\Orchestrating a brighter world **NEC**



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

2022年 6月23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
- 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. 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	データ共有と連携の柔軟性	スパコンと一体のストレージ		
 ◆ 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 	 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(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 ONION				

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



\Orchestrating a brighter world **NEC**

ー次ストレージ連携





U



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



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
 - 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

■ 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(予定)/S3API などデータ転送方式を拡充

■ S3 API対応による クラウド連携対応のストレージ

• 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

\Orchestrating a brighter world NEC

EXAScaler

ONION-object

ONION-object

ONION-file

EXAScaler

各サブシステムの役割

ONIONでは用途別にデータが ONION-object(アーカイブ)とExaScaler(高速解析) に格納され、ONION-file にて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







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。



<u>デスクトップ クライアント</u>

<u>モバイルアプリケーション</u>

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





ONION-objectは、Cloudian HyperStore で構成されたオブジェクトストレージで、 665.9TB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	950 TB
データ保護方式	イレイジャー・コーディング (データチャンク:4+パリティチャンク:2)

※ 2021年12月拡張作業実施。順次拡張予定。



■ 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のS3 API と高い互換性を持つストレージです。 S3 APIの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/

	Deutsch Englis	sh Español Português 日本語				
	サインイン グルーブID: Groupname ロシステム管理					
ONION	ユーザー ID: Username パスワード:	バケット オブジェクト パケット名 testbucket	ファイルをアップロー	-ド + フォルダーを作成 Q プレ	フィックスで検索	
– object –	Password	osakau : testbucket 合前 日本 test2 (4) tyt	サイズ 961 MB	最終史新 Oct. 28-2021 02:40 PM +0000	# プロパーィ	會 淵祥
		test 20211104 tvt	2 B	Oct-01-2021 12:07 PM +0900	樹 プロパティ	
				100 042021 10.40 AM 10500		リストア 削除





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。



<u>NVMe領域:DDN ES400NVX</u>



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して、DDN S3 Data Service (S3DS) が 稼働しています。





DDN S3 Data Service x4

※ 2021年12月拡張作業実施。順次拡張予定。

- 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/ダウンロードを行う上で必要な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

EXAScaler

- ログイン
- バケット作成、アクセスキー作成

- ✓ 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。
- ✓ 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 のブラウザでのログイン手順です。



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の使い方: ログイン



ONION-file のログイン方法が、2022年4月より変更されています。

■ 変更内容

- 重要情報をまもるため、ONION-file ログイン時に、二要素認証を必須としました。
- 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 専用のものでり、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

ルのドラッグ&ドロップでアップロードが可能です。 をクリックし、[ダウンロード]を選択します。

	۹ 😃 🕕
\equiv \bigstar \rangle \cdots \rangle demo \ll \rangle +	
□ 名前 ▲	サイズ 更新日時
cloud-test4.sh.o544595.log	< 3 KB 17時間前
cloud-test4.sh.o545147.log	
2 個のファイル	5 KB

✓ ダウンロードの場合 アップロードしたいディレクトリに移動し、ファイ ダウンロードしたいファイルのメニュー(...)の部分

	Q	4 (
1	ナイズ	更新日時
<	3 KB	17時間前
★ お気に入りに追加	3 KB	16時間前
1 詳細	5 VD	
✓ 名前の変更	J ND	
Ⅰ 移動またはコピー		
♦ ダウンロード		
	・ ★ お気に入りに追加 i 詳細 メ 名前の変更 C 移動またはコピー ↓ ダウンロード 章 ファイルを削除	サイズ く ・・・ 3 KB ★ お気に入りに追加 1 詳細 メ 名前の変更 ビ 移動またはコピー ↓ ダウンロード 夏 ファイルを削除

