

大阪大学 サイバーメディアセンター
利用者講習会

ONION活用 講習会

2024年 6月26日
日本電気株式会社
文教・科学ソリューション統括部



\Orchestrating a brighter world

NECは、安全・安心・公平・効率という社会価値を創造し、
誰もが人間性を十分に発揮できる持続可能な社会の実現を目指します。

目次

1. ユースケース
2. ONIONの概要
3. 基本的な使い方
 - ONION-file
 - ONION-object
 - ExaScaler

1 3 : 3 0 分開始

4. サブシステム間連携
5. S3 APIの概要
6. S3 APIの利用
7. 情報入手先

1 4 : 4 0 分再開

1. ユースケース

ユースケース

ONIONは、大阪大学サイバーメディアセンターが提供するストレージサービスで、下記のような特徴を持っています。

エクスペリエンス(UX)の向上

- ◆ WEBブラウザ対応による簡単なファイル操作
- ◆ モバイル対応により、外出先からのデータ参照

データ共有と連携の柔軟性

- ◆ スパコンユーザ、及び未登録のユーザ・研究者とデータを共有
- ◆ 豊富なクラウド対応アプリを使ってデータ連携

スパコンと一体のストレージ

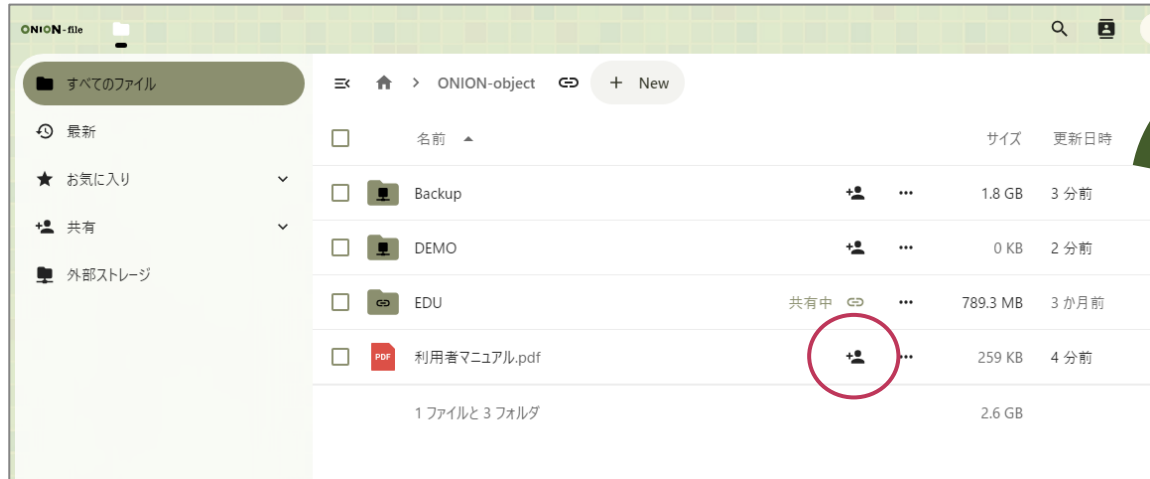
- ◆ スパコン(SQUID)直結の高速ストレージにデータを簡単に転送
- ◆ スパコンの解析結果を即座に確認

まずONIONの**特徴的な使い方(ユースケース)**をご紹介します



研究データのブラウザ操作

ブラウザから直感的なデータ操作、共有が可能なオンラインストレージ



① ブラウザ上でデータ操作

② データを共有



③ 共有データにアクセスする

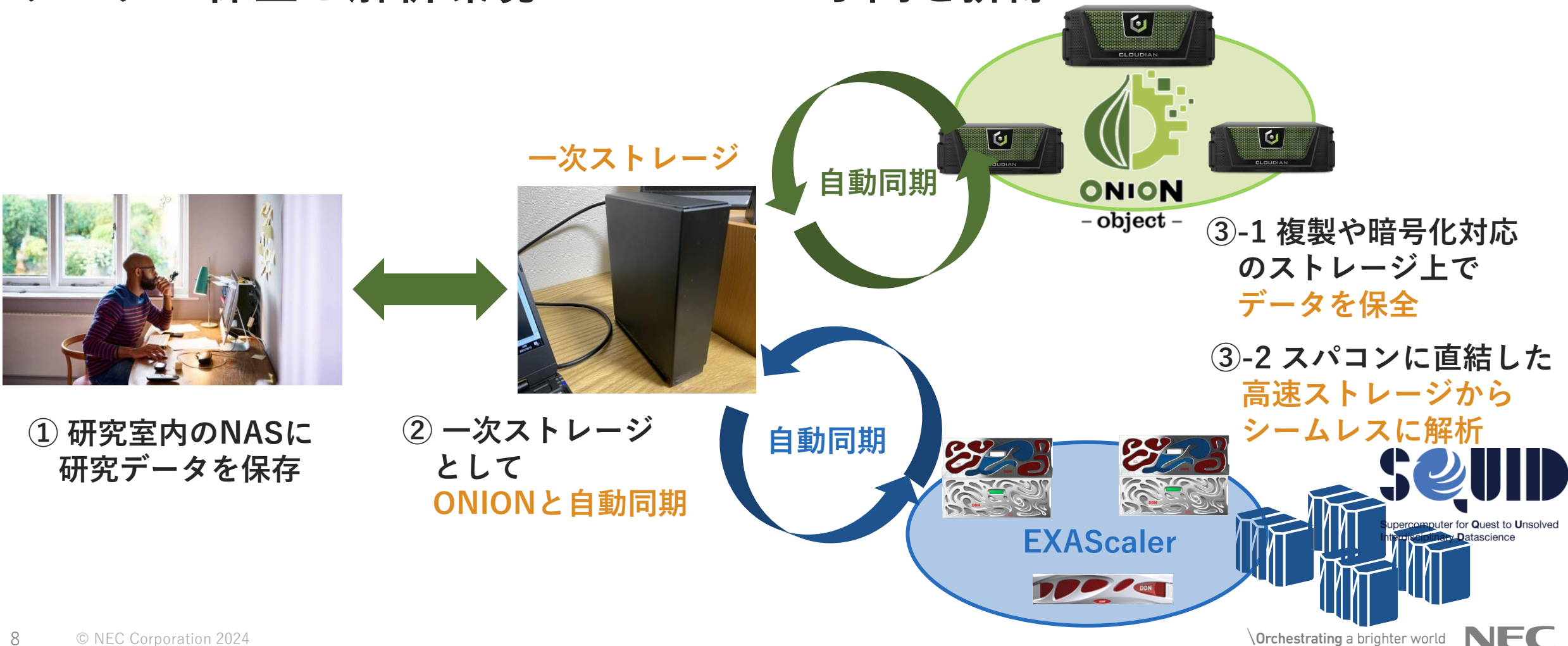
スマートフォン連携

スパコン解析結果をスマートフォンで即時確認 & その場でデータ共有



一次ストレージ連携

研究室やデスクトップ脇の**一次ストレージ**を**ONION**と**自動同期**
データの保全と解析環境へのコピーの手間を排除



GitLab 連携

GitLab のバックエンドとしてONIONを活用 重要資産であるソースコードのデータを保護



```
gitlab_rails['backup_upload_connection'] = {  
  'provider' => 'AWS',  
  'region' => 'eu-west-1',  
  'aws_access_key_id' => 'AKIAKIAKI',  
  'aws_secret_access_key' => 'secret123'  
  # If using an IAM Profile, don't configure aws_access_key_id & aws_secret_access_key  
  # 'use_iam_profile' => true  
}  
gitlab_rails['backup_upload_remote_directory'] = 'my.s3.bucket'
```

2. ONIONの概要

ONIONのシステム概要

ONIONは下記の特徴を持ったサブシステムが一体となったストレージサービスです。

■ WEBブラウザ/モバイル対応ストレージ

- 利便性やエクスペリエンス向上のための、WEBブラウザやモバイルアプリで利用可能なストレージサービス
- GUIからのデータのアップロード/ダウンロード、データ共有が可能

ONION-file

■ 複製や暗号化に対応したオブジェクトストレージ

- 複製や暗号化など、柔軟にポリシーを指定可能なオブジェクトストレージ

ONION-object

■ スパコンと一体となった高速ストレージ

- スパコンの解析環境として利用されるLustre による高速ストレージ
- SCP/SFTP/NFS/CIFS/S3API などデータ転送方式を拡充

EXAScaler

■ S3 API対応による クラウド連携対応ストレージ

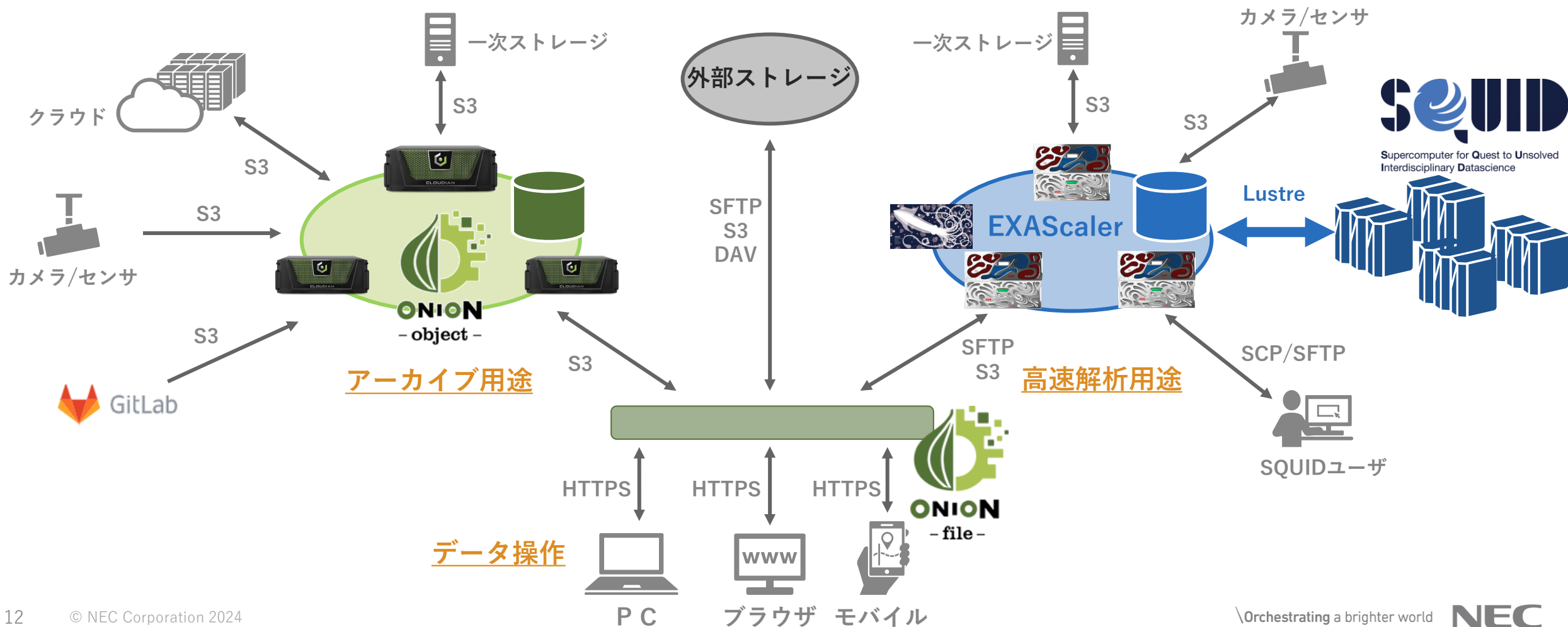
- ストレージはS3 API に対応しており、クラウド連携のエコシステムを利用可能

ONION-object

EXAScaler

各サブシステムの役割

ONIONでは、データがONION-object(アーカイブ)とExaScaler(高速解析)に格納され、ONION-file から多様なデータ操作を可能とします。



ONIONの利用申請

SQUIDの一部であるEXAScalerと ONION-fileは、**SQUIDの利用申請**をお願いします。**ONION-object**は別途申し込みが必要となります。



