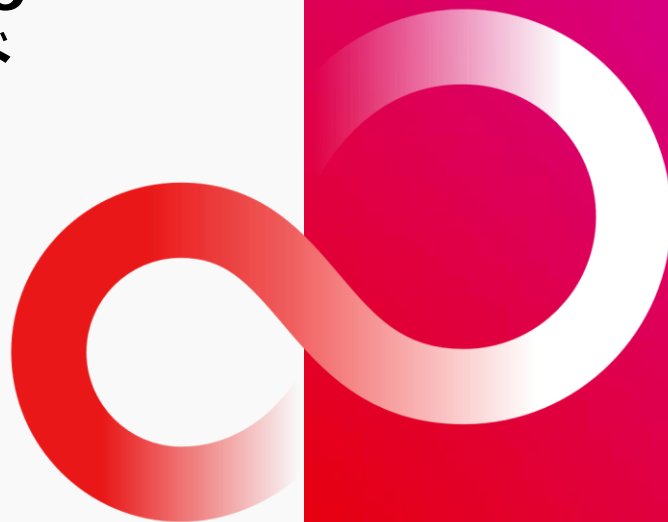


FUJITSU Hybrid IT Service FJcloud-O オブジェクトストレージ データ移行ガイド (SwiftからS3)

対象リージョン
東日本第3リージョン
西日本第3リージョン

2025年4月16日 1.0版
富士通株式会社



版数	改訂日	概要
1.0	2025/4/16	新規作成

- 本書の概要
- SwiftからS3へのデータ移行概要
- SwiftからS3へのデータ移行手順
- 付録

本書の概要

語句	定義
Swift API	FJcloud-O 東日本第3、西日本第3リージョンで提供するオブジェクトストレージ Swift互換APIのこと。 東日本第1、東日本第2、西日本第1、西日本第2リージョンで提供されるSwift互換APIは含まない。
S3 API	FJcloud-O 東日本第3、西日本第3リージョンで提供するオブジェクトストレージ S3互換APIのこと。

- 本ガイドではFJcloud-O[東日本第3、西日本第3リージョン]で提供しております、オブジェクトストレージ Swift互換API（以降Swift API）で格納したデータを、オブジェクトストレージ S3互換API（以降S3 API）に移行する手順を記載しています。
- 各サービスの詳細は下記ドキュメントをご参照ください。
 - オブジェクトストレージ概要 : [機能説明書](#)
 - Swift API : [機能説明書/APIリファレンス](#)
 - S3 API : [機能説明書/APIリファレンス](#)

Swift APIの提供終了について

- FJcloud-O[東日本第3、西日本第3リージョン]で提供しております、オブジェクトストレージ Swift APIは2025年12月31日で提供を終了いたします。
- 提供終了後、Swiftデータの格納、格納済みSwiftデータの参照を含む Swift APIの一切の操作が使用できなくなるため、データ移行をお願いいたします。
- スケジュール（2025年4月16日時点、最新のスケジュールは別途広報を参照）

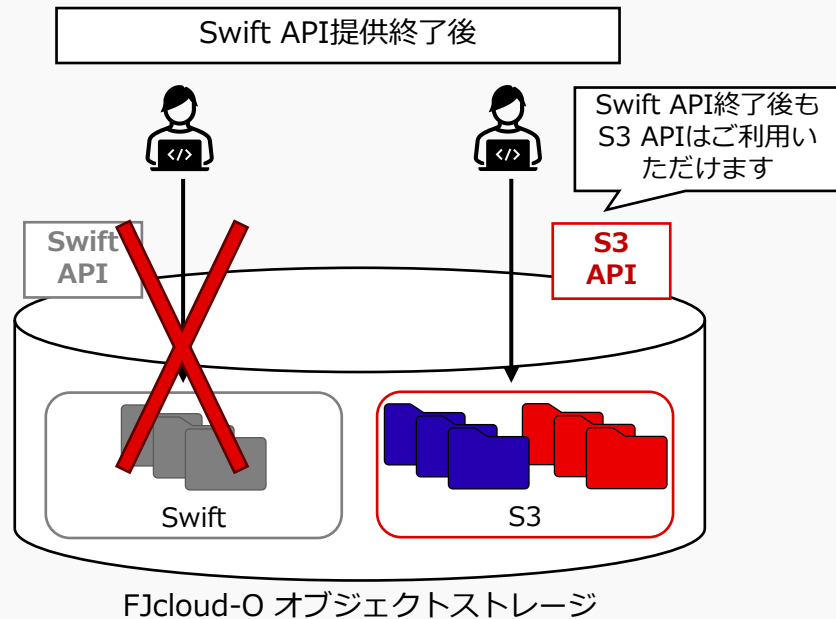
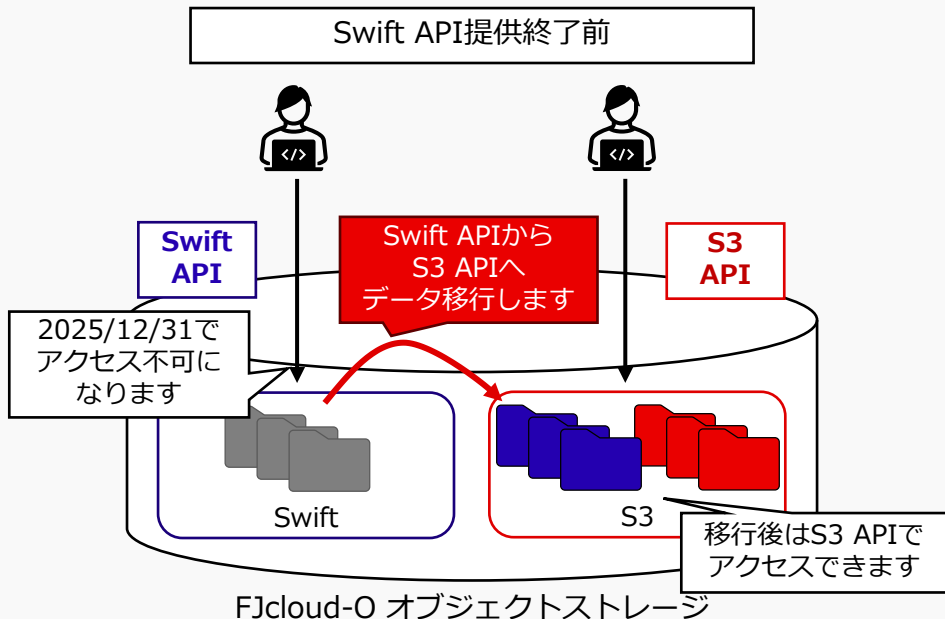
日付	イベント	備考
2025/3/17	Swift API 提供終了の事前告知	Swift APIの利用が非推奨となります
2025/12/31	Swift API 提供終了	Swift APIが利用不可となります

※ Swift APIだけが提供終了となります。S3 APIは引き続きご利用いただけます。

SwiftからS3へのデータ移行概要

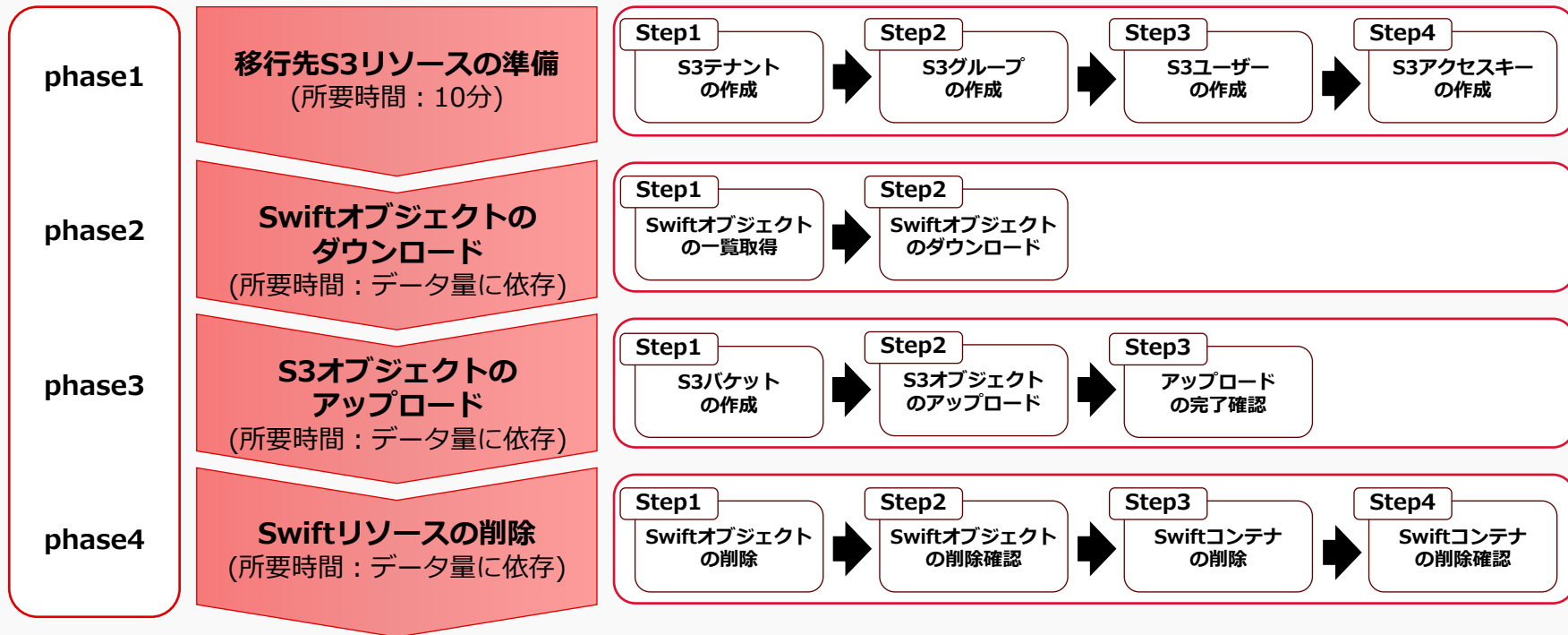
データ移行作業概要

- Swift API提供終了後には、Swift APIの一切の操作が使用できなくなります。Swift API提供終了前にS3 APIへのデータ移行をお願いいたします。



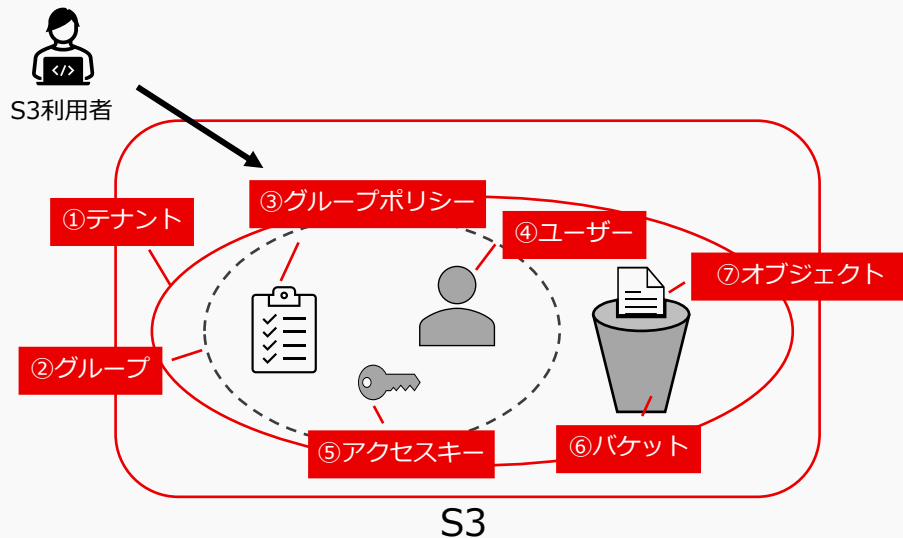
データ移行作業フロー

- 下記の流れでデータ移行を実施します。



- S3 APIはAWSのS3との互換性を重視したAPIです。
 - Swift APIと比較してより多岐にわたる操作が可能です。
 - SwiftリソースをS3 APIで、S3リソースをSwift APIで操作することはできません。

<S3 APIのイメージ>



<各リソースの概要>

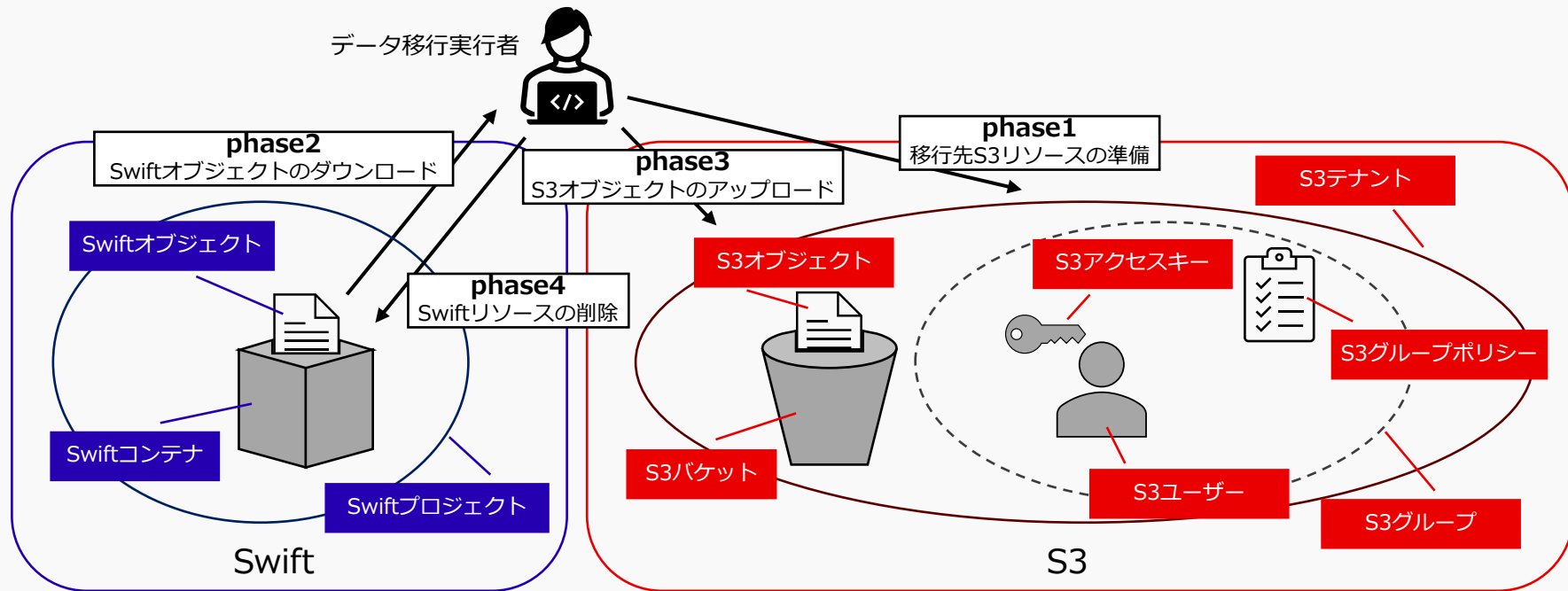
①テナント	S3リソースの管理単位 Swiftのプロジェクトに相当する
②グループ	ユーザーを所属させることができるグループ
③グループポリシー	グループに設定できるアクセス権や実行権限
④ユーザー	S3利用時に使用するユーザーアカウント
⑤アクセスキー	ユーザーに紐づけて発行される、 バケット・オブジェクト操作に必要なアクセス鍵
⑥バケット	オブジェクトを格納する入れ物 Swiftのコンテナに相当する
⑦オブジェクト	オブジェクトストレージに格納するデータ

SwiftからS3へのデータ移行手順

- 本作業はAPI(REST API)で実施します。
APIを実行できるクライアント(curlなど)をご利用ください。
- **オブジェクト削除後の復元はできません。**
S3への移行完了を確認したうえでSwiftオブジェクトの削除をお願いいたします。

データ移行のシナリオ

- 本ガイドでは以下の移行例を想定して、移行手順を説明します。



※構成、設定は一例のため、利用環境に応じて適宜読み替えて作業してください。

- 本作業で使用するAPIは下記2種類の認証方法を用いて実行します。

- トークン認証
- アクセスキー認証

	トークン認証	アクセスキー認証
認証情報	FJcloud-Oのユーザー名、パスワードなどを使用して取得したトークン文字列	S3のアクセスキー、シークレットキーを使用して生成した署名文字列
認証方法	・ リクエストヘッダーにトークン文字列を付加	以下2種類のいずれかで認証 ・ リクエストヘッダーに署名文字列を付加 ・ URLクエリパラメータに署名文字列を付加
対象API	・ 全てのSwift API ・ バケット・オブジェクト以外を操作するS3 API <本ガイドでは以下マークを記載> Swift API (トークン認証) S3 API (トークン認証)	・ バケット・オブジェクトを操作するS3 API <本ガイドでは以下マークを記載> S3 API (アクセスキー認証)
参考ドキュメント	・ 利用者管理	・ アクセスキー・シークレットキー ・ 署名情報

phase1 : 移行先S3リソースの準備

phase1

移行先S3リソースの準備
(所要時間 : 10分)

Step1

S3テナント
の作成

Step2

S3グループ
の作成

Step3

S3ユーザー
の作成

Step4

S3アクセスキー
の作成

データ移行実行者



phase2

Swiftオブジェクトのダウンロード

Swiftオブジェクト

Swiftコンテナ

Swiftプロジェクト

Swift

phase3

S3オブジェクトのアップロード

S3オブジェクト

S3バケット

phase1

移行先S3リソースの準備

S3テナント

S3アクセスキー

S3グループポリシー

S3ユーザー

S3グループ

S3

1-1 : S3テナントの作成

- APIリファレンス : [Create Tenant](#)

S3 API
(トークン認証)

<リクエストサンプル>

POST /v2/admin/accounts

<リクエストボディサンプル>

```
{"name": "<テナント名>"}
```

<レスポンスボディサンプル>

```
{  
  "apiVersion": "2.2",  
  "code": 201,  
  "data": {  
    "id": "<テナントID>",  
    "name": "<テナント名>"  
  },  
  "responseTime": "<レスポンス日時>",  
  "status": "success"  
}
```

以降の手順で使用

1-2 : S3グループの作成

- APIリファレンス : [Create Group](#)

S3 API
(トークン認証)

<リクエストサンプル>

POST /v2/admin/ <テナントID>/groups

<リクエストボディサンプル>

```
{
  "uniqueName": " <グループ名>",
  "Statement": [
    {
      "Sid": " <グループポリシー名>",
      "Action": " <許可/拒否する操作>",
      "Effect": " <許可/拒否>",
      "Resource": " <操作対象のリソース>"
    }
  ]
}
```

※[ポリシー設定](#)は利用環境に応じて適切に設定してください。

※設定例は[ポリシーの設定例](#)を参照してください。

<レスポンスボディサンプル>

```
{
  (中略)
  "Statement": [
    {
      "Action": " <許可/拒否する操作>",
      "Effect": " <許可/拒否>",
      "Resource": " <操作対象のリソース>",
      "Sid": " <グループポリシー名>"
    }
  ],
  "accountId": " <テナントID>",
  "displayName": " <グループ名>",
  (中略)
  "id": " <グループID>",
  (中略)
  "status": "success"
}
```

以降の手順で使用

1-3 : S3ユーザーの作成

- APIリファレンス : [Create User](#)

S3 API
(トークン認証)

<リクエストサンプル>

POST /v2/admin/ <テナントID>/users

<リクエストボディサンプル>

```
{
  "uniqueName": " <ユーザー名>",
  "memberOf": [
    " <グループID>"
  ]
}
```

<レスポンスボディサンプル>

```
{
  "apiVersion": "2.2",
  "code": 201,
  "data": {
    "accountId": " <テナントID>",
    "fullName": " <ユーザー名>",
    "id": " <ユーザーID>",
    "memberOf": [
      " <グループID>"
    ],
    "uniqueName": "user/ <ユーザー名>",
    "userURN": " <ユーザーURN>"
  },
  "responseTime": " <レスポンス日時> ",
  "status": "success"
}
```

以降の手順で使用

1-4 : S3アクセスキーの作成

- APIリファレンス : [Create s3-access-key](#)

S3 API
(トークン認証)

<リクエストサンプル>

POST /v2/admin/ <テナントID>/users/ <ユーザID>/s3-access-keys

<リクエストボディサンプル>

```
{
  "expires": <アクセスキーの有効期限>
}
```

<レスポンスボディサンプル>

```
{
  "apiVersion":"2.2",
  "code":201,
  "data":{
    "accessKey": "<アクセスキー>",
    "accountId": "<テナントID>",
    "displayName": "<マスクされたアクセスキー>",
    "expires": "<アクセスキーの有効期限>",
    "id": "<アクセスキーID>",
    "secretAccessKey": "<シークレットアクセスキー>",
    "userURN": "<ユーザーURN>",
    "userUUID": "<ユーザーID>"
  },
  "responseTime": "<レスポンス日時>",
  "status":"success"
}
```

以降の手順で使用

以降の手順で使用

phase2 : Swiftオブジェクトのダウンロード

phase2

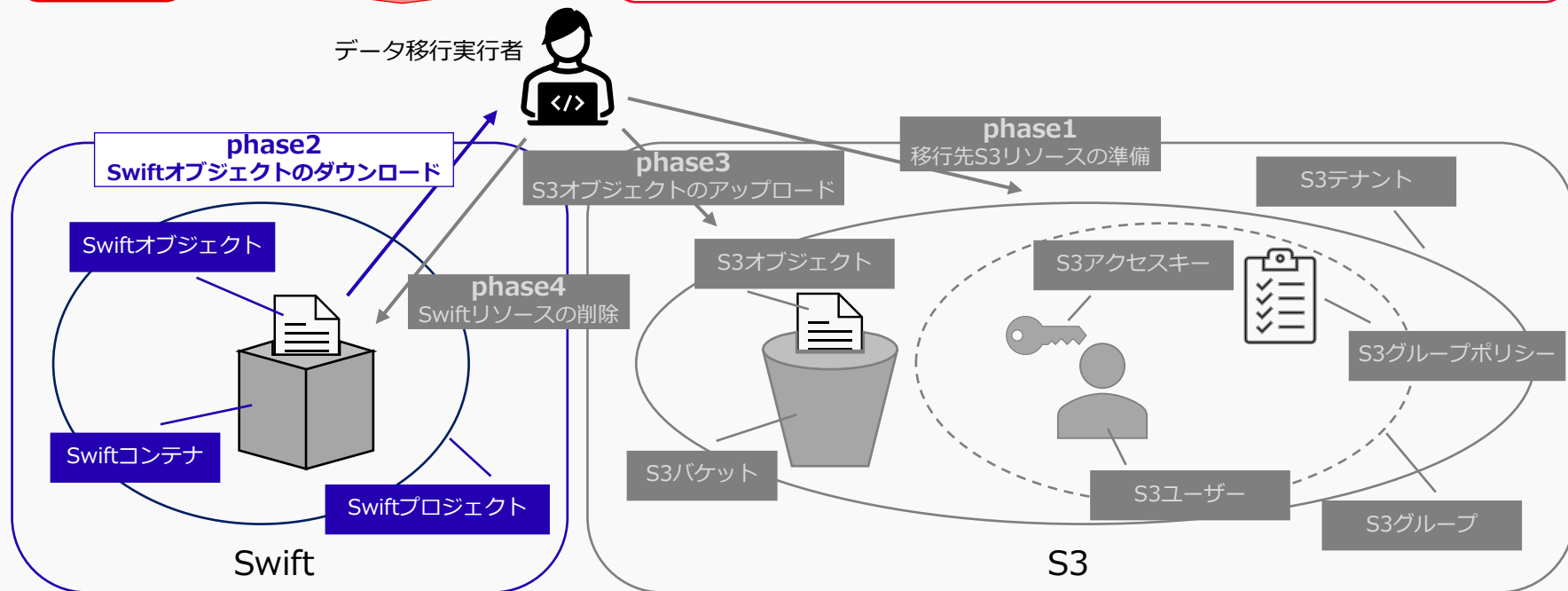
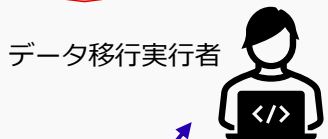
**Swiftオブジェクトの
ダウンロード**
(所要時間：データ量に依存)

Step1

Swiftオブジェクト
の一覧取得

Step2

Swiftオブジェクト
のダウンロード



2-1 : Swiftオブジェクトの一覧取得

- APIリファレンス : [List container objects, statistics, and metadata](#)

Swift API
(トークン認証)

<リクエストサンプル>

```
GET /v2/ <AUTH_プロジェクトID>/ <コンテナ名>?format=json
```

<レスポンスボディサンプル>

```
[  
  {  
    "hash": " <オブジェクトのハッシュ値>",  
    "last_modified": " <オブジェクトの最終更新日時>",  
    "bytes": <オブジェクト容量>,  
    "name": " <オブジェクト名>",  
    "content_type": " <オブジェクトのMIME>"  
  },  
  {  
    (中略)  
  }  
]
```

※データ移行前にSwiftのコンテナ総数、オブジェクト総数、オブジェクト総容量を確認する場合は下記APIをご利用ください。

[Show Account information and statistics metadata](#)

レスポンスヘッダーを確認することで上記のSwift使用状況を確認できます。

2-2 : Swiftオブジェクトのダウンロード

- APIリファレンス : [Show object content and metadata](#)

Swift API
(トークン認証)

<リクエストサンプル>

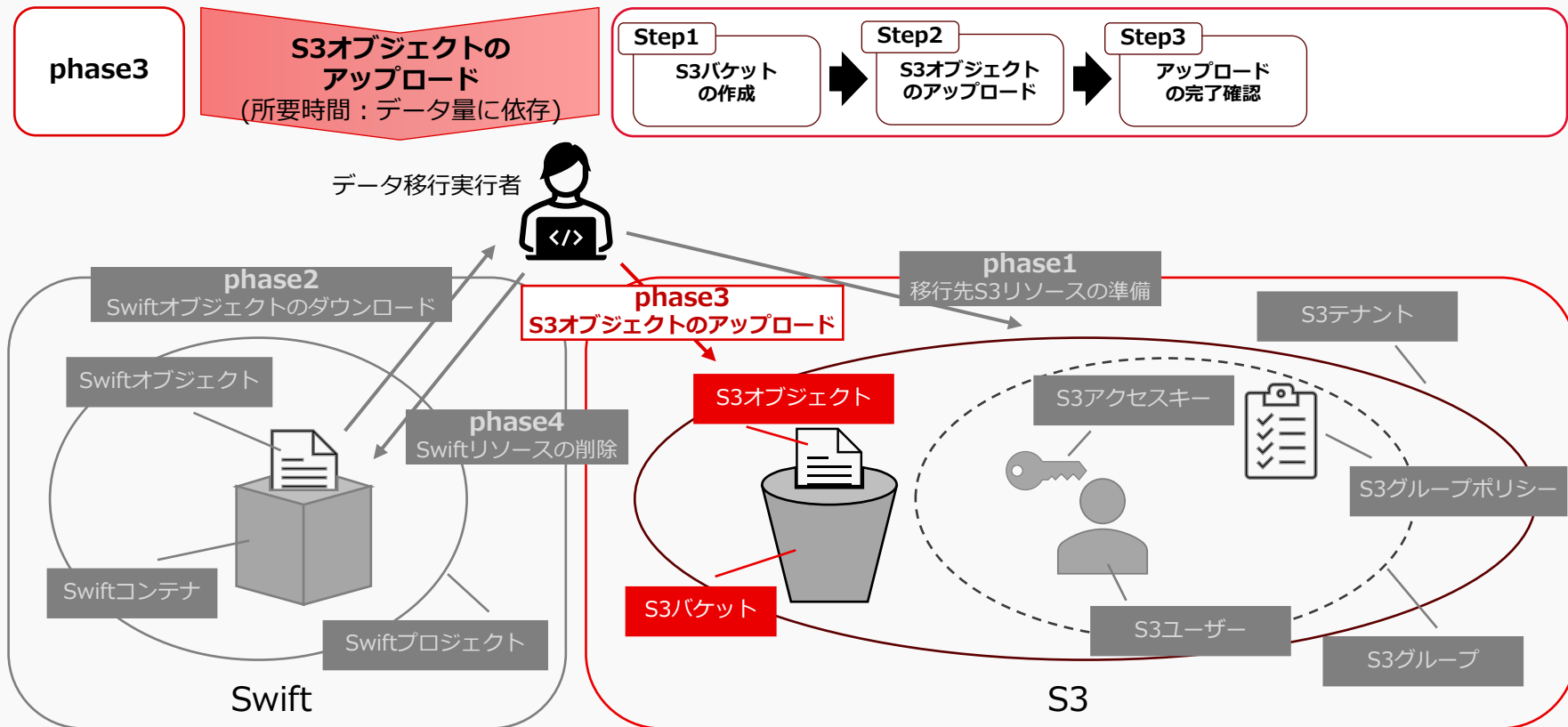
GET /v2/ <AUTH_プロジェクトID>/ <コンテナ名>/ <オブジェクト名>