■ 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



<u>グループ内での</u>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(<)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
 す。 共有された側は、[共有]-[他ユーザがあなたと共有
 リスト 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。SQUID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。 パスワードを設定し、[Create share]を押します。 ONION-file Q 4 $\rangle \cdots \rangle$ demo < \rangle + cloud-test4.sh.... × 名前 🔺 サイズ 更新日時 Name or email .. ••• 3 KB 17時間前 cloud-test4.s....log **<** 井有中 URLで共有 ••• cloud-test4.s....log 16時間前 *S* 共有中 ••• 3 KB 共有を作成する前に、 次の必要な情報を入力 してください ▲ パスワード保護(強制) tgjdSYbYC6ys \rightarrow Create share × キャンセル
- ✓ 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。 パスワード入力が求められるため、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。





(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。

フォルダー名	外部ストレージ	ROBL	設定
SQUID_HOME	SFTP	ログイン認証情報は、データベースに保存されます。	
フォルダー名	ストレージを追か ▼ Amazon S3 SFTP WebDAV		

(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。

フォルダー名	外部ストレージ	P2音 <u>正</u>	設定
SQUID_HOME	SFTP	ログイン認証情報は、データベースに保存されます。	
フォルダー名	ストレージを追か▼		
	Amazon S3 SFTP WebDAV		

(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。

ここにファイルをアップロードまたは作成する権限がありません	
□ 名前 _	
SQUID_HOME	<
2 個のフォルダー	



ONION-file 利用に関する補足事項です。

■ アカウントの管理

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.h	oc.cmc.osaka-u.ac.jp:8443/

	サインイン ガループロ・ Groupname ロシステム管理 ユーザー ID: Username
ONION – object –	アスワード: 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	o Iam	Sqtest -	₀ ∿レプ
バケット						
					+ 新規	い、ケット追加
新規パケット追加						
バケット名	リージョン		オブジェク	クトロック		
Bucket Name	osakau		🔹 💿 無効			
ストレージボリシー						
*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

バケット			
			+ 新規バケット追加
新規パケット追加			
パケット名	リージョン	オブジェクトロック	
sqtestdemo20211109	osakau	💲 💿 無効	
ストレージボリシー			
*EC42			
			キャンセル 作成

(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。

Г	バケット	オブジェクト							
L									+ 新規バケット追加
	名前		リージョン		ストレージボリシー				
L	sqtestdemo2	0211109	osakau		EC42		tit :	プロパティ	■ 削除
L	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ン	プリケーション	バージョニング	ロギング
	Ď	バープ・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	1						
	Au	thenticated Users	(
		Log Delivery							
					╋ 新規追加				

■ 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。

S3アクセスクレデンシャル		
作成済	アクセスキーID	アクション
11月-09-2021 03:45 +0900	00a313235d4dc7183e34 *	シークレットキーを見る ^製 除
	・アクティブ	なアクセスキー新しいキーを作成

(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。

S3アクセスクレデンシャル		
作成済	アクセスキーID	アクション
11月-09-2021 03:45 +0900	00a313235d4dc7183e34 *	シークレットキーを見る 無効にする 削除
11月-09-2021 04:17 +0900	0c742bf939b59e173efe *	シークレットキーを見る 無効にする ^{割除}
	・アクティブなアク	セスキー新しいキーを作成

- 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ます。

 扱し、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。

ファイルアップロード	□ _{名前}	サイズ	最終更新
+ ファイル追加	🗆 🖺 TestFile.pdf	1.5 MB	Nov-09-2021 04:27 AM +0900
1.50 MB TestFile.pdf			

■ 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の使い方: ログイン



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

- EXASxaler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。
- <u>SQUIDの利用方法 > ログイン方法</u>

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。

フロントエンド上でs3dskey コマンドを実行します
 ※ 使用データ量をクォータに加算するグループを引数にします

\$ s3dskey create --group=(グループ名)

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	AKIA7X1KXXXYYYZZZ000
enabled	True
fspaths	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
3sdskey 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
s3dskeyhelp	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願います。



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
3sdskey enable	アクセスキーの有効化
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
s3dskeyhelp	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。

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以下に保存されます。
- Lustre ファイルシステムから上記のPath以下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	<u>s3-osakau.oniongw.hpc.cmc.osaka-u.ac.jp</u>	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 のS3 APIでのバ ケット作成は非推奨
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 \rightarrow **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できるようになります。

✓ 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/



NEC

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

フォルダー名	ストレージを追加 ・
	Amazon S3 SFTP WebDAV

(5) [Amazon S3] 用メニューで必要な情報を入力

			バケット名	
			ホスト名	
			オート	
			リージョン	
AmazonS3	Amazon S3	アクセスキー ・	□ SSLを有効	🗸
			□ パス形式を有効	
			□ レガシー認証(v2)	
			アクセスキー]
			シークレットキー]

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			
パス形式を有効	パス形式でのバケット アクセス			
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。

		Q	a.	
ここにファイルをアップロードまたは作成する権限がありませ	ъ			==
□ 名前 ▲	ţ	サイズ	更新日時	ŧ
		保留中	2時間前	
ONION-object		2.7 GB	数秒前	
SQUID_HOME <		保留中	3時間前	
3 個のフォルダー		保留中		



WEBブラウザの画面

S3 API対応ストレージ間のデータ移動

 $ONION-object \leftarrow \rightarrow EXAScaler$

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

Q このガイド内で検索	日本語
AWS 〉 ドキュメント 〉 Amazon Simp	ole Storage Service (S3) 〉 ユーザーガイド
Amazon Simple X Storage Service ユーザーガイド	Amazon S3 とは
Amazon S3 とは	
▶ 開始方法	Amazon Simple Storage Service(Amazon S3)は、インターネット用
▶ バケットの使用	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
▶ オブジェクトの使用 © NEC Corporation 2022	できるように設計されています。

46

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 アプリケーションとの親和性 クラウド・エコシステムの恩恵 > オブジェクトアクセス WEBサービスでの実装 ◆ クラウド連携の流用 高スケーラビリティを実現す REST APIで整備されている。 S3 APIを用いて、データをク るオブジェクトストレージを ストレージ操作を言語依存し ラウド連携する製品が増加。 操作可能 ない統一的なインタフェース S3 APIをサポートすればクラ で利用可能 ウド連携の恩恵を得られる ▶ 広範なSDKの提供 ◆ コミュニティの活用 柔軟なアクセス権/ポリシー 多数のプログラミング言語向 AWSはクラウドシェアの最大 フォルダ階層に捉われない柔 けにSDKが整備されおり、効 手となっており、S3 APIに関 軟なアクセス権の設定 率的にAP開発が可能 する情報も得られやすい データ保護のポリシーなど、 JS/Python/PHP/.NET/Ruby/J NASではできないデータ操作 ava/Go/Node.js/C++ が可能

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。



オブジェクト

bucketb

アクセスコントロールの最小単位

バージョニング時はバージョンIDが付与

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

S3 API の概念 – バケットの機能

- 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	0	0
デフォルト暗号 化	自動暗号化の指定	0	×
アクセスログ	アクセスログの記録有無	0	×
静的WEBサイ トホスティング	静的WEBとしての公開有無	0	×
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	0	×
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)	0	0
ポリシー	バケットへのアクセスポリ シー	0	\bigtriangleup



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)	0	0

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

※ 2021年11月現在、AWS では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。

アカウントプロファイル	
ユーザーID: CANONICAL ID: 512a2ff0fde91187b6274fc5a71f521a	グループID:

 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
	True	ΥΥΥΥΥΥΥΥΥ	0001:ZZ:XX	abcdefghijksss

✓ 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	き込み可能 ACP
Public			
Authenticated Users			
Log Delivery			

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	APIレベルのツール操作	プログラムからSDKを利用			
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	 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 	 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 			
■ アプリケーション例 aws cli、rclone、cloudberry	■ ツール例 s3curl、 aws s3api、 s3cmd	 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 はブランク

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を利用する

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。

1. リモートの定義(ONION-object)

\$ rclone config n) New remote n/s/q > nname>(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>(7222+)AWS Secret Access Key (password) secret access key> $(\dot{\nu} - \dot{\rho} \nu \nu + \dot{\tau} - \dot{\rho})$ region> osakau endpoint> s3-osakau.oniongw.hpc.cmc.osaka-u.ac.jp