ONION-file は、NextCloudをベースとしたWEBストレージです。
GUIによる簡単なファイル操作や、ファイル共有が可能です。

<https://onionweb.hpc.cmc.osaka-u.ac.jp>



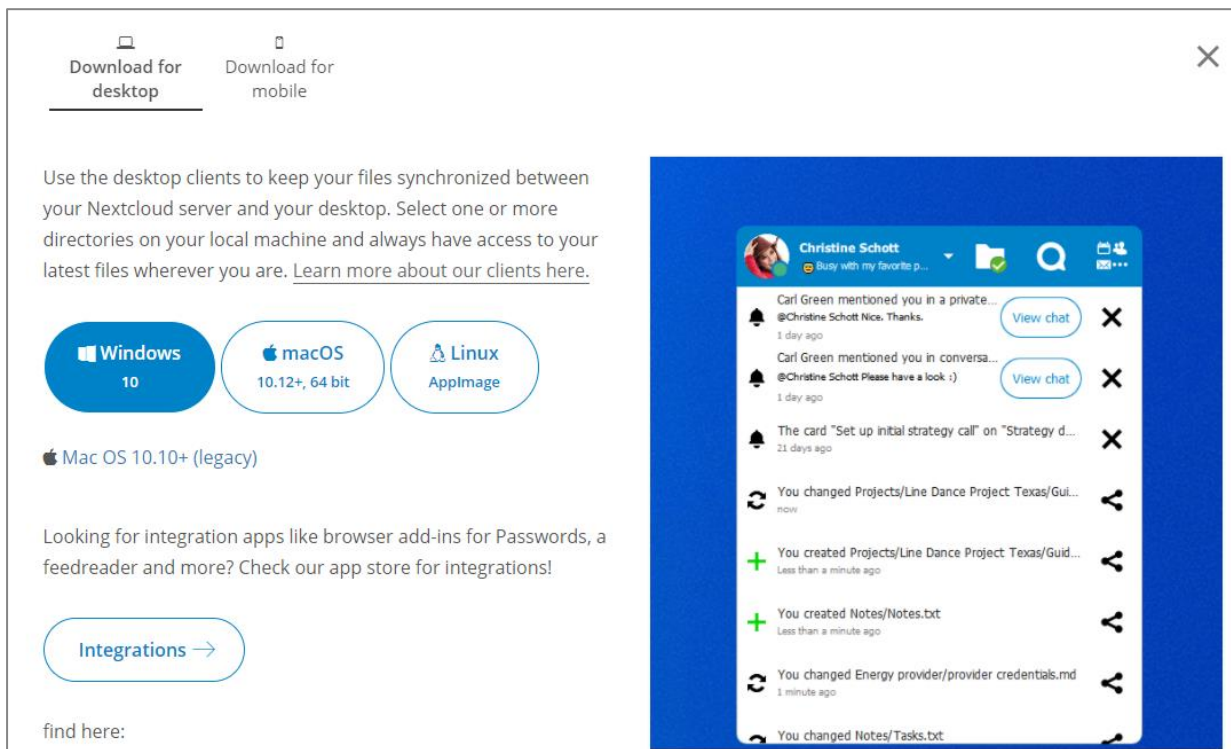
The screenshot displays the ONION-file web interface. On the left, the login page features the ONION-file logo, input fields for 'ユーザー名またはメールアドレス' (Username or email address) and 'パスワード' (Password), a 'ログイン' (Login) button, and a link for 'パスワードをお忘れですか?' (Forgot your password?). Below the login section, it states 'デバイスを使ってログインします。' (Log in using your device).

On the right, the file management interface is shown for the 'SQUID_HOME' directory. It includes a sidebar with navigation options: 'すべてのファイル' (All files), '最新' (Recent), 'お気に入り' (Favorites), '共有' (Sharing), 'URLリンクで共有中' (Sharing via URL link), '共有を解除しました' (Unshared), '保留中の共有' (Pending sharing), '外部ストレージ' (External storage), and 'ゴミ箱' (Trash). The main area lists files and folders with columns for '名前' (Name), 'サイズ' (Size), and '更新日時' (Update time).

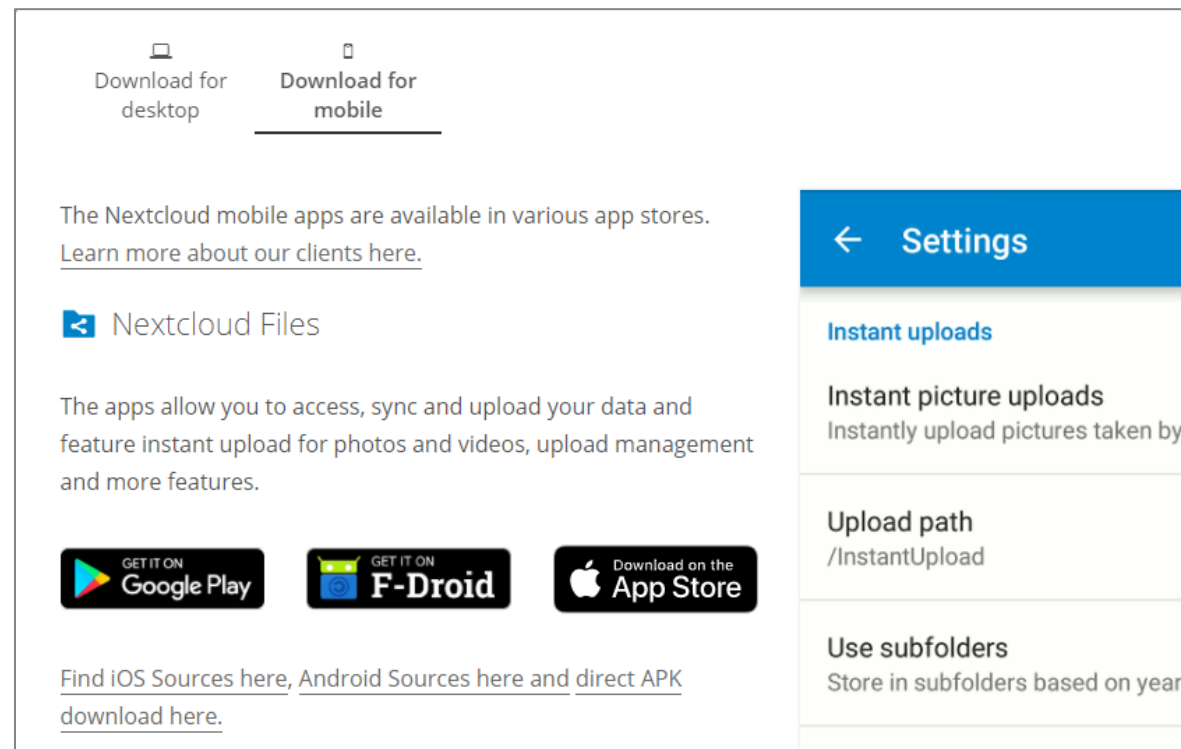
名前	サイズ	更新日時
20210723	1.4 MB	4ヶ月前
20211108	43.3 MB	2分前
cpu	保留中	数秒前
demo	5 KB	2時間前
gpu	360.8 MB	3ヶ月前
picture	3.4 MB	2分前

At the bottom of the file list, it indicates '6 個のフォルダー' (6 folders) and '保留中' (Pending).

ONION-file(NextCloud) は、コミュニティで開発された多様なクライアントアプリが利用可能です。



デスクトップクライアント



モバイルアプリケーション

<https://nextcloud.com/install/#install-clients>

ONION-objectは、Cludian HyperStore で構成されたオブジェクトストレージで、768TBの実効容量を有します。EC42(イレジャーコーディング4+2)の符号化方式でデータを保持します。

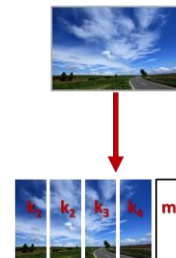


Cludian HyperStore Appliance 1610

項目	内容
実効容量	768 TB
データ保護方式	イレジャー・コーディング (データチャンク:4+パリティチャンク:2)

イレジャー・コーディング (EC) 方式

大きなデータや格納効率を確保する場合



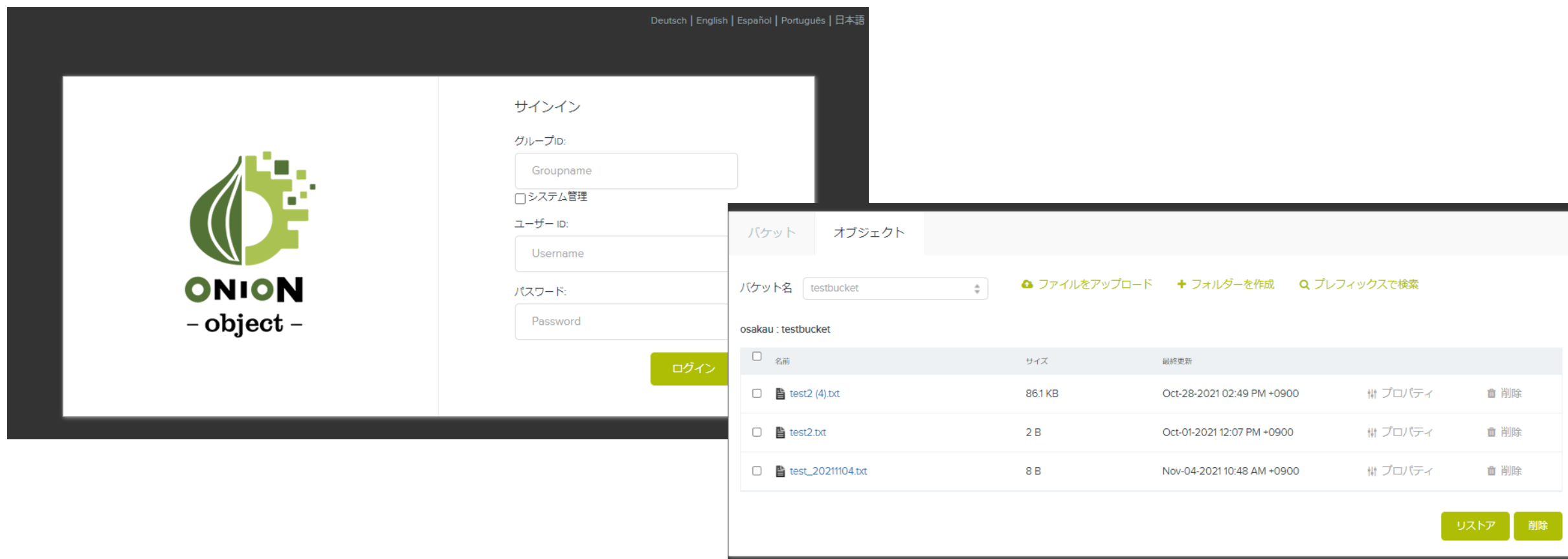
4 + 2 コーディング = 6/4または
1.5倍の格納領域

■ イレジャーコーディング方式(EC42)

- データを4KB単位のブロックサイズに分割し、4つのデータチャンクと2つのパリティチャンクに保存
- 大きなデータを容量効率良く保存する用途に向いている
- 符号化処理を毎回行うため、**更新が頻発する利用には不向き。アーカイブ用途。**

ONION-objectは、AWSと高い互換性をもつS3 APIが利用可能なストレージです。
簡易操作を行うための、GUIも備えています。

<https://onionportal.hpc.cmc.osaka-u.ac.jp:8443/>



The screenshot displays the ONION-object web interface. On the left is the login page with the ONION-object logo and fields for Groupname, Username, and Password, along with a 'ログイン' (Login) button. On the right is a preview of the 'testbucket' contents, showing a list of files with their names, sizes, and last update times.

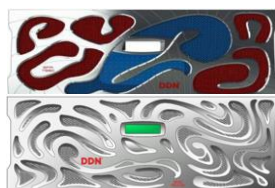
名前	サイズ	最終更新
test2 (4).txt	86.1 KB	Oct-28-2021 02:49 PM +0900
test2.txt	2 B	Oct-01-2021 12:07 PM +0900
test_20211104.txt	8 B	Nov-04-2021 10:48 AM +0900



EXAScalerは、DDN 社製のLustre ファイルシステムです。高速大容量のストレージとして、スパコン(SQUID)の主ファイルシステムとして動作しています。



NVMe領域 : DDN ES400NVX



HDD領域 : DDN ES7990X

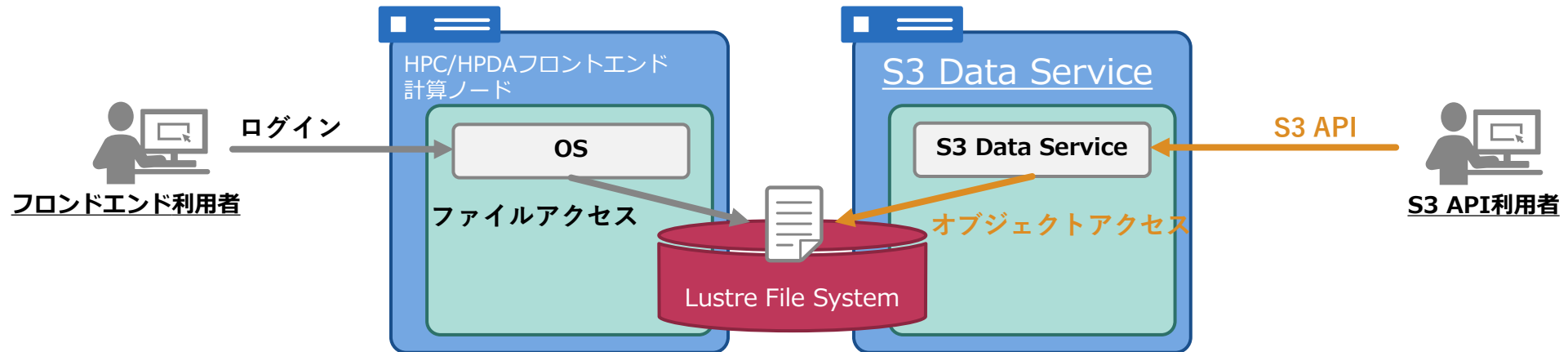
項目	内容
実効容量(HDD)	20PB
実効容量(NVMe)	1.2PB
最大合計inode数	約 88億個
最大想定実効スループット(HDD)	160 GB/s 以上
最大想定実効スループット(NVMe)	Write : 160 GB/s 以上 Read : 180 GB/s 以上

■ Lustre ファイルシステム

- 多数のMDSとOSSが一体となって動作する並列ファイルシステムで、多数のノードからの一斉アクセスで性能を発揮
- 大規模なIOに対しても高速なアクセスが可能であることから、スパコンなどの**解析用途に利用される**



SQUIDのEXAScalerは、**S3 APIに対応**しています。Lustre ファイルシステムにS3 APIからアクセスするためのゲートウェイとして、**DDN S3 Data Service (S3DS)** が稼働しています。



DDN S3 Data Service x4

■ S3 Data Service

- Lustre上のファイルに**S3 APIからのオブジェクト操作を行うようにする**ためのゲートウェイ
- **ONION-object に比べ、S3 APIの互換性は劣る**が、オブジェクトのアップロード/ダウンロードに必要なAPIに対応

3. 基本的な使い方

基本的な使い方

ONIONのサブシステム毎に基本的な使い方をご説明します。

■ ONION-file

- ログイン、ファイルのアップロード/ダウンロード
- データ共有(グループ内共有、URLリンク共有)
- 外部ストレージ連携

■ ONION-object

- ログイン
- バケット作成、アクセスキー作成
- オブジェクトのアップロード/ダウンロード

✓ グループ管理者操作は、対象者が限られるため本日の内容上は省略しています。

■ EXAScaler

- ログイン
- バケット作成、アクセスキー作成

✓ オブジェクト操作について、
【6. S3 APIの利用】にて説明します。

ONION-fileの使い方：ログイン

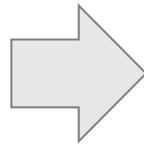
ONION-file ヘブラウザからのログイン手順です。

(1) ONION-file のログイン画面にアクセスします。

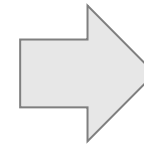
<https://onionweb.hpc.cmc.osaka-u.ac.jp>



The login screen for ONION-file. It features the ONION-file logo at the top. Below the logo, there are two input fields: 'ユーザー名またはメールアドレス' (Username or email address) and 'パスワード' (Password). A 'ログイン' (Login) button with a right arrow is positioned below the password field. At the bottom, there is a link 'パスワードをお忘れですか?' (Forgot your password?) and a note 'デバイスを使ってログインします。' (Log in using your device).



The TOTP (Authenticator app) screen. It displays the ONION-file logo and the title 'TOTP (Authenticator app)'. Below the title, there is an illustration of a smartphone and the text 'デバイスの二段階認証アプリで認証コードを取得してください。' (Please obtain the authentication code using the two-step authentication app on your device). A '認証コード' (Authentication code) input field is shown, followed by a '送信' (Send) button. At the bottom, there is a 'ログインをキャンセル' (Cancel login) link.



(2) **SQUID**の利用者番号とパスワード、及び
認証コードを入力してログインします。



The ONION-file file browser interface. It shows a sidebar with navigation options: 'すべてのファイル' (All files), '最新' (Latest), 'お気に入り' (Favorites), '共有' (Sharing), and '外部ストレージ' (External storage). The main area displays a list of files and folders. A folder named 'SQUID_HOME' is highlighted with an orange box. The interface also includes a search bar at the top right and a list of sharing options on the left.

■ SQUID_HOME

- 標準状態のONION-fileからは、**EXAScaler上のHome 領域のサブフォルダのみ**参照可能です。
- SQUIDフロントエンドからは、下記のパスで参照可能です。
`/sqfs/home/【利用者番号】/OnionWeb`

ONION-file ではログインに二要素認証を必須としています。

■ 概要

- 重要情報を守るため、ONION-file ログイン時に、**二要素認証が必須となっています。**
- 二要素目は、TOTP Authenticatorを使用します。Google Authenticator 等の対応アプリを準備ください。
- 利用方法の詳細は、下記のページを参照ください。

http://www.hpc.cmc.osaka-u.ac.jp/system/manual/squid-use/transfer_web/#login

■ ログインに必要な情報

- 利用者番号、パスワード：SQUID利用申請に伴い発行される利用者番号とパスワードです。
- 認証コード：ONION-file 初回ログイン時に設定する、ワンタイムパスワードです。
※ ONION-file 専用のもので、SQUIDフロントエンド、ポータルサイトとは**別コードになります。**

ONION-fileの使い方：ファイル操作

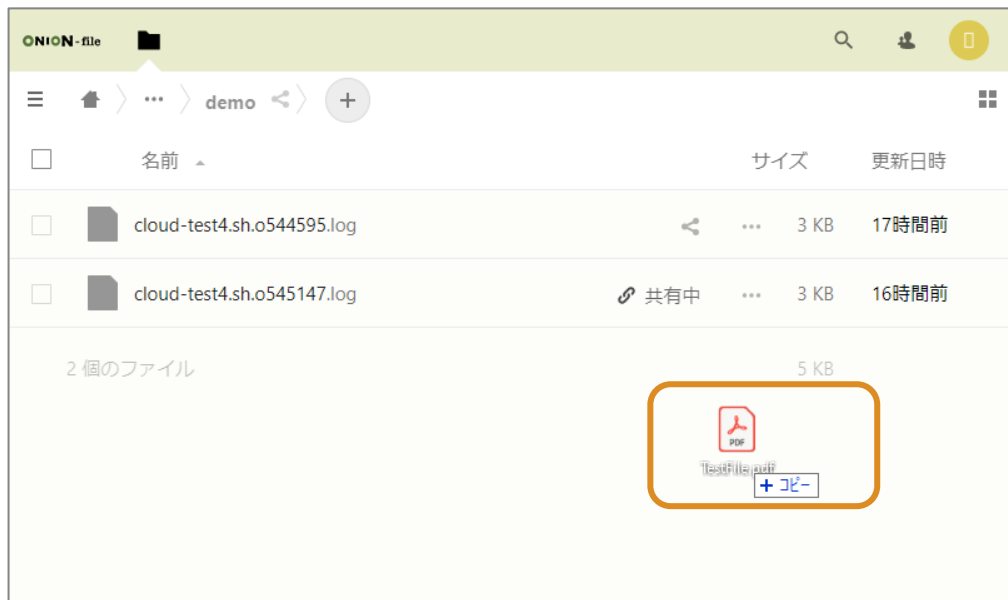


ファイルのアップロード/ダウンロード手順です。

(1) SQUID_HOME等のディレクトリ内に移動します。

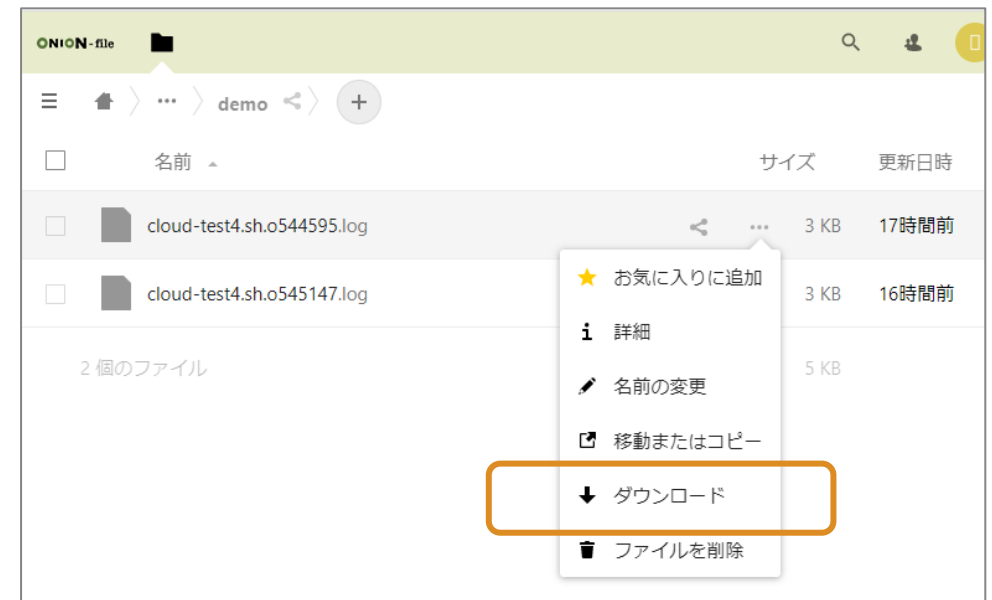
✓ アップロードの場合

アップロードしたいディレクトリに移動し、**ファイルのドラッグ&ドロップでアップロード**が可能です。



✓ ダウンロードの場合

ダウンロードしたいファイルの**メニュー(...)**の部分をクリックし、**[ダウンロード]**を選択します。



■ 一括操作

- 左のチェックボックスを選択した後、上部メニューの[アクション]より、一括操作も可能です。

ONION-fileの使い方：ファイル共有



グループ内でのファイル共有手順です。

(1) 共有したいファイルの共有メニュー(◀, +人)を開きます。

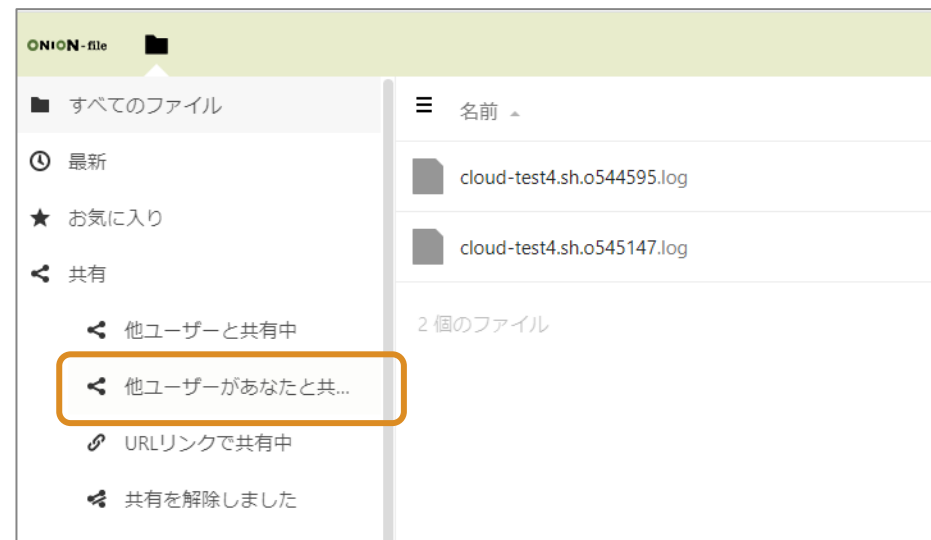
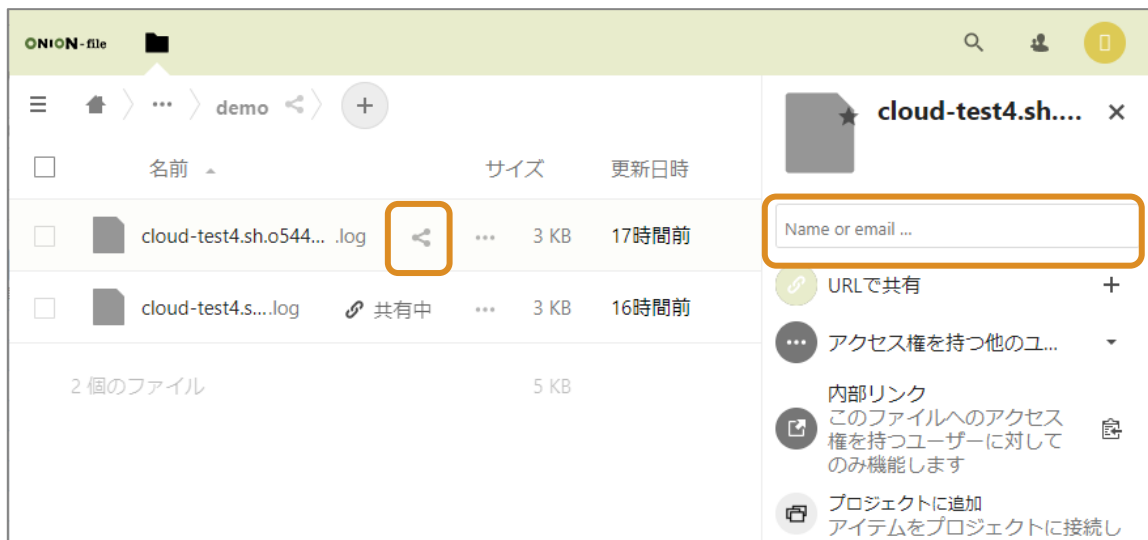
✓ **グループ内の別ユーザに共有する場合**

検索BOXに共有相手の利用者番号を入力します。

検索結果に表れたユーザを選択すると、許可リストに追加されます。

✓ **共有されたファイルを参照する**

共有された側は、**[共有]-[他ユーザがあなたと共有中]**のメニューから該当ファイルを参照可能です。



■ 共有範囲

- 利用者番号指定の共有範囲は、**同じグループに所属するユーザに限定**されています。

ONION-fileの使い方：ファイル共有



URLリンクでのファイル共有手順です。システム外ユーザともデータ共有可能です。

(1) 共有したいファイルの共有メニュー(🔗, +👤)を開きます。

✓ URLリンクで別ユーザに共有する場合

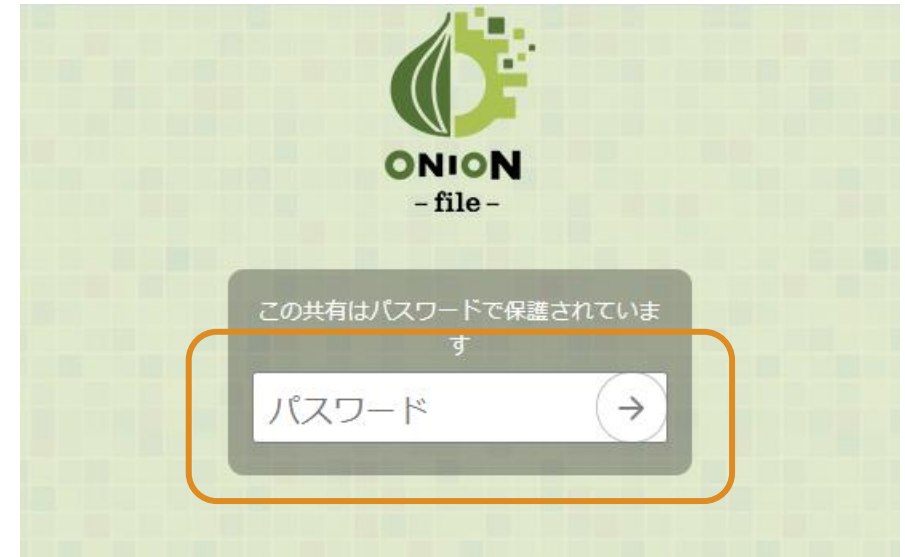
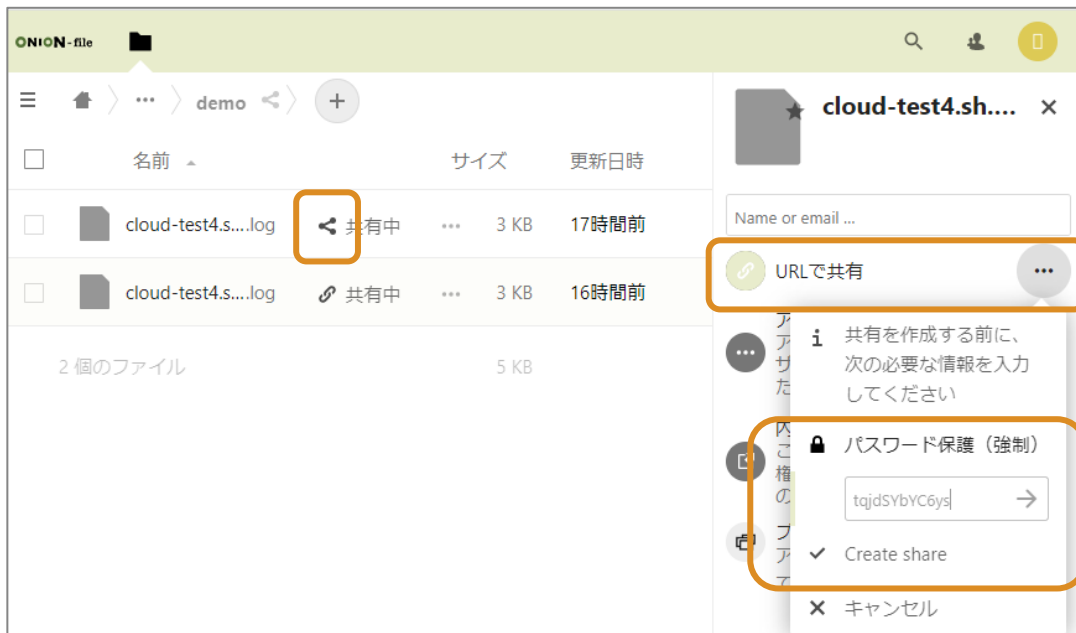
[URLで共有]のメニューをクリックします。

パスワード設定後、[共有を作成(Create share)]を押します。

✓ 共有されたファイルを参照する

共有時に作成されたURLにブラウザアクセスします。

パスワード入力求められるため、設定したパスワードを入力してアクセスします。



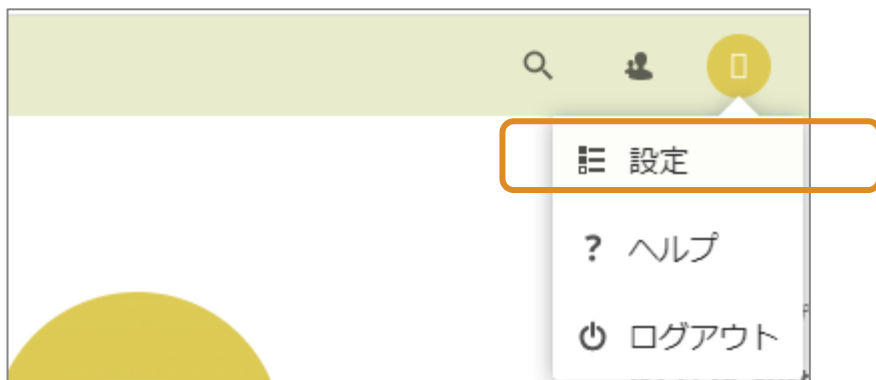
■ URLリンク制限

- URLリンクのパスワードは、**必ず12文字以上を指定してください。**

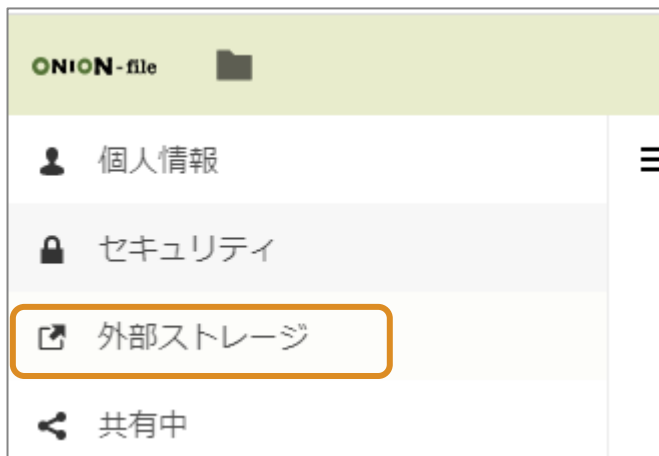
ONION-fileの使い方：外部ストレージ連携

ONION-file 上で、システム外のユーザストレージを接続することが可能です。

(1) 右上アイコンから**[設定]**を選択します。



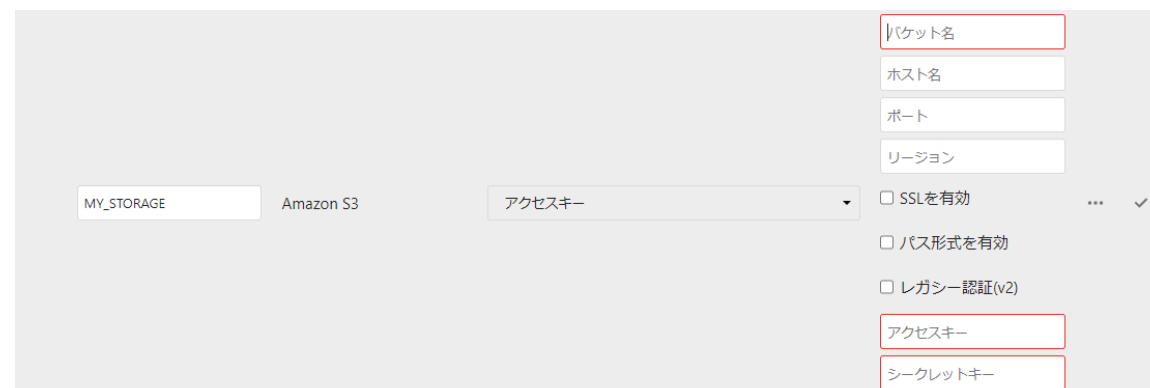
(2) 左メニューから**[外部ストレージ]**を選択します。



(3) 外部ストレージの項目から[フォルダー名]を入力し、[ストレージを追加]で追加したいタイプを選びます。



(4) タイプ毎に必要な情報を入力して、**外部ストレージを追加**します。(以下はAmazon S3の例)



ONION-fileの使い方：外部ストレージ連携



OCTOPUSのホーム領域を外部ストレージとして追加する例です。

(1) [フォルダー名]に任意の名前を入力し、
[ストレージを追加]で**SFTPを選択**します。

(2) 下記の項目を入力しチェック(レ)ボタンを押します。

項目	内容
フォルダー名	(任意) 例：OCTOPUS_HOME
外部ストレージ	SFTP
認証	ユーザー名とパスワード
ホスト	octopus.hpc.cmc.osaka-u.ac.jp
ルート	/octfs/home/ 【利用者番号】
ユーザ名	【利用者番号】
パスワード	【パスワード】

(3) 接続に成功すると、緑色のチェックが入ります。

(4) トップ画面に戻ると、新しいストレージが追加され、
アクセスできるようになります。

※ OCTOPUSは、次期システムに向けたサービス停止中です。SFTPを提供するシステムでは、同じ手順で接続可能です。

ONION-fileの使い方：補足事項

ONION-file 利用に関する補足事項です。

■ アカウソトの管理

- ・ 外部ストレージ連携機能で、ストレージをつなげるとパスワード漏洩時のリスクが高まります。パスワードの定期的な変更や、複雑なパスワードの設定などアカウント管理にはご注意ください。

■ 更なるセキュリティ強化方法

- ・ セキュリティ強化のために、2022年4月よりONION-file 上で二要素認証が必須となりました。

ONION-objectの使い方：ログイン



ONION-object の管理画面へのログイン手順です。

- (1) ONION-object のログイン画面にアクセスします。
<https://onionportal.hpc.cmc.osaka-u.ac.jp:8443/>

サインイン

グループ名
Groupname

☐ システム管理

ユーザーID
Username

パスワード
Password

ログイン

- (2) **グループ管理者から発行された**、
[グループID][ユーザーID][パスワード]を入力して
ログインする

ONION-object

分析 オブジェクト IAM Squest ヘルプ

バケット

+ 新規バケット追加

新規バケット追加

バケット名
Bucket Name

リージョン
osakau

オブジェクトロック
無効

ストレージポリシー
*EC42

キャンセル 作成

■ ログイン準備

- 一般ユーザがログインするためには、ONION-object の利用申請の後、**グループ管理者による一般ユーザの追加が必要**となります。

ONION-objectの使い方：バケット作成

バケット(データの入れ物)の作成手順です。

- (1) オブジェクトメニューで[新規バケットの追加]を選択します。
- (2) バケット名を入力し、作成を選択します。
※ストレージポリシー/リージョンは選択肢が1つ



- (3) バケットが作成されます。
必要に応じて[プロパティ]を表示しバケットの設定を変更します。



名前	リージョン	ストレージポリシー	
sqtestdemo20211109	osakau	EC42	プロパティ 削除

個別アクセス権	バケットのCANNED ACL	ストレージポリシー	ライフサイクルポリシー	静的WEBサイトホスティング	クロスリージョンレプリケーション	バージョンニング	ロギング
グループ・ユーザー		読み出し可能	書き込み可能	ACP読み出し可能	ACP書き込み可能		
Public		☐	☐	☐	☐		
Authenticated Users		☐	☐	☐	☐		
Log Delivery		☐	☐	☐	☐		

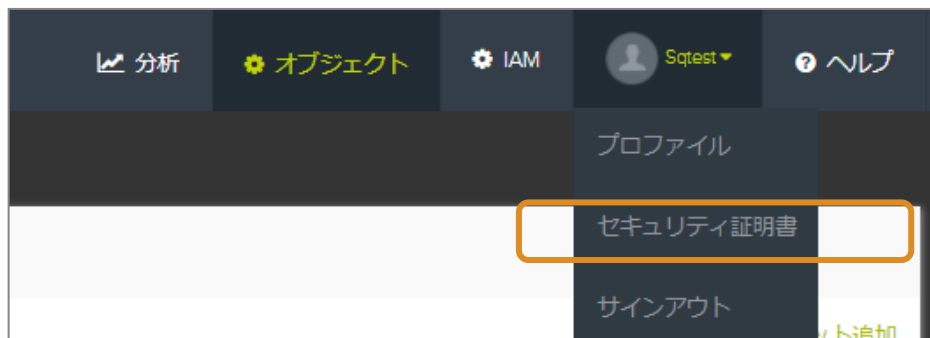
■ バケットの詳細説明

- バケットに関する下記の詳細な説明は、【5. S3 APIの概要】で説明します。
バケットの命名規則、バケット設定の重要な概念(ACL等)

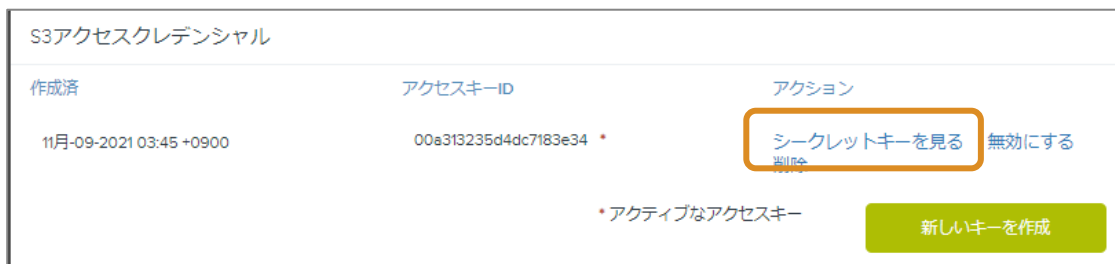
ONION-objectの使い方：アクセスキーの生成

アクセスキーの生成手順です。完了後、S3 APIアクセスが可能となります。

(1) アイコンから**[セキュリティ証明書]**を選択します。



(2) アクセスキーは標準で1つ生成されています。
そのまま使う場合は、**[シークレットキーを見る]**を選択し、キーを確認します。



(3) 新たなキーを作る場合は、**[新しいキーを作成]**を選択します。キーが1つ増えます。



■ アクセスキーとシークレットキー

- アクセスキーとシークレットキーの組は、**S3 APIで認証に利用するIDとパスワードに相当**します。
キーの管理にはご注意願います。

ONION-objectの使い方：オブジェクト操作



管理画面を用いたオブジェクトのアップロード/ダウンロード手順です。

(1) 操作を行うバケットを選択します。

✓ アップロードの場合

[ファイルのアップロード]を選択します。
ポップアップ画面の**[ファイル追加]**で**ファイルを選択し、アップロード開始**を選択します。

✓ ダウンロードの場合

ダウンロードしたいオブジェクトをクリックすると、オブジェクトのダウンロードが始まります。



☐	名前	サイズ	最終更新	
☐	TestFile.pdf	1.5 MB	Nov-09-2021 04:27 AM +0900	

■ オブジェクト操作

- ONION-object 管理画面でのオブジェクト操作は、補助的な位置づけになります。
- S3 API対応アプリケーションを利用することで、より簡単にオブジェクト操作が可能となります。具体的な手順については、【6. S3 APIの利用】で説明します。



EXAScaler の利用にはSQUIDシステムへのログインが必要です。

■ SQUIDのログイン

- EXAScaler の操作は、SQUIDのフロントエンドで行います。まず、SQUIDフロントエンドにログインしてください。
- SQUIDのログイン方法の詳細は、下記を参照ください。

- **SQUIDの利用方法 > ログイン方法**

<http://www.hpc.cmc.osaka-u.ac.jp/system/manual/squid-use/login/>

EXAScalerの使い方：アクセスキーの生成



アクセスキーの生成手順です。

1. SQUIDのフロントエンド上でs3dskey コマンドを実行します

※ 使用量をクォータに加算するグループを引数に指定します

```
$ s3dskey create --group=(グループ名)
```

2. アクセスキーの発行

下記のような出力と共にアクセスキーが生成されます。

accesskey	AKIA7X1KXXYYYYZZZ000
enabled	True
fspath	None
fsuid	60101:10
secretkey	*****
tag	(連番):(ユーザ名):(グループ名)
uuid	ea60f00a60hufaweofapo12813nfawe9506e1216

■ s3dskeyコマンド

項目	内容
s3dskey list	作成済みアクセスキーの一覧表示
s3dskey create	アクセスキーの新規作成
s3dskey disable	アクセスキーの無効化
s3dskey enable	アクセスキーの有効化
s3dskey reset	シークレットキーの初期化
s3dskey --help	コマンドヘルプの表示

■ アクセスキーとシークレットキー

- ・ アクセスキーとシークレットキーの組は、S3 APIで認証に利用するIDとパスワードに相当します。
キーの管理にはご注意ください。

EXAScalerの使い方 : バケット作成



バケット(データの入れ物)の作成手順です

1. フロントエンド上でs3dsbucket コマンドを実行します

```
$ s3dsbucket create --key=(アクセスキー) ¥  
--bucket=(バケット名)
```

■ バケットの詳細説明

- バケットに関する下記の詳細な説明は、【5. S3 APIの概要】で説明します。
バケットの命名規則、バケット設定の重要な概念

■ s3dsbucketコマンド

項目	内容
s3dsbucket list	作成済みバケットの一覧表示
s3dsbucket create	バケットの新規作成
s3dsbucket sync	ファイルシステムのファイルとS3 API のオブジェクトを同期する
s3dsbucket delete	バケットの削除 ※バケット内にオブジェクトが無い場合のみ可能
s3dskey --help	コマンドヘルプの表示

EXAScalerの使い方 : バケット作成



バケット(データの入れ物)の作成手順です

1. バケット作成後、s3dsbucket list コマンドを実行するとバケットのパスが確認できます。

```
$ s3dsbucket list --key=(アクセスキー)
```

CreationDate	Name	Path
2021-04-11 18:17:24	(バケット名)	/sqfs/s3/(アクセスキーのUUID)/(バケット名)

■ オブジェクトアクセスとLustre アクセス

- 外部からS3 APIでアップロードされたファイルは、**上記のPathに保存されます。**
- Lustre ファイルシステムから上記のPathにファイルをコピーし、S3 APIから参照することも可能です。
- S3DS は、オブジェクトアクセスとLustre アクセスの同期をsync メソッドを実行して保ちます。
整合性をとるためには、s3dsbucket syncコマンドを実行してください。

```
$ s3dsbucket sync --key=(アクセスキー) --bucket=(バケット名)
```

基本的な使い方：S3 APIの情報

ONION-object 並びに、EXAScaler のS3 API に関する基本情報をまとめます。

項目	ONION-object	EXAScaler	備考
エンドポイント名	s3-osakau.oniongw.hpc.cmc.osaka-u.ac.jp	squidgw.hpc.cmc.osaka-u.ac.jp	
リージョン名	osakau	(なし)	EXAScaler はリージョンの概念無し。ダミー値 us-east-1 を設定
アクセスキー /シークレットキー	管理コンソール上で発行	フロントエンド s3dskey コマンドで発行	
バケット作成	管理コンソールまたは、S3 APIで作成	フロントエンド s3dsbucket コマンドで作成	EXAScaler は、S3 APIでバケット作成する操作は非推奨
個別ACL対象	Canonical ID (管理コンソール上で確認) グループ (管理コンソールのみ)	UUID	

4. サブシステム間連携

サブシステムを連携して利用する

ONIONのサブシステム間を連携して使用する例を紹介します。

■ ONION-file からのS3 API 連携

ONION-file → ONION-object/EXAScaler

- データのブラウザ表示やモバイルアクセスが可能なONION-fileと、S3 APIを連携することで、ストレージ上のデータに、より柔軟にアクセスが可能になります。

■ S3 API対応ストレージ間のデータ移動

ONION-object ↔ EXAScaler

- ONIONでは、ONION-object と、EXAScaler の用途の異なるS3 API対応ストレージがあります。
- 両者のストレージ間でデータを移動することで、2つのストレージの長所を使い分けることができます。

✓ 本項で出てくるツールの詳細な使い方の説明は、【6. S3 APIの利用】で説明します。

外部ストレージ連携機能により、S3 対応ストレージの操作が可能です。

(1) ONION-fileにSQUIDユーザでログイン

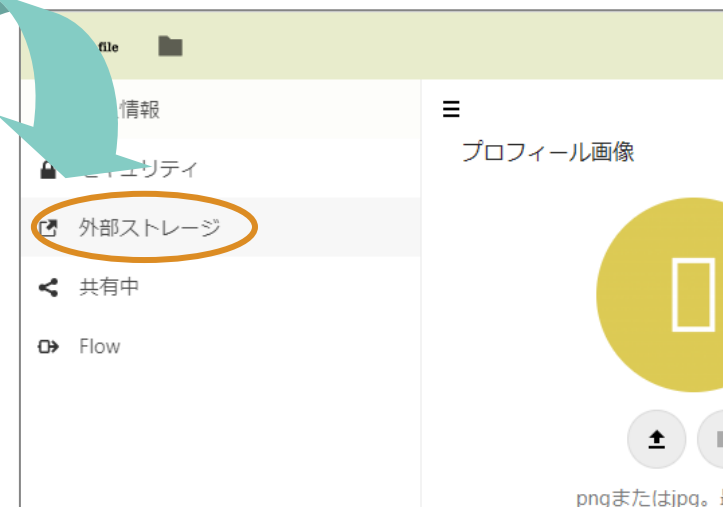
<https://onionweb.hpc.cmc.osaka-u.ac.jp/>



(2) 右上のアイコンから[設定]を選択



(3) [外部ストレージ]を選択



外部ストレージ連携機能により、S3 対応ストレージの操作が可能です。

(4) [ストレージを追加]メニューで[Amazon S3]を選択



(5) [Amazon S3]用メニューで必要な情報を入力



外部ストレージ連携での入力内容

項目	内容	ONION-object の場合	EXAScaler の場合	備考
フォルダー名	OnionWeb上で表示されるフォルダ名	(任意の名前)	(任意の名前)	
外部ストレージ	外部ストレージの種類	Amazon S3	Amazon S3	固定
バケット名	バケットの名前	(事前作成したバケット名)	(事前作成したバケット名)	
ホスト名	エンドポイント	s3-osakau.oniongw.hpc. cmc.osaka-u.ac.jp	squidgw.hpc. cmc.osaka-u.ac.jp	
ポート	ポート番号	443	443	
リージョン	S3 リージョン名	osakau	(なし)	
SSLを有効	SSL有無	レ	レ	
パス形式を有効	パス形式でのバケットアクセス	レ	レ	
レガシー認証(v2)	古い(v2)認証方式の利用	(なし)	❗ (S3DSはv4認証で一部メソッドが未対応)	
アクセスキー	アクセスキー	(事前発行したアクセスキー)	(事前発行したアクセスキー)	
シークレットキー	シークレットキー	(アクセスキーに紐づくキー)	(アクセスキーに紐づくキー)	

設定後は、WEBブラウザやスマートフォンアプリ等から表示可能です。



WEBブラウザの画面



スマートフォンアプリの画面

SQUIDフロントエンド経由でデータ移動が可能です。
S3 API対応コマンドにより、相互にバケットデータを同期します。

※ AWS CLI の実施方法を説明します。

1. aws cli の設定

```
$ aws configure
```

AWS Access Key ID : (ONION-object アクセスキー)

AWS Secret Access Key : (ONION-object シークレットキー)

Default region name : osakau

Default output format : (なし)

✓ EXAScaler → ONION-object へのデータ同期

```
$ aws --endpoint-url=https://s3-osakau.oniongw.hpc.cmc.osaka-u.ac.jp ¥  
s3 sync /sqfs/s3/(EXAScalerのUUID)/(EXAScalerのバケット名)/ s3://(ONION-objectバケット名)/
```

✓ ONION-object → EXAScalerへのデータ同期

```
$ aws --endpoint-url=https://s3-osakau.oniongw.hpc.cmc.osaka-u.ac.jp ¥  
s3 sync s3://(ONION-objectのバケット名)/ /sqfs/s3/(EXAScalerのUUID)/(EXAScalerバケット名)/  
$ s3dsbucket sync -k (S3DSアクセスキー) -b (S3DS バケット名)
```

5. S3 APIの概要

S3 APIとは

AWSのストレージ(Amazon S3)を利用するためのAPI
高スケーラビリティのクラウドストレージを簡単に利用するために設計
され、RESTやSOAPインタフェース、SDK等が提供されている

https://docs.aws.amazon.com/ja_jp/AmazonS3/latest/userguide/Welcome.html

The screenshot shows the AWS S3 User Guide page. At the top, there is the AWS logo, a search bar with the text 'このガイド内で検索', and a language selector set to '日本語'. Below the header, a breadcrumb trail reads 'AWS > ドキュメント > Amazon Simple Storage Service (S3) > ユーザーガイド'. On the left side, there is a sidebar with the title 'Amazon Simple Storage Service ユーザーガイド' and a list of topics: 'Amazon S3 とは' (highlighted in orange), '開始方法', 'バケットの使用', and 'オブジェクトの使用'. The main content area has the title 'Amazon S3 とは' and links for 'PDF' and 'RSS'. Below the title, a paragraph states: 'Amazon Simple Storage Service (Amazon S3) は、インターネット用のストレージです。ウェブスケールのコンピューティングを簡単に利用できるように設計されています。'

なぜ S3 APIを使うのか

S3 API の利点

オブジェクトストレージの操作

◆ オブジェクトアクセス

高スケーラビリティを実現するオブジェクトストレージを操作可能

◆ 柔軟なアクセス権/ポリシー

フォルダ階層に捉われない柔軟なアクセス権の設定
データ保護のポリシーなど、NASではできないデータ操作が可能

アプリケーションとの親和性

◆ WEBサービスでの実装

REST APIで整備されている。
ストレージ操作を言語依存しない統一的なインタフェースで利用可能

◆ 広範なSDKの提供

多数のプログラミング言語向けにSDKが整備されており、効率的にAP開発が可能
JS/Python/PHP/.NET/Ruby/Java/Go/Node.js/C++

クラウド・エコシステムの恩恵

◆ クラウド連携の流行

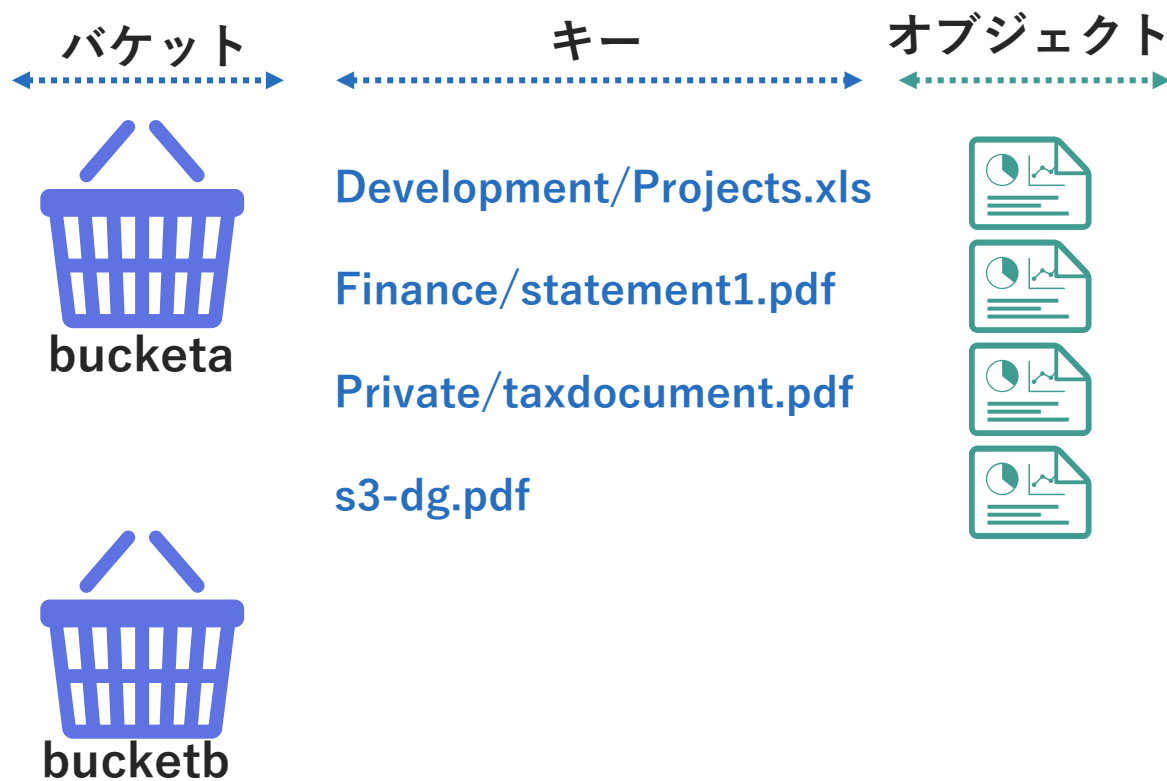
S3 APIを用いて、データをクラウド連携する製品が増加。
S3 APIをサポートすればクラウド連携の恩恵を得られる

◆ コミュニティの活用

AWSはクラウドシェアの最大手となっており、S3 APIに関する情報も得られやすい

S3 API の概念 – バケット / キー / オブジェクト

Amazon S3 では、バケット/キー/オブジェクトの組み合わせでデータを管理します。オプションで、バージョンIDがこれに加わります。



バケット

オブジェクトの入れ物(コンテナ)
アクセスコントロールの単位
アカウントの集計単位

キー

バケット内でオブジェクトを特定するための識別子
区切り記号"/"にて階層構想を表現

オブジェクト

データの基本単位(エンティティ)
オブジェクトデータとメタデータの組
アクセスコントロールの最小単位
バージョンニング時はバージョンIDが付与

S3 API の概念 – バケットの機能

バケットは、オブジェクトを格納するコンテナ
S3 にオブジェクトを格納する際は、最初に作成する



■ バケットの命名規則

- 3~63文字
- 英小文字、数字、ドット(.)、ハイフン(-)のみ
※ ドット(.)は非推奨
- 文字または数字で開始/終了する必要あり
- IPアドレス形式(192.168.1.1等)は不可
- xn-- のプレフィックス不可
- -s3alias のサフィックス不可
- **バケット名は、パーティション(リージョン内のグループ)で一意**

■ バケットに指定可能な内容(例)

項目	内容	HS	S3DS
バージョニング	オブジェクトの履歴管理	○	○
デフォルト暗号化	自動暗号化の指定	○	×
アクセスログ	アクセスログの記録有無	○	×
静的WEBサイトホスティング	静的WEBとしての公開有無	○	×
ライフサイクル	バケット内オブジェクトの破棄ポリシー	○	×
個別アクセス許可(ACL)	バケットへのアクセス許可 (Canned ACL 含む)	○	○
ポリシー	バケットへのアクセスポリシー	○	△

S3 API の概念 – オブジェクトの機能

オブジェクトは、データを格納する単位
バケット内で一意のキーにより識別されます



■ キーの命名規則

- UTF-8文字の使用可。
※アプリケーションによっては未対応ケースあり
- 安全な文字として、下記が推奨される
英小文字、英大文字、数字、スラッシュ(/)、感嘆符(!)、ハイフン(-)、下線(_)、ピリオド(.)、アスタリスク(*)、一重引用符(')、丸カッコ()

■ バケットに指定可能な内容(例)

項目	内容	HS	S3DS
オブジェクト ACL	オブジェクトへのアクセス許可 (Canned ACL含む)	○	○

オブジェクトへのアクセス

仮想ホスティング形式とパス形式のアクセス方法があります

■ 仮想ホスティング形式のアクセス

`https://bucket-name.s3-endpoint-name/key-name`

例) `https://mybucket.s3-osakau.oniongw.hpc.cmc.osaka-u.ac.jp/mykey/myobject.png`

■ パス形式のアクセス

`https://s3-endpoint-name/bucket-name/key-name`

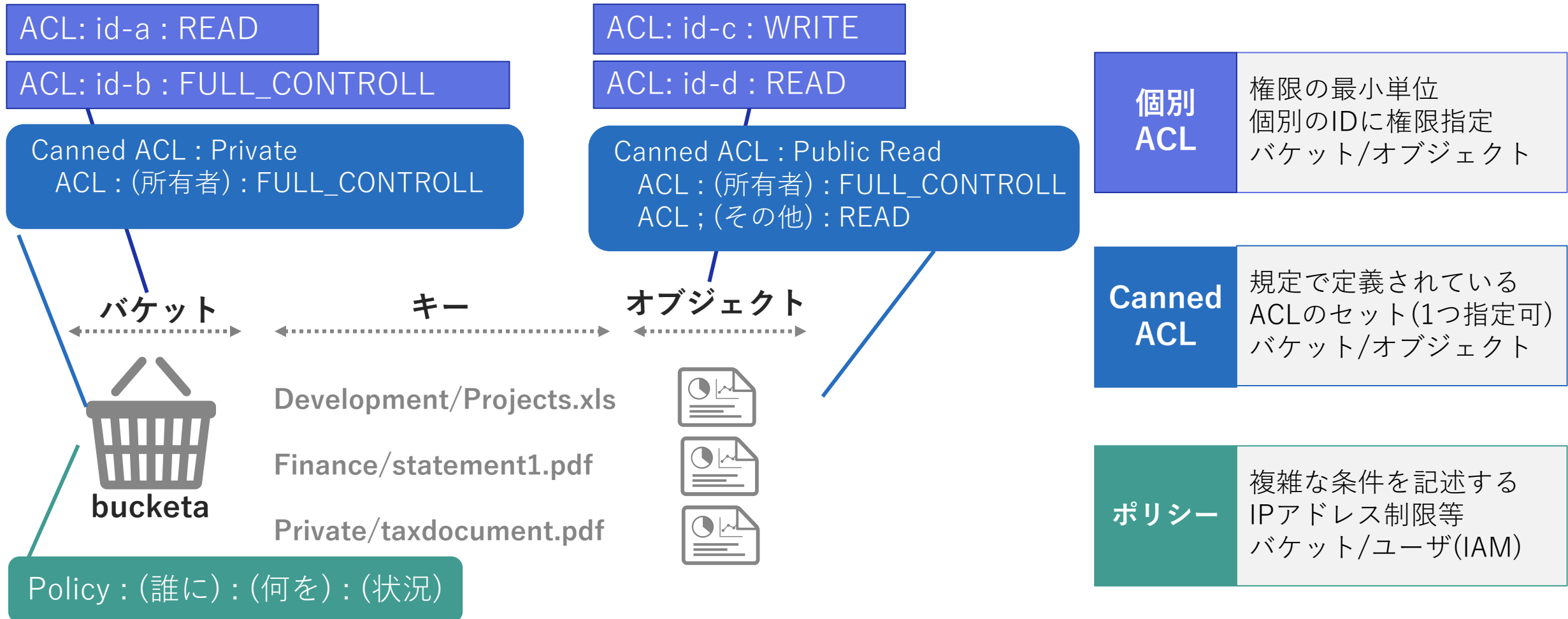
例) `https://s3-osakau.oniongw.hpc.cmc.osaka-u.ac.jp/mybucket/mykey/myobject.png`

※ AWS では、将来的にパス形式を非推奨化するプランが出されていますが、移行に必要な時間確保のため、延期されています。

参考URL: https://docs.aws.amazon.com/ja_jp/AmazonS3/latest/userguide/access-bucket-intro.html

アクセス制御 - 概念

バケット/オブジェクトのアクセス制御に関わる主要概念を説明します。



アクセス制御 - Canned ACL

利用可能なCanned ACL について説明します。

名称	対象	内容	備考
Private	バケット/オブジェクト	所有者のみにフルコントロール	規定
Public Read	バケット/オブジェクト	所有者にフルコントロール public 扱いの全ユーザに 読み取り	
Authenticated Read	バケット/オブジェクト	所有者にフルコントロール 認証済みユーザに読み取り	
Bucket Owner Read	バケット/オブジェクト	所有者にフルコントロール バケット所有者に読み取り	S3DSはバケットに指定不可
Bucket Owner Full Control	バケット/オブジェクト	所有者にフルコントロール バケット所有者にフルコントロール	S3DSはバケットに指定不可
Public Read-Write	バケット/オブジェクト	所有者にフルコントロール public 扱いの全ユーザに 読み取り/書き込み	S3DS のみ

アクセス制御：設定例

アクセス制御の設定例について説明します。Canned ACLの設定例です

※ アクセスキーの発行/バケット作成が実施され、AWS CLIの設定が完了している前提です。詳細は後述。

- ✓ **ONION-object の バケット (testbucket) に Authenticated Read のCanned ACL を付与**

```
$ aws --endpoint-url=https://s3-osakau.oniongw.hpc.cmc.osaka-u.ac.jp ¥  
s3api put-bucket-acl --acl authenticated-read --bucket testbucket
```

- ✓ **EXAScaler のオブジェクト (test.txt)にPublic Read のCanned ACLを付与**

```
$ aws --endpoint-url=https://squidgw.hpc.cmc.osaka-u.ac.jp ¥  
s3api put-object-acl --acl public-read --bucket testbucket --key test.txt
```

※ 上記は実行例であり、Public アクセス許可は非推奨です。

■ Canned ACLの指定

- Canned ACLは、**バケット(オブジェクト)当たり1つのみ指定可能**で、aws cli では --acl オプションで指定します。

アクセス制御：設定例

ACL設定時のID に関する詳細です。

✓ ONION-objectの場合

idは**CANNONICAL ID**を指定します。
管理コンソール[プロファイル]より確認が可能です。

アカウントプロフィール

ユーザーID: [REDACTED] グループID: [REDACTED]

CANONICAL ID:
512a2ff0fde91187b6274fc5a71f521a

✓ より簡単な操作のため、管理コンソール上では、**グループ名とユーザ名の直接指定が可能です。**

個別アクセス権	バケットのCANNED ACL	ストレージポリシー	ライフサイクルポリシー	静的WEBサイトホスティング
グループ・ユーザー		読み出し可能	書き込み可能	ACP
Public		☐	☐	
Authenticated Users		☐	☐	
Log Delivery		☐	☐	
		☐	☐	

✓ EXAScalerの場合

idは**アクセスキーのUUID**を指定します。
s3dskey コマンドで確認が可能です。

```
$ s3dskey list
```

accesskey	enabled	secretkey	tag	uuid
XXXXXXXXXX	True	YYYYYYYYYY	0001:ZZ:XX	abcdefghi jksss

ユーザ指定の場合：

グループ名|ユーザ名

例) group1|user1

グループ指定の場合：

グループ名|

例) group1

6. S3 API の利用

S3 APIを利用する

S3 API の利用ケースを下記の分類で説明します。

S3 対応アプリケーション利用

- **対応アプリケーション利用**
一連のタスクを処理するように開発されたアプリケーションを通しての利用
- アプリケーション例
aws cli、rclone、cloudberry

APIレベルのツール操作

- **API操作するツールを利用**
S3 APIを発行するツールを使用し、APIレベルの動作確認や細かい操作を実施
- ツール例
s3curl、aws s3api、s3cmd

プログラムからSDKを利用

- **開発プログラムからのSDK呼び出し**
開発プログラムとS3 APIとの連携処理や、システム間連携
- SDK 例(AWS SDK)
Python(BOTO3)

<https://aws.amazon.com/jp/getting-started/tools-sdks/>

AWS の標準CLIツール(AWS CLI) には、S3向け高レベルコマンドが用意されています。

<https://docs.aws.amazon.com/cli/latest/reference/s3/index.html>

1. aws cli の設定

```
$ aws configure
```

AWS Access Key ID : (アクセスキー)

AWS Secret Access Key : (シークレットキー)

Default region name : **osakau**

Default output format : (なし)

✓ 左記はONION-objectの設定例。
EXAScalerの場合はregion name は空白

2. aws s3 にファイル操作

```
$ aws --endpoint-url=s3-endpoint-name s3 (サブコマンド) (オプション)
```

※ 利用可能なサブコマンドは、下記の通りです。

cp / ls / mb / mv / presign / rb / rm / sync / website

詳細は、 **aws s3 help** コマンドにてご確認ください。

AWS CLI による、コマンド実行例を紹介します。

✓ ローカル → ONION-object へのファイルコピー

```
$ aws --endpoint-url=https://s3-osakau.oniongw.hpc.cmc.osaka-u.ac.jp ¥  
s3 cp (ローカルパス) s3://(バケット名)/(キー)
```

✓ ローカル → EXAScaler へのファイルコピー

```
$ aws --endpoint-url=https://squidgw.hpc.cmc.osaka-u.ac.jp ¥  
s3 cp (ローカルパス) s3://(バケット名)/(キー)
```

✓ ONION-object → ローカルへのファイルコピー

```
$ aws --endpoint-url=https://s3-osakau.oniongw.hpc.cmc.osaka-u.ac.jp ¥  
s3 cp s3://(バケット名)/(キー) (ローカルパス)
```

✓ ローカルとONION-object のディレクトリ同期

```
$ aws --endpoint-url=https://s3-osakau.oniongw.hpc.cmc.osaka-u.ac.jp ¥  
s3 sync (ローカルパス) s3://(バケット名)/(キー)/
```

rclone は、多数のクラウドストレージとデータ同期が可能なツールです。リモートサイトを複数定義できる特徴があります。

<https://rclone.org/>

1. リモートの定義(ONION-object)

```
$ rclone config
n) New remote
n/s/q> n
name> (任意の名前 例: oniongw)
4 / Amazon S3 Compliant Storage Providers including AWS, Alibaba, Ceph,
Digital Ocean, Dreamhost, IBM COS, Minio, SeaweedFS, and Tencent COS
¥ "s3"
Storage> s3
13 / Any other S3 compatible provider
¥ "Other"
provider> other
1 / Enter AWS credentials in the next step
¥ "false"
env_auth> false
AWS Access Key ID.
access_key_id> (アクセスキー)
AWS Secret Access Key (password)
secret_access_key> (シークレットキー)
region> osakau
endpoint> s3-osakau.oniongw.hpc.cmc.osaka-u.ac.jp
```

```
(続き)
location_constraint> (ブランク)
1 / Owner gets FULL_CONTROL. No one else has access rights
(default).
¥ "private"
acl> private
Edit advanced config?
n) No (default)
y/n> n
-----
[oniongw]
type = s3
provider = Other
access_key_id = (アクセスキー)
secret_access_key = (シークレットキー)
region = osakau
endpoint = s3-osakau.oniongw.hpc.cmc.osaka-u.ac.jp
acl = private
-----
y) Yes this is OK (default)
y/e/d> y
```

rclone は、多数のクラウドストレージとデータ同期が可能なツールです。リモートサイトを複数定義できる特徴があります。

<https://rclone.org/>

1. リモートの定義(EXAScaler)

```
$ rclone config
n) New remote
n/s/q> n
name> (任意の名前 例: squidgw)
4 / Amazon S3 Compliant Storage Providers including AWS, Alibaba, Ceph,
Digital Ocean, Dreamhost, IBM COS, Minio, SeaweedFS, and Tencent COS
¥ "s3"
Storage> s3
13 / Any other S3 compatible provider
¥ "Other"
provider> 13
1 / Enter AWS credentials in the next step
¥ "false"
env_auth> false
AWS Access Key ID.
access_key_id> (アクセスキー)
AWS Secret Access Key (password)
secret_access_key> (シークレットキー)
2 / Use this only if v4 signatures don't work, e.g. pre Jewel/v10 CEPH.
¥ "other-v2-signature"
region> other-v2-signature
endpoint> squidgw.hpc.cmc.osaka-u.ac.jp
```

```
(続き)
location_constraint> (ブランク)
1 / Owner gets FULL_CONTROL. No one else has access rights
(default).
¥ "private"
acl> private
Edit advanced config?
n) No (default)
y/n> n
-----
[squidgw]
type = s3
provider = Other
access_key_id = (アクセスキー)
secret_access_key = (シークレットキー)
region = other-v2-signature
endpoint = squidgw.hpc.cmc.osaka-u.ac.jp
acl = private
-----
y) Yes this is OK (default)
y/e/d> y
```

rclone による、コマンド実行例を紹介します。

✓ ローカル → ONION-object の同期

```
$ rclone sync (ローカルパス) oniongw:(バケット名)/(キー)/
```

✓ ローカル → EXAScaler の同期

```
$ rclone sync (ローカルパス) squidgw:(バケット名)/(キー)/
```

✓ EXAScaler → ONION-object の同期

```
$ rclone sync squidgw:(バケット名)/(キー)/ oniongw:(バケット名)/(キー)/
```

※ 前ページの例に従い、リモート定義を以下で登録している前提の実行例です。

ONION-object : oniongw EXAScaler : squidgw

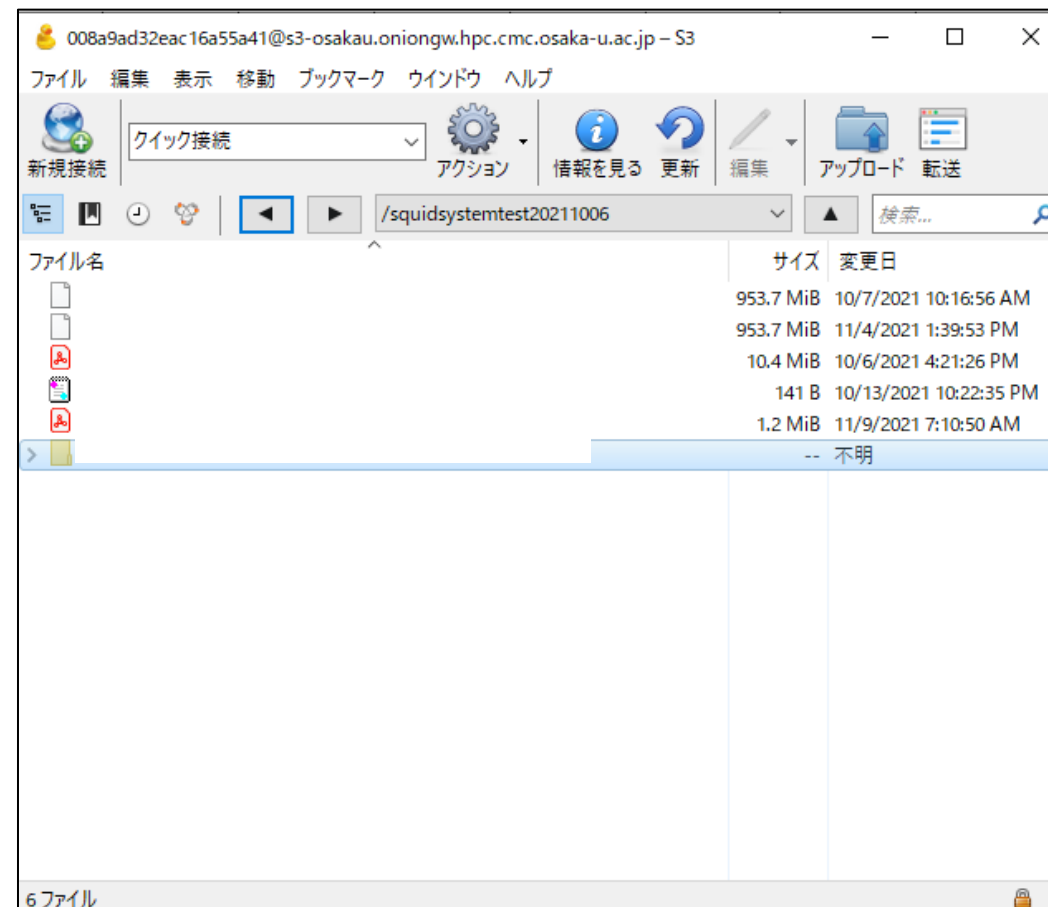
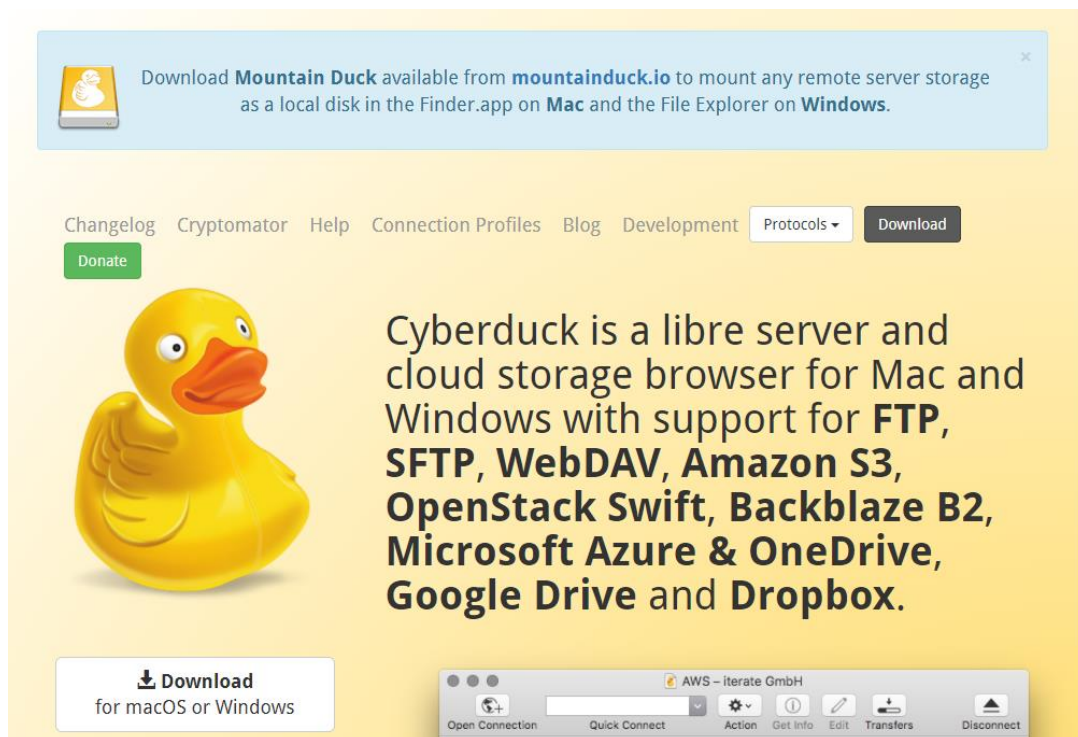
※ rclone は、sync以外にも多数のコマンドが実装されています。利用可能なコマンドの詳細は、**rclone help** コマンドにてご確認ください。