※オブジェクトのダウンロード先を指定してください。

指定方法をご利用のクライアントソフトの仕様に則ってください。

※完了後は指定したダウンロード先にオブジェクトが存在していることを確認してください。

phase3 : S3オブジェクトのアップロード



3-1 : S3バケットの作成

- APIリファレンス : [Put Bucket](#)

S3 API
(アクセスキー認証)

<リクエストサンプル>

PUT / <バケット名>

<リクエストヘッダーサンプル>

- H "x-amz-content-sha256 : <リクエスト内容のチェックサム値>"
- H "x-amz-date: <リクエスト日時>"
- H "Authorization : <リクエスト認証用文字列>"

※バケット名はオブジェクトストレージ全体で一意である必要があるため、Swiftコンテナと同名のS3バケットは作成できません。
※本リクエストには[署名情報](#)を付与する必要があります。

3-2 : S3オブジェクトのアップロード

- S3 APIでのオブジェクトアップロードは[マルチパートアップロード](#)をご利用ください
 - S3では5GiBを超えるオブジェクトをアップロードする場合、マルチパートアップロードを利用して分割アップロードする必要があります。
5GiBを超えるオブジェクトの[PutObject](#) APIによるアップロードはサポートされません。
 - マルチパートアップロードは[下記3ステップ](#)で実施可能です
 1. マルチパートアップロードの開始
 2. 分割したオブジェクトのアップロード
 3. マルチパートアップロードの完了

※5GiB未満のオブジェクトは[Put Object](#) APIを利用したアップロードも可能です。

3-2-1 : マルチパートアップロードの開始

- APIリファレンス : [Initiate Multipart Upload](#)

S3 API
(アクセスキー認証)

<リクエストサンプル>

POST / <バケット名> / <オブジェクト名> ?uploads

<リクエストヘッダーサンプル>

-H "x-amz-content-sha256 : <リクエスト内容のチェックサム値>"
-H "x-amz-date: <リクエスト日時>"
-H "Authorization : <リクエスト認証用文字列>"

※本リクエストには[署名情報](#)を付与する必要があります。

<レスポンスボディサンプル>

```
<?xml version="1.0" encoding="UTF-8"?>
<InitiateMultipartUploadResult xmlns="...">
  <Bucket> <バケット名> </Bucket>
  <Key> <オブジェクト名> </Key>
  <UploadId> <アップロードID> </UploadId>
</InitiateMultipartUploadResult>
```

以降の手順で使用

※InitiateしてからCompleteするまでに15日以内に完了する必要があります。

※一つのバケットで同時実行可能なマルチパートアップロードは1000件までです。

※マルチパートアップロードを中止する場合は[Abort Multipart Upload](#) APIを実行してください。

3-2-2 : 分割したオブジェクトのアップロード

- APIリファレンス : [Upload Part](#)

S3 API
(アクセスキー認証)

<リクエストサンプル>

PUT / <バケット名> / <オブジェクト名> ?partNumber= <分割したオブジェクトの通し番号> &uploadId= <アップロードID>

<リクエストヘッダーサンプル>

-H "x-amz-content-sha256: <リクエスト内容のチェックサム値>"
-H "x-amz-date: <リクエスト日時>"
-H "Authorization: <リクエスト認証用文字列>"

※本リクエストには署名情報を付与する必要があります。
※アップロードするオブジェクトを指定してください。
指定方法はご利用のクライアントソフトの仕様に則ってください。

<レスポンスヘッダーサンプル>

HTTP/1.1 200 OK
Date: <レスポンス日時>
Content-Length: 0
Connection: keep-alive
x-amz-request-id: <リクエストID>
x-amz-id-2: <リクエストID>
x-ntap-sg-trace-id: <リクエストID>
ETag: " <オブジェクト部分のEタグ文字列> "
X-Fcx-Endpoint-Request: . . .

以降の手順で使用

※アップロードするオブジェクトは事前に分割しておく必要があります。

例) Linux環境でオブジェクトs3testを500MBずつに分割する場合

split -b 500MB -d ./s3test s3test

→分割後のオブジェクト : s3test00、s3test01、 . . .

