storage
Weka storage on the SENSA (Secure sENSitive data processing plAtform) infrastructure. +

Additional external paid accounts for Tresorit
Tresorit is a storage solution for sensitive data. Tresorit accounts for UNIL members, as well as three accounts for people external to UNIL (by PI), are free. +

Finally, specify your requirements:



Resources information*

Please fill the required data for each article below :

Object Storage, NO BACKUP

Object Storage quota in GB* 500

S3 Data Access from outside of UNIL* ● Yes ○ No

Read-Only Access from outside of UNIL campus* ○ Yes ● No

External Access IP based Limitation* { allow all; }

Software to access S3 bucket

From Curnagl cluster (command line tools)

Rclone

awscli

From your laptop or a workstation

Command line tools

Following the official documentation, you can install Rclone (https://rclone.org/install/) or awscli (https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html) on your laptop/workstation.

Then both tools can be used as described in the above part dedicated to use from Curnagl cluster.

Cyberduck

Cyberduck can be installed from https://cyberduck.io/download/.

Share files from a bucket with presign keys

Purpose

AWS presigned URLs (or presign keys) are used to grant temporary access to objects in Amazon S3 without requiring the recipient to have AWS security credentials. Here are the main purposes:

- **Download Access**: You can generate a presigned URL to allow someone to download an object from your S3 bucket without needing their own AWS credentials
- **Upload Access**: Similarly, you can create a presigned URL to permit someone to upload a specific object to your S3 bucket
- **Time-Limited Access**: The access provided by a presigned URL is time-limited, meaning it will expire after a specified duration

This is particularly useful for sharing files securely or allowing temporary uploads without exposing your AWS credentials.

Warning: only files can be shared this way, so if you want to share a folder, you have to create an archive of this folder first.

Prerequisite

The bucket has to be opened on the outside, if it's not the case, send an email to helpdesk@unil.ch with the subject starting with "DCSR S3 bucket".

Create a presign key with `aws-cli`

You can either install awscli on your laptop since it's a simple Python package, or use it from the cluster (there is a module called `awscli-v2` that you can simploy load).

Configuration

If it's not configured yet, you can run `aws configure` command. This will ask you for:

- AWS Access Key ID: you have to provide the read/write access key corresponding to your bucket

- AWS Secret Access Key: you have to provide the read/write secret key corresponding to your bucket
- Default region name: you can put `us-east-1`
- Default output format: you can put `json`

This will create two files:

- ~/.aws/config
- ~/.aws/credentials

Presign key creation

Let's consider that you want the file `to_share/important_file.gz` from the bucket `recn-fac-fbm-dep-greatpi-data`, you can use the following command:

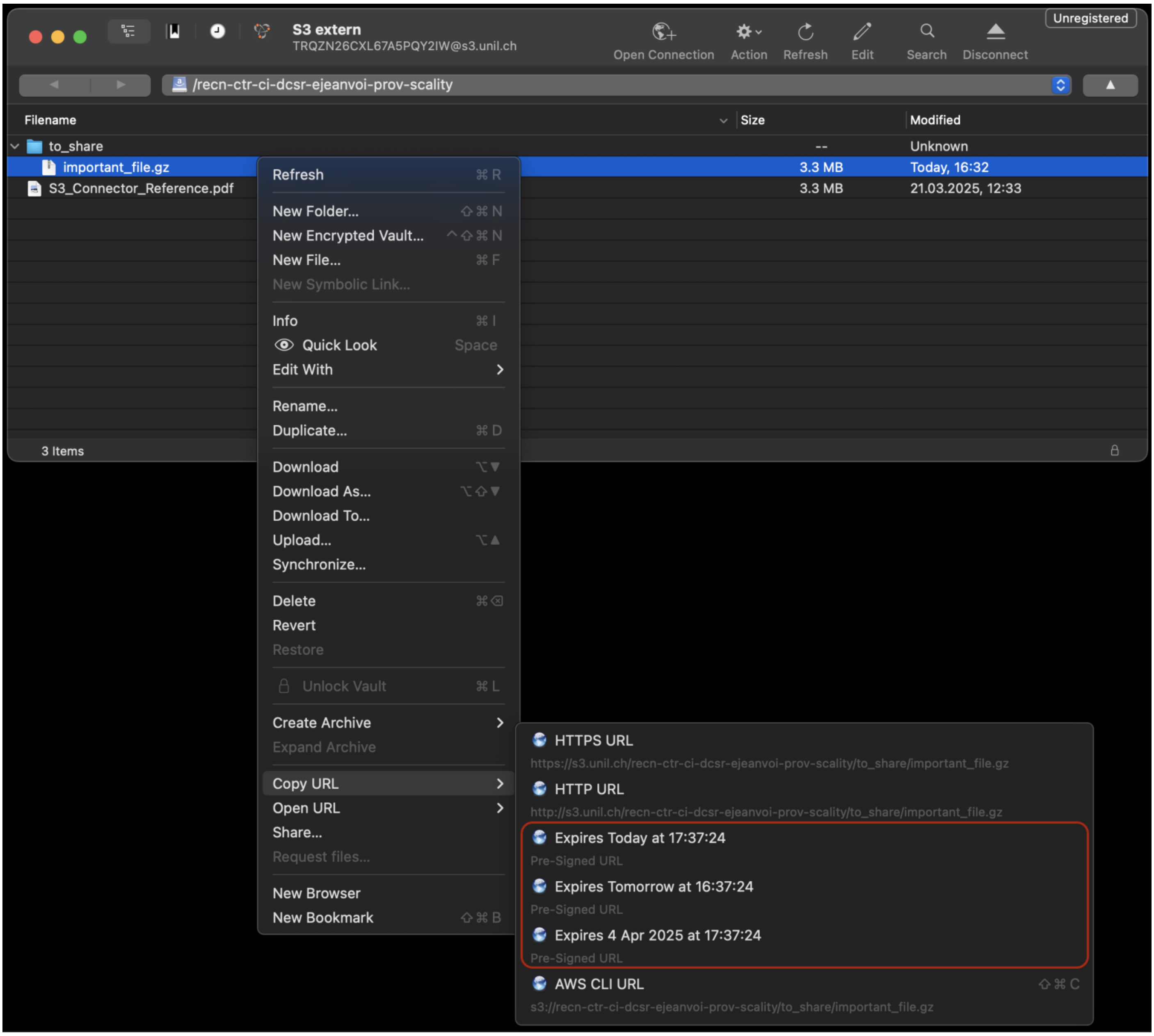
```
aws --endpoint-url=https://s3.unil.ch s3 presign \
    s3://recn-fac-fbm-dep-greatpi-data/to_share/important_file.gz \
    --expires-in 604800
```

The value defined with --expires-in parameter is the validity of the link expressed in seconds. The maximum validity is 7 days.

This command will return a link that can be shared with your external collaborator.

Create a presign key with Cyberduck

Once your connection to the bucket is configured with Cyberduck, you can browse it. Then right click to file you want to share, then choose "Copy URL", and choose one of the 3 expiration options (1 hour, 1 day, 1 week). Finally, just paste the result somewhere to get the link.



Unregistered
S3 extern
TRQZN26CXL67A5PQY2IW@s3.unil.ch
Open Connection
Action
Refresh
Edit
Search
Disconnect
/recn-ctr-ci-dcsr-ejeanvoi-prov-scality

Filename	Size	Modified
to_share	--	Unknown
important_file.gz	3.3 MB	Today, 16:32
S3_Connector_Reference.pdf	3.3 MB	21.03.2025, 12:33

3 Items

HTTPS URL
https://s3.unil.ch/recn-ctr-ci-dcsr-ejeanvoi-prov-scality/to_share/important_file.gz

HTTP URL
http://s3.unil.ch/recn-ctr-ci-dcsr-ejeanvoi-prov-scality/to_share/important_file.gz

Expires Today at 17:37:24
Pre-Signed URL

Expires Tomorrow at 16:37:24
Pre-Signed URL

Expires 4 Apr 2025 at 17:37:24
Pre-Signed URL

AWS CLI URL ⇧ ⌘ C
s3://recn-ctr-ci-dcsr-ejeanvoi-prov-scality/to_share/important_file.gz