Cyberduckを利用する

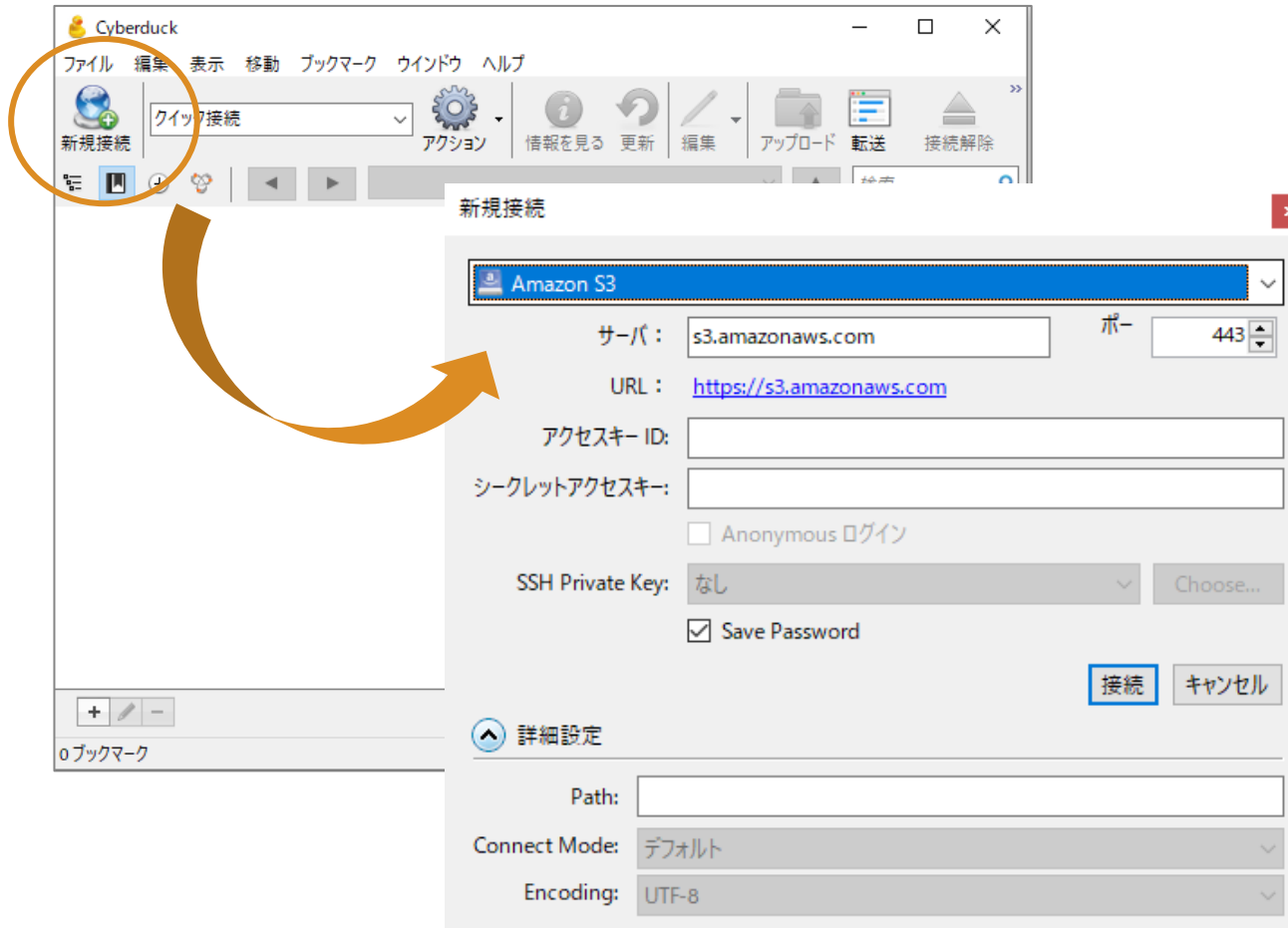
S3 対応アプリケーション利用

Cyberduck は、Windows OS/Mac OS環境で使えるファイル転送アプリケーションです。多数のプロトコルを同じインタフェースで扱える点に特徴があります。

<https://cyberduck.io/>



Cyberduck のAmazon S3 の接続先として、ONIONを登録することが可能です。



項目	内容
タイプ	Amazon S3
サーバ	ONION-objectの場合： s3-osakau.oniongw.hpc.cmc.osaka-u.ac.jp EXAScalerの場合： squidgw.hpc.cmc.osaka-u.ac.jp
ポート	443
アクセス キー	(発行したアクセスキー)
シーク レット キー	(発行したシークレットキー)

アプリケーション利用にあたっては、EXAScaler はS3 APIに対応していない部分もあり、エラーとなるケースもあります。下記の既知の不具合と回避策が確認されています。

■ 署名バージョン v4の際に、日本語ファイルの操作がエラーとなる

- S3 APIの署名バージョンv4認証を使用するアプリケーションで、日本語を含むファイル操作でエラーとなるケースがあります。

【回避策1】

アプリケーション側で、署名バージョンv2認証(レガシ認証) が利用可能であれば、そちらを利用ください。

【回避策2】

ファイル名を英語のみのものに変更し、エラーが出なくなるかお試してください。

s3curl は、curl コマンドにS3向けの機能を拡張した、コマンドラインユーティリティです。特殊ヘッダの指定などが行える特徴があります。

<https://github.com/rtdp/s3curl>

1. s3curl の設定

```
$ vi ~/.s3curl
%awsSecretAccessKeys = (
    (定義名) => {
        id => (アクセスキー),
        key => (シークレットキー),
    },
);
```

2. s3curl の実行

```
$ s3curl.pl --id=(定義名) --endpoint=(エンドポイント名) (s3curlオプション) ¥
-- (curlオプション) (URL)
```

※ オプションの指定方法が特殊で、-- の前後で指定可能なオプションが異なります。
s3curl 固有のオプション(s3curl.pl --help) は前半、curl のオプション(curl --help)を後半に指定します。

s3curl による実行例です。

✓ S3DSへファイルアップロード (PUT Object)

```
$ s3curl.pl --id=(定義名) --endpoint=squidgw.hpc.cmc.osaka-u.ac.jp --put=(ローカルファイル名) ¥  
-- https://squidgw.hpc.cmc.osaka-u.ac.jp/(バケット名)/(キー)
```

✓ S3DSからファイルダウンロード (GET Object)

```
$ s3curl.pl --id=(定義名) --endpoint=squidgw.hpc.cmc.osaka-u.ac.jp ¥  
-- https://squidgw.hpc.cmc.osaka-u.ac.jp/(バケット名)/(キー) --output=(ローカルファイル名)
```

✓ バケットとファイルシステムの名前空間同期 (PUT BucketSync)

```
$ s3curl.pl --id=(定義名) --endpoint=squidgw.hpc.cmc.osaka-u.ac.jp --put=emptyPayload ¥  
-- https://squidgw.hpc.cmc.osaka-u.ac.jp/(バケット名)?sync ¥  
-H "x-ddn-bucket-sync-ops:WRITE,UPDATE,DELETE"
```

※ emptyPayload というダミーのファイルを指定していますが、ファイルの内容は空でも問題ありません。

※ S3DS の特殊操作を呼び出すために、-H オプションで拡張ヘッダを指定しています。

AWS の標準CLIツール(AWS CLI)の `s3api` コマンドは、APIレベルの操作を行うコマンドです。ACLの細かな制御等が可能となります。

<https://docs.aws.amazon.com/cli/latest/reference/s3api/index.html>

1. aws cli の設定

```
$ aws configure
```

AWS Access Key ID : (アクセスキー)

AWS Secret Access Key : (シークレットキー)

Default region name : **osakau**

Default output format : (なし)

✓ 左記はONION-objectの設定例。
EXAScalerの場合はregion name は空白

✓ ONION-objectのバケットポリシーを確認する

```
$ aws --endpoint-url=https://s3-osakau.oniongw.hpc.cmc.osaka-u.ac.jp ¥  
s3api get-bucket-policy --bucket (バケット名)
```

✓ EXAScaler のオブジェクトにACLを追加する

```
$ aws --endpoint-url=https://squidgw.hpc.cmc.osaka-u.ac.jp ¥  
s3api put-object-acl --bucket (バケット名) --key (オブジェクトキー) --grant-write id=(UUID)
```

Python 向けSDK (BOT03) を利用する

プログラムからSDKを利用

Python言語向けのSDK として、BOT03 が用意されています。下記にサンプルコードを示します。
APIレベルのclient オブジェクトと、高レベルのresource オブジェクトがあります。

client_test.py

```
import boto3

# create bot3 session object
sess = boto3.session.Session(
    aws_access_key_id='アクセスキー',
    aws_secret_access_key='シークレットキー',
)

# create s3 service client
s3cl = sess.client('s3',
    endpoint_url='https://s3-osakau.oniongw.hpc.cmc.osaka-u.ac.jp'
)

# exec low level api
response = s3cl.list_objects(Bucket='バケット名')

# print object list
for obj in response.get('Contents'):
    print( obj.get('Key') )
```

バケット操作をS3の
LIST_OBJECTS API
を発行して操作

resource_test.py

```
import boto3

# create bot3 session object
sess = boto3.session.Session(
    aws_access_key_id='アクセスキー',
    aws_secret_access_key='シークレットキー',
)

# create s3 resource client
s3 = sess.resource('s3',
    endpoint_url='https://s3-osakau.oniongw.hpc.cmc.osaka-u.ac.jp'
)

# exec high level function
bucket = s3.Bucket('バケット名')

# print object list
for obj in bucket.objects.all():
    print( obj.key )
```

バケットをオブジェク
トとして操作
(API名が不要)

7. 情報入手先

情報参照先 - ONIONの利用全般

センターホームページで、ONIONの利用に関するページが公開されています。「ONIONの利用方法」を参照ください。

<http://www.hpc.cmc.osaka-u.ac.jp/system/manual/onion-use/>

The screenshot shows the homepage of the Osaka University Cybermedia Center. The header includes the center's name in Japanese and English, along with links for 'アクセス' (Access), 'サイトマップ' (Site Map), and '日本語' (Japanese). A navigation bar contains links for various user groups: '利用を検討中の方' (For those considering use), '一般利用の方' (For general use), '産業利用の方' (For industrial use), '公募利用の方' (For public use), 'HPCI利用の方' (For HPCI use), and 'JHPCN利用の方' (For JHPCN use). Below this is a secondary navigation bar with links for 'システム' (System), '利用案内・申請' (Usage Guide/Apply), '利用支援' (Usage Support), 'イベント' (Event), '公開資料' (Public Materials), '成果報告' (Research Results), and 'お問い合わせ' (Contact Us). The main content area is titled 'ONIONの利用方法' (ONION Usage Method) and includes a breadcrumb trail: 'Cybermedia Center, Osaka University > システム > 利用方法 > ONIONの利用方法'. The text explains that the page provides instructions for using the ONION storage service and directs users to specific links for more information or to apply for ONION-object usage. A sidebar on the left lists additional topics: 'ONION-objectの利用方法' (ONION-object Usage Method), '基本的な利用方法' (Basic Usage Method), 'ユーザ登録、削除方法' (User Registration/Deletion Method), and '利用状況の確認方法' (Usage Status Confirmation Method).

大阪大学 サイバーメディアセンター
大規模計算機システム

アクセス サイトマップ 日本語

利用を検討中の方 一般利用の方 産業利用の方 公募利用の方 HPCI利用の方 JHPCN利用の方

システム 利用案内・申請 利用支援 イベント 公開資料 成果報告 お問い合わせ

ONIONの利用方法 Cybermedia Center, Osaka University > システム > 利用方法 > ONIONの利用方法

ストレージサービスONIONの利用方法について解説します。
ONIONについて知りたい方は[こちら](#)をご覧ください。
ONION-objectの利用申請や問い合わせは[こちら](#)をご参照ください。

ONION-objectの利用方法

基本的な利用方法
ユーザ登録、削除方法
利用状況の確認方法

S3 APIの概念、関連ツール類のマニュアルは、オンラインマニュアルが充実しています。



■ Amazon Simple Storage Service

- User Guide

<https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html>

■ AWS CLI

- AWS CLI User Guide

<https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-welcome.html>

- AWS CLI Command Reference

<https://awscli.amazonaws.com/v2/documentation/api/latest/reference/index.html>

■ AWS Boto3

- Boto3 Docs 1.20.1

<https://boto3.amazonaws.com/v1/documentation/api/latest/guide/quickstart.html>

\Orchestrating a brighter world

NEC