※アップロードしたパートの一覧は[List Parts](#) APIで確認可能です。

3-2-3 : マルチパートアップロードの完了

- APIリファレンス : [Complete Multipart Upload](#)

S3 API
(アクセスキー認証)

<リクエストサンプル>

POST / <バケット名> / <オブジェクト名> ?uploadId= <アップロードID>

<リクエストヘッダーサンプル>

-H "x-amz-content-sha256 : <リクエスト内容のチェックサム値>"
-H "x-amz-date: <リクエスト日時>"
-H "Authorization : <リクエスト認証用文字列>"

<リクエストボディサンプル>

```
<CompleteMultipartUpload xmlns="http://...">
  <Part>
    <ETag> <オブジェクトパート1のEタグ> </ETag>
    <PartNumber> 1 </PartNumber>
  </Part>
  (中略)
</CompleteMultipartUpload>
```

※本リクエストには署名情報を付与する必要があります。

<レスポンスボディサンプル>

```
<CompleteMultipartUploadResult xmlns="...">
  <Location> <オブジェクトのURI> </Location>
  <Bucket> <バケット名> </Bucket>
  <Key> <オブジェクト名> </Key>
  <ETag> " <オブジェクトのEタグ文字列> " </ETag>
</CompleteMultipartUploadResult>
```

※マルチパートアップロード後のオブジェクトのEタグ文字列はSwiftのオブジェクトダウンロード時に出力されるEタグ文字列と異なりますのでご注意ください。

3-3 : アップロードの完了確認

- APIリファレンス : [Head Object](#)

S3 API
(アクセスキー認証)

<リクエストサンプル>

HEAD / <バケット名> / <オブジェクト名>

<リクエストヘッダーサンプル>

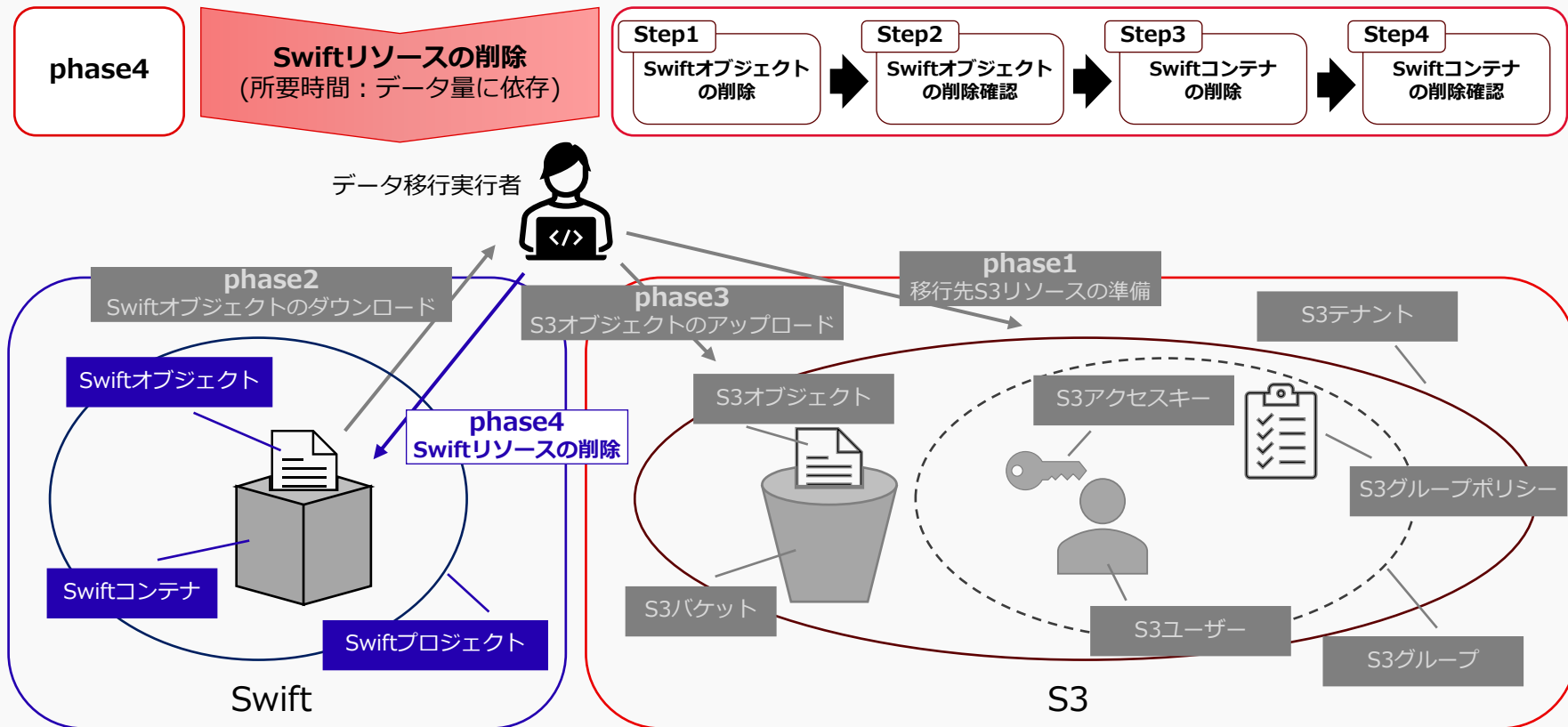
-H "x-amz-content-sha256 : <リクエスト内容のチェックサム値>"
-H "x-amz-date: <リクエスト日時>"
-H "Authorization : <リクエスト認証用文字列>"

※本リクエストには[署名情報](#)を付与する必要があります。

<レスポンスヘッダーサンプル>

HTTP/1.1 200 OK
Date: <レスポンス日時>
Content-Type: <MIMEタイプ>
Content-Length:
Connection: keep-alive
x-amz-request-id: <リクエストID>
x-amz-id-2: <リクエストID>
x-ntap-sg-trace-id: <リクエストID>
ETag:" <オブジェクトのEタグ>"
x-amz-server-side-encryption: <暗号化方式>
Last-Modified: <オブジェクトの最終更新日時>
Accept-Ranges: bytes
X-Fcx-Endpoint-Request: . . .

phase4 : Swiftリソースの削除



4-1 : Swiftオブジェクトの削除

- APIリファレンス : [Delete Object](#)

Swift API
(トークン認証)

<リクエストサンプル>

DELETE / < AUTH_プロジェクトID > / <コンテナ名> / <オブジェクト名>

<レスポンスヘッダーサンプル>

HTTP/1.1 204 No Content
Date: <レスポンス日時>
Content-Length: 0
Connection: keep-alive
X-Trans-Id: <リクエストID>
x-ntap-sg-trace-id: <リクエストID>
X-Fcx-Endpoint-Request: . . .

4-2 : Swiftオブジェクトの削除確認

- APIリファレンス : [Show object metadata](#)

Swift API
(トークン認証)

<リクエストサンプル>

HEAD / <AUTH_プロジェクトID> / <コンテナ名> / <オブジェクト名>

<レスポンスヘッダーサンプル>

HTTP/1.1 404 NotFound
Date: <レスポンス日時>
Content-Type: <MIMEタイプ>
Content-Length: 102400
Connection: keep-alive
X-Trans-Id: <リクエストID>
ETag: <リクエストID>
Accept-Ranges: bytes
Last-Modified <オブジェクトの最終更新日時>
X-Timestamp: <リクエスト処理日時>
X-Fcx-Endpoint-Request: . . .

削除済みのため、
404 NotFoundになる

4-3 : Swiftコンテナの削除

- APIリファレンス : [Delete a container](#)

Swift API
(トークン認証)

<リクエストサンプル>

DELETE / < AUTH_プロジェクトID > / <コンテナ名>

<レスポンスヘッダーサンプル>

HTTP/1.1 204 No Content

Date: <レスポンス日時>

Content-Length: 0

Connection: keep-alive

X-Trans-Id: <リクエストID>

X-Fcx-Endpoint-Request: . . .

※コンテナにオブジェクトが格納されている場合、

下記のエラーでリクエストが失敗します。

オブジェクトをすべて削除してから再試行してください。

"There was a conflict when trying to complete your request."

4-4 : Swiftコンテナの削除確認

- APIリファレンス : [List container statistics and metadata](#)

Swift API
(トークン認証)

<リクエストサンプル>

HEAD / < *AUTH_プロジェクトID* > / <コンテナ名>

<レスポンスヘッダーサンプル>

HTTP/1.1 404 NotFound
Date: <レスポンス日時>
Content-Type: <MIMEタイプ>
Content-Length: 102400
Connection: keep-alive
X-Trans-Id: <リクエストID>
Accept-Ranges: bytes
Last-Modified: <オブジェクトの最終更新日時>
X-Timestamp: <リクエスト処理時刻>
X-Fcx-Endpoint-Request: . . .

削除済みのため、
404 NotFoundになる

付録

- API対応表
- Swift互換API/S3互換APIの非互換一覧
- 移行作業のログ

No.	操作対象	SwiftAPI	対応するS3API
1	Accounts/ Tenants	Show Container and Account Usage Statistics	Get Service *1
2		Show Account information and statistics metadata	対応APIなし
3	Containers/ Buckets	Delete a container	DELETE Bucket
4		List container objects, statistics, and metadata	Get Bucket(List Objects) version2
5		List container statistics and metadata	Head Bucket
6		Create, modify, or delete container ACL metadata	Put Bucket policy *2
7		Create a container	PUT Bucket
8	Objects	Delete an object	Delete Object
9		Show object content and metadata	Get Object
10		Show object metadata	Head Object
11		Create or replace an object	Put Object
12	Consistency	Show container consistency	Get bucket consistency *3
13		Set container consistency	Put bucket consistency

*1 : バケット一覧のみを取得可能でオブジェクト総数、総容量の取得はできません。(S3の仕様に準拠しています。)

*2 : ACL設定と同等のアクセス権設定が可能です。詳細は[ポリシー設定](#)を参照ください。

*3 : レスポンスヘッダーには整合性レベルが表示されず、レスポンスボディだけに整合性レベルが表示されます。

移行時に注意が必要なSwift API/S3 APIの非互換

非互換項目	Swift	S3
バケット/コンテナ命名時の 使用可能文字	- 大文字・小文字アルファベット - 数字 ・ <ピリオド> - <ハイフン> _ <アンダースコア>	- 小文字アルファベット - 数字 - <ハイフン>
バケット/コンテナ名のbyte数	255byte以下	3~63byte
オブジェクト命名時の 使用可能文字	- 大文字・小文字アルファベット - 数字 ! <エクスクラメーション> # <ハッシュマーク> \$ <ドルマーク> % <パーセント> - <ハイフン> _ <アンダースコア> ・ <ピリオド> * <アスタリスク> / <カンマ> @ <アットマーク> = <イコール> ~ <チルダ> + <プラス> : <コロン>	- 大文字・小文字アルファベット - 数字 ! <エクスクラメーション> - <ハイフン> _ <アンダースコア> ・ <ピリオド> * <アスタリスク> ' <シングルクォーテーション> () <左右丸括弧> / <スラッシュ> & <アンド> \$ <ドル> @ <アットマーク> = <イコール> ; <セミコロン> : <コロン> + <プラス> <スペース> / <カンマ> ? <クエスチョンマーク> ※URLエンコードが必要です %26 ※URLエンコードが必要です %24 ※URLエンコードが必要です %40 ※URLエンコードが必要です %3D ※URLエンコードが必要です %3B ※URLエンコードが必要です %3A ※URLエンコードが必要です %2B ※URLエンコードが必要です %20 ※URLエンコードが必要です %2C ※URLエンコードが必要です %3F

Thank you