(続き)

location constraint> $(\overline{7}\overline{7}\overline{2})$ 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 = s3provider = Other access key id = $(P \rightarrow z \rightarrow z \rightarrow z)$ secret_access_key = $(\mathbf{\hat{\nu}} - \mathbf{\hat{\nu}} \mathbf{\hat{\nu}} + \mathbf{\hat{\nu}})$ region = osakau endpoint = s3-osakau.oniongw.hpc.cmc.osaka-u.ac.jp acl = private

y) Yes this is OK (default) y/e/d> y https://rclone.org/

rcloneを利用する

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。

1. リモートの定義(EXAScaler)

https://rclone.org/

\$ rclone config n) New remote n/s/q > nname>(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>(7222+)AWS Secret Access Key (password) secret access key> $(\dot{\nu} - \dot{\rho} \nu \nu + \dot{\tau} -)$ 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> $(\overline{7}\overline{7}\overline{2})$ 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 = s3provider = Other secret access key = $(\dot{\nu} - \rho \nu)$ 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を利用する

S3 対応アプリケーション利用

- 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/



👶 008a9ad32eac16a55a41@s3-osakau.oniongw.hpc.cmc.osaka-u.ac.jp – S3		—		×
ファイル 編集 表示 移動 ブックマーク ウインドウ ヘルプ				
かうかいのです。 アクション でのです。 でのです。 でのです。 でのです。 新規接続 アクション でのです。	 編集 ▼	「」 アップロード	正 転送	>>
Image: Image	~	▲ 検索	ī	Q
~ ファイル名	サイズ	変更日		
	953.7 MiB	10/7/2021	10:16:56	AM
	953.7 MiB	11/4/2021	I 1:39:53 P	M
	10.4 MiB	10/6/2021	I 4:21:26 P	M
	141 B	10/13/202	21 10:22:3	5 PM
	1.2 MIB	11/9/2021 不明	1 7:10:50 A	M
6 771N				<u>a</u> .:

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

 Cyberduck - □ × ファイル 編集 表示 移動 ブックマーク ウインドウ ヘルブ ジー ② ・ ② ・ ○ ・ ○ ・ ○ ・ ○ ・ ○ ・ ○ ・ ○ ・ ○ ・		
新規接続 アクション 情報を見る 更新 編集 アップロード 転送 接続解除	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
シークレットアクセスキー:	ポート	443
SSH Private Key: なし く Choose	アクセス キー	(発行したアクセスキー)
+ / - 接続 キャンセル o ブックマーク 詳細設定 Path:	シーク レット キー	(発行したシークレットキー)
Connect Mode: デフォルト 〜		
Encoding: UTF-8		

アプリケーション利用に関する注意事項

S3 対応アプリケーション利用

アプリケーションを利用にあたっては、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ください。

仮想ホスティング形式のアクセス時に証明書エラーが発生する

仮想ホスティング形式でのみアクセスするアプリケーションにて、証明書エラーが発生するケースがあります。
 → この問題は2022年6月現在解消されています。

s3curl を利用する

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

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 を利用する

APIレベルのツール操作

s3curl による実行例です。

✓ S3DSヘ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 CLI S3API を利用する

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 はブランク

✓ 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 むけSDK (BOTO3) 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、BOTO3 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 revel 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 service 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APIを発行せ ず操作

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ま した。順次更新されています。「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/

大阪大学 サイバーメディアセンター Q 大規模計算機システム アクセス サイトマップ								Q 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			
ONIC	Nの利用方法			Cyt	ermedia Center, O	saka University > シ	ステム > 利	用方法 > ONIONØ	D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 ユーザ登 利用状況	利用方法 録、削除方法 2の確認方法								

情報参照先 - S3 API関連

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 <u>https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-welcome.html</u>
- AWS CLI Comand Reference <u>https://awscli.amazonaws.com/v2/documentation/api/latest/reference/index.html</u>

AWS Boto3

• Boto3 Docs 1.20.1

https://boto3.amazonaws.com/v1/documentation/api/latest/guide/quickstart.html
Orchestrating a brighter